

004  
0-62

U.R.Xamdamov, Dj.B.Sultanov,  
S.S.Parsiyev, U.M.Abdullayev

Linux



# OPERATION TIZIMLAR

O‘ZBEKISTON RESPUBLIKASI AXBOROT  
TEXNOLOGIYALARI VA KOMMUNIKATSIYALARINI  
RIVOJLANTIRISH VAZIRLIGI

MUHAMMAD AL-XORAZMIY NOMIDAGI  
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI

U.R.XAMDAMOV, DJ.B.SULTANOV,  
S.S.PARSIYEV, U.M.ABDULLAYEV

# OPERATSION TIZIMLAR

O‘zbekiston Respublikasi  
Oliy va o‘rtamaqsusta’lim vazirligi  
tomonidan o‘quv qo‘llanmasifatida  
tavsiya etilgan.

Toshkent –2021

**UDK: 004.45(075.8)**

**KBK: 32.973**

**U.R.Xamdamov, Dj.B.Sultanov, S.S.Parsiyev, U.M.Abdullayev**  
**Operatsion tizimlar. (O‘quv qo‘llanma). – T.: «Nihol print» OK, 2021. – 436 b.**

**ISBN 978–9943–7343-2-6**

Operatsion tizimlar hisoblash mashinalari va hisoblash tizimlari uchun dasturiy ta’minotning asosidir. Hozirgi kunda kompyuterlashtirish hayotimizning barcha sohalarini qamrab oldi. Ushbu o‘quv qo‘llanmada operatsion tizimlarning asosiy tushunchalari va ularning ahamiyati, operatsion tizimlarning asosiy funksiyalari va imkoniyatlarini o‘rganish masalalarini o‘z ichiga olgan.

Mazkur fan talabalarga maxsus fanlarni o‘zlashtirishda, keyinchalik ishlab chiqarish, loyihalash va tadqiqot ishlarida kerak bo‘ladigan asosiy negiz tushunchalarni o‘rgatadi. O‘quv qo‘llanmada “Operatsion tizimlar” fanini o‘zlashtirish jarayonida talabalar tomonidan o‘zlashtirilishi kerak bo‘lgan bilimlar: operatsion tizim tushunchasi va ularning ahamiyati, ularning tuzilishi va asosiy komponentlari, operatsion tizimlarning asosiy funksiyalari bo‘lgan jarayonlarni boshqarish, protsessorni boshqarish, xotirani tashkil etish va boshqarish usullari, tashqi qurilmalarni boshqarish, fayl tizimi va fayllarni boshqarish masalalari, tarmoq operatsion tizimlari va ularning imkoniyatlari, tarmoq xizmatlari va ularni tashkil etish, xavfsizlik va himoya vositalari va dasturlari, amaliyotda keng qo‘llaniladigan Linux, Windows, Android operatsion tizimlari, ularning qurilish tamoyillari va imkoniyatlari haqidagi ma’lumotlar keltirilgan.

O‘quv qo‘llanma 5350100 - “Telekommunikatsiya texnologiyalari” (Telekommunikatsiyalar, Teleradioeshittirish, Mobil tizimlar) yo‘nalishi bo‘yicha oliy o‘quv yurtlari talabalari, hamda kredit ta’lim tizimi bo‘yicha ta’lim olayotgan talabalarning “Tarmoqni dasturlash asoslari”, “O‘rnatilgan tizimlar” v.b. fanlarni tayanch fani hisoblanadi va shu fanlar bilan mantiqan bog‘langan.

**UDK: 418-371**

**KBK: 32.973**

**Taqrizchilar:** B.N.Raximov;  
X.N.Zaynidinov.

**Mas’ul muharrir:** S.S.Parsiyev.

**Musahhih:** M.X.Axmedova.

**ISBN 978–9943–7343-2-6**

© «Nihol print» OK nashriyoti, 2021.

## Kirish

Operatsion tizimlar har qanday kompyuter tizimining ajralmas qismidir. Shu sababli, operatsion tizimlar kursi har qanday kompyuter ta'limining ajralmas qismi hisoblanadi. Ushbu sohada tez o'zgarishlar ro'y bermoqda, chunki hozirgi kunda kompyuterlar kundalik hayotning deyarli barcha sohalarida - avtomashinalar ichiga o'rnatilgan qurilmalardan tortib, hukumatlar va transmilliy kompaniyalar uchun eng zamonaviy rejalashtirishning eng murakkab vositalariga qadar keng tarqalgan. Zamonaviy kompyuter bir yoki bir nechta protsessorlardan, operativ xotiradan, disklardan, printerdan, klaviaturadan, sichqonchadan, displeydan, tarmoq interfeyslaridan va boshqa har xil kiritish/chiqarish qurilmalaridan iborat. Natijada esa ancha murakkab tizim hosil bo'ladi. Bundan tashqari, komponentlarning barchasini boshqarish va ulardan maqbul foydalanish juda qiyin vazifadir. Shu sababli, kompyuterlar operatsion tizim deb ataladigan maxsus darajadagi dasturiy ta'minot bilan jihozlangan, ularning vazifasi foydalanuvchi dasturlarini va barcha manbalarini boshqarishdir.

Ushbu o'quv qo'llanmada operatsion tizimlarning asosiy tushunchalari, tuzilishi, asosiy funksiyalari, imkoniyatlari, ularda tarmoq xizmatlari va boshqalar ko'rib chiqilgan. Uning maqsadi zamonaviy operatsion tizimlarning tabiati va xususiyatlarini iloji boricha aniq va to'liq taqdim etishdir. Bu vazifa bir necha sabablarga ko'ra qiyin. Operatsion tizimlar o'rnatilgan ko'plab kompyuter tizimlari va xilma-xilligi mavjud. Bular ichiga operatsion tizimlar ishlaydigan o'rnatilgan tizimlar: smartfonlar, bir foydalanuvchili ishchi stansiyalar va shaxsiy kompyuterlar, o'rta o'lchamdagi umumiy tizimlar, superkompyuterlar, shuningdek real vaqt tizimlari kabi ixtisoslashtirilgan mashinalar kiradi. O'quv qo'llanmada keltirilgan asosiy tushunchalar va algoritmlar ham tijorat, ham ochiq manbali operatsion tizimlarda qo'llaniladigan tushunchalarga asoslanadi. Bizning maqsadimiz ushbu tushunchalar va algoritmlarni bitta operatsion tizimga bog'lamagan holda umumiy sharoitda taqdim etishdir. Bundan tashqari biz hozirgi kundagi eng mashhur va eng innovatsion Linux, Microsoft Windows va Android kabi operatsion tizimlarni ko'rib chiqamiz.

Ushbu kitobning mazmuni to‘qqizta asosiy bobdan iborat:

Birinchi bob operatsion tizimlar va ularning ahamiyatini o‘rganishga bag‘ishlanadi. Ushbu bobda operatsion tizimlar nima, ular qanday vazifani bajaradi, operatsion tizimlarning tuzilishi va ularning asosiy komponentlari qanday yaratilganligi tushuntiriladi. Ushbu boblarda operatsion tizimning umumiy xususiyatlari va foydalanuvchi uchun operatsion tizimning vazifalari muhokama qilinadi. Bundan tashqari operatsion tizimlarning rivojlanish davrlari, kompyuter tizimlari tuzilishi va ahamiyati masalalari ko‘rib chiqilgan.

Ikkinchi bobda operatsion tizimda jarayonlarni va protsessorni boshqarish masalalari ko‘rib o‘tilgan. Ushbu bobda zamonaviy operatsion tizimlarning yuragi sifatida jarayon, oqimlar va protsessorni boshqarish masalalari tasvirlangan. Jarayon - bu tizimdagi ish birligi. Bunday tizim bir vaqtning o‘zida ishlaydigan jarayonlar to‘plamidan iborat bo‘lib, ularning ba’zilari operatsion tizimning jarayonlaridir (tizim kodini bajaradiganlar), qolganlari foydalanuvchi jarayonlaridir (foydalanuvchi kodini bajaradiganlar). Bundan tashqari ushbu bobda jarayonlarni rejalashtirish usullari, jarayonlar o‘rtasidagi o‘zaro ta’sirlar, jarayonlarni sinxronlashtirish va berklklarga ishlov berish usullari muhokama qilinadi.

Uchinchi bob xotirani tashkil etish va boshqarish usullariga bag‘ishlangan. Ushbu bobda jarayonni bajarilish vaqtida asosiy xotirani boshqarish masalalari ko‘rib chiqilgan. Protssorni ishlatish va foydalanuvchilarga javob berish tezligini oshirish uchun kompyuter bir necha jarayonlarni xotirada saqlashi kerak. Xotirani boshqarishning turli xil yondashuvlarini aks ettiruvchi juda ko‘p turli xil xotirani boshqarish sxemalari ko‘rib chiqilgan.

To‘rtinchi bobda kompyuter tashqi qurilmalarini boshqarishga doir operatsion tizimlarning funksiyalari ko‘rib o‘tilgan. Ushbu bobda zamonaviy kompyuter tizimida ma’lumotlarni saqlash qurilmalari va kiritish/chiqarishni qanday ishlashini tavsiflaydi. Kompyuterga ulanadigan kiritish/chiqarish qurilmalari juda xilma-xil bo‘lgani sababli, operatsion tizim ushbu qurilmalarning barcha jihatlarini boshqarishga imkon beradigan keng funksiyalarga ega ilovalarni taqdim etishi kerak. Kiritish/chiqarish tizimini batafsil ko‘rib chiqamiz, jumladan, kiritish/chiqarish tizimining dizayni, interfeyslari,

shuningdek tizimning ichki tuzilishi va funksiyalarini ham o'rganamiz.

Beshinchi bob fayl tizimi va fayllarni boshqarish jarayoniga bag'ishlanadi. Fayl tizimi operatsion tizimning fayllarni boshqarish uchun javobgar bo'lgan qismidir. Fayl tizimi ma'lumotlarni saqlash, ma'lumotlar va dasturlarga kirish mexanizmini ta'minlaydi. Ushbu bobda operatsion tizimlarda fayl tushunchasi, fayl tizimlari, tuzilishi va tashkil etilishi, fayllarni fizik tashkil etish, ularni o'qish va yozish usullari, fayl turlari va atributlari, fayllarni boshqarish va ular ustida amallar bajarish, fayllarni himoyalash va ularga murojaatlarni boshqarish masalalari ko'rib chiqilgan.

Oltinchi bobda kompyuter tarmoqlarini qurish va boshqarish masalalari ko'rib o'tilgan. Kompyuter tarmoqlari zamonaviy AKT sohasida keng tarqalgan. Kompyuter tarmoqlari bu ma'lumotlar bazalari, veb va mobil ilovalar, taqsimlangan tizimlar va boshqa ko'plab dasturlarning asosidir. Tarmoqning asosiy tamoyillarini yaxshi tushunish yanada ishonchli va samarali dasturiy mahsulotlarni yaratishga yordam beradi. Ushbu bobda kompyuter tarmoqlari, ularning turlari, asosiy tushunchalari, kompyuter tarmoqlarining tuzilishi va ularni qurish topologiyalari, ma'lumotlar kommutatsiyasi va marshrutlash mexanizmi, kompyuter tarmoqlari arxitekturasi va protokollari, operatsion tizimning tarmoq funksiyalari va turlari masalalari ko'rib chiqilgan.

Yettinchi bob operatsion tizimlarning tarmoq xizmatlarini o'rganishga bag'ishlangan. Kompyuter tarmoqlarida tarmoq xizmati - bu tarmoq ilovalari darajasida va undan yuqori darajada ishlaydigan dastur bo'lib, u ma'lumotlarni saqlash, boshqarish, taqdim etish, aloqa yoki boshqa imkoniyatlarni ta'minlaydi, ular ko'pincha dastur darajasidagi tarmoq protokollariga asoslangan mijoz-server yoki "peer-to-peer" arxitekturasidan foydalanib amalga oshiriladi. Ushbu bobda mijoz-server arxitekturasi tuzilishi va asosiy tushunchalari, tarmoq operatsion tizimlari va ularning asosiy tarmoq xizmatlari, tarmoq kompyuterlarini tashkil etish va boshqarish mexanizmlari, tarmoq fayllarini tashkil etish va boshqarish mexanizmlari, domen tizimi va internet xizmatlarini tashkil etish masalalari ko'rib chiqilgan.

Sakkizinchi bobda operatsion tizimlarda xavfsizlik masalalari ko'rib o'tilgan. Ushbu bobda kompyuter tizimlarining himoyasi va xavfsizligi uchun zarur bo'lgan mexanizmlar muhokama qilinadi.

Operatsion tizimdagi jarayonlar bir-birining harakatlaridan himoyalangan bo'lishi kerak va bunday himoyani ta'minlash uchun biz faqat operatsion tizimdan tegishli avtorizatsiyani olgan jarayonlar fayllar, xotira, protsessor va boshqa tizim manbalari bilan ishlashini ta'minlashimiz kerak. Himoya bu dasturlar, jarayonlar yoki foydalanuvchilarning kompyuter tizimining resurslariga kirishini boshqarish mexanizmidir. Ushbu mexanizm kiritilishi kerak bo'lgan boshqarish vositalarini, shuningdek, bajarish vositalarini aniqlashni ta'minlashi kerak. Xavfsizlik tizimda saqlanadigan ma'lumotlarning yaxlitligini (ma'lumotlar va kodlar), shuningdek tizimning fizik resurslariga ruxsatsiz kirishdan, ma'lumotlarni yo'q qilish yoki o'zgartirishdan himoya qiladi.

To'qqizinchi bob amaliyotda operatsion tizimlardan foydalanish haqidagi bilimlarni o'z ichiga oladi. Ushbu bobda biz Linux, Windows va Android operatsion tizimlarini ko'rib chiqamiz. Linux va Windows operatsion tizimlari tuzilishi, ularning imkoniyatlari va ularda jarayonlarni boshqarish, xotirani boshqarish, qurilmalarni boshqarish, fayllarni boshqarish, ularda qo'llaniladigan fayl tizimlari va ularning imkoniyatlari, fayllarga kirish usullari va ularda xavfsizlikni boshqarish, shuningdek, Android operatsion tizimi tuzilishi, uning qo'llanilish sohasi va imkoniyatlari, fayl tizimi tuzilishi va Android Debug Bridge dan foydalanish masalalari muhokama qilingan.

## **IBOB. OPERATSION TIZIMLAR VA ULARNING AHAMIYATI**

### **1.1. Operatsion tizim tushunchasi va ularning ahamiyati**

Ushbu kursda operatsion tizim va ularning asosiy funksiyalari protsessorni boshqarish, jarayonlarni boshqarish, xotirani boshqarish, qurilmalarni boshqarish, fayllarni boshqarish, tarmoqni boshqarish, operatsion tizimlar xavfsizligi va xavfsizlikni boshqarish masalalari, Unix, Linux, Windows va Android operatsion tizimlari, shuningdek, tarmoq xizmatlari va ularning turlari o'rganiladi. Zamonaviy kompyuter bir yoki bir nechta protsessorlardan, operativ xotiradan, disklardan, printerdan, klaviaturadan, sichqonchadan, displaydan, tarmoq interfeyslaridan va boshqa kiritish/chiqarish qurilmalaridan iborat. Natijada ancha murakkab tizim kelib chiqadi. Bundan tashqari, ushbu komponentlarning barchasini boshqarish va ulardan maqbul foydalanish juda qiyin vazifadir. Shu sababli, kompyuterlar operatsion tizim deb ataladigan maxsus darajadagi dasturiy ta'minot bilan jihozlangan bo'lib, ularning vazifasi foydalanuvchi dasturlarini, shuningdek ilgari aytib o'tilgan barcha resurslarni boshqarishdir.

Operatsion tizim (OT, ingliz tilidagi shakli - **operating system**) qurilmalar (**hardware**), amaliy dasturlar (**application software**) va foydalanuvchi (**user**) kompyuteri o'rtasida vositachilik (**interface**) qiladigan va kompyuter resurslarini boshqarish va foydalanuvchilarning o'zaro munosabatlarini tashkil qilish uchun mo'ljallangan bir-biriga bog'langan dasturlar to'plamidir. Foydalanuvchi nuqtai nazaridan qaralganda operatsion tizim – bu qurilmalarning davomi bo'lib kompyuter va kompyuter tarmoqlari uchun qulay, ishonchli va xavfsiz foydalanishni ta'minlaydigan va qurilmaga o'rnatiladigan dasturiy vosita hisoblanadi.

Operatsion tizimda ishlashdan asosiy maqsad:

❖ foydalanuvchi dasturlarini bajarishda qulaylik, samaradorlik, ishonchlilik, xavfsizlikni ta'minlash. Foydalanuvchi birinchi navbatda – foydalanuvchi xoxlagandagidek uni qanoatlantiridigan darajada ishlashi uchun unga so'rov natijalarini to'g'ri taqdim etishi, rad etishlar bo'lmasligi, tashqi xujumlardan himoyalangan bo'lishi kerak.



Buni bajarish uchun hisoblash vositalari dasturni ishlata olishi va operatsion tizim bilan ta'minlangan bo'lishi kerak.

❖ kompyuterdan foydalanishda qulaylik, samaradorlik, ishonchlilik, xavfsizlikni ta'minlash. Operatsion tizim kompyuter va uning resurslaridan foydalanishda maksimal foydalilik va samaradorlikni ta'minlashi, uzilishlarni qayta ishlash, rad etilishlar va buzg'unchilar hujumidan kompyuterni himoyalashi kerak. OTning bu ishi foydalanuvchi uchun ko'rinmasligi mumkin, lekin u doimiy ravishda mavjud bo'ladi.

❖ kompyuterga ulangan tarmoq, disk va boshqa tashqi qurilmalardan foydalanishda qulaylik, samaradorlik, ishonchlilik, xavfsizlikni ta'minlash. Operatsion tizimning asosiy funksiyasi – bu tashqi qurilmalar bilan ishlash. Masalan, OT qattiq diskga istalgan murojaatni qayta ishlashi, drayver (disk bilan axborotlarni almashish uchun past darajali dastur) va kontrollerlarning (disk bilan kiritish/chiqarishni bajaruvchi) birgalikda ishlashini ta'minlashi kerak. Kompyuter USB – portiga ulangan istalgan “fleshka” operatsion tizimda aniqlanishi, o'zining mantiqiy nomini olishi va u chiqarilmagunga qadar butun vaqt davomida kompyuter tizimi fayl qismi bo'lib qolishi mumkin.

❖ bugungi kundagi OT larni muhim bo'lgan xususiyatlari, ya'ni xavfsizlik, ishonchlilik va ma'lumotlar himoyalanganlikni ta'minlash. Tarmoq doirasida kompyuter va operatsion tizim bilan ishlashda ishtirok etuvchilar unga va uning dasturlariga buzg'unchilar tomonidan bo'ladigan xujumlarga (reklama yoki virus va boshqa ko'rinishlarda kompyuterlarga kirib, foydalanuvchi maxfiy saqlaydigan login, parollarni o'g'irlash, kompyuter ish foaliyatini buzish va shunga o'xshash boshqa xodisalar) doimiy ravishda uchraydilar. Ushbu kursda biz bugungi kundagi OTda qabul qilingan ishonchlilik, xavfsizlik va ma'lumotlarni himoyalashni ta'minlash usullarini ham ko'rib chiqamiz.

Operatsion tizim quyidagi sifatlarga ega bo'lishi kerak:

**1. Ishonchlilik.** Tizim o'zi boshqarayotgan kompyuter qurilmalari kabi ishonchli bo'lishi kerak. Agar dasturda yoki qurilmada biror xatolik uchrasa, uni tizim topa olishi va bu holatni tuzatishga harakat qilishi, hech bo'lmasa shu xatolik tufayli foydalanuvchi dasturiga yetkaziladigan zararning oldini olishi kerak.

**2. Himoyalash.** Ixtiyoriy foydalanuvchi o‘z ishiga boshqa foydalanuvchi-larning ta’sir qilishini hohlamaydi. Shu sababli tizim foydalanuvchilarni dastur va ma’lumotlarini boshqa shaxslar xatoliklari ta’siridan hamda aralashuvidan himoya qilishi lozim.

**3. Samaradorlik.** Odatda operatsion tizimning o‘zi EHM ning katta resursini egallaydi. Bu resurslar foydalanuvchi ixtiyoriga berilmaydi. Demak, tizimning o‘zi ancha ixcham bo‘lishi va EHM ning resurslarini samarali boshqarishi lozim.

**4. Qulaylik.** Operatsion tizimda ko‘p hollarda bir paytda ikki va undan ortiq foydalanuvchi ishlaydi. Ular operatsion tizim orqali turli maqsadli va turli algoritimli masalalarni hal qiladilar. Ma’lumki, bunday holda har bir foydalanuvchiga keng qulayliklar yaratilishi talab etiladi. Shu bois, mazkur funktsiya operatsion tizimning muhim xususiyati hisoblanadi.

**5. Bashorat.** Operatsion tizim foydalanuvchi so‘roviga bashoratchilik bilan javob berishi kerak. Foydalanuvchi buyruqlari tizimda qabul qilingan qoidalar asosida yozilgan bo‘lsa, ularning ketma-ketligi qanday bo‘lishidan qat’iy nazar natija bir xil bo‘lishi kerak.

**6. Moslashuvchanlik.** Tizim amallari foydalanuvchiga qarab sozlanishi mumkin. Resurslar majmuasi operatsion tizim samaradorligini oshirish maqsadida ko‘paytirilishi yoki kamaytirilishi mumkin.

**7. Kengayuvchanlik.** Evolyutsiya jarayonida operatsion tizimga yangi texnik va dasturiy resurslar qo‘shilishi mumkin.

**8. Aniqlik.** Foydalanuvchi tizim interfeys darajasidan pastda sodir bo‘ladigan jarayondan bexabar qolishi mumkin. Shu bilan birga foydalanuvchi tizim haqida qancha bilgisi kelsa, shuncha bilish imkoniyatiga ega bo‘lishi kerak.

## **1.2. Operatsion tizimlarning rivojlanish davrlari**

### ***Birinchi davr (1945-1955 yillar)***

Hammaga ma’lumki, kompyuter ingliz matematigi Charlz Bebbij tomonidan XVIII asr oxirida kashf etildi. Uning “analitik mashina”si haqiqatda ishlay olmadi, chunki u vaqtdagi texnologiyalar hisoblash texnikasi uchun zarur bo‘lgan aniq mexanika qismlarini tayyorlash bo‘yicha zarur talablarni qondiradigan texnologiyalar

mavjud bo'lmagan. Yana eng asosiy narsa, u vaqtda kompyuter operatsion tizimga ega bo'lmagan.

Raqamli hisoblash mashinalarini yaratishda, ikkinchi jahon urushidan keyin ma'lum taraqqiyot-rivojlanish yuz berdi. 1940yillar o'rtalarida birinchi lampali mashinalar yaratildi. U vaqtda ayni bir guruh mutaxassislar hisoblash mashinalarini ham loyihalashda, ham ekspluatatsiya qilishda va dasturlashda ham shu guruh mutaxassislari ishtirok etganlar. Bu jarayon ko'proq, kompyuterdan asbob-uskuna sifatida turli amaliy sohalar masalalarini yechishda foydalanish emas, balki hisoblash texnikasi sohasidagi ilmiy-tadqiqot ishiga yaqinroq edi.

Dasturlash faqat mashina tilida amalga oshirilar edi. OT to'g'risida gap ham yo'q edi, chunki hisoblash jarayoni tashkil qilish masalalari, har bir dasturchi tomonidan boshqaruv pulti orqali "qo'lda" yechilar edi.

Pult oldida faqat bitta foydalanuvchi o'tirishi mumkin edi. Dastur mashina xotirasiga eng yaxshi holatda perfokarta kolodasidan kiritilar edi, odatda esa o'tkazish paneli yordamida yuklanar edi.

Hisoblash tizimi bir vaqtning o'zida faqat bitta operatsiyani (kiritish/chiqarish yoki hisoblashlarni) bajarar edi. Dasturni sozlash boshqarish panelidan xotira va mashina registri holatini o'rganish yordamida olib borilar edi. Bu davr oxirida birinchi tizimli dasturiy ta'minot yuzaga keldi. 1951-1952 yillar belgili tillar (Fortran va boshqa) dan birinchi kompilyatorlar versiyalari yuzaga keldi, 1954yilda esa IBM-701 uchun Assembler tili ishlab chiqildi.

Vaqtning eng ko'p qismi dasturni ishga tushirishga ketib qoldi, dasturlarning o'zi esa qat'iy ravishda ketma-ket ishlov berish rejimi deb ataldi. Xulosa qilib aytganda, birinchi davr, hisoblash tizimlarining yuqori narxi, ularning soni kamligi va foydalanishning past samarasi bilan belgilandi.

### ***Ikkinchi davr (1955-1965 yillar)***

1950yil o'rtalariga kelib, hammaga ma'lumki yangi texnik baza yarim o'tkazgich elementlarni yuzaga kelishi bilan, hisoblash texnikasi rivojlanishida yangi davr boshlandi. Ikkinchi avlod kompyuterlari ishonchliroq bo'lib qoldi, chunki ular amaliy muhim masalalarni bajarish darajasida uzluksiz ravishda uzoq ishlay oladigan imkoniyatga ega bo'ldilar. Aynan shu davrda hisoblash texnikasi bilan ishlaydigan mutaxassislar, dasturchilar, operatorlar,

ekspluatatsiyachilar va hisoblash mashinasini ishlab chiqaruvchilarga ajraldilar.

Shu yillarda birinchi alogritmik tillar yuzaga keldi va natijada birinchi tizimli dasturlar kompilyatorlar ham yaratildi. Protsessor vaqti qiymati (narxi) oshdi, bu esa dasturlar orasidagi vaqtni qisqartirishni talab qildi.

Birinchi paketli ishlov berish tizimlari yuzaga keldi, bu tizimlarda dasturlarni ishga tushirish ketma-ketligini avtomatlashtirildi va shu bilan birga protsessor yuklanish koeffitsienti oshdi. Paketli ishlov berish tizimlarini zamonaviy OT larining birinchi variantlari deyish mumkin, chunki ular hisoblash tizimini boshqarishga mo'ljallangan birinchi tizimli dasturlar edi.

Paketli ishlov berish tizimlarini amalga oshirishda, topshiriqlarni boshqarishning formallashtirilgan tili ishlab chiqildi, uning yordamida dasturchi tizimga va operatorga hisoblash mashinasida qaysi ishni bajarmoqchi ekanligi haqida ma'lumot beradi. Bir nechta topshiriqlar majmuasi, qoida bo'yicha perfokartalar "koloda"si ko'rinishida bo'lib, topshiriqlar paketi nomini oldi.

### ***Uchinchi davr (1965-1980 yillar)***

Hisoblash mashinalari rivojlanishida keyingi muhim davri shu yillarga to'g'ri keladi. Bu vaqtda, texnik bazada quyidagi o'zgarishlar yuz berdi: alohida yarim o'tkazgichli elementlardan (tranzistor turidagi) integral mikrosxemalarga o'tildi, bu esa yangi uchinchi avlodga, yangi imkoniyatlar yaratdi. Bu davrning o'ziga xos xususiyatlaridan biri, integral mikrosxemalarda yaratilgan birinchi dasturiy-mutanosib mashinalardir, ya'ni IBM/360 mashinalari seriyasidir. 1960yillar boshida yaratilgan bu mashinalar oilasi ikkinchi avlod mashinalaridan baho/unumdorlik ko'rsatkichi bo'yicha oldinga anchagina o'tib ketdi. Tezda, dasturiy-mutanosib mashinalar g'oyasi tan olindi.

Dasturiy mutanosiblik OT larni ham mutanosibligini talab qildi. Bunday operatsion tizimlar ham katta EHM da ham, kichik hisoblash tizimlarida ham, turli yordamchi (periferik) qurilmalarning kam soni va ko'p soni bilan ham, tijorat sohasida ham, ilmiy-tadqiqot sohasida ham ishlay olishi kerak.

Shunday hamma qarama-qarshi talablarni qondiradigan asosda quriladigan operatsion tizimlar juda murakkab "monstr"lar bo'lib

chiqdi. Ular ko‘p millionli assembler qatorlaridan iborat bo‘lib, minglab dasturchilar tomonidan yozilgan bo‘lib, minglab xatolarni o‘z ichiga oladi, ular minglab tuzatishlarga olib keladi. Operatsion tizimning har bir yangi versiyasida biror xatolar tuzatilib, yangisi yuzaga keldi. Ko‘pgina muammolar va juda katta o‘lchamga qaramasdan OS/360 va unga o‘xshash 3-avlod operatsion tizimlari haqiqatdan ham iste‘molchilarning ko‘pgina talablarini qondirdilar. Bu avlodning eng katta erishgan yutuqlaridan bir multidasturlashni amalga oshirishdir.

Multidasturlash – bu hisoblash jarayonning tashkil qilish usuli bo‘lib, bitta protsessorida navbat bilan bir nechta dastur bajariladi.

Bitta dastur kiritish/chiqarishni amalga oshirguncha keyingi dasturlarni oldingi ketma-ket bajarilishdagi kabi (bir dasturli rejim) protsessor to‘xtab turmaydi, balki boshqa dasturni bajaradi (ko‘p dasturli rejim). Bunda har bir dastur operativ xotiradagi bo‘lim deb ataluvchi o‘z qismiga yuklanadi.

Boshqa yangilik – buferlash (spooling) deb ataladi. Buferlash u vaqtda hisoblash jarayonini tashkil etish usullaridan biri bo‘lib, unga mos ravishda topshiriq perfokartadan diskga hisoblash markazida paydo bo‘lish tartibida yoziladi, keyin esa navbatdagi topshiriq tugallanishi bilan, yangi topshiriq diskdan bo‘shagan bo‘limga yuklanadi.

Paketli ishlov berishni multidasturlashli amalga oshirish bilan birga, OTlarning yangi turi – vaqtni taqsimlash tizimlari yuzaga keldi. Vaqtni taqsimlash tizimlarida qo‘llaniladigan multidasturlash varianti, har bir foydalanuvchi uchun hisoblash mashinasidan yagona foydalanish tasavvurini hosil qilishga imkon beradi.

Multidasturlashni yuzaga kelishi hisoblash tizimi tuzilishiga chuqur o‘zgartirishlar kiritishni talab qiladi. Bunda asosiy rol ni apparat tomonidan qo‘llanish o‘ynaydi, uning asosiy xususiyatlari quyida keltirilgan:

Multidasturlashning xususiyatlari

❖ Himoya mexanizmini amalga oshirish. Dasturlar mustaqil ravishda resurslarni taqsimlash imkoniga ega bo‘lishi kerak emas, bu imtiyozli va imtiyozsiz buyruqlarni keltirib chiqardi. Imtiyozli buyruqlar OT tomonidan bajariladi.

❖ Uzilishlar mavjudligi. Tashqi uzilishlar OT ni asinxron hodisa, masalan kiritish/chiqarish operatsiyasi tugallanganligi haqida

ogoxlantiradi. Ichki uzilish, OT aralashuvi zarur bo'lganda yuz beradi, masalan himoyani buzishga bo'lgan harakat.

❖ Arxitekturada parallellashni rivojlantirish. Xotiraga bevosita murojaat va kiritish/chiqarish kanalini tashkil etish, markaziy protsessorni qiyin operatsiyalarni bajarishdan xalos etadi.

Albatta, multidasturlashni tashkil etishda OT roli juda muhimdir. U quyidagi operatsiyalar uchun javob beradi:

❖ Tizimli chaqiriqlar yordamida OT va amaliy dasturlar orasida interfeysni tashkil etish;

❖ Xotiradagi topshiriqlardan navbat tashkil etish va topshiriq uchun protsessorni ajratish uchun protsessordan foydalanishni rejalashtirish;

❖ Bir topshiriqdan ikkinchisiga o'tish, hisoblashlarni to'g'ri tashkil etish uchun kontekstni saqlash;

❖ Xotira chegaralangan resurs bo'lganligi uchun, xotirani boshqarish algoritmi zarur, ya'ni xotiradan ma'lumotlarni olish, joylashtirish va almashtirish jarayonlarini tartibga solish talab qilinadi;

❖ Ma'lumotlarni tashqi qurilmalarda fayl ko'rinishida saqlashni va ma'lum fayllardan faqat aniq foydalanuvchilar foydalana olishini tashkil etish;

❖ Dasturlarga ruhsatli (sanksiyali) ma'lumot almashish talab etilgani uchun, ularni kommunikatsiya vositalari bilan ta'minlash zarur;

❖ Ma'lumotlarni to'g'ri taqsimlash uchun, ziddiyatli holatlarni yechishga to'g'ri keladi, bu ko'pincha turli resurslar bilan ishlashda ro'y beradi, shuning uchun harakatlarni dasturlar bilan sinxronlashtirish talab qilinadi.

Vaqtning taqsimlash tizimlarida foydalanuvchi, dasturni interaktiv rejimda sozlash imkoniga ega bo'ldi, bunda u ma'lumotli diskga perfokarta orqali emas, bevosita klaviaturadan kiritishi mumkin bo'ldi. On-line fayllarni yuzaga kelishi rivojlangan fayl tizimlarini ishlab chiqish zaruriyatini keltirib chiqardi.

### ***To'rtinchi davr (1980 yildan – hozirgi vaqtgacha)***

Operatsion tizimlar rivojlanishidagi keyingi davr katta integral sxemalarni (KIS) yuzaga kelishi bilan bog'liq bo'lgan davrdir. Bu yillarda integratsiya darajasi keskin o'sishi va mikrosxemalar arzonlashishi yuz berdi. Kompyuterdan mutaxassis bo'lmagan

foydalanuvchilar foydalanishi imkoni yuzaga keldi, va shaxsiy kompyuterlar davri boshlandi.

Arxitektura jihatidan, shaxsiy kompyuterlar, minikompyuterlar turlari sinflaridan hech narsasi bilan farq qilmas edi, faqat ularning narxlarida farq bo'ldi. Minikompyuterlar korxonalar va universitetning bo'limlariga shaxsiy hisoblash markaziga ega bo'lishiga imkon bergan bo'lsa, shaxsiy kompyuter esa bunday imkoniyatni alohida inson uchun yaratdi.

Kompyuterlardan hisoblash texnikasi sohasida mutaxassis bo'lmaganlar ham keng ko'lamda foydalana boshladilar, bu esa o'z navbatida "do'st" dasturiy ta'minotni yaratishni talab etdi, bu dasturchilarni alohida o'rnidan qo'zg'atdi.

Operatsion tizimlar bozorida ikkita tizim ustunlik qila boshladilar: MS-DOS va UNIX OT lari. Bir foydalanuvchili MS-DOS OT lari Intel 8088 asosida qurilgan mikroprotessorlar, va keyin 80286, 80386 va 80486 asosida qurilgan kompyuterlarda foydalanildi.

Multidasturli, ko'p foydalanuvchili UNIX operatsion tizimi Intel bo'lmagan kompyuterlar muhitida ustunlik qila boshladi, ayniqsa yuqori unumdorlikka ega bo'lgan RISC-protessorlar uchun.

Tarmoq OTlarida, foydalanuvchi tarmoqda boshqa kompyuterlar mavjudligi haqida bilishlari va boshqa kompyuterga uning resurslaridan, asosan fayllaridan foydalanish uchun boshqa kompyuterga mantiqan kirishlari kerak edi.

Tarmoqdagi har bir mashina, kompyuterning avtonom operatsion tizimidan tarmoqda ishlashga imkon beradigan qo'shimcha vositalarga ega bo'lgan lokal operatsion tizimi vazifasini bajaradi.

Tarmoq operatsion tizimi, bir protessorli kompyuter operatsion tizimidan asosli farq qilmaydi. Ularning tarkibida, albatta, tarmoq interfeysini qo'llovchi (tarmoq adapteri drayveri) va shu bilan birga tarmoqdagi boshqa kompyuterlarga masofadagi kirish vositalari va masofadagi fayllarga murojaat vositalari mavjuddir, ammo bu qo'shimchalar operatsion tizimni tuzilishini tubdan o'zgartirmaydi.

Hisoblash tizimlarini rivojlanishi bosqichlarini ko'rib chiqib, biz rivojlanish jarayonida klassik OT lar bajargan 6 ta asosiy funksiyalarni ajratishimiz mumkin:

❖ vazifalarni (yoki topshiriq) rejalashtirish va protsessordan foydalanish;

❖ dasturlarni kommunikatsiya va sinxronizatsiya vositalari bilan ta'minlash;

❖ xotirani boshqarish;

❖ fayl tizimini boshqarish;

❖ kiritish/chiqarishni boshqarish;

❖ xavfsizlikni ta'minlash.

Har bir keltirilgan funksiyalar odatda OT tarkibidagi komponentalaridan biri sifatida amalga oshirilgan. Ular boshidanoq, OT komponenti sifatida yaratilgan emas, ular rivojlanish jarayonida yuzaga keldi. Inson yaratgan hisoblash tizimi rivojlanishi (evolyutsiyasi) shu yo'ldan ketdi, ammo hech kim bu yo'l rivojlanishning yagona mumkin bo'lgan yo'li deb isbot qila olmaydi.

### **1.3. Operatsion tizimlarning tuzilishi va asosiy komponentlari**

Operatsion tizim dasturlar bajariladigan muhitni ta'minlaydi. Ichki tomondan, operatsion tizimlar turli xil yo'nalishlarda tashkil etilganligi sababli, ularning tuzilishida juda katta farq bor. Yangi operatsion tizimning dizaynini yaratish katta vazifadir. Dizaynni boshlashdan oldin tizimning maqsadlari aniq belgilanishi juda muhimdir. Ushbu maqsadlar turli xil algoritmlar va strategiyalar orasida tanlov uchun asos bo'lib xizmat qiladi. U dasturlarga va dasturlardan foydalanuvchilarga ma'lum xizmatlarni taqdim etadi. Taqdim etilayotgan o'ziga xos xizmatlar, albatta, bitta operatsion tizimda boshqa operatsion tizimlarga qaraganda farq qiladi, ammo biz umumiy sinflarni aniqlashimiz mumkin. Ushbu operatsion tizim xizmatlari dasturchiga qulaylik yaratish, dasturlash vazifasini osonlashtirish uchun taqdim etiladi.

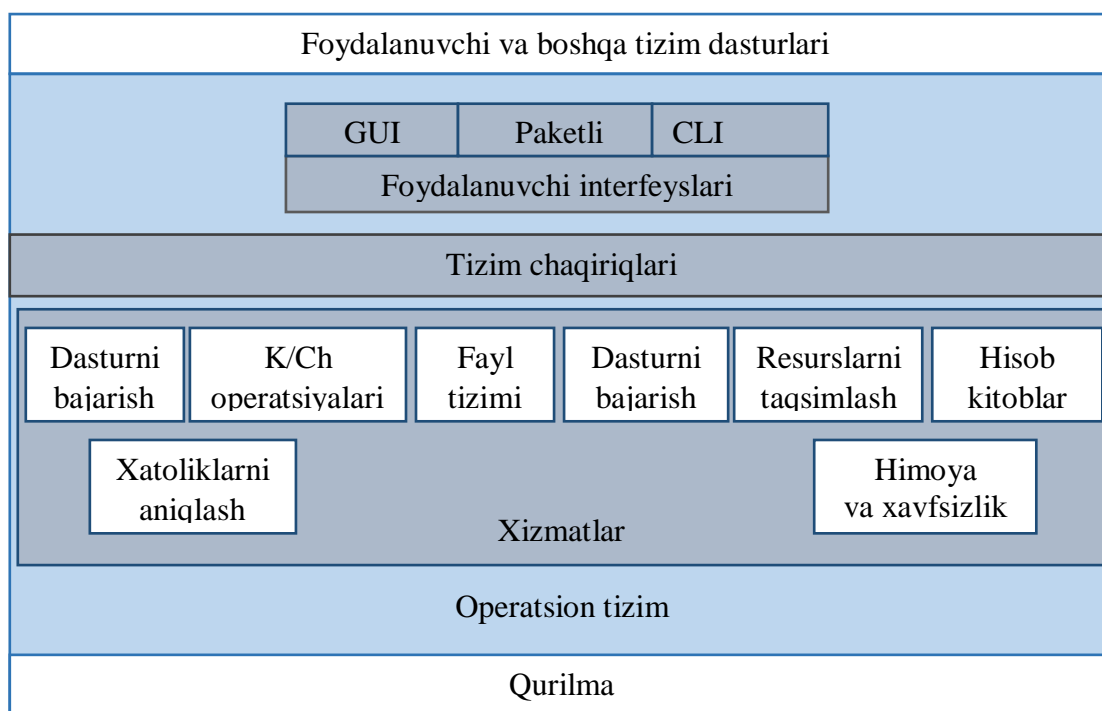
1.1- rasmda turli xil operatsion tizimlar xizmatlarining ko'rinishi va ularning o'zaro bog'liqligi ko'rsatilgan.

Operatsion tizim xizmatlarining to'plami foydalanuvchi uchun foydali bo'lgan xususiyatlarni taqdim etadi.

**Foydalanuvchi interfeysi** (user interface). Deyarli barcha operatsion tizimlarda foydalanuvchi interfeysi mavjud. Ushbu interfeys bir nechta shakllarni olishi mumkin. Ulardan biri bu **buyruqlar qatori interfeysi** (Command Line Interface - CLI) bo'lib, unda matnli buyruqlar va ularni kiritish usuli qo'llaniladi (aytaylik, ma'lum parametrlarga ega buyruqlarni ma'lum formatda terish uchun



klaviatura). Ikkinchisi - **paketli interfeys** bo'lib, unda buyruqlar va ko'rsatmalar fayllarga kiritiladi va ushbu fayllar bajariladi. Ko'pincha **foydalanuvchi grafik interfeysi** (Graphic User Interface - GUI) ishlatiladi. Bu yerda interfeys kiritish/chiqarishni yo'naltirish, menyudan tanlash, klaviaturadan matnni kiritish va klaviaturani yo'naltirish moslamasi bo'lgan oyna tizimi.



1.1- rasm. Operatsion tizim xizmatlarining ko'rinishi

**Dasturning bajarilishi** (program execution). Tizim dasturni xotiraga yuklashi va ushbu dasturni ishga tushirishi kerak. Dastur odatiy yoki noodatiy ravishda (xatolik natijasida) o'z ishini yakunlashi kerak.

**Kiritish/chiqarish (K/Ch) operatsiyalari** (I/O operations). Bajarilayotgan dastur K/Ch talab qilishi mumkin, bu fayl yoki K/Ch qurilmalarini o'z ichiga olishi mumkin. Muayyan qurilmalar uchun maxsus funksiyalar talab qilinishi mumkin (masalan, CD yoki DVD kabi disklarga yozish yoki displey ekranni o'chirish). Samaradorlik va himoya qilish uchun foydalanuvchilar odatda kiritish/chiqarish qurilmalarini to'g'ridan-to'g'ri nazorat qila olmaydilar. Shuning uchun operatsion tizim kiritish/chiqarishni bajarish uchun vositalarni taqdim etishi kerak.

**Fayl tizimi bilan ishlash** (File-system manipulation). Fayl tizimi alohida qiziqish uyg'otadi. Shubhasiz, dasturlar fayllar va kataloglarni o'qish va yozishni talab qiladi. Shuningdek, ularni nomlari bo'yicha yaratishi va yo'q qilishi, berilgan faylni qidirish va fayl ma'lumotlarini ro'yxatga olishi kerak. Va nihoyat, ba'zi operatsion tizimlar fayllarga asoslangan fayllar yoki kataloglarga kirishni taqiqlash uchun ruxsatlarni (yoki kirishni) boshqarishni o'z ichiga oladi. Ko'pgina operatsion tizimlar ba'zida shaxsiy tanlovga, ba'zan esa o'ziga xos xususiyatlarga ega bo'lish uchun turli xil fayl tizimlarini taqdim etadi.

**Kommunikatsiya** (aloqa) (Communication). Bitta jarayon boshqa jarayon bilan ma'lumot almashishi kerak bo'lgan ko'plab holatlar mavjud. Bunday aloqa bitta kompyuterda ishlaydigan jarayonlar yoki kompyuter tarmog'iga ulangan turli xil kompyuter tizimlarida ishlaydigan jarayonlar o'rtasida sodir bo'lishi mumkin. Aloqa umumiy xotira orqali amalga oshirilishi mumkin, bunda ikki yoki undan ko'p jarayonlar o'qiladi va umumiy xotira bo'limiga yoziladi yoki xabarlarni yuborish orqali amalga oshiriladi, bunda oldindan belgilangan formatdagi ma'lumot paketlari operatsion tizim tomonidan jarayonlar o'rtasida uzatiladi.

**Xatoliklarni aniqlash** (error detection). Operatsion tizim doimiy ravishda xatoliklarni aniqlashi va tuzatishi kerak. Xatoliklar protsessor va xotira qurilmalarida (masalan, xotiradagi xatolik yoki elektr ta'minotidagi uzilishlar), kiritish/chiqarish qurilmalarida (masalan, diskdagi xatolik, tarmoqqa ulanishda xatolik yoki printerda qog'ozni mavjud emasligi) va foydalanuvchi dasturida (masalan, arifmetik to'lish, mavjud bo'lmagan xotira maydoniga kirishga urinish yoki juda ko'p protsessor vaqtidan foydalanish). Xatoliklarning har bir turi uchun operatsion tizim to'g'ri va kelishilgan hisoblashni ta'minlash uchun tegishli choralarni ko'rishi kerak. Ba'zida uni tizimni to'xtatishdan boshqa chorasi bo'lmaydi. Boshqa holatlarda, u xatolik keltirib chiqaradigan jarayonni to'xtatishi yoki aniqlashi va tuzatishi mumkin bo'lgan xatolik kodini jarayonga qaytarishi mumkin. Operatsion tizim funksiyalarining yana bir to'plami, foydalanuvchiga yordam berish uchun emas, balki tizimning o'zi samarali ishlashini ta'minlash uchun kerak. Ko'p foydalanuvchilarga ega tizimlar foydalanuvchilar orasida kompyuter resurslarini almashish orqali samaraga erishishlari mumkin.

**Resurslarni taqsimlash** (resource allocation). Bir vaqtning o'zida bir nechta foydalanuvchi yoki bir nechta vazifalar bajarilganda, ularning har biri uchun resurslar ajratilishi kerak. Operatsion tizim ko'plab turli xil resurslarni boshqaradi. Ba'zilarida (masalan, protsessor sikllari, asosiy xotira va fayllarni saqlash) maxsus taqsimlangan kodi bo'lishi mumkin, boshqalarida (kiritish/chiqarish qurilmalari kabi) umumiy so'rov va ozod qilish kodi bo'lishi mumkin. Masalan, protsessorni qanday qilib yaxshiroq ishlatish mumkinligini aniqlashda, operatsion tizimlar protsessorni rejalashtirishda protsessor tezligini, bajarilishi kerak bo'lgan vazifalarni, mavjud registrlar sonini va boshqa omillarni hisobga oladi. Shuningdek, printerlarni, USB qurilmalarni va boshqa yordamchi qurilmalarni taqsimlashning tartib-qoidalari mavjud bo'lishi mumkin.

**Hisoblash** (accounting). Biz qaysi foydalanuvchilar qancha va qanday turdagi kompyuter resurslaridan foydalanishini kuzatishni istaymiz. Hisobni yuritish buxgalteriya hisobida (foydalanuvchilar hisob-kitoblarni olishlari uchun) yoki foydalanish statistikasini to'plash uchun ishlatilishi mumkin.

**Xavfsizlik va himoya** (Protection and security). Ko'p foydalanuvchili yoki tarmoqqa ulangan kompyuter tizimida saqlanadigan ma'lumotlarning egalari ushbu ma'lumotlardan foydalanishni va boshqarishni xohlaydilar. Bir vaqtning o'zida bir nechta alohida jarayonlar sodir bo'lganda, bitta jarayon boshqa jarayonga yoki operatsion tizimning o'ziga xalaqit bermasligi kerak. Himoyalash – tizim resurslariga barcha kirishlar boshqarilishini ta'minlashni o'z ichiga oladi. Tizimni begona odamlardan himoyalash ham muhim. Bunday xavfsizlik har bir foydalanuvchidan tizim resurslariga kirish huquqini olish uchun, odatda, parol yordamida tizimda autentifikatsiya (o'zini haqiqiylikini tasdiqlashni) qilishni talab qilishdan boshlanadi.

#### **1.4. Kompyuter tizimlari, ularning tuzilishi va ahamiyati**

Hisoblash jarayonida operatsion tizimlarining vazifasini yaxshiroq tushunish uchun kompyuter tizimini ko'rib chiqamiz. U quyidagi komponentlardan tashkil topgan:

❖ Kompyuter qurilmalari (**hardware**), uning asosiy qismi – markaziy protsessor (**Central Processor Unit - CPU**) – kompyuter

buyruqlarini bajarish yo‘riqnomasi, dasturlar va ma’lumotlar saqlanadigan xotira, kompyuterga axborotlarni kiritishni ta’minlovchi va foydalanuvchi – inson qabul qilgan yoki boshqa dastur shaklida dastur natijalarini chiqaruvchi kiritish/chiqarish qurilmasi yoki tashqi qurilma.

❖ **Operatsion tizim (operating system)** – kursimizning asosiy predmeti. Foydalanuvchili va turli xil dasturli kompyuter apparatlaridan foydalanishni boshqaruvchi tizim dasturiy ta’minoti.

❖ **Amaliy dasturiy ta’minot (applications software)** – turli xil masalalarni yechish uchun mo‘ljallangan dasturlar. Ularga ko‘proq dasturlash tillari (masalan C++) bilan dasturlarni translyatsiya qilishni ta’minlovchi kompilyatorlar, mashina kodi (buyruq), ma’lumotlar omborini boshqarish tizimi, grafik kutubxona, o‘yin dasturlari, ofis dasturlari kiradi. Amaliy dasturiy ta’minot operatsion tizim bilan solishtirilganda nisbatan yuqori pog‘onada joylashadi, va kompyuterdagi turli xil amaliy va kundalik masalalarni yechishga yordam beradi.

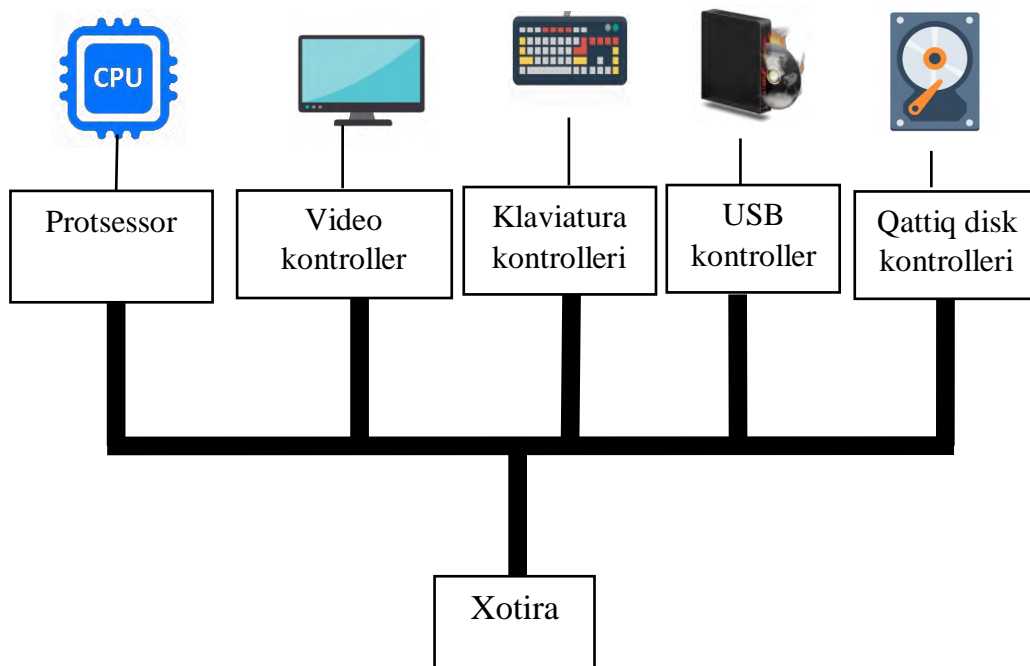
❖ **Foydalanuvchilar (users)** – insonlar va boshqa kompyuterlar. Kompyuter tizimlari komponentlariga foydalanuvchi inson sifatida – kompyuterda o‘z ish faoliyati jarayonining hisoblash tizimlari bir qismi bo‘lgan istalgan inson bo‘lishi mumkin. Tarmoqdagi boshqa kompyuterlar ushbu kompyuterga bog‘lanish (serverga ulanish) bo‘yicha foydalanuvchi rolini bajarishi mumkin. Masalan, katta dasturlarni bajarish yoki fayllarni saqlash.

Kompyuter tizimlarining ishlash tafsilotlarini o‘rganishdan oldin bizga kompyuter tizimining tuzilishi haqida umumiy ma’lumot kerak. Ushbu bo‘limda biz ushbu strukturaning bir nechta qismlarini ko‘rib chiqamiz.

### ***Kompyuter tizimining ishlashi***

Zamonaviy umumiy maqsadli kompyuter tizimi umumiy xotiraga ulanishni ta’minlaydigan bitta yoki bir nechta protsessorlardan va bir nechta qurilma kontrollerlaridan iborat (1.2-rasm). Har bir qurilma boshqaruvchisi qurilmaning ma’lum bir turi uchun javobgardir (masalan, diskovodlar, audio qurilmalar yoki video displeylar). Protsessor va kontroller qurilmalari parallel ravishda ishlashi mumkin, xotira sikli uchun raqobatlashadi. Birgalikda ishlatiladigan xotiraga muntazam ravishda kirishni ta’minlash uchun,

xotira kontrolleri xotiraga kirishni sinxronlashtiradi. Kompyuterni ishga tushirish uchun, masalan, u yoqilganda yoki qayta ishga tushirilganda, uni ishga tushirish uchun boshlang'ich dastur bo'lishi kerak. Ushbu boshlang'ich dastur yoki boshlang'ich dasturi oddiy ko'rinishga ega. Qoida tariqasida, u kompyuter qurilmalarida faqat o'qish uchun mo'ljallangan xotirada (ROM) saqlanadi. Bu protsessor registrilaridan tortib, qurilma kontrolleriga qadar xotira tarkibiga qadar tizimning barcha jihatlarini initsializatsiya qiladi.



1.2- rasm. Zamonaviy kompyuter tizimi

Boshlang'ich yuklash (Bootstrap) dasturi operatsion tizimni qanday yuklashni va uni qanday ishga tushurishni bilishi kerak. Bunga erishish uchun yuklash dasturi operatsion tizimning yadrosini topishi va uni xotiraga yuklab qo'yishi kerak. Yadro yuklangan va ishga tushirilgandan so'ng, u tizim va uning foydalanuvchilariga xizmat ko'rsatishni boshlash mumkin. Ba'zi bir xizmatlar yadrodan tashqarida, ya'ni tizim jarayonlariga aylanish uchun yuklash vaqtida xotiraga yuklanadigan yoki yadro ishlayotganda doimo ishlaydigan tizim demonlari tomonidan taqdim etiladi. UNIX-da birinchi tizim jarayoni "init" bo'lib, u boshqa ko'plab demonlarni ishga tushiradi. Ushbu bosqich tugallangandan so'ng, tizim to'liq yuklanadi va tizim biron bir hodisani sodir bo'lishini kutadi. Hodisaning ro'y berishi,

odatda, apparat yoki dasturiy ta'minot tomonidan uzilish bilan signal beradi. Qurilma istalgan vaqtda protsessorga signal yuborib, uzilishni keltirib chiqarishi mumkin, bu odatda tizim shinasini orqali amalga oshiriladi. Dasturiy ta'minot tizim chaqirig'i deb ataladigan maxsus operatsiyani (monitor chaqirig'i deb ham ataladi) bajarib, uzilishni keltirib chiqarishi mumkin.

### ***Xotira tuzilishi***

Protsessor ko'rsatmalarni faqat xotiradan yuklay oladi, shuning uchun ishga tushuriladigan har qanday dastur o'sha yerda saqlanishi kerak. Umumiy maqsadli kompyuterlar o'zlarining ko'p dasturlarini qayta yoziladigan asosiy xotiradan ishga tushuradi (operativ tezkor xotira yoki RAM (random-access memory) deb ham ataladi). Asosiy xotira odatda yarim o'tkazgich texnologiyasida amalga oshiriladi, bu dinamik operativ xotira (DRAM-dynamic random-access memory) deb ataladi. Xotiraning barcha shakllari baytlarning ketma-ketligini ta'minlaydi. Har bir baytning o'z manzili bor. O'zaro aloqaga muayyan xotira manzillariga ko'rsatmalarni yuklash yoki saqlash ketma-ketligi orqali erishiladi. Yuklash ko'rsatmasi bayt yoki so'zni asosiy xotiradan CPU ichki registriga o'tkazadi, saqlash bo'yicha ko'rsatma esa registr tarkibini asosiy xotiraga o'tkazadi. Yuklash va saqlashdan tashqari, protsessor protsedurani bajarish uchun, ko'rsatmalarni asosiy xotiradan avtomatik ravishda yuklaydi.

Fon Neyman arxitekturasi bilan ishlaydigan tizimda bajariladigan odatiy buyruqlarni bajarish sikli, avval ko'rsatmani xotiradan tanlaydi va ushbu ko'rsatmani buyruq registrida saqlaydi. Keyin ko'rsatma dekodrlanadi va operandalarni xotiradan olish va ba'zi bir ichki registrda saqlanishi mumkin. Operandlardagi ko'rsatma bajarilgandan so'ng, natija xotirada saqlanishi mumkin. E'tibor bering, xotira bloki faqat xotira manzillari oqimini ko'radi. U ularning qanday yaratilganligini (buyruq hisoblagichi, indekslash, bilvosita kirish, so'zma-so'z manzillar yoki har qanday boshqa vositalar orqali) yoki ular nimaga mo'ljallanganligini (ko'rsatmalar yoki ma'lumotlar) bilmaydi. Shunga ko'ra, dastur tomonidan xotira manzili qanday yaratilishini e'tiborsiz qoldiramiz. Bizni faqat ishlaydigan dastur tomonidan yaratilgan xotira manzillarining ketma-ketligi qiziqtiradi. Ideal holda, biz dasturlar va ma'lumotlar doimiy ravishda asosiy

xotirada bo'lishini xohlaymiz. Odatda, quyidagi ikkita sababga ko'ra tartibga solish mumkin emas:

1. Asosiy xotira barcha kerakli dasturlar va ma'lumotlarni doimiy saqlash uchun odatda juda kichikdir.

2. Asosiy xotira bu o'zgaruvchan xotira qurilmasi bo'lib, u o'chirilganda o'z tarkibini yo'qotadi.

Shunday qilib, ko'pgina kompyuter tizimlari asosiy xotiraning kengaytmasi sifatida ikkilamchi xotirani ta'minlaydi. Ikkilamchi xotira qurilmasining asosiy sharti shundaki, u katta hajmdagi ma'lumotlarni doimiy ravishda saqlashi mumkin. Eng keng tarqalgan ikkilamchi xotira qurilmasi ikkala dastur va ma'lumotlarni saqlashni ta'minlaydigan magnit diskdir. Ko'pgina dasturlar (tizimlar va ilovalar) ular asosiy xotiraga yuklanguncha diskda saqlanadi. Diskni to'g'ri boshqarish kompyuter tizimida muhim o'rin tutadi, shuning uchun u IV-bobda batafsil muhokama qilinadi.

### ***Kiritish/chiqarish qurilmasi***

Saqlash qurilmasi - kompyuterdagi kiritish/chiqarish qurilmalarining ko'pgina turlaridan faqat bittasi. Ko'pincha operatsion tizim kodi tizimning ishonchliligi va ishlashi uchun muhimligi va qurilmalarning o'zgaruvchanligi sababli kiritish/chiqarishni boshqarish uchun mo'ljallangan. Keyinchalik biz kiritish/chiqarish haqida qisqacha ma'lumot beramiz. Umumiy maqsadlar uchun mo'ljallangan kompyuter tizimi umumiy shina orqali ulangan protsessorlardan va bir nechta qurilma kontrollerlaridan tashkil topadi. Har bir qurilma kontrolleri qurilmaning ma'lum bir turi uchun javobgardir. Kontroller qurilmasiga qarab, bir nechta qurilmalarni ulash mumkin. Masalan, kichik kompyuter tizimlari interfeysi (SCSI - small computer systems interface) kontrolleriga yettita yoki undan ortiq qurilmalar ulanishi mumkin. Qurilma kontrolleri ba'zi bir mahalliy bufer saqlash qurilmasi va maxsus funksiyali registrlar to'plamini qo'llab-quvvatlaydi. Odatda, operatsion tizimlarda har bir qurilma kontrolleri uchun qurilma drayveri mavjud. Ushbu qurilma drayveri qurilma kontrollerini tushunadi va operatsion tizimning qolgan qismini qurilmaga yagona interfeys bilan ta'minlaydi. Kiritish/chiqarish operatsiyasini boshlash uchun qurilma drayveri qurilmalarni kontrolleriga tegishli registrlarni yuklaydi. Qurilma kontrolleri, o'z navbatida, qaysi registrlar (masalan, "klaviaturadan

belgi o‘qish”) ni aniqlash uchun ushbu registrlar tarkibini tekshiradi. Kontroller qurilmadan ma’lumotlarni mahalliy lokal buferga uzatishni boshlaydi.

Ma’lumotni uzatish tugagandan so‘ng, qurilma kontrolleri uzilish orqali qurilma drayveriga uning ishlashi tugaganligi to‘g‘risida xabar beradi. Keyin qurilma drayveri boshqaruvni operatsion tizimga qaytaradi, agar operatsiya o‘qilgan bo‘lsa, ma’lumot yoki ko‘rsatgich ma’lumoti qaytariladi. Boshqa operatsiyalar uchun qurilma drayveri holat to‘g‘risidagi ma’lumotlarni qaytaradi. Uzilishga asoslangan kiritish/chiqarish bu kichik hajmdagi ma’lumotlarni ko‘chirish uchun juda yaxshi, ammo katta hajmdagi ma’lumotni, masalan, kiritish/chiqarish diskini ko‘chirishda foydalanilganda juda katta xarajatlarga olib kelishi mumkin. Ushbu muammoni hal qilish uchun to‘g‘ridan-to‘g‘ri xotiraga kirish (DMA - direct memory access) ishlatiladi. Kiritish/chiqarish qurilmasi uchun buferlar, ko‘rsatkichlar va hisoblagichlarni o‘rnatgandan so‘ng, qurilma boshqaruvchisi butun ma’lumotlar blokini CPU aralashuvisiz to‘g‘ridan-to‘g‘ri yoki o‘z buferli saqlash xotirasidan xotiraga uzatadi. Har bir blok uchun qurilma drayverini operatsiya tugaganligi to‘g‘risida xabardor qilish uchun faqat bitta uzilish yaratiladi. Qurilma kontrolleri ushbu operatsiyalarni bajarayotganda, boshqa vazifalarni bajarish uchun markaziy protsessor mavjud.

### ***Bir protsessorli tizimlar***

Yaqin vaqtgacha ko‘pgina kompyuter tizimlari bitta protsessoridan foydalangan. Bir protsessorli tizimda umumiy maqsadli ko‘rsatmalar to‘plamini, shu jumladan foydalanuvchi jarayonlarining ko‘rsatmalarini bajarishga qodir bitta asosiy protsessor mavjud bo‘lgan. Deyarli barcha bitta protsessor tizimlarida boshqa ixtisoslashtirilgan protsessorlar mavjud. Ular disk, klaviatura va grafik kontrollerlar kabi maxsus qurilmalar uchun protsessorlar shaklida taqdim etilishi mumkin; yoki, asosiy kadrlarda, ular ma’lumotni tizim komponentlari o‘rtasida tezkor ravishda ko‘chiradigan kiritish/chiqarish protsessorlari kabi ko‘proq umumiy maqsadli protsessorlar shaklini olishlari mumkin. Bularning barchasi ixtisoslashtirilgan protsessorlar cheklangan ko‘rsatmalar to‘plamini bajaradilar va foydalanuvchi jarayonlarini ishga tushurmeydilar. Ba’zan ularni operatsion tizim boshqaradi, chunki operatsion tizim



ularga keyingi vazifalari haqida ma'lumot yuboradi va ularning holatini kuzatadi. Masalan, disk kontrollerining mikroprotssessori asosiy protssesordan so'rovlar ketma-ketligini oladi va o'z diskini navbatga qo'yish va rejalashtirish algoritmini amalga oshiradi. Ushbu tartibga solish asosiy protsessorni diskni rejalashtirish xarajatlaridan ozod qiladi. Maxsus maqsadli mikroprotssesordlardan foydalanish keng tarqalgan va u bir protsessorli tizimni ko'p protsessorli tizimga aylantirmaydi. Agar bitta umumiy maqsadli protsessor mavjud bo'lsa, unda tizim bir protsessorli tizimdir.

### ***Ko'p protsessorli tizimlar***

So'nggi bir necha yil ichida hisoblash tizimida ko'p protsessor tizimlari (parallel tizimlar yoki ko'p protsessorli tizimlar deb ham ataladi) ustunlik qila boshladi. Bunday tizimlarda ikki yoki undan ko'p protsessorlar o'zaro aloqa qiladilar, kompyuter shinasini, ba'zan esa soat, xotira va tashqi qurilmalarni almashadilar. Ko'p protsessorli tizimlar birinchi marta serverlarda paydo bo'ldi va shu vaqtdan boshlab ish stoli va ko'chma tizimlarga o'tdi. Yaqinda smartfonlar va planshet kompyuterlari kabi mobil qurilmalarda bir nechta protsessorlar paydo bo'ldi. Ko'p protsessorli tizimlar uchta asosiy afzalliklarga ega:

❖ **O'tkazish hajmi oshdi** (Increased throughput). Protsessorlar sonini ko'paytirib, biz kamroq vaqt ichida ko'proq vazifalarni bajarishni kutamiz. Biroq,  $N$  protsessorlari bilan tezlashtirish koeffitsienti  $N$  ga teng emas; aksincha, u  $N$  dan kam. Bir nechta protsessorlar biron bir vazifani bajarish uchun birgalikda ishlaganda, uning barcha qismlari to'g'ri ishlashini ta'minlash uchun ma'lum miqdordagi qo'shimcha xarajatlar kelib chiqadi. Ushbu xarajatlar va umumiy resurslar uchun raqobat, qo'shimcha protsessorlardan kutilayotgan yutuqlarni kamaytiradi.

❖ **Masshtabni tejankorligi** (Economy of scale). Ko'p protsessorli tizimlar ekvivalent ko'p protsessorli tizimlarga qaraganda arzonroq bo'lishi mumkin, chunki ular periferik qurilmalar, saqlash qurilmalari va quvvat manbalaridan birgalikda foydalanishlari mumkin. Agar bir nechta dastur bir xil ma'lumot to'plamida ishlayotgan bo'lsa, mahalliy disklarga ega ko'plab kompyuterlarga va ma'lumotlarning ko'p nusxalariga ega bo'lishdan ko'ra, ushbu

ma'lumotni bitta diskda saqlash va uni barcha protsessorlarda ishlatish arzonroq.

❖ **Ishonchlilik oshdi** (Increased reliability). Agar vazifalarni bir nechta protsessorlar o'rtasida to'g'ri taqsimlash mumkin bo'lsa, bitta protsessorning ishdan chiqishi tizimni to'xtatmaydi, faqat uni sekinlashtiradi. Agar bizda o'nta protsessor mavjud bo'lsa va ulardan bittasi ishdan chiqsa, qolgan to'qqiz protsessorning har biri ishdan chiqqan protsessor vazifasining bir qismini olishi mumkin. Shunday qilib, butun bir tizim ishdan chiqmaydi va faqatgina 10 foizga sekinroq ishlaydi. Kompyuter tizimining ishonchliligini oshirish ko'plab dasturlarda juda muhimdir.

### ***Parallel kompyuter tizimlari va ularni OTlarining o'ziga xos xususiyatlari***

Parallel kompyuter tizimlari bu bir necha to'g'ridan-to'g'ri o'zaro ta'sirlashishadigan multiprotsessorli tizimlar hisoblanadi. Hozirgi vaqtda multiprotsessorli ishchi stansiyalar chiqarilmoqda. Masalan, ish stoli kompyuteriga ega bo'lish bilan siz uning tarkibida ikkita yoki hatto to'rtta protsessorlarni aniqlashingiz mumkin.

Mos ravishda, OT bunday tizimni rekonfiguratsiyalash, yangi protsessorlarni ulash yoki tizimdan protsessorlarni olib tashlash, masalalarni yechishni bir nechta protsessorlarda parallellashtirish va ularni yechadigan parallel jarayonlarni sinxronlashtirishni ta'minlashi kerak.

Parallel kompyuterlar orasidan protsessorlar umumiy xotira va taymerga (taktlarga) ajratiladigan, uzviy bog'langan (tightly coupled) tizimlar ajralib turadi, ular orasidagi o'zaro ta'sirlashish umumiy xotira orqali bo'lib o'tadi.

Very Long Instruction Word (VLIW) va Explicit Parallelism Instruction Computer (EPIC) arxitekturalardagi parallel kompyuterlarda hisoblashlarni parallellashtirish buyruqlar darajasida bo'lib o'tadi.

Ko'p yadroli (multi-core) kompyuterlar – assotsiativ xotirani (keshni) ajratadigan va umumiy xotirada ishlaydigan bitta kristallda joylashgan bir-birlari bilan uzviy bog'langan protsessorlarga (yadrolarga) asoslangan kompyuter tizimlari hisoblanadi.

#### **Parallel kompyuter tizimining afzalliklari**

1. Yaxshilangan unumdorlik (throughput) – ma’lumki, masalan yechish algoritmini parallellashtirish uni yechishga ketadigan vaqtni kamaytirishga imkon berishi mumkin;

2. Tejamkorlik – parallel OTda ishning qismini boshqa protsessor yoki yadroga berish mumkin;

3. Oshirilgan ishonchlilik – protsessorlardan biri ishdan chiqqanda OT hisoblashlarni boshqa protsessorga o’tkazishi mumkin;

4. Foydalanuvchiga “do’stona” unumdorlikni kamaytirish (graceful degradation) – agar protsessorlardan biri ishdan chiqsa va konfiguratsiyadan chiqarilgan bo’lsa, foydalanuvchi kompyuter va OTni to’g’ri tashkil etishda hisoblashlarning sekinlashishini sezmasligi mumkin.

5. Xatoliklarga barqarorlik (fail-soft system) – qurilma va dasturdagi xatoliklarda ko’p protsessorli tizimning barqaror ishlashi.

Very Long Instruction Word (VLIW) va Explicit Parallelism Instruction Computer (EPIC) arxitekturalardagi parallel kompyuterlarda hisoblashlarni parallellashtirish buyruqlar darajasida bo’lib o’tadi.

Ko’p yadroli (multi-core) kompyuterlar – assotsiativ xotirani (keshni) taqsimlaydigan va umumiy xotirada ishlaydigan bitta kristalda joylashgan bir-birlari bilan uzviy bog’langan protsessorlarga (yadrolarga) asoslangan kompyuter tizimlari hisoblanadi.

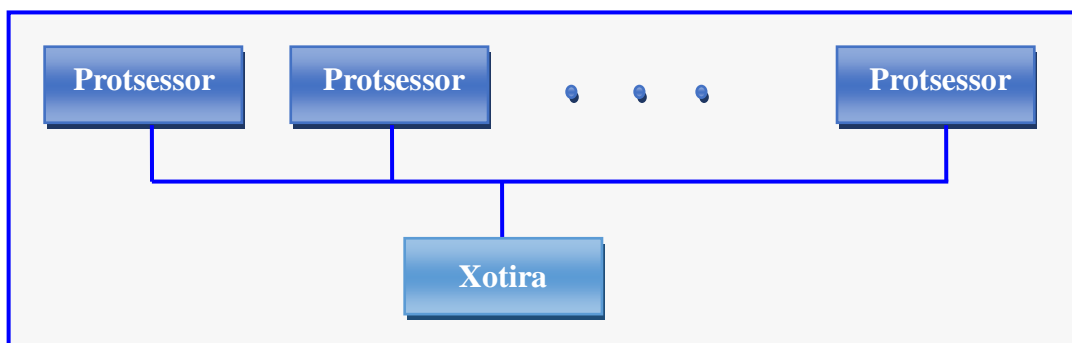
### ***Simmetrik va assimmetrik ko’p protsessorli tizimlar***

Simmetrik ko’p protsessorli tizim (symmetric multiprocessing (SMP)) – bu ko’p protsessorli kompyuter tizimi bo’lib, uning barcha protsessorlari teng huquqli va o’sha bir OT nusxasini ishlatadi. Operatsion tizim bunda istalgan protsessorida bajarilishi mumkin. Bunday tizimda istalgan bo’sh protsessorga istalgan topshiriq berilishi mumkin.

Barcha protsessorlar umumiy xotira va umumiy disklar resurslarini ishlatadi. Bir necha jarayonlar (yoki oqimlar) unumdorli sezilarli buzilmasdan bir vaqtda bajarilishi mumkin. Ko’plab zamonaviy OTlar SMP arxitekturasini qo’llaydi. Simmetrik multiprotsessorli tizimda OT (masalan, Linux) o’rnatilgandan keyin foydalanuvchi menyuda boot loader ni ko’rishi mumkin, bu haqiqatda uning kompyuteriga bitta emas, balki ikkita SMP qo’llanadigan va qo’llanmaydigan OT versiyalari o’rnatilganligini bildiradi.

Assimmetrik ko‘p protsessorli tizim (asymmetric multiprocessing) – bu ko‘p protsessorli kompyuter tizimi bo‘lib, unda protsessorlar o‘z funksiyalari bo‘yicha maxsuslashtirilgan. Har bir protsessorga o‘ziga xos topshiriq beriladi. Bosh protsessor (master processor) bo‘ysunuvchi jarayonlarning (slave processors) ishlashini rejalashtiradi. Bunday tizimda OT bitta ma’lum unga biriktirilgan markaziy protsessorida bajariladi.

Bunday arxitektura juda katta tizimlar uchun o‘ziga xos. Misol uchun "Elbrus" tizimi o‘z tarkibida konfiguratsiyaga bog‘liq ravishda bittadan o‘ntagacha markaziy protsessorlarga, bittadan to‘rttagacha maxsuslashtirilgan kiritish/chiqarish protsessorlariga (KChP), bittadan to‘rttagacha ma’lumotlarni uzatish protsessorlariga (MUP) ega. Kompyuterlarning SMP-arxitekturasini tashkil etish sxemasi 1.3- rasmda keltirilgan.



1.3- rasm. Kompyuterlarning SMP-arxitekturasini tashkil etish sxemasi

### ***Taqsimlangan kompyuter tizimlari va ularning OTlarini o‘ziga xos xususiyatlari***

Taqsimlangan tizimda (distributed system) hisoblashlar o‘zaro tarmoqqa birlashtirilgan bir necha fizik protsessorlar (kompyuterlar) orasida taqsimlanadi. Kuchsiz bog‘langan tizim (loosely coupled system) – taqsimlangan kompyuter tizimi bo‘lib, unda har bir protsessor o‘z lokal xotirasiga ega bo‘ladi, turli protsessorlar esa o‘zaro aloqa liniyalari – yuqori tezlikli shinalar, telefon liniyalari, simsiz aloqa (Wi-Fi, EVDO, Wi-Max va boshqalar) orqali o‘zaro ta’sirlashishadi.

Taqsimlangan tizimlarning afzalliklariga quyidagilar kiradi:

1. **Resurslarni ajratish** (birgalikda foydalanish): taqsimlangan tizimda turli resurslar turli kompyuterlarda saqlanishi mumkin. Dasturlar yoki ma'lumotlarning nusxalarini bir necha kompyuterlarda saqlash bilan zahiralashning zarurati yo'q.

2. **Birgalikda yuklash** (load sharing): taqsimlangan tizimda har bir kompyuterga ma'lum topshiriq berilishi mumkin, uni u boshqa kompyuterlar o'z topshiriqlarini bajarishi bilan parallel bajaradi.

3. **Ishonchlilik**: taqsimlangan tizimning kompyuterlaridan biri rad etganda yoki uzilganda uning topshirig'i uzilish yakuniy natijaga minimal darajada ta'sir etishi yoki umuman ta'sir etmasligi uchun boshqa kompyuterga taqsimlanishi mumkin.

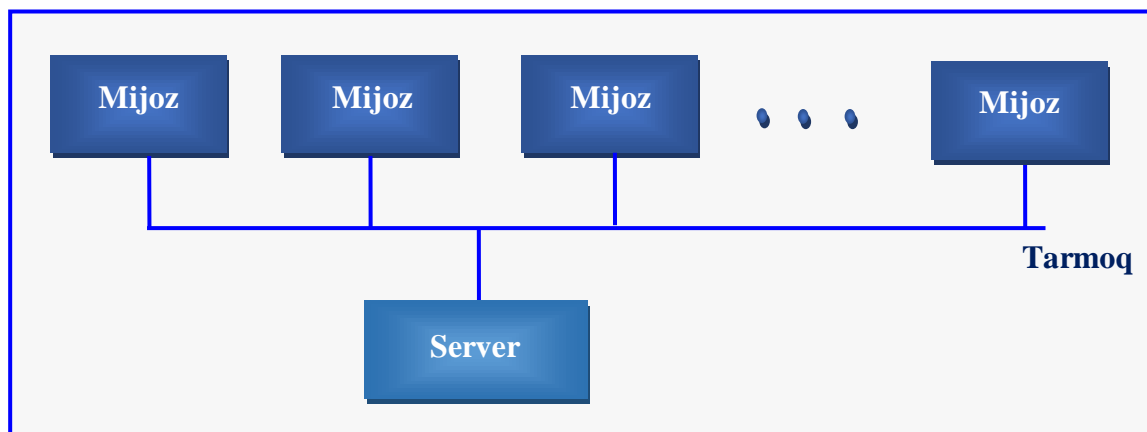
4. **Aloqa**: taqsimlangan tizimda barcha kompyuterlar bir-birlari bilan bog'langan, shunday ekan, masalan, zarurat bo'lganda quvvatliroq kompyuterning resurslaridan foydalanish maqsadida bitta kompyuterdan boshqa kompyuterga masofadan kirish mumkin.

Taqsimlangan tizimda kompyuterlar tarmoq infratuzilmasiga bog'langan, u quyidagilar bo'lishi mumkin:

1. Lokal tarmoq (local area network - LAN);
2. Global yoki hududiy tarmoq (wide area network - WAN).

O'zining tashkil etilishi bo'yicha taqsimlangan tizimlar mijoz-serverli (client-server) yoki bir darajali (peer-to-peer) tizimlar bo'lishi mumkin. Mijoz-serverli tizimda ma'lum kompyuterlar serverlar rolini, boshqalari esa ularning xizmatlaridan foydalanadigan mijozlar rolini o'ynaydi.

Taqsimlangan tizimlarning bunday tashkil etilishi eng keng tarqalgan va uni atroflicha ko'rib chiqamiz. Bir darajali taqsimlangan tizimda barcha kompyuterlar teng huquqli. Mijoz-server tizimining tuzilishi 1.4- rasmda keltirilgan.



## 1.4- rasm.Mijoz-server tizimi tuzilishi

### *Mijoz-server kompyuter tizimlaridagi serverlarning turlari*

Taqsimlangan tizimlarning mijoz-server arxitekturasi juda keng tarqalgan va operatsion tizimlar bilan qo'llanadi. Shuning uchun zamonaviy taqsimlangan tizimlar qanday serverlar turlari va funksiyalarini tashkil etishini bilish juda muhim.

**Fayl-server (file server)** – kompyuter va dasturiy ta'minot bo'lib, u kompyuter-serverning disklarida joylashgan fayllar tizimlari to'plamiga, lokal tarmoqning (LAN) boshqa kompyuterlariga ulanishni taqdim etadi. Misol uchun, UNIX (Linux, FreeBSD, Solaris va boshqalar) turdagi OT uchun SAMBA (SMB – Server Message Block) serverlar dasturiy ta'minoti lokal tarmoqning Windows-kompyuterlaridan UNIX-mashinalarning fayllar tizimlariga ulanishni ta'minlaydi. Samba shuningdek, Macintosh/MacOS platformasi uchun ham amalga oshirilgan.

**Ilovalar serveri (application server)** – kompyuter va dasturiy ta'minot bo'lib, hisoblash resurslari (xotira va protsessor) va lokal tarmoqning boshqa kompyuterlaridan ilovalarning ma'lum sinflarini olisdan ishga tushirish uchun zarur muhitni taqdim etadi. Misollar Java Enterprise Editionda (JEE) ishlaydigan ma'lum ilovalar serverlaridan eng yaxshilari WebSphere (IBM), WebLogic (BEA) ilovalar serverlari hisoblanadi.

**Ma'lumotlar ombori serveri (database server)** – kompyuter va dasturiy ta'minot bo'lib, kompyuter-serverda joylashgan ma'lumotlar omboriga tarmoqning boshqa kompyuterlariga ulanishni taqdim etadi. Misol, Microsoft SQL Server ma'lumotlar omboriga ulanish uchun serverlar dasturiy ta'minoti hisoblanadi.

**Veb-server (Web server)** – kompyuter va dasturiy ta'minot bo'lib, mijozlarga kompyuter-serverda joylashgan Web-sahifalarga WWW orqali ulanishni taqdim etadi. Misol, erkin tarqatiladigan Web-server Apache hisoblanadi.

**Proksi-server (Proxy server)** – kompyuter va dasturiy ta'minot bo'lib, u lokal tarmoqning qismi hisoblanadi va Internetga lokal tarmoq kompyuterlaridan samarali kirish, trafikni filtrlash, tashqi

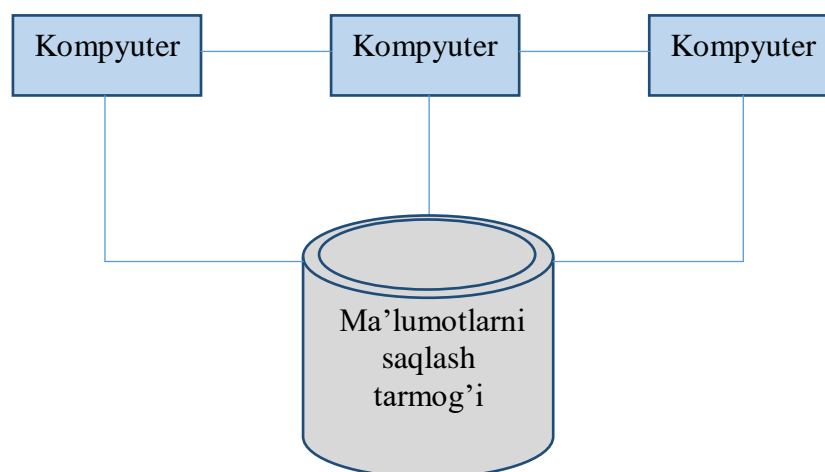
hujumlardan himoyalashni ta'minlaydi. Proxy-server odatda operatsion tizimga o'rnatiladi.

**Elektron pochta serveri** – kompyuter va dasturiy ta'minot bo'lib, u qandaydir lokal tarmoq kompyuterlari elektron pochtasini jo'natish, olish va tarqatishni bajaradi. Shuningdek pochta shifrlash (email encryption) – ma'lum tarmoq domenidan manzillarga (buyurtmachiga) xatlarni jo'natishdan oldin shifrlash va ularni buyurtmachi olganidan keyin deshifrlashni ta'minlashi mumkin.

**Server qismi (Server back-end)** – katta ishonchlilik va katta resurslar hajmini taqdim etish maqsadlarida bitta server o'rniga ishlatiladigan server kompyuterlarning lokal tarmoqqa birlashtirilgan guruhi hisoblanadi. Bunga yaqin bo'lgan boshqa atama ma'lumotlarga ishlov berish markazi (data center) hisoblanadi. Bu tushuncha bulutli hisoblashlarning yanada keng tarqalishi bilan juda dolzarb, bu nuqtai nazardan eng zamonaviy mijoz-server o'zaro ta'sirlashishi hisoblanadi.

### ***Klasterli hisoblash tizimlari va ularning O'nlari***

Kompyuterlar klasterlari ilmiy hisoblashlar uchun juda ommalashgan. Kompyuterlar klasterlari tezkor lokal tarmoq orqali bog'lanadi. Klasterlashtirish ikki yoki undan ortiq tizimlarga umumiy xotiradan foydalanishga imkon beradi. Klasterlashtirish yuqori ishonchlilikni ta'minlaydi. Klasterli hisoblash tizimi ko'rinishi 1.5-rasmda keltirilgan.



1.5- rasm. Klasterli hisoblash tizimi ko'rinishi

Kompyuterlar klasterlari quyidagi ikkita turlarga bo‘linadi:

❖ **assimmetrik klasterlashtirish (asymmetric clustering)** – bitta kompyuter ilovani bajaradi, qolganlari esa turib qoladigan kompyuterlar klasterini tashkil etish;

❖ **simmetrik klasterlashtirish (symmetric clustering)** – klasterning barcha mashinalari bir vaqtda bitta katta ilovaning turli qismlarini bajaradigan kompyuterlar klasterini tashkil etish.

Shuningdek, quyidagilarga bo‘linadi:

❖ **yuqori tezlikli ulanishli klasterlar (high-availability clusters)** – klaster kompyuterlari resurslariga, masalan, ma’lumotlar omborlariga optimal ulanishni ta’minlaydigan kompyuterlar klasterlari;

❖ **yuklanish balanslanadigan klasterlar (load-balancing clusters)** – so‘rovlarni balanslaydigan (front-ends), serverlar qismlari (server back-end) kompyuterlari orasida topshiriqni taqsimlaydigan bir necha kompyuterlarga ega bo‘lgan kompyuterlar klasterlari.

Klasterlar ko‘pincha universitetlarda va tadqiqotlar markazlarida (masalan, CERN, Shveytsariya) o‘rnatilgan. Klasterlar uchun operatsion tizimlarga Windows 2003 for clusters, Windows 2008 High-Performance Computing kiradi.

### ***Real vaqt tizimlari va OTlari***

Real vaqt tizimlari maxsus ilovalar uchun, masalan, ilmiy tajribalar uchun, tasvirlarga bog‘liq tibbiy tizimlarda, sanoatdagi boshqarish tizimlarida, aks ettirish (display) tizimlarida, kosmik uchishlarni boshqarish tizimlari, AESlarda va boshqalarda, boshqarish qurilmalari sifatida ko‘p ishlatiladi. Bunday tizimlar uchun aniq aniqlangan vaqt cheklashlarining bo‘lishi va bajarilishi (javob vaqti - response time, buzilishsiz ishlash vaqti va h.k.) o‘ziga xos. Real vaqt tizimlari hard real-time va soft real-time turlarga bo‘linadi.

Real vaqt tizimlarining ikki turi mavjud. Ular quyidagilar:

**Hard real-time - tizimlar** – vaqt bo‘yicha cheklashlar buzilganda u boshqaradigan obyektida xatolik (rad etish) vujudga kelishi mumkin bo‘lgan real vaqt tizimlari hisoblanadi. Masalan, avtomobil dvigatelini boshqarish tizimi, kardiostimulyatorni boshqarish tizimi va boshqalar. Bunday tizimlarda ikkilamchi xotira cheklangan yoki mavjud emas. Ma’lumotlar operativ xotirada (RAM) yoki doimiy xotira qurilmasida (DXQ, ROM) saqlanadi.



Bunday tizimlardan foydalanishda umumiy maqsadlardagi OT uchun o'z o'rniga ega bo'lmagan vaqtni bo'lish tizimlari bilan nomuvofiqliklar vujudga kelishi mumkin. Oddiy til bilan aytganda, bunday tizimlarning ishlashida uzilishlarga ruxsat etilmaydi. Tizimning asosiy ishlash sikli uchun barcha zarur ma'lumotlar oldindan xotiraga yuklanishi kerak. Bunday tizimning kodini bajaradigan jarayon diskka ko'chirilishi mumkin emas.

Bunday tizimlar uchun OT odatda soddalashtirilgan, virtual xotira o'rniga fizik xotira ajratiladi, qolgan barcha resurslarni virtuallashtirish turlari bo'lishi mumkin emas. Real vaqt OTlarini ommaviy ishlab chiqish amaliyoti umumiy maqsadlardagi OTlar ochiq dastlabki kodlari asosida "barcha ortiqchaliklarni olib tashlash" yo'li bilan bunday OTlarni ishlab chiqish amaliyoti hisoblanadi. Lekin bunda ehtiyotkorlikka rioya etish kerak bo'ladi.

**Soft real-time - tizimlar** – vaqt bo'yicha cheklashlar buzilganda u boshqaradigan obyektga xatolikka (rad etishga) olib kelmaydigan real vaqt tizimlari hisoblanadi. Odatda bu doimiy o'zgaradigan vaziyatli bir necha o'zaro bog'langan tizimlarni boshqarish tizimi hisoblanadi. Misol, tijorat havo yo'llaridagi parvozlarni rejalashtirish tizimi hisoblanadi.

Bunday tizimdagi qandaydir kechikishda, eng yomon holda, ayrim parvozlarning yo'lovchilariga aeroportda biroz kutishga to'g'ri keladi, lekin hech qanday halokatli oqibatlar bo'lmaydi. Bunday tizimlar sanoat boshqarish tizimlar va robotlar texnikasi uchun cheklangan foydalilikka ega. Ular yana OTlar rivojlangan imkoniyatlarini talab qiladigan zamonaviy ilovalar uchun (masalan, multimedia va virtual reallik uchun) foydali bo'ladi.

### ***Cho'ntak kompyuterlari (handhelds) va ularning OTlari***

Bu sinfdagi qurilmalarga cho'ntak personal kompyuterlari (ChPK) yoki Personal Digital Assistants (PDA) va mobil telefonlar kiradi. Bu sinfdagi kompyuterlarning o'ziga xos xususiyatlari va muammolariga quyidagilar kiradi:

- ❖ cheklangan xotira hajmi;
- ❖ nisbatan sekin ishlaydigan protsessorlar: mobil qurilmalar uchun eng oddiy komandalarni bir necha sekundlarda bajarilishini kutish oddiy hol, bu noqulay;

❖ monitorlar (displeylar) ekranlarining kichik o'lchami, bu yerdan GUIning qo'llash uchun maxsuslashtirilgan dasturiy ta'minot zarurati kelib chiqadi. Masalan, Java Micro Editionda (JME) mobil qurilmalar uchun Java versiyalarida GUIning ishlab chiqish uchun qulay AWT va Swing umumiy paketlaridan foydalanish zarur. Ularning o'rniga JMEda ishlab chiquvchiga Java (JSE) standart nashri bilan moslashuvchan bo'lmagan javax.microelectronics...lcdui turdagi maxsuslash-tirilgan paketlar taklif etiladi, bu kodni qurilmaning turiga, ba'zan esa ekranlarning turli o'lchamlariga ega bo'lgan mobil telefonlarning aniq bir modellariga bog'liq bo'ladi;

❖ Internet orqali uncha yuqori bo'lmagan aloqa tezligi. Masalan, mobil telefonlarning GPRS-modemi taxminan sekundiga 3-5 kilobayt – dial-up tezligidagi aloqani ta'minlaydi;

❖ ma'lumotlarni uzatish uchun aloqa Bluetooth yoki IrDA (binobarin, u ko'pincha bo'lmaydi) orqali amalga oshiriladi. Barcha zarur portlar mavjud emas. Masalan, mobil qurilmalarning qismida USB port mavjud emas, ya'ni "fleshkadan" foydalanish mumkin emas, bu juda noqulay va Smart Media turdagi o'ta kichik maxsus disklardan foydalanishga to'g'ri keladi, ular uchun stol kompyuterlarida o'qish uchun adapterlar hamma vaqt ham bo'lavermaydi.

Shunga qaramay, zamonaviy kommunikatsiya vositalari (masalan, Wi-Fi) va personal kompyuterlar bilan moslashuvchan portlar va tashqi xotira modullari cho'ntak va mobil qurilmalarda ham ishlatila boshlanmoqda. Cho'ntak va mobil qurilmalar uchun operatsion tizimlar va boshqa tizim dasturiy ta'minotida bu barcha cheklashlarni, xususan, cheklangan xotira hajmini hisobga olishga to'g'ri keladi. Shuning uchun qator qulay kundalik dasturiy imkoniyatlarni mobil qurilmalar uchun ta'qiqlashga to'g'ri keladi.

### ***Bulutli hisoblashlar va bulutli hisoblashlar uchun OTlar***

**Bulutli hisoblashlar (cloud computing).** ATning eng ommaviy rivojlanishi yo'nalishlaridan biri hisoblanadi. "Bulut" (cloud) bu endi Internet yoki boshqa kommunikatsion tarmoq orqali (masalan, ATM tarmoq orqali) taqdim etiladigan servislarni aks ettirish uchun o'nlab yillardan beri ishlatib kelinayotgan metafora hisoblanadi. Bulutli hisoblashlar Internet orqali servislarning sifatida mumkin bo'lgan va ishlatiladigan va yuqori unumdor ma'lumotlarga ishlov berish

markazlari (data centers) yordamida amalga oshiriladigan dinamik masshtablashtiriladigan (scalable) va virtuallashtirilgan resurslarga (ma'lumotlar, ilovalar, OT va boshqalarga) asoslangan hisoblash modeli hisoblanadi.

Foydalanuvchilar nuqtai nazaridan individual foydalanuvchida bo'lmagan quvvatli hisoblash resurslaridan foydalanish uchun turli kompaniyalar tomonidan taqdim etiladigan "bulutlar" to'plami (umumiy mumkin bo'lgan, korporativ, xususiy va boshqalar) mavjud. "Bulut" servislari pullik.

Bepul servislardan Windows Live'ni aytish mumkin (<http://www.live.com>;). Bulut hisoblashlarning kamchiligi foydalanuvchi u foydalanadigan "bulutga" (u foydalanadigan ma'lumotlar va dasturlar mumkin bo'lgan) to'liq bog'liq bo'lib qolishi va nafaqat "bulutdagi" kompyuterlarni, balki hatto o'z ma'lumotlardan zahira nusxa ko'chirishni boshqara olmasligidan iborat.

Shu sababli bulut hisoblashlarning xavfsizligi, foydalanuvchilar ma'lumotlarining konfidensialligi va boshqalar haqidagi qator muhim masalalar vujudga keladi. Ularning hammasi hozirgi vaqtda yechilmagan. Ma'lumotlarga ishlov berish markazlari apparaturalari nuqtai nazaridan bulut hisoblashlarni tashkil etishning jiddiy muammosi elektr energiyasini tejash va yuklanishni taqsimlash muammosi hisoblanadi, chunki har bir ma'lumotlarga ishlov berish markazida bulutli hisoblashlarga millionlab olisdagi foydalanuvchilar ega.

Eng ommaviy "bulut" platformalari Microsoft Windows Azure (bulutli OT) va Microsoft Azure Services Platform (Microsoft.NET asosida amalga oshirilgan) hisoblanadi. Foydalanuvchiga buning uchun zarur resurslarga ega bo'lmasligi mumkin bo'lgan kompyuterida uni o'rnatish haqida zarurat bo'lmaydi.

Web-brauzer va brauzer orqali bulut servislarni ishga tushirish foydalanish uchun minimal sozlashlar paketi (plug-ins) talab qilinadi xalos. Hozirgi vaqtda yirik kompaniyalar (Microsoft, IBM, HP, Dell, Oracle va boshqalar) o'z bulutli hisoblashlar tizimlarini ishlab chiqishmoqda. Ushbu korporativ tizimlarni foydalanuvchiga mumkin bo'lgan yagona "bulutga" integratsiyalash (birlashtirish) yo'nalishi mavjud.

## Nazorat savollari

1. Operatsion tizim nima?
2. Operatsion tizimning asosiy funksiyalari?
3. Operatsion tizimda ishlashdan asosiy maqsad nima?
4. Operatsion tizimlar qanday sifatlarga ega bo'lishi kerak?
5. Operatsion tizimlar davrlarini tushuntiring.
6. Kompyuter tizim komponentlari.
7. Operatsion tizimning asosiy komponentlari.
8. Operatsion tizim tuzilishini tushuntiring.
9. Operatsion tizimlarning klassifikatsiyalanishini tushuntiring.
10. Mijoz-server kompyuter tizimlaridagi serverlarning turlari
11. Parallel kompyuter tizimlar va ularning operatsion tizimlarining o'ziga xos xususiyatlari nima?
12. Klasterli hisoblash tizimlari va ularning operatsion tizimlari.
13. Real vaqt tizimlari va ularning operatsion tizimlarini tushuntiring.
14. Bulutli hisoblash operatsion tizimlaridan qanday maqsadlarda qo'llaniladi?

## **IIBOB. OPERATSION TIZIMDA JARAYONLARNI VA PROTSESSORNI BOSHQARISH**

Ushbu mavzu jarayonlarni boshqarish, jarayonlarni rejalashtirish va rejalashtirishning asosiy tushunchalarini o'z ichiga oladi. Ushbu ma'ruzada batafsil muhokama qilingan birinchi konsepsiya, jarayon va jarayonlarni boshqarish tushunchasi. Quyidagi masalalar ko'rib chiqiladi:

- ❖ jarayon tushunchasi;
- ❖ jarayon holati;
- ❖ jarayonni boshqarish bloki;
- ❖ jarayonlarni rejalashtirish;
- ❖ jarayonlar bo'yicha operatsiyalar;
- ❖ oqim tushunchasi;
- ❖ oqim turlari va boshqalar.
- ❖ protsessorni boshqarish;
- ❖ tizim modeli;
- ❖ resurslarni taqsimlash grafi;
- ❖ berliklarning xarakteristikalarini;
- ❖ berliklarga ishlov berish;
- ❖ berliklarni oldini olish.

### **2.1. Operatsion tizimda jarayonlar va ularning holatlari**

#### ***Jarayon tushunchasi***

Jarayon - bu foydalanuvchi tomonidan ishga tushirilgan dastur, jarayon buyruqlarni bajaruvchi to'plam majmui (xotirani bajarish uchun ajratish va fayllar va kiritish/chiqarish qurilmalaridan foydalaniladigan manzil va boshqalar) va operatsion tizim boshqaruvida joylashgan joriy vaqtda uni bajarilishini (registr, dastur hisoblagich, stek holati va qo'llash belgilari) tavsiflaydi. Ish paytida operatsion tizim ko'plab dasturlarni bajaradi: paketli vazifalar, vaqtni taqsimlash rejimida foydalanuvchi dasturlari, tizim dasturlari va jarayonlari. Foydalanuvchi dasturlarini tavsiflovchi bir nechta o'xshash atamalar (terminlar) mavjud: jarayon (process), vazifa (job),

topshiriq (task). Ushbu atamalarni sinonim deb hisoblashimiz mumkin.

Jarayonning muhim xususiyati: bu ketma-ket bajarilishi kerak bo'lgan hisoblash birligi, ya'ni, har bir jarayon o'zining ketma-ket boshqarish oqimiga (control flow) ega - bu jarayon tomonidan bajariladigan buyruqlar ketma-ketligidir. Operatsion tizim tomonidan yaratilgan va boshqariladigan jarayon quyidagi asosiy ma'lumotlarni o'z ichiga oladi:

- ❖ dastur hisoblagichi (program counter - PC) - bajarilayotgan joriy buyruq manzili; odatda qurilmalarning maxsus tizim registrida saqlanadi;

- ❖ Stek - bu operatsion tizim tomonidan jarayon protseduralarining lokal ma'lumotlari, ularning parametrlari (argumentlari) va hisob-kitoblarni tashkil qilish uchun zarur bo'lgan ma'lumotlar o'rtasida saqlanadigan, jarayonni yaratishda operatsion tizim tomonidan ajratilgan asosiy xotiraning doimiy maydoni. Keyingi protsedurani ishga tushirishda, faollashtirish yozuvi (activation record) stekda beriladi, shuningdek, protseduraning hozirgi avlodining lokal ma'lumotlarini saqlash uchun stek kadri (stack frame) va lokal ma'lumotlar maydoni (local data area) deb ataladi.

- ❖ Ma'lumotlar bo'limi (data section) – bu operatsion tizim tomonidan global o'zgaruvchilar, massivlar, tuzilmalar, obyektlar saqlanadigan jarayonga ajratilgan asosiy xotiraning statik (doimiy ravishda ajratilgan, doimiy o'lchamdagi) maydoni. Jarayonning bajariladigan kodi (buyruqlar) dastlab ikkilamchi xotirada (diskda) saqlanadi va unga kirishda to'liq yoki qisman asosiy xotiraga yuklanadi.

### ***Jarayon holatlari***

Jarayon ishga tushganda, jarayon o'z holatini quyidagicha o'zgartirishi mumkin:

- ❖ **Yangi (new):** jarayon operatsion tizim tomonidan yaratilgan, ammo hali ishga tushmagan;

- ❖ **Bajarish (running):** Jarayon buyruqlari protsessorida bajariladi yoki OT ishlaydigan kompyuter tizimining protsessorida;

- ❖ **Kutish (waiting):** jarayon qandaydir hodisa sodir bo'lishini kutmoqda, masalan, kiritish/chiqarishni (I/O) tugallanishini. Kutish holatida jarayon protsessorini egallamaydi;

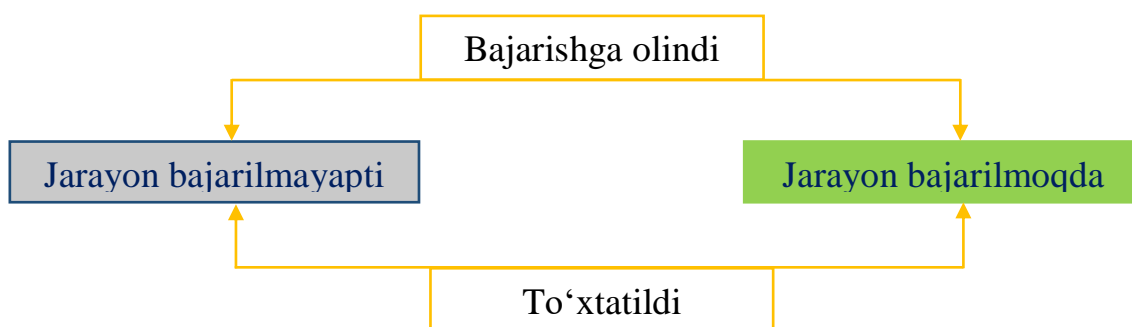
❖ **Tayyorlik (ready):** jarayon bajarilishi uchun protsessor resurslarini olishni kutmoqda. Jarayon odatda u yaratilganda ham, yoki kiritish/chiqarish (I/O) tugagandan so‘ng ham (kutish holatidan) bajarilishga tayyor holatga keladi;

❖ **Tugatish (finished):** Jarayonni bajarish tugallandi.

### *Jarayon holat diagrammasi*

Kiritilgan topshiriqlarni hisoblash tizimlarda bajarilishi (faqat foydalanuvchi dasturlari emas, balki operatsion tizimning belgilangan qismi bo‘lishi mumkin) jarayon majmuini tashkil qiladi. Shubxasiz, har bir vaqt momentida bir protsessorli kompyuter tizimida faqat bitta jarayon bajarilishi mumkin. Multidasturli hisoblash tizimlarida protsessoridagi bir jarayonni boshqasiga ulash orqali bir nechta jarayonlarni parallel qayta ishlanadi. Bitta jarayon bajarilgunga qadar qolganlari o‘z navbatini kutadi.

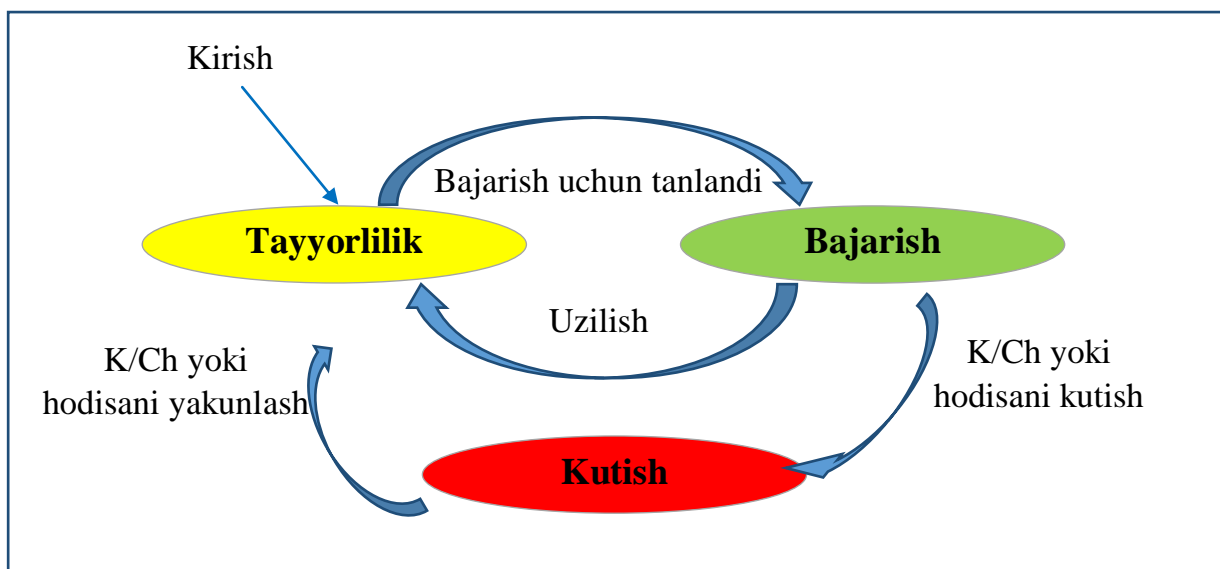
Har bir jarayon kamida ikki holatdan iborat bo‘lishi mumkin: *bajariluvchi jarayonlar va bajarilmaydigan jarayonlar*. 2.1- rasmda *jarayon holat diagrammasi* modeli ifodalangan.



2.1- rasm. Jarayon holat diagrammasi modeli

Bajariluvchi jarayon holatida joylashgan jarayon ma‘lum vaqtdan so‘ng operatsion tizimni tugatishi yoki ma‘lum vaqtga to‘xtashi va bajarilmaydigan jarayon holatiga o‘tishi mumkin. Ma‘lum vaqtga to‘xtash jarayoni ikki sabab tufayli yuz berishi mumkin: ishni davom ettirish uchun qandaydir harakat (masalan, kiritish/chiqarish operatsiyasini yakunlash) yoki ushbu jarayonda ishlash uchun operatsion tizimda ajratilgan vaqt oralig‘ining tugashi. Shundan so‘ng belgilangan algoritm bo‘yicha operatsion tizim bajarilmaydigan jarayon holatida joylashgan jarayonlardan birini bajarish uchun

bittasini tanlaydi. Va uni bajariluvchi jarayon holatiga o'tkazadi. Tizimda paydo bo'lgan yangi jarayon dastlab bajarilmaydigan jarayon holatida joylashadi. Bu juda ham qo'pol model bo'lib bajarish uchun jarayonlarni tanlashda u ma'lum vaqtga to'xtashi va real holatda bajarish uchun tayyor bo'lmasligi bois kutish holatida bo'lishi mumkin. Bajarilmaydigan jarayon holatidan chiqib ketish uchun ikkita yangi holatdan foydalaniladi: tayyorlilik va kutish (2.2- rasm).



2.2- rasm. Tayyorlilik va kutish holatlari

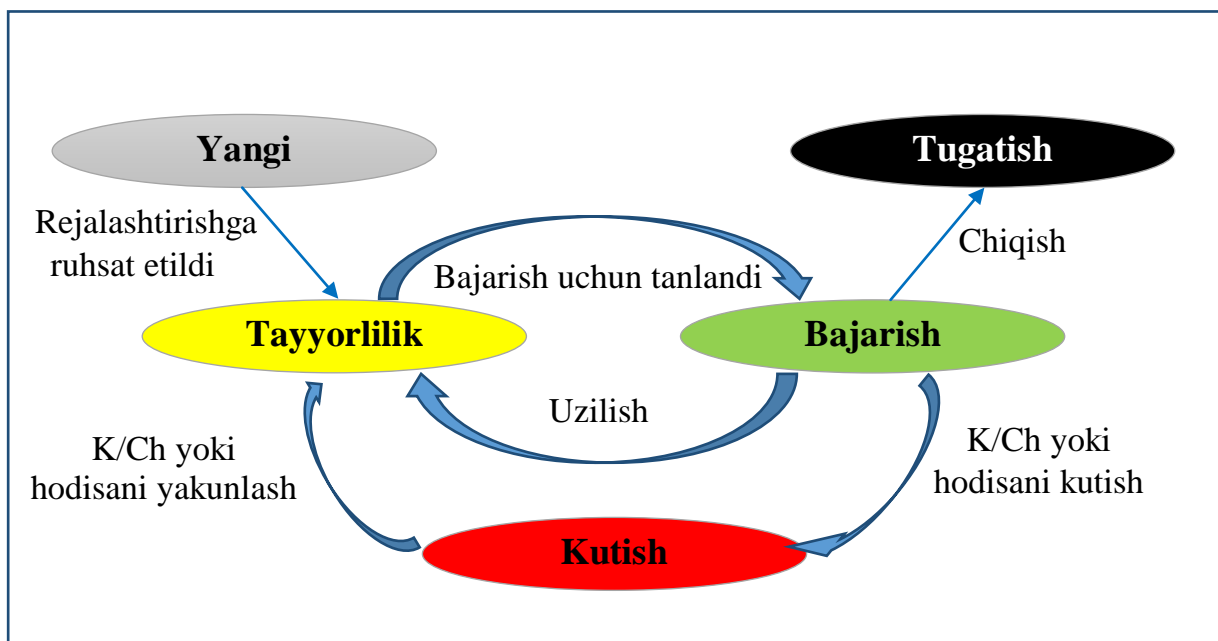
Tizimda paydo bo'lgan har qanday jarayon tayyorlilik holatida bo'ladi. Operatsion tizim tayyor jarayonlardan birini tanlab foydalanadigan algoritmnini rejalashtiradi va uni bajarish holatiga o'tkazadi. Bajarish holatida jarayonni dasturiy kodini bevosita bajarishga o'tkaziladi. Bu jarayon holatidan chiqish uchun uchta sabab bo'lishi mumkin:

- ❖ operatsion tizim uning mavjudligini tugatishi;
- ❖ ayrim voqealar sodir bo'lmasligi uchun u o'z ishini davom ettirmasligi mumkin va operatsion tizim uni kutish holatiga o'tkazadi;
- ❖ uni tayyorlilik holatiga qaytishda hisoblash tizimlarida uzilishlarning paydo bo'lishi natijasida (masalan, bajarilish uchun ajratilgan vaqtni tugab qolishi sababli vaqtni uzilishi).

Jarayon kutish holatidan tayyorlilik holatiga o'tgandan so'ng u qaytadan bajarish uchun tanlanishi mumkin.

2.3- rasmda yana ikkita jarayon holati kiritilgan: yangi (jarayon tug'ilishi) va tugatish (bajarishni tugallanishi).





2.3- rasm. Tug‘ilish va bajarishni yakunlash

Endilikda jarayon hisoblash tizimida paydo bo‘lishi uchun tug‘ilish holati orqali o‘tishi kerak bo‘ladi.

2.3- rasmdan ko‘rinib turibdiki, tizimda yaratilgan yangi jarayon rejalash-tirishga ruhsat etish bosqichidan o‘tadi. OT tizimdagi barcha jarayonlarning navbatiga kiradi, shundan so‘ng OT uni bajarishga tayyor holatga keltiradi. Shuni belgilaymizki, bajarishga tayyor bo‘lgan jarayonlar navbati – jarayonlarni boshqarish uchun eng ko‘p ishlatiladigan tizim tuzilmalaridan biri. Bajarilishga tayyor holatdan bajarilish holatiga o‘tish jarayoni rejalashtirish asosida protsessorga vaqt kvantini ajratish natijasida OT rejalashtiruvchisi tomonidan uzatiladi. Jarayon bajarilish paytida to‘xtatilishi mumkin (taymer tomonidan, xatolik natijasida va hokazo) va uzilishni qayta ishlaganidan so‘ng, operatsion tizim yana ishlashga tayyor holatga qaytadi. Agar jarayonda sinxron kiritish-chiqirish amalga oshirilsa yoki jarayon biron bir hodisa sodir bo‘lishini kutishi kerak bo‘lsa (masalan, ma‘lum bir vaqtda), jarayon kutish holatiga o‘tadi. Kiritish/chiqarish (I/O) tugallanganda yoki kutilayotgan hodisa yuz berganda, protsessor vaqt kvantini darhol qabul qilmaydi, lekin bajarishga tayyor holatga o‘tadi. Jarayon dasturi tugashi bilan jarayon tugallangan holatga kiradi, masalan, exit(c) tizim chaqirig‘ining natijasida, c - chiqish kodi. Agar c=0 bo‘lsa, jarayon muvaffaqiyatli yakunlandi deb hisoblanadi. Masalan, Windows NT operatsion tizimi

uchun jarayon holati modeli 7 ta, Unix operatsion tizimi esa 9 ta turli xil jarayon holati modelini o'z ichiga oladi.

### ***Jarayon ichidagi operatsiyalar va ular bilan bog'liq tushunchalar***

Jarayon bir holatdan boshqa bir holatga mustaqil ravishda o'ta olmaydi. Jarayon holatlarini o'zgartirish, operatsion tizimga kiradigan operatsiyalarda amalga oshiriladi. Bizning modelimiz bunday operatsiyalarning soni hozirchalik holat diagrammasidagi strelkalar soniga teng. Operatsiyani uchta juft bilan bog'lash mumkin:

❖ jarayonni yaratish – jarayonni tugatish;

❖ jarayonnima'lumbirvaqtgato'xtatish

(bajarilishholatidantayyorlilikholatigao'tish ) – jarayonniishgatushirish (tayyorlilikholatidanbajarilishholatigao'tish).

❖ Jarayonni bloklash (bajarilish holatidan kutish holatiga o'tish) – bloklangan jarayonni ochish (kutish holatidan tayyorlilik holatiga o'tish).

Algoritmni rejalashtirish haqida gapirar ekanmiz, bizning modelimizda o'z juftiga ega bo'lmagan yana bir operatsiya paydo bo'ladi: jarayon muximligini o'zgartirish.

Operatsiyalarni tashkil qilish va tugatish jarayoni bir martalik bo'ladi. Va u jarayondan faqat bir marotaba foydalaniladi (hisoblash tizimi jarayonida ayrim tizimlar hech qachon ishlashdan to'xtamaydi). Jarayon holatini o'zgartirish bilan bog'liq qolgan barcha operatsiyalar ko'p martalik hisoblanadigan ishga tushirish yoki bloklash bo'ladi.

### ***Jarayonni boshqarish tizim osti***

Asosiy tushunchalar:

Jarayon tizim resurslaridan foydalanishda ba'zi talabnomalarni aniqlashi mumkin.

❖ Multidasturli operatsion tizimlar bir vaqtda bir nechta jarayonlarni bajarishi mumkin.

❖ Jarayonlar ko'pincha foydalanuvchi tashabbusi va uning ilovalari bo'yicha paydo bo'lishi mumkin, va bu foydalanuvchi jarayoni deyiladi.

❖ O'zining funksiyalarini bajarish uchun operatsion tizim tashabbusi bilan bajariladigan jarayonlar tizim jarayoni deb ataladi.

❖ Jarayonga ajratilgan operatsion tizim tezkor xotirasi sohasi majmui, manzil maydoni deb ataladi.

### ***Jarayonni boshqarish tizim ostining asosiy funksiyasi:***

- ❖ Jarayonni yaratish va yo‘q qilish (jarayon bilan bog‘liq bo‘lgan ma‘lumotlar tuzilishi);
- ❖ Resurslarga jarayon talabi navbatini qo‘llab-quvvatlash;
- ❖ Birgalikda resurslardan foydalanishga tashkil qilingan boshqa jarayonlardan joriy jarayonga ajratilgan resurslarni himoya qilish;
- ❖ Ayrim jarayonlarni to‘xtalishi va yangidan paydo bo‘lishini ta‘minlash;
- ❖ Jarayonlarni bog‘lanish vositasini taqdim etish.

### ***Jarayonni boshqarish bloki***

Jarayonni boshqarish bloki (Process Control Block – PCB) – bu har bir jarayon bilan bog‘liq quyidagi ma‘lumotlarni o‘z ichiga olgan jarayonni boshqarish uchun foydalanadigan tizim ma‘lumotlari tuzilishi hisoblanadi:



2.4- rasm. Jarayonni boshqarish bloki tuzilishi

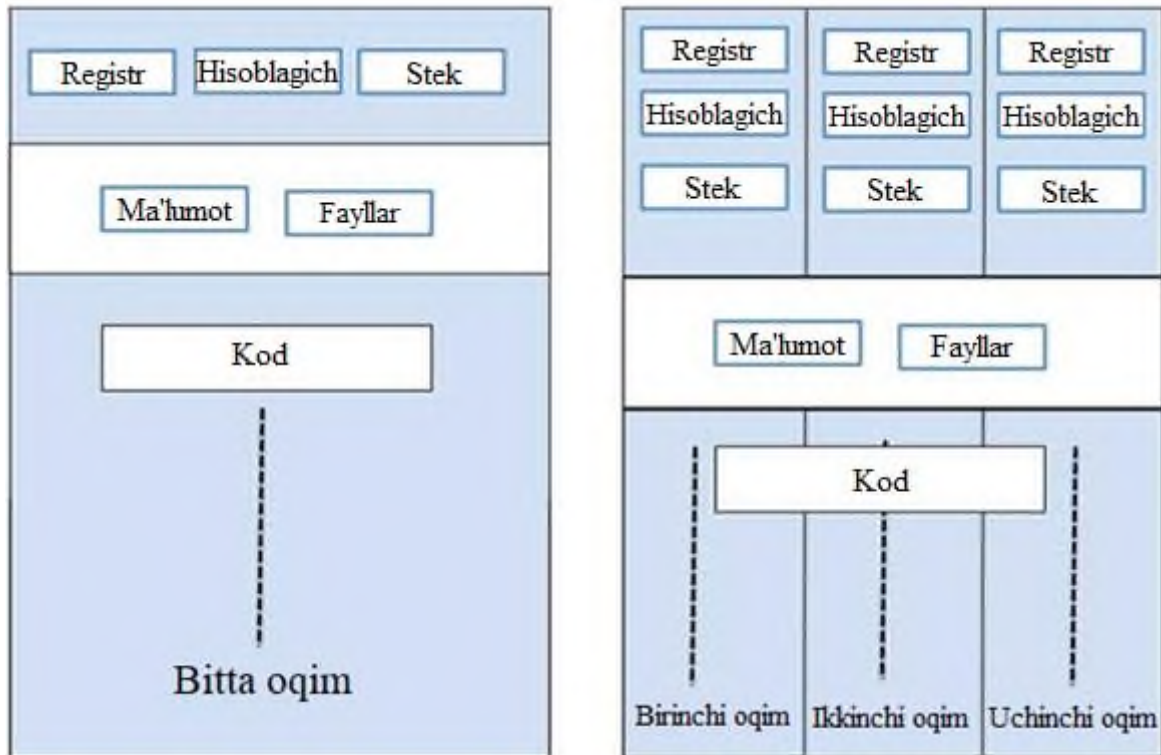
- ❖ Jarayon identifikator raqami (ID si);
- ❖ Jarayon holati;
- ❖ Buyruq hisoblagichining joriy qiymati (jarayonni bajarilishini davom ettirishda foydalaniladi);
- ❖ Protessor registri qiymatlari (jarayonni qayta tiklashda ham foydalaniladi);

- ❖ Protsessorni rejalashtirish uchun ma'lumot (jarayon stekiga ko'rsatgich, jarayon raqami);
- ❖ Xotirani boshqarish uchun ma'lumot (jarayon xotira maydonining chegaralari);
- ❖ Resurslar (jarayon bajarilishi uchun ajratilgan);
- ❖ Hisob ma'lumotlari (e'lon qilingan ijro vaqtidan qolgan jarayonning umumiy bajarilish vaqti, jami kiritish/chiqarish vaqti va boshqalar);
- ❖ Kiritish/chiqarish holatlari to'g'risidagi ma'lumotlar (ochiq fayllar ro'yhati).

Jarayonni boshqarish bloki tuzilishi 2.4- rasmda ko'rsatilgan.

## 2.2. Operatsion tizimdagi oqimlar va ularning turlari

Oqim – bu operatsion tizimning mohiyati, protsessorda ko'rsatmalar to'plamini bajarish jarayoni, aniqrog'i dastur kodi. Oqimlarning umumiy maqsadi ikki yoki ko'plab turli xil vazifalarni protsessorda parallel bajarishdir.



2.5- rasm. Bitta oqimli bitta jarayon, uchta oqimli bitta jarayon ko'rinishi

Oqim o'zining kod segmenti, ma'lumotlar segmenti va ochiq fayllarini teng huquqli oqimlari bilan almashadi. Agar bir oqim kod segmentining xotira elementini o'zgartirsa, qolgan barcha oqimlar buni ko'radi. Oqim yengil jarayon deb ham ataladi. Oqimlar parallellash orqali dastur samaradorligini oshirish usulini taqdim etadi. Quyidagi rasmda bir oqimli va ko'p oqimli jarayonlarning ishlashi ko'rsatilgan.

Oqimlar - bu klassik jarayonga mos keladigan ishchi oqimlarni kamaytirish hisobiga operatsion tizimning ish faoliyatini yaxshilashga qaratilgan dasturiy yondashuv.

2.1- jadval. Jarayon va oqim o'rtasidagi farq

<b>№</b>	<b>Jarayon</b>	<b>Oqim</b>
1	Jarayon og'ir yoki ko'p resurs talab qiladi	Oqim yengil va jarayonga qaraganda kamroq resurslarni talab qiladi
2	Jarayonni almashtirish operatsion tizim bilan o'zaro aloqani talab qiladi	Oqimlarni almashtirish operatsion tizim bilan o'zaro aloqani talab qilmaydi
3	Bir nechta qayta ishlash muhitida har bir jarayon bir xil kodni bajaradi, ammo o'z xotirasi va fayl resurslariga ega	Barcha oqimlar farzand jarayonini bir xil ochiq fayllar to'plamini bo'lishishi mumkin
4	Agar bitta jarayon bloklangan bo'lsa, unda birinchi jarayon blokdan chiqarilmaguncha boshqa jarayon bajarilmaydi	Bitta oqim bloklangan va kutayotgan paytda, xuddi shu vazifadagi ikkinchi oqim ishlashi mumkin
5	Bir nechta jarayonlar oqimlardan foydalanmasdan ko'proq resurslardan foydalanadi	Ko'p oqimli jarayonlar kamroq resurslardan foydalanadi
6	Bir nechta jarayonlarda har bir jarayon mustaqil ravishda amalga oshiriladi	Bitta oqim boshqa oqim ma'lumotlarini o'qishi, yozishi yoki o'zgartirishi mumkin

Har bir oqim aniq bitta jarayonga tegishli va jarayondan tashqarida hech qanday oqim mavjud bo'lmaydi. Har bir oqim alohida boshqaruv oqimini anglatadi. Oqimlar tarmoq serverlari va veb-

serverlarni amalga oshirishda muvaffaqiyatli ishlatilgan. Ular shuningdek, umumiy xotiraga ega bo'lgan ko'p protsessorlarda ilovalarni parallel ravishda bajarish uchun munosib asosni ta'minlaydi.

### ***Oqim afzalliklari***

- ❖ Oqimlar kontekstni almashtirish vaqtini kamaytiradi;
- ❖ Oqimlardan foydalanish jarayon ichida parallellikni ta'minlaydi;
- ❖ Samarali aloqa;
- ❖ Kontekst oqimlarini almashtirish va yaratishda yanada tejamkor;
- ❖ Oqimlar ko'p protsessorli arxitekturalardan keng miqyosda va samarali foydalanishga imkon beradi.

### ***Oqim turlari***

Oqimlar quyidagi ikki usulda amalga oshiriladi:

- ❖ **Foydalanuvchi darajasidagi oqimlar** (User Level Threads) - foydalanuvchi boshqaradigan oqimlar;
- ❖ **Yadro darajasidagi oqimlar** (Kernel Level Threads) – yadroda ishlaydigan operatsion tizim boshqaradigan oqimlar, operatsion tizim yadrosi.

### ***Foydalanuvchi darajasidagi oqimlar***

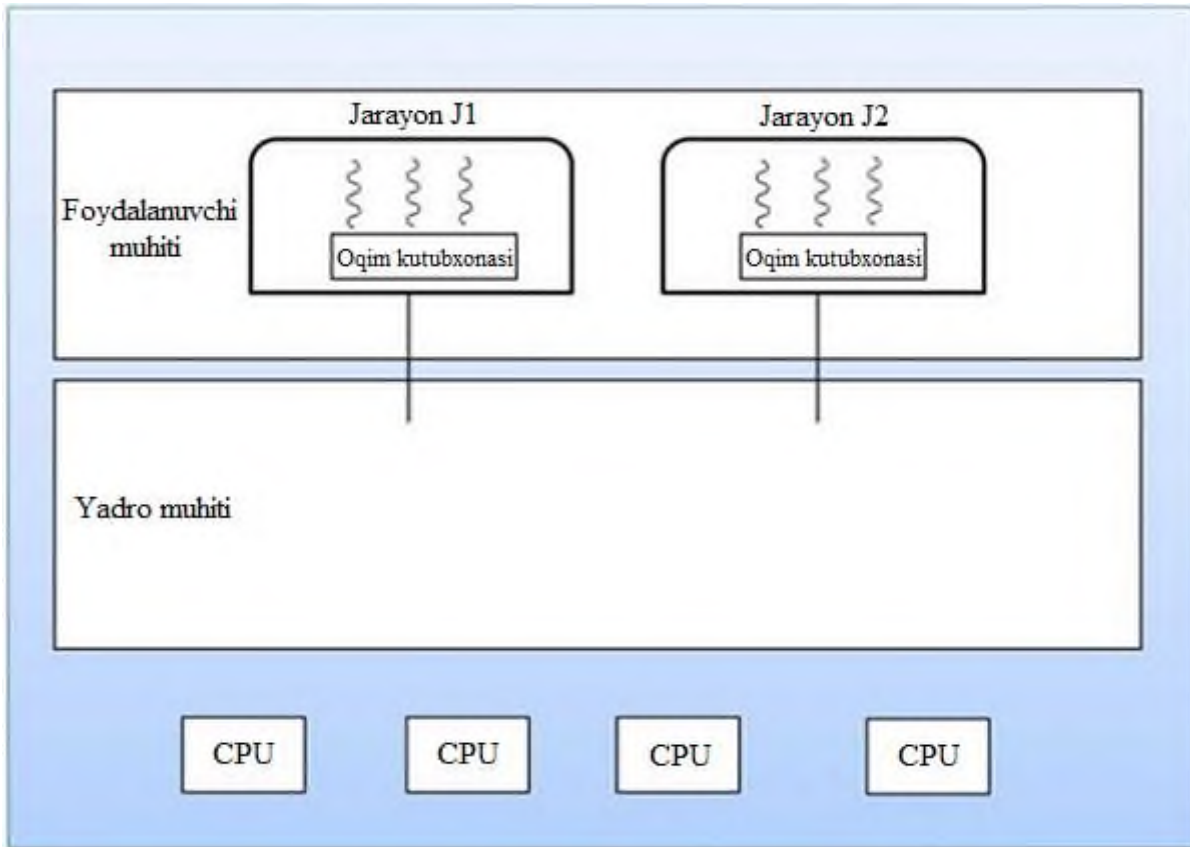
Bunday holda, oqimlarni boshqarish yadrosi oqimlarning mavjudligi haqida xabardor bo'lmaydi. Oqimlar kutubxonasi oqimlarni yaratish va yo'q qilish, xabarlar va ma'lumotlarni oqimlar o'rtasida uzatish, oqimlarni bajarishni rejalashtirish va oqim kontekstlarini tiklash va saqlash kodlarini o'z ichiga oladi.

Foydalanuvchi darajasidagi oqimlarning afzalliklari:

- ❖ Oqimni almashtirish yadro rejimi imtiyozlarini talab qilmaydi;
- ❖ Foydalanuvchi darajasidagi oqim har qanday operatsion tizimda ishlashi mumkin;
- ❖ Rejalashtirish foydalanuvchi darajasidagi oqimga xos bo'lgan dastur bo'lishi mumkin;
- ❖ Foydalanuvchi darajasidagi oqimlar tezda yaratiladi va boshqariladi.

Foydalanuvchi darajasidagi oqimlarning kamchiliklari:

- ❖ Oddiy operatsion tizimda tizim qo‘ng‘iroqlarining aksariyati bloklanadi;
- ❖ Ko‘p oqimli ilovalar ko‘p protsessorli ishlov berishning afzalliklaridan foydalana olmaydi.



2.6- rasm. Foydalanuvchi darajasidagi oqimlar

### ***Yadro darajasidagi oqimlar***

Bunday holda, oqimni boshqarish yadro tomonidan amalga oshiriladi. Ilova sohasida oqimlarni boshqarish kodi mavjud emas. Yadro oqimlari to‘g‘ridan-to‘g‘ri operatsion tizim tomonidan qo‘llab-quvvatlanadi. Har qanday ilovani ko‘p oqimli dasturlash mumkin. Ilovadagi barcha oqimlar bitta jarayonda qo‘llab-quvvatlanadi.

Yadro butun jarayon uchun va jarayon tarkibidagi alohida oqimlar uchun kontekst ma‘lumotlarini saqlab turadi. Yadro tomonidan rejalashtirish oqimlar asosida amalga oshiriladi. Yadro oqimlarni yaratish, rejalashtirish va boshqarishni yadro sohasida

amalga oshiradi. Yadro oqimlari odatda foydalanuvchi oqimlariga qaraganda sekinroq yaratiladi va boshqariladi.

Afzalliklari

❖ Yadro bir vaqtning o'zida bir xil jarayondan bir nechta jarayonga bir nechta oqimlarni rejalashtirishi mumkin;

❖ Agar jarayonda bitta oqim bloklangan bo'lsa, yadro xuddi shu jarayonda boshqa oqimni rejalashtirishi mumkin;

❖ Yadro modulini o'zi ko'p oqimli qilishi mumkin;

Kamchiliklari

❖ Yadro oqimlari odatda foydalanuvchi oqimlariga qaraganda sekinroq yaratiladi va boshqariladi;

❖ Xuddi shu jarayon davomida boshqaruvni bitta oqimdan boshqasiga o'tkazish, yadro rejimiga o'tishni talab qiladi.

### ***Ko'p oqimli model***

Ba'zi bir operatsion tizimlar birlashtirilgan foydalanuvchi darajasidagi oqimlar va yadro darajasidagi oqimlarni birlashtiradigan vositani ta'minlaydi. Solaris ushbu birlashtirilgan yondashuvga yaxshi misol hisoblanadi. Birlashtirilgan tizimda bitta dasturdagi bir nechta oqimlar bir nechta protsessorlarda parallel ravishda ishlashi mumkin, va bloklash tizim chaqiruvini butun jarayonni bloklamasligi kerak.

Ko'p oqimli modellar uch turga bo'linadi:

❖ Many to many modeli;

❖ Many to one modeli;

❖ One to one modeli.

### ***Many to many modeli***

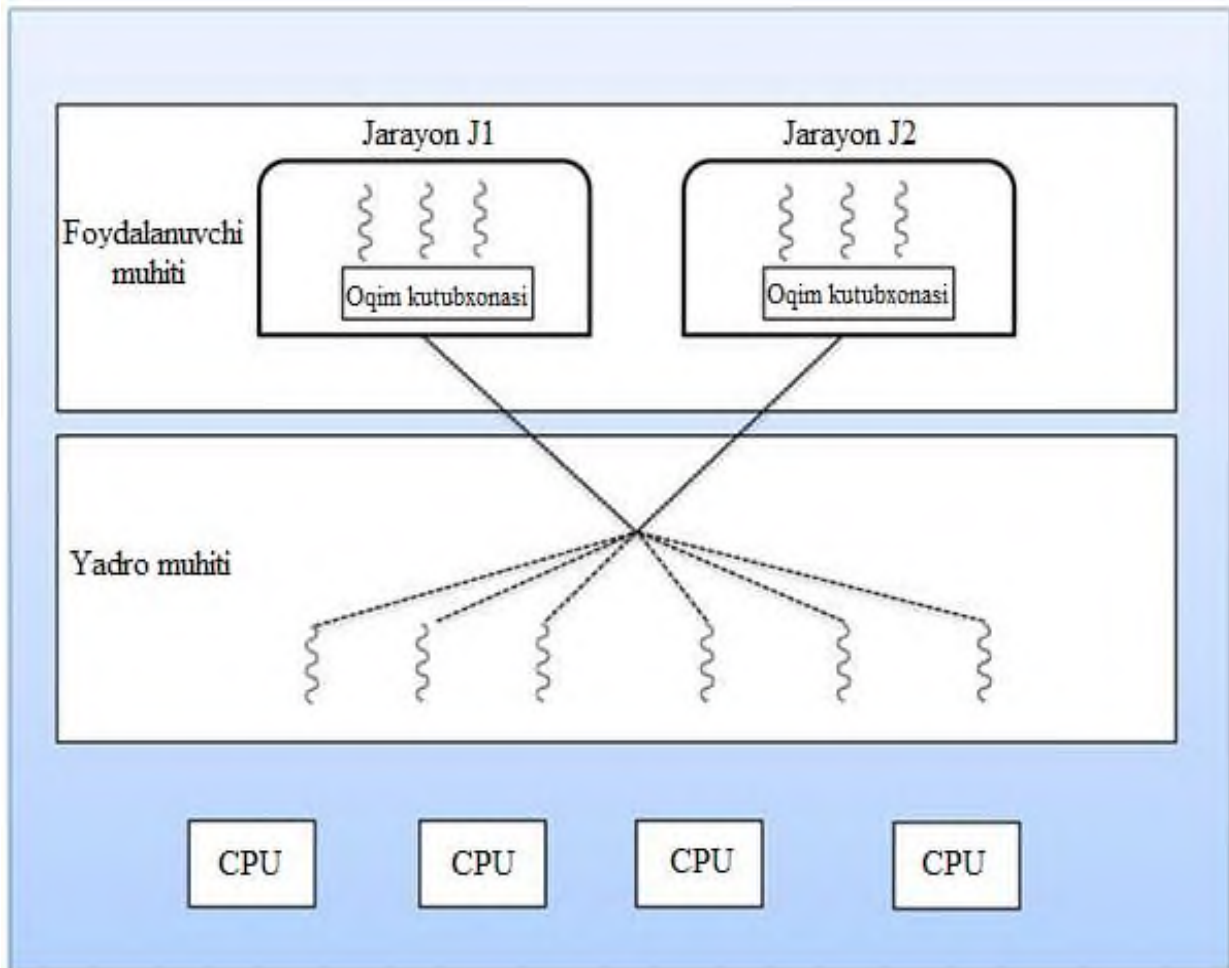
*Many to many modeliga* misol quyidagi 2.7- rasmda keltirilgan. Many to many modeli har qanday foydalanuvchi oqimlarini teng yoki kichik sonli yadro oqimlariga ko'paytiradi.

2.7- rasmda ko'p oqimli model ko'rsatilgan, unda 6 foydalanuvchi darajasidagi oqimlar 6 yadro darajasidagi oqimlar bilan ko'paytiriladi.

Ushbu modelda ishlab chiquvchilar, qancha foydalanuvchi oqimlari kerak bo'lsa, shuncha yaratishi mumkin va mos keladigan yadro oqimlari ko'p protsessorli mashinada parallel ravishda ishlashi



mumkin. Ushbu model parallellash bilan eng yaxshi aniqlikni ta'minlaydi, va oqim bloklash tizim chaqirig'i amalga oshirganda, yadro yana bitta oqimni bajarishi mumkin.



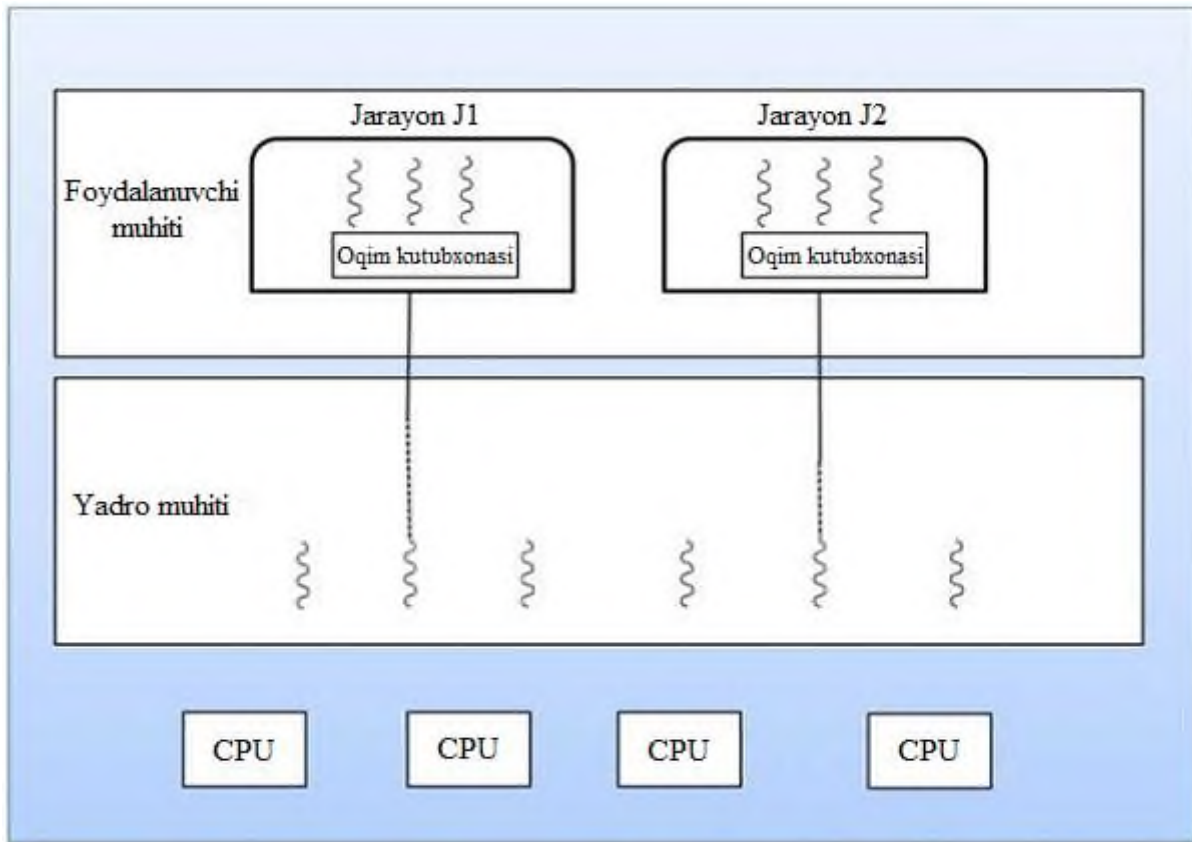
2.7- rasm. Many to many modeli

### ***Many to One Modeli***

*Many to one modeli* ko'p foydalanuvchi darajasidagi oqimlarni bitta yadro darajasidagi oqimlar bilan taqqoslaydi. Oqimlarni boshqarish foydalanuvchi maydonida oqimlar kutubxonasi tomonidan amalga oshiriladi.

Agar oqim bloklash tizim chaqiruvini amalga oshirsa, butun jarayon bloklanadi. Bir vaqtning o'zida faqat bitta oqim yadroga kirishi mumkin, shuning uchun ko'p protsessorli tizimlarda bir nechta oqimlar parallel ravishda ishlay olmaydi. Agar foydalanuvchi darajasidagi oqimlar kutubxonalari operatsion tizimda tizim ularni

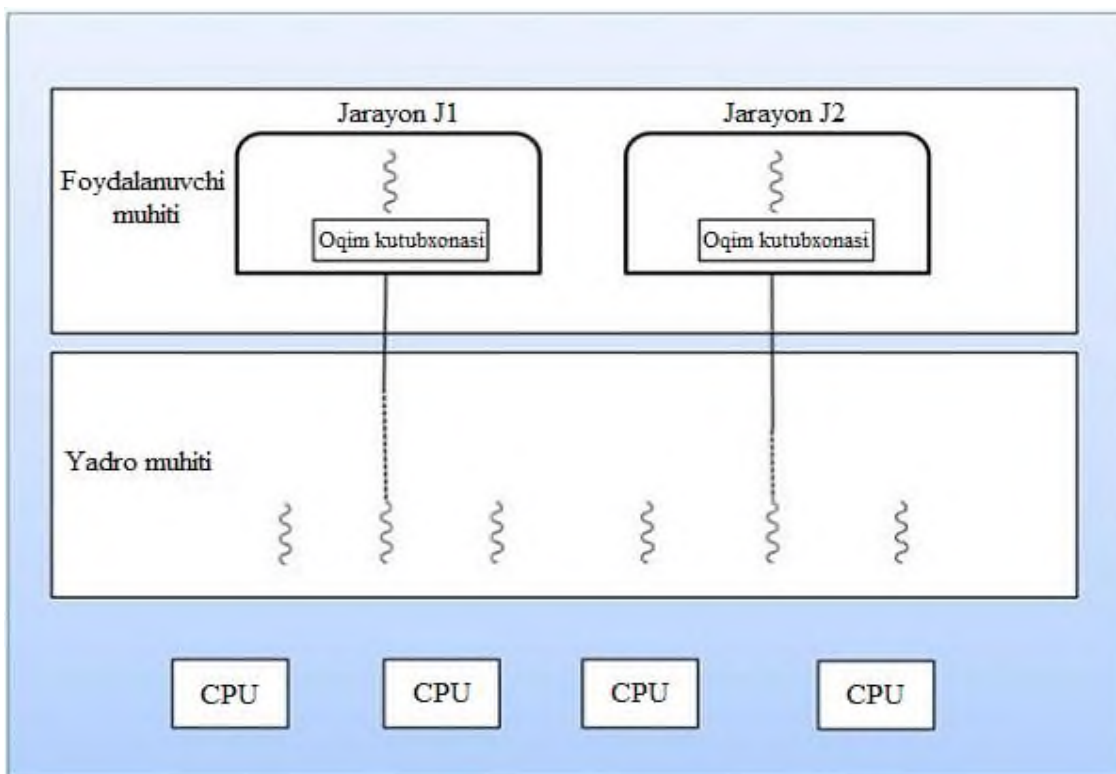
qo‘llab-quvvatlamaydigan tarzda amalga oshirilsa, unda yadro oqimi many to one modelini qo‘llaydi.



2.8- rasm. Many to One modeli

### ***One to One Modeli***

Foydalanuvchi darajasidagi oqim va yadro darajasidagi oqim o‘rtasida one to one munosabatlar mavjud. Ushbu model many to one modelga qaraganda ko‘proq parallellashni ta‘minlaydi. Bundan tashqari, u bloklash tizim chaqirig‘i amalga oshirilganda, boshqa oqimni ishga tushirishga imkon beradi. Mikroprotssessorlarda parallel bajarilishi uchun bir nechta oqimlarni qo‘llab-quvvatlaydi. Ushbu modelning kamchiligi shundaki, foydalanuvchi oqimini yaratish uchun yadroga mos keladigan oqim kerak bo‘ladi. OS/2, Windows NT va Windows 2000 Otlari one to one modelidan foydalanadi.



2.9- rasm. One to One modeli

***Foydalanuvchi darajasi va yadro darajasidagi oqim o'rtasidagi farq***

2.2- jadval

№	<b>Foydalanuvchi darajasidagi oqimlar</b>	<b>Yadro darajasidagi oqimlar</b>
1	Foydalanuvchi darajasidagi oqimlar tezroq yaratiladi va boshqariladi	Yadro darajasidagi oqimlar ekinroq yaratiladi va boshqariladi
2	Foydalanuvchi darajasidagi oqim kutubxonasi tomonidan amalga oshiriladi	Operatsion tizim yadro oqimlarini yaratishni qo'llab-quvvatlaydi
3	Foydalanuvchi darajasidagi oqim umumiydir va har qanday operatsion tizimda ishlashi mumkin	Operatsion tizim uchun o'ziga xos yadro darajasidagi oqim
4	Ko'poqimli ilovalar ko'pprotsessorliqay taishlashimkoniyatidan foydalana olmaydi	Yadro modulini o'ziko'poqimlibo'lishim mumkin

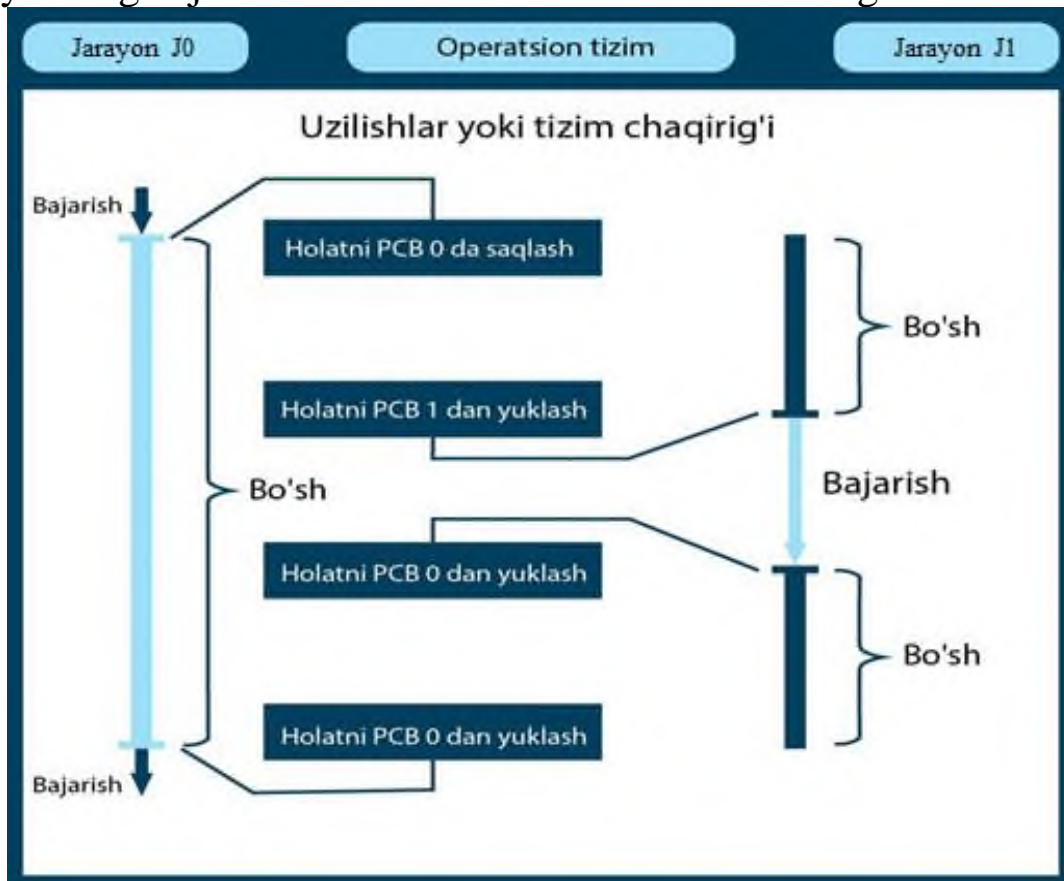
### 2.3. Operatsion tizimda vazifalar va jarayonlarni rejalashtirish

#### *Bir jarayondan boshqasiga o'tish*

Jarayonlarni boshqarishda operatsion tizim ularning ketma-ket bajarilishini ta'minlaydi. Ushbu vazifa OT rejalashtiruvchisi tomonidan hal qilinadi. Protsessorni bir jarayondan boshqa jarayonga o'tkazish sxemasi 2.10- rasmda keltirilgan.

Rasmda ikkita jarayon ko'rsatilgan - J0 va J1.

J0 jarayonida uzilishlar yoki tizim chaqirig'i bo'lsa, operatsion tizim o'z holatini PCBda saqlaydi va protsessorni J1 jarayoniga o'tkazadi (J1 jarayoni ham shunga o'xshash tarzda ishlanadi). Har bir jarayonning bajarilish va to'xtash muddatlari ko'rsatilgan.



2.10- rasm. Protsessorni bir jarayondan boshqa jarayonga o'tkazish sxemasi

#### *Jarayonlarni rejalashtirish bilan bog'liq navbatlar*

Jarayonlarni boshqarish uchun OT quyidagi navbatlarni tashkil qiladi:

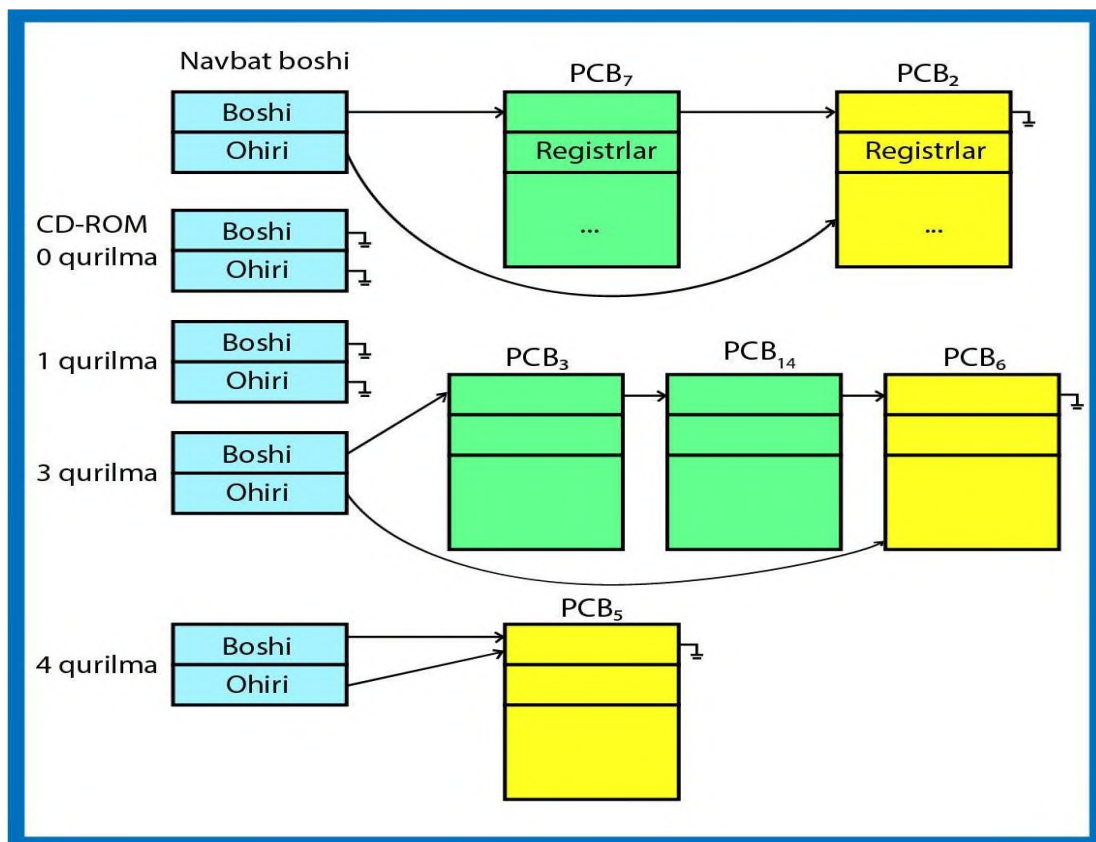
❖ **Vazifalar navbati** (job queue) – tizimdagi barcha jarayonlar to‘plamini o‘z ichiga oladi. Har bir yangi jarayon unga kelib tushadi va butun tizim davomida unda qoladi.

❖ **Tayyor jarayonlar navbati** (ready queue) – asosiy xotirada joylashgan, tayyor va bajarilishini kutayotgan barcha jarayonlar to‘plamini o‘z ichiga olgan, eng ko‘p ishlatiladigan va o‘zgartiriladigan navbat. Unga har bir yangi jarayon, shuningdek, kiritish/chiqarishni tugallanishini kutayotgan har bir jarayon kelib tushadi.

❖ **Qurilmalar navbati** (device queue) – kiritish/chiqarish qurilmalarini ishini tugallanishini kutayotgan jarayonlar navbati (har bir qurilma o‘z navbatiga ega).

Operatsion tizim tomonidan jarayonlarni boshqarish va tizimdagi jarayonlarning xatti-harakatlari turli navbatlar orasidagi ko‘chish sifatida ko‘rib chiqilishi mumkin.

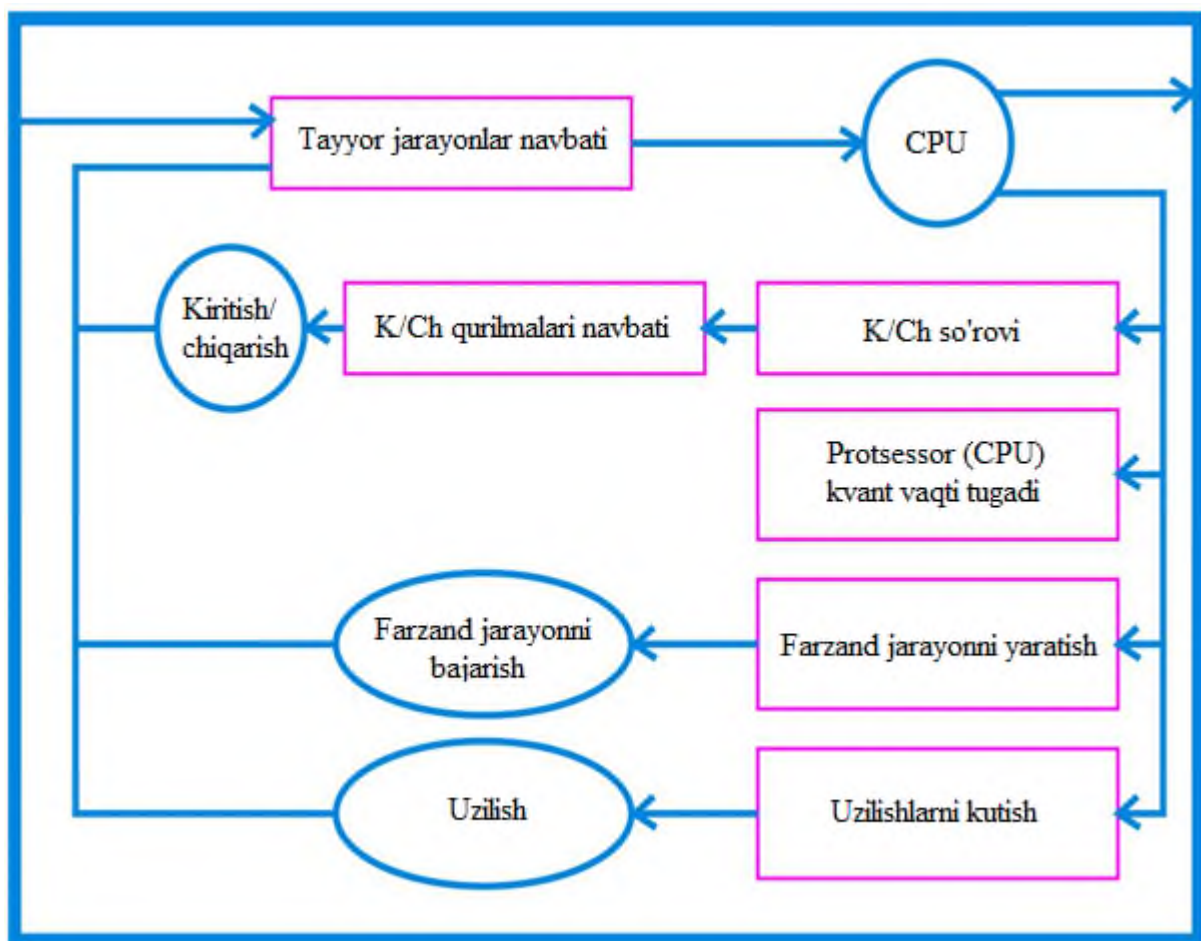
2.11- rasmda tayyor jarayonlar va kiritish/chiqarish qurilmalari uchun navbat-ning diagrammasi ko‘rsatilgan.



2.11- rasm. Tayyor jarayonlar navbati va kiritish/chiqarish qurilmalari navbati

Jarayonlarni rejalashtirish va navbatlar bilan ishlashi quyidagi 2.12- rasmda keltirilgan. Har bir to‘rtburchak navbatni anglatadi. Ikki turdagi navbatlar mavjud: tayyor jarayonlar navbati va qurilmalar navbati. Doiralalar navbatlarga xizmat qiladigan resurslarni anglatadi va strelkalar tizimdagi jarayonlar oqimini ko‘rsatadi. Dastlab yangi jarayon tayyorlik navbatiga qo‘yiladi. U bajarilgunga qadar yoki ubajarilish uchun yuborilgunga qadar u yerda kutadi. Jarayon protsessor (CPU) bajarilgandan so‘ng, bir nechta hodisalardan biri sodir bo‘lishi mumkin:

- ❖ Jarayon so‘rovni K/Ch qurilmasiga yuborishi, va keyin uni K/Ch navbatiga qo‘yishi mumkin.
- ❖ Jarayon yangi farzand jarayonini yaratishi va uning tugatilishini kutishi mumkin.
- ❖ Jarayon uzilishlar natijasida protsessoridan majburiy ravishda olib tashlanishi va yana tayyorlik navbatiga qo‘yilishi mumkin.



2.12- rasm. Jarayonni rejalashtirishning grafik ko‘rinishi

Tayyor jarayonlarning navbatidan boshlab, har bir jarayon oxir-oqibat protsessorga kelib tushadi va bajariladi. Bajarishda quyidagi holatlar yuzaga kelishi mumkin, ya'ni, bajarilish to'xtatilishi mumkin: sinxron kiritish/chiqarish so'rovi, jarayonga ajratilgan vaqt miqdorining tugashi, uzilish yoki farzand jarayonini yaratilishi. Klassik UNIX sxemasida, farzand jarayoni tizimni **fork** bilan chaqirish orqali yaratilganda, farzand jarayoni uchun yangi virtual xotira maydoni yaratiladi, unga ona xotirasi tarkibi ko'chiriladi. Shundan so'ng, ona jarayoni farzand jarayoniga yo'l beradi. Farzand jarayonini yaratish uchun bunday sxema, albatta, har doim ham qulay emas.

### ***Jarayonlarni rejalashtirishni amalga oshiradigan rejalashtiruvchilar***

Operatsion tizimda jarayonlarni rejalashtirish odatda bir nechta rejalashtiruvchilar tomonidan amalga oshiriladi, ularning har biri o'z chaqiruvlar davriga va o'zi hal qiladigan o'ziga xos vazifasiga ega.

Rejalashtiruvchi (**scheduler**)bu – maxsus tizim dasturi bo'lib, jarayonlarni turli usullar bilan rejalashtirishni amalga oshiradi. Asosiy maqsadi vazifani tizimga taqdim qilish uchun tanlash va qaysi jarayonlarni bajarishni hal qiladi.

Rejalashtiruvchilar 3 turda bo'ladi:

- ❖ Uzoq muddatli rejalashtiruvchi (**Long-Term Scheduler**);
- ❖ O'rta muddatli rejalashtiruvchi (**Medium-Term Scheduler**);
- ❖ Qisqa muddatli rejalashtiruvchi (**Short-Term Scheduler**).

**Uzoq muddatli rejalashtiruvchi** (vazifalarni rejalashtiruvchi) – qaysi jarayonlarni tayyor jarayonlar navbatiga o'tkazish kerakligini belgilaydi.

**Qisqa muddatli rejalashtiruvchi** (protsessorni rejalashtiruvchi) – keyingi bajarilishi kerak bo'lgan jarayon va qaysi jarayonlar protsessor bilan ta'minlanishi kerakligini belgilaydi.

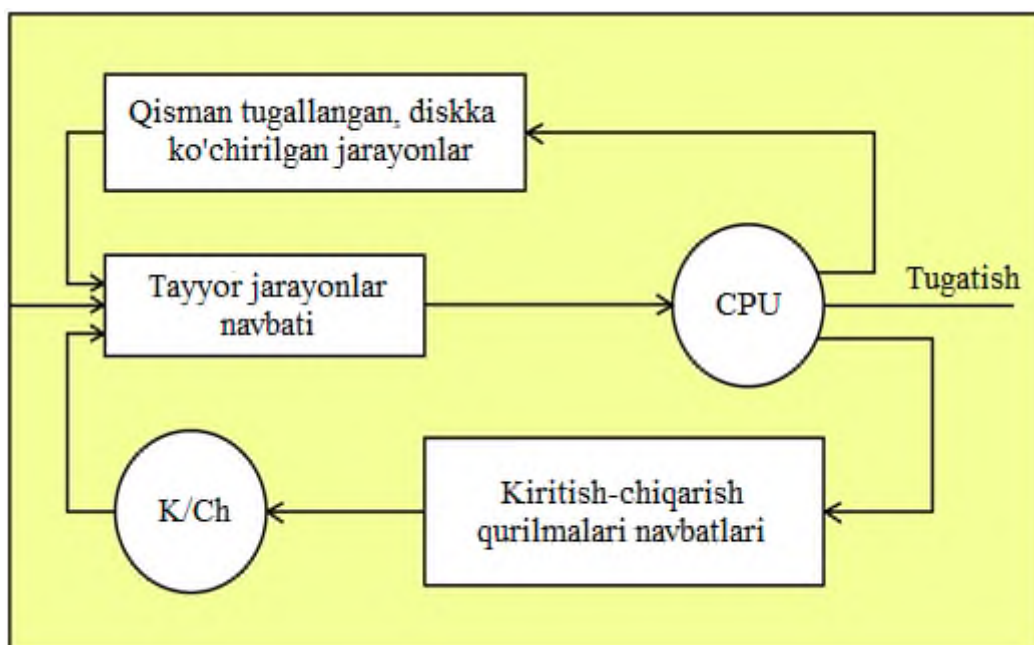
**O'rta muddatli rejalashtiruvchi** (jarayonlarni almashtirishni rejalashtiradi) – vaqtni taqsimlash rejimini amalga oshirish uchun tizimga yozishni va almashtirishni rejalashtiruvchi qo'shilishi mumkin, u qaysi foydalanuvchi jarayonlarini xotiraga va diskka ko'chirish (**swapping**) kerakligini belgilaydi.

Rejalashtiruvchilar va jarayonlarning xususiyatlari. Har bir rejalashti-ruvchining, shuningdek har bir jarayonning o'ziga xos xususiyatlari bor.

Qisqa muddatli rejalashtiruvchi juda tez-tez chaqiriladi, masalan, kamida keyingi protsessor kvant vaqtining tugashi, shuning uchun u juda tez va samarali amalga oshirilishi kerak.

Uzoq muddatli rejalashtiruvchi nisbatan kamdan-kam hollarda chaqiriladi, chunki tizim jarayonni tayyor jarayonlar holatiga o'tkazish to'g'risida qaror qabul qilmaydi. Shuning uchun u nisbatan sekin bajarilishi mumkin, unchalik samarali amalga oshirilmaydi.

Bunday rejalashtiruvchini o'z ichiga olgan tizimning sxemasi quyidagi 2.13- rasmda keltirilgan.



2.13- rasm. Foydalanuvchi jarayonlarini xotiraga va diskka ko'chirish sxemasi

Ammo, umuman olganda tizimning asosiy vazifasi - iloji boricha ko'proq jarayonlarga xizmat ko'rsatish, aynan uzoq muddatli rejalashtirish multidasturlash darajasini (koeffisientini) – tizimning vaqt birligida xizmat qiladigan jarayonlar sonini aniqlaydi.

Jarayonlarning o'zi ham ularni rejalashtirish nuqtai nazaridan boshqacha harakat qilishi mumkin. Jarayonlar quyidagicha bo'lishi mumkin:



❖ Kiritish/chiqarishga yo‘naltirilgan (I/O bilan bog‘langan - **(I/O bound)**) – hisoblashga qaraganda kiritish/chiqarishga ko‘proq vaqt sarflaydigan jarayonlar. Bunday jarayonlar odatda protsessor vaqtining ko‘pgina qisqa kvantlarini sarflaydi.

❖ Protsessordan foydalanishga yo‘naltirilgan (CPU bilan bog‘langan – **CPU bound**) - bu ko‘p vaqtini hisoblash uchun sarflaydigan jarayonlardir. Bunday jarayonlar protsessor vaqtining oz sonli uzoq muddatli kvantlarini sarflaydi.

### ***Kontekst jarayoni***

Jarayonni boshqarish blokiga mo‘ljallangan axborotlarni saqlash uchun uni ikki qismga ajratib ishlatish qulayroq. Protsessorning o‘z ichiga olgan barcha registrlar (dasturiy hisoblagich belgilarini o‘z ichiga olgan bo‘ladi) kontekstli jarayon registrlari deb ataladi, qolganlari esa – tizimli kontekstli jarayon deb ataladi. Registr va kontekstli jarayon tizimini bilish operatsion tizimda uning ishlarini amalga oshirishni boshqarish uchun yetarli hisoblanadi. Biroq bu jarayonni to‘liq tavsiflash uchun yetarli emas. Operatsion tizimni aynan jarayon qanday hisoblanishi ya‘ni uning manzilida qanday kod va qanday ma‘lumotlar borligi qiziqitirmaydi. Foydalanuvchi nuqtai nazaridan qaralganda, teskarisi bo‘ladi, ya‘ni ma‘lumotlar o‘zgarishini aniqlovchi registrlar kontekstning qator imkoniyatlari va olingan natija jarayon manzilini o‘z ichiga olishi ko‘proq qiziqtiradi. Jarayon manzilida joylashgan kod va ma‘lumotlarni foydalanuvchi konteksti deb ataymiz. Qisqa vaqtga qabul qiladigan registr, tizim va foydalanuvchi majmui *kontekst jarayoni* deb ataladi. Jarayonning istalgan vaqt momenti o‘zining konteksti bilan tavsiflanadi.

### ***Kontekstni almashtirish***

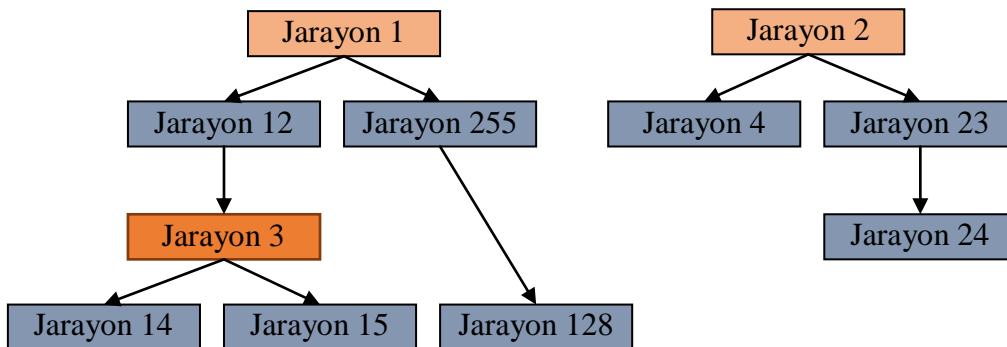
Protsessor bir jarayondan boshqa jarayonga o‘tganda jarayonni bajaruvchi kontekstni saqlashi va protsessorga ulanadigan jarayon kontekstini qayta tiklashi (jarayonlar bajarilishini to‘xtatilgan bosqichdan davom ettirish uchun) zarur. Jarayonni ishlash unumdorligini saqlash, qayta tiklash protsedurasi kontekstni almashtirish deb ataladi. Bu texnikadan foydalanib bir protsessor bir nechta jarayonlar o‘rtasida taqsimlanadi. Kontekstni almashtirishga sarflangan vaqt foydali ishlarni bajarish uchun hisoblash mashinalarida foydalanilmaydi. U mashinadan mashinagacha

o'zgaradi va odatda 1 dan 1000 mikrosekundgacha oraliqda tebranadi. O'z ichiga *threads of execution* (oqimdan foydalanish) tushunchasini olgan jarayon modelini kengaytirish yordami bilan zamonaviy operatsion tizimlarga sarflanadigan xarajatlar kamaytirilmoqda. Jarayon tushunchasi ularning resurslari bilan bog'langan bajariladigan buyruqlar to'plami va joriy vaqtda uni bajarilish bilan tavsiflanadi. Jarayon istalgan vaqt momentida registr, tizim va foydalanuvchi qismidan tashkil topgan o'zining kontekstini to'liq tavsiflaydi. Jarayonlar operatsion tizimlarda registrli va tizimli kontekstini o'z ichida ifodalagan PCB–ma'lumotlar tuzilishini belgilashni taqdim etadi. Jarayonlar beshta asosiy ko'rinishda bo'lishi mumkin: yangi (tug'ilish), tayyorlik, bajarish, kutish, tugatish. Operatsion tizimda holatdan jarayon holatiga o'tish natijasida operatsiyalar bajariladi. Operatsion tizim jarayonda quyidagi operatsiyalarni bajarishi mumkin: jarayoni yaratish, jarayonni tugatish, ma'lum bir vaqtga jarayonni to'xtatish, jarayoni ishga tushirish, jarayoni bloklash, bloklangan jarayoni qayta ochish, jarayon ustuvorligini o'zgartirish. PCBni o'z tarkibiga olgan jarayonlar o'rtasida o'zgarish bo'lmaydi. Multidasturli operatsion tizimlar ishlashi turli xil jarayonlarni bajaradigan operatsiyalarni o'tkazish zanjiridan tuzilgan va jarayonni saqlash, qayta tiklash protseduralari hamda kontekstni o'zgartirishdan iborat bo'ladi. Kontekstni o'zgartirish jarayoni amalga oshiradigan foydali ishga aloqasi bo'lmaydi va unga sarflangan vaqt protsessor ishlashi foydali vaqtini qisqartiradi.

### ***Bir marotabalik operatsiyalar***

Kompyuterda jarayonning hayoti uning tug'ilishidan boshlanadi. Istalgan operatsion tizim jarayon konsepsiyasini qo'llab-quvvatlaydi, va ularni tashkil qilish uchun zarur vositalarga ega bo'lishi kerak. Barcha jarayonlar uchun eng sodda tizim (masalan, tizim faqat aniq biror bir ilovada ishlash uchun mo'ljallangan bo'lsin), tizimning tug'ilish jarayonida turgan bo'lishi mumkin. Nisbatan murakkab operatsion tizimlarda zaruriyat tufayli dinamik jarayonlar ishlab chiqiladi. Operatsion tizimlar ishlashi boshlangandan so'ng yangi jarayoni yaratish tashabbuschisi sifatida maxsus chaqiruv tizimi yoki operatsion tizimning o'zini chaqiruv tizimining istalgan jarayonini foydalanuvchi ishga tushirishi mumkin, u holda albatta ba'zi jarayonlar mavjud bo'ladi. Yangi jarayon tug'ilishiga sabab bo'lgan

jarayon – ona jarayon (*parent process*), qaytadan yangi yaratilgan jarayon – farzand jarayon (*child process*) deb ataladi. Farzand jarayonlari o‘z navbatida yangi farzandlarni yaratishi mumkin. Umumiy holatda tizim ichi jarayon daraxtidan iborat bo‘ladi. 2.14-rasmda geneologik daraxtga misol keltirilgan. Shuni qayd etish kerakki operatsion tizimdagi ba’zi jarayonlar bilan birgalikda foydalanuvchi jarayonlari bitta daraxtda joylashgan bo‘lishi mumkin. Ko‘plab hisoblash tizimlarida daraxtlar bitta daraxtdan tug‘ilgan bo‘ladi.



2.14- rasm. Jarayonning soddalashtirilgan geneologik daraxti

*Izoh: Ko‘rsatgich ona - farzand munosabatini ko‘rsatadi*

Tizimda jarayonning tug‘ilish jarayoni holati yangi PCB (*Process Control Block*) bilan nomlanadi va uni to‘ldirish boshlanadi. Yangi jarayon o‘zining noyob identifikatsiya raqamini oladi. Operatsion tizimda jarayonlar identifikatsiya raqamini saqlash uchun chegaralangan jarayonlarda bir vaqtda ishtirok etishi uchun noyob raqam ajratiladi. Jarayon tugagandan so‘ng uni boshqa jarayonda ishlatish uchun identifikatsiya raqamidan ozod qilinadi.

Odatda o‘zining farzand jarayoni funksiyasini bajarish uchun ma’lum bir resurslar talab qilinadi: xotira, fayl, kiritish/chiqarish qurilmasi va boshqalar. Ularni ajratish bo‘yicha ikki xil usul mavjud. Yangi jarayon ba’zi qismlari ona resurslarida joylashgan bo‘lishi mumkin, ona jarayoni va boshqa farzand jarayonlari bilan ajratilgan bo‘lishi mumkin, yoki operatsion tizimdan bevosita o‘zining resurslarini olishi mumkin. Ajratilgan resurslar to‘g‘risida axborotlar PCB ga kiritiladi.

Farzand jarayon resurslari ajratilgandan so‘ng unga dasturiy kod manzili, ma’lumot belgilari, dastur hisoblagichni o‘rnatish kerak bo‘ladi. Bu yerda ikki xil yechim mavjud. Farzand jarayon birinchi

holatida registr va foydalanuvchi konteksti bo'yicha ona jarayon nusxasi bo'lib shakllanadi, shu sababli qaysi biri ona jarayoni nusxasi ekanligini aniqlash kerak bo'ladi. Ikkinchi holatda farzand jarayon istalgan biror fayli yangi dasturni yuklaydi. Unix operatsion tizimi faqat birinchi usuldagi jarayon tug'ilishiga ruhsat beradi; ya'ni ona jarayon nusxasini yaratish uchun dastlab yangi dasturni yuklaydi, so'ng farzand jarayoni maxsus chaqiriq tizimi yordami bilan o'zining foydalanuvchi kontekstini almashtirishi kerak bo'ladi. VAX/VMS operatsion tizimi faqat ikkinchi usuldan foydalanadi. Windows NT ikki xil rejimdan ham foydalanadi (API turlariga qarab).

Ona jarayon nusxasiga o'xshab yangi jarayonni paydo bo'lishi bittadan ko'proq jarayonni tashkil qilishda ishlatish uchun mavjud dastur imkoniyatlaridan foydalanadi. Uning ishi bo'yicha foydalanuvchi kontekst jarayonini almashtirishi mumkin. Ya'ni bir jarayon doirasida turli xil bir nechta dasturlarni bajarilish ketma-ketligiga o'tkazish mumkin.

Jarayonlar taqsimlanib berilgandan so'ng, PCBda qolgan axborotlar yozib qo'yiladi va yangi jarayon holati tayyorlik holatiga o'tkaziladi. Endi farzand jarayonlari paydo bo'lgandan so'ng ona jarayoni holatini aytib o'tish qoldi. Ona jarayoni farzand jarayonlarini bajarish bilan bir vaqtda o'z ishini ham bajarishi mumkin, ayrim ishlarni yakunlashni yoki barcha farzand jarayonlarning paydo bo'lishini kutishi mumkin.

Biz jarayon hayot davrini tugashini batafsil ko'rib chiqmaymiz. Jarayon o'z ishini tugatgandan so'ng operatsion tizim uni bajarish yakunlangan holatga o'tkazadi va jarayon boshqaruv blokidagi yozuvlarga muvofiq u bilan bog'liq barcha resurslarni bo'shatadi. Bu yerda PCB yo'q qilinmaydi, tizimda ma'lum bir vaqtgacha saqlanadi. Bu ona jarayonidan farzand jarayoni tug'ilishi yakunlangandan so'ng operatsion tizim jarayonning tugallanishi "o'limi" to'g'risida va/yoki uning ishlashi to'g'risida statistik axborotlarni so'rashi mumkin. Ona jarayoni so'rovigacha qayta ishlovchi jarayonlar yoki uning faoliyatining oxirigacha, ya'ni tizimda jarayonlarning tugashigacha bo'lgan batafsil axborot PCB da saqlanadi. Unix operatsion tizimlarida jarayonlar bajarilishi tugallangan holatida bo'ladi.

Shuni aytib o'tish kerakki ko'plab operatsion tizimlarda (masalan, VAX/VMS da) ona jarayonlari o'zining "farzandlari" uchun *yakunlangan* ko'rinishga o'tadi. Boshqa operatsion tizimlarda

(masalan, Unix) ona jarayon vazifasi tugagandan so'ng farzand jarayonlari o'zining mavjudligini davom ettiradi. Shu sababli farzand jarayon o'z ishini davom ettirishi uchun PCB da axborotlarni o'zgartirish zarurati paydo bo'ladi. 20 raqamli jarayondan 25 raqamli jarayon tug'ilgan bo'lsin, va uning vazifasi tugagandan so'ng uzoq vaqt hisoblash tizimida saqlanib qoladi. 20 raqamli jarayon operatsion tizimda boshqa jarayonda qatnashishi bundan mustasno emas. Agar 25 jarayonining ona jarayoni to'g'risida axborot o'zgartirilmasa, u holda jarayon geneologik daraxti noto'g'ri deb hisoblaniladi – 25 jarayoni yangi 20 jarayonini o'zining onasi deb hisoblaydi, 20 jarayon ochilganda esa kutilmagan holat yuz beradi. “Yetim qolish” jarayonini “asrab qolish” qoidasi bo'yicha operatsion tizim ishini davom ettiradi va butun vaqt davomida ishlaydi.

### ***Ko'p martalik operatsiyalar***

Bir martalik operatsiyalar operatsion tizim boshqaruvi ostida joylashgan jarayonlar sonining o'zgarishiga olib keladi va har doim ajratilgan yoki belgilangan resurslarni ozod qilish jarayonlariga bog'liq bo'ladi. Ko'p martalik operatsiyalar operatsion tizimlarda jarayonlar sonini o'zgartirmaydi va ajratilgan yoki bo'shatilgan resurslarga bog'liq emas.

**Jarayonni ishga tushirish.** Operatsion tizim tayyorlik holatida bo'lgan jarayon sonlaridan birini keyingi vazifani bajarish uchun tanlab oladi. Operatsion tizim tanlagan jarayonlar axborotlarni tezkor xotirada mavjud bo'lishini ta'minlaydi. So'ng jarayon holati ushbu jarayon uchun registr belgilarini tiklaydi, bajarilishini o'zgartiradi va jarayon buyruqlarini hisoblagich ko'rsatishiga muvofiq boshqaruv buyruqlarini uzatadi. Kontekstni qayta tiklash uchun kerakli barcha ma'lumotlar, operatsiyani amalga oshiradigan PCB jarayonidan olinadi.

**Jarayonni ma'lum bir vaqtga to'xtatish.** Bajarish holatida bo'lgan jarayon ishini istalgan biror bir uzilish natijasida to'xtatish mumkin. Protessor buyruq hisoblagichini avtomatik ravishda saqlaydi va bajarilayotgan jarayonda bir yoki bir nechta registr bo'lishi mumkin, so'ng esa joriy uzilishni qayta ishlash uchun maxsus manzil bo'yicha boshqaruvga uzatiladi. Uzilishni qayta ishlash bo'yicha ishni qurilma (hardware) oxiriga yetkazadi. Odatda ko'rsatilgan manzilda operatsion tizimning bir qismi joylashgan

bo‘ladi. U tayyorlik holatidagi jarayonga o‘tkazishda uning PCB da tizimli va registrlil kontekst jarayoni dinamik qismida saqlanadi va uzilishlarni qayta ishlaydi, u holda paydo bo‘lgan uzilish bilan bog‘liq belgilangan ishlar bajariladi.

**Jarayonni bloklash.** Jarayon biror bir hisoblash tizimlarida paydo bo‘lishi kutilmayotganda va ishini davom ettirish mumkin bo‘lmaganda bloklanib qo‘yiladi. Buning uchun belgilangan chaqiriq tizimi yordami bilan operatsion tizimga murojaat qiladi. Operatsion tizim chaqiriq tizimini qayta ishlaydi (kiritish/chiqarish operatsiyalarini nomlaydi, jarayonlar navbatiga jarayonni qo‘shadi, qurilmada kutayotganlarni bo‘shatadi va boshqalar) va bajarish holatidan kutish holatiga o‘tkazib uning PCB si kontekst jarayoni kerakli qismiga yozib qo‘yadi.

**Jarayonni qayta ochish.** Istalgan biron bir operatsion tizimda paydo bo‘lgan biror bir holatda aynan qanday hodisa yuz berganini aniqlash zarur bo‘ladi. So‘ng operatsion tizim ushbu holat uchun kutish holatida ayrim jarayonlar joylashganmi yoki yo‘qligini tekshiradi va agar topilsa, hodisa bilan bog‘liq bo‘lgan ishlarni bajarish uchun uni tayyorlik holatiga o‘tkazadi.

### ***Jarayonni yaratish***

Jarayonni yaratish – jarayonlardagi asosiy operatsiyalardan biridir. Ona jarayoni farzand jarayonlarini yaratadi, bu esa o‘z navbatida boshqa jarayonlarni keltirib chiqaradi va shu bilan jarayon daraxtini shakllantiradi. Jarayonlarni yaratishda turli xil yondashuvlar mavjud, ona va farzand jarayoni tomonidan resurslardan birgalikda foydalanish (almashish), ularni parallel ravishda bajarish va manzillar va xotiradan foydalanish.

### ***Resurslarni taqsimlash***

Quyidagi yondashuvlar bo‘lishi mumkin:

- ❖ Ona va farzand jarayoni barcha resurslarni almashadi (bo‘lishadi);
- ❖ Farzand jarayonlari ona jarayonlari resurslarining bir qismini almashadi;
- ❖ Ona va farzand jarayonlari umumiy resurslarga ega emas.

### ***Bajarish***

Quyidagi yondashuvlar bo‘lishi mumkin:

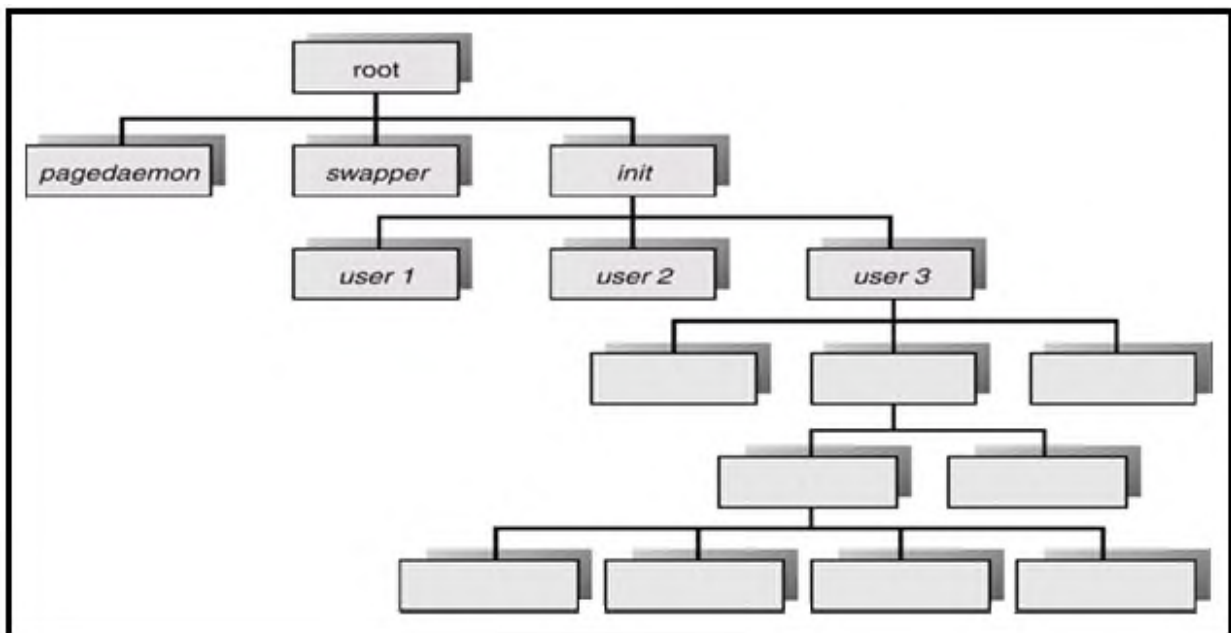
- ❖ Ona jarayoni va farzand jarayonlari birgalikda bajariladi;
- ❖ Ona jarayoni farzand jarayonining tugashini kutmoqda.

### *Manzil va xotiradan foydalanish*

Quyidagi yondashuvlar bo‘lishi mumkin:

- ❖ Farzand jarayonining manzil maydoni ona jarayonining manzil maydonidan nusxa ko‘chiradi;
- ❖ Farzand jarayoni unga yuklangan dasturga ega.
- ❖ Farzand jarayoni ona jarayoni (yengil jarayon) bilan bir xil xotira maydonida ishlaydi.

Unix operatsion tizimda bu quyidagicha amalga oshiriladi: **fork** – bu yangi jarayonni yaratadigan tizim chaqiruvi. Ona jarayonining xotirasini nusxalashtiradi va farzand jarayoni uchun yangi virtual manzillar maydonini yaratadi. Shundan so‘ng, tizim xotirasida yangi dastur bilan almashtirish uchun yana bir tizim chaqiruvi amalga oshiriladi - **exec (execve)** tizim chaqiruvi. Farzand jarayoni ona jarayonining o‘rniga davom etadi.



2.15- rasm. UNIX tizimidagi jarayon daraxti

Tizim ishga tushganda ildiz (**root**) jarayon yaratiladi. U o‘z navbatida, uchta farzand jarayonini yaratadi:

- ❖ **init** - tizimni ishga tushirish;

❖ **pagedaemon** - xotirani sahifali tashkil qilish jarayonini boshqaruvchi demon jarayoni (tizim qayta tiklanmaguncha doimiy ravishda amalga oshiriladigan jarayon);

❖ **swapping** - jarayon, ko'chirishni (yoki almashtirishni) boshqarish.

❖ **init** - jarayonitizimni ishga tushirgandan so'ng foydalanuvchi jarayonlarini boshlaydi. Keyin, o'z navbatida, yangilarini va boshqalarni ishga tushirishi mumkin.

### ***Jarayonni yo'q qilish***

Bu shuningdek, jarayonlardagi asosiy operatsiyalardan biridir. Qoida sifatida, yakuniy vazifani bajarganda va tugatish uchun operatsion tizimga **exit** tizim chaqiruvi yordamida murojaat qilganda jarayon yo'q qilinadi (tugatiladi).

O'z faoliyatida ona jarayoni **abort** (UNIX) tizim chaqiruvi bilan farzand jarayonlarini yo'q qilishi mumkin. Linuxda esa **kill** tizim chaqiruvi asosida amalga oshiriladi.

Bu quyidagi hollarda bo'lishi mumkin:

❖ Farzand jarayon unga ajratilgan resurslardan oshib ketsa;

❖ Farzand jarayoniga topshirilgan vazifani hal qilish boshqa talab qilinmasa;

❖ Ona jarayonidan chiqish amalga oshirilsa, shu sababli farzand jarayonlari ham to'xtatilishi kerak. Agar ona jarayoni yo'q qilinsa, operatsion tizim farzand jarayonining bajarilishini davom ettirishga ruxsat bermaydi. Oxirgi hodisa jarayonlarning kaskadli yo'q qilinishi deb ataladi.

## **2.4. Operatsion tizimda protsessorni boshqarish**

Multidasturli muhitda OT protsessor qaysi jarayonni, qachon va qancha vaqt olishini hal qiladi. Bu funktsiya jarayonni rejalashtirish deb nomlanadi. Operatsion tizim protsessorni boshqarish uchun quyidagi amallarni bajaradi:

- ❖ Protsessor va jarayon holatini kuzatadi;
- ❖ Jarayonga protsessorni (CPU) ajratadi;
- ❖ Jarayon talab qilinmasa, protsessorni o'chiradi.

Protsessorni (CPU ni) rejalashtirish multidasturli operatsion tizimlarning asosidir. Protsessorni jarayonlar o'rtasida almashtirish orqali operatsion tizim kompyuterni unumdorligini oshirishi mumkin.



Protsessorni rejalashtirish va dispetcherlashtirish operatsion tizimning eng muhim funksiyalaridan biri hisoblanadi. Ushbu bo‘limda quyidagi masalalar ko‘rib chiqilgan:

- ❖ Jarayonlarni rejalashtirishning asosiy tushunchalari;
- ❖ Dispetcherlashtirish mezonlari;
- ❖ Dispetcherlashtirish algoritmlari (FCFS, SJF, RR va boshqalar);
- ❖ Bir necha jarayonlarni dispetcherlashtirish;
- ❖ Real vaqtda dispetcherlashtirish;
- ❖ Ko‘p darajali navbatlar.

### ***Jarayonlarni rejalashtirish***

Kompyuter ko‘p vazifalik rejimda ishlayotganda markaziy protsessorda bir vaqtning o‘zida bir nechta jarayon yoki oqimlar ishga tushadi. Agar bunday vaziyatda tayyorlik holatda bo‘lsa bir vaqtning o‘zida ikki yoki undan ko‘p jarayon yoki oqimlarga duch kelinadi. Agar faqat bitta protsessorga kirish ruhsat etilgan bo‘lsa, bu jarayonlardan qaysi biri birinchi bajarilishi tanlaniladi. Ko‘plab operatsion tizimlarda bu tanlov **rejalashtirish** deb ataladi, u foydalanadigan algoritm esa **rejalashtirish algoritmi** deb ataladi.

Jarayonlarni rejalashtirishni qo‘llash, shuningdek bir biridan farqli bo‘lgan ayrim oqimlarni rejalashtirishga qo‘llash mumkinmi degan ko‘plab bir xil bo‘lgan savollar mavjud. Yadro oqimni boshqarsa, u holda odatda har bir oqimni, u aynan qaysi jarayonda yotganiga qaramay rejalashtiriladi. Dastlab biz jarayonlar va oqimlar qanday rejalashtirishni ko‘rib chiqamiz. Shundan so‘ng biz oqimlarni rejalashtirish va qator paydo bo‘ladigan noyob savollarni ko‘rib chiqamiz. Magnitli lentada keltirilgan perfokart ko‘rinishli shakldagi ma’lumotlarni kiritishni amalga oshirgan paketli tizim davrida rejalashtirish algoritmlari soddaroq bo‘lgan: faqat lentadagi navbatdagi topshiriqni ishga tushirish talab qilingan. Ko‘p topshiriqli tizimlarning paydo bo‘lishi rejalashtirish algoritmlarini takomillashtirdi, ushbu holatda odatda xizmat ko‘rsatishni kutayotgan bir nechta foydalanuvchiga, bir vaqtning o‘zida xizmat ko‘rsatiladi. Ayrim universal mashinalar haligacha vaqtni ajratish rejimida topshiriqlarni paketli topshiriq usulida hisoblaydi, va rejalashtiruvchi navbatdagi bajariladigan ish qanday bo‘lishini hal qilishi kerak bo‘ladi: paketli topshiriqni bajaradi yoki terminalda o‘tiruvchi

foydalanuvchi bilan interaktiv aloqani ta'minlaydi. Bunday vaqtli protsessorli mashinalari uchun resurslar yetishmovchiligi bo'ladi, kuchli rejalashtiruvchi mashinalar unumdorligini his qilishi va foydalanuvchilarni qanoatlantira olishi mumkin.

Shaxsiy kompyuterlarning paydo bo'lishi vaziyatni ikki holatga yo'naltirdi. Birinchisi, asosiy vaqtni bitta faol jarayonga qaratdi. Foydalanuvchi hujjatga matnni kiritishda protsessor bir vaqtning o'zida fonli rejimda dasturni kompilatsiyalagan. Foydalanuvchi protsessor matni uchun buyruqni ter ganda, rejalashtiruvchi qaysi jarayoni ishga tushirishni tanlamagan, chunki matnli protsessor yagona nomzod bo'lgan.

Kompyuterlarning ikkinchi davrida, kompyuterlar tez ishlashi natijasida, resurslar yetishmovchiligi kuzatilgan. Shaxsiy kompyuterlar uchun ko'plab dasturlar foydalanuvchiga kiruvchi axborotlar belgilangan (chegaralangan) tezlikda taqdim etiladi, lekin bu tezlik bilan emas. Markaziy protsessor uni qayta ishlashiga bog'liq. Endilikda xatto bir vaqtning o'zida bir nechta dasturlar bilan ishlash imkoniga ega bo'lindi, masalan, matnli jarayon va elektron jadval. Shu sababli shaxsiy kompyuterlarda jarayonlarni rejalashtirish muxim rol o'ynamaydi. Albatta mavjud ilovalar, markaziy protsessorning barcha resurslaridan foydalanadi: masalan, bir soatli yuqori sifatli videoda har bir 108 000 kadrni (NTSC da) yoki 90 000 kadrni (PAL da) rangli gammalarda to'g'irlash uchun katta quvvatga ega sanoat hisoblash kompyuterlari talab qilinadi, lekin ilova qoidalari bundan mustasno.

Tarmoq xizmatlari talab qilinganda vaziyat o'zgaradi. Bu yerda protsessor vaqtida bir nechta jarayonlarga qarshi kurashiladi, shu sababli yangi topshiriqlar olinadi. Masalan, markaziy protsessorida ishga tushirilgan jarayonlardan birontasini tanlash zarurati paydo bo'lganda kunlik statistikani to'playdi va foydalanuvchi so'rovlariga xizmat ko'rsatish jarayonlaridan birini tanlaydi (agar navbatdagi jarayonga muximlilik berilsa, foydalanuvchi o'zini yaxshi xis qiladi).

“To'g'ri” jarayonni tanlashdan tashqari rejalashtiruvchi markaziy protsessorning to'g'ri ishlashi haqida o'ylashi kerak. Dastlab foydalanuvchi rejimida yadro rejimiga o'tkazilishi kerak. So'ng ularni keyingi qayta yuklanish uchun uning registeri jarayon jadvalida saqlashni o'z ichiga olgan joriy jarayon holati saqlanishi kerak. Shundan so'ng, keyingi jarayonni tanlash uchun rejalashtirish

algoritmi ishga tushiriladi. So‘ng yangi jarayon xotira kartasiga muvofiq xotirani boshqarish blokiga yuklanadi. Va nihoyat yangi jarayon ishga tushiriladi.

### ***Qachon rejalashtiriladi***

Rejalashtirishning kalit so‘zi qarorni qabul qilish vaqti bo‘ladi. Rejalashtirish talab qilinadigan turli xil holatlar mavjud.

Birinchidan, jarayonlarning qaysi biri birinchi bajarilishini aniqlashtiruvchi yangi jarayonni yaratish. Har ikkala jarayon tayyorlik holatida joylashganligi sababli, rejalashtiruvchi jarayonning onasi (parent process) yoki uning farzandlarini (child process) tanlashga asoslanib qaror qabul qilishi kerak bo‘ladi.

Ikkinchidan, rejalashtiruvchi jarayon tugayotganda qaror qabul qilishi kerak bo‘ladi. Jarayon boshqa bajarilmaydi (endilikda u mavjud bo‘lmaydi), shu bois bajarishga tayyor bo‘lgan jarayonlar ichidan birortasini tanlash kerak bo‘ladi. Agar jarayonlar bajarishga tayyor bo‘lmasa, odatda tizim bo‘sh jarayonni ishga tushuradi.

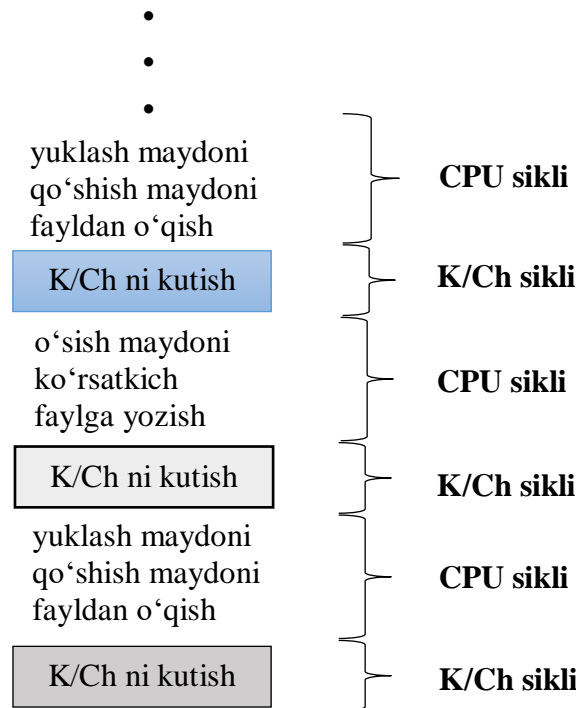
Uchinchidan, jarayon belgiga yoki biron bir boshqa jarayonni tanlash asosida kiritish/chiqarish operatsiyasini yakunlanishini kutishda bloklansa jarayonni bajarish uchun boshqa biron bir jarayonni tanlash kerak bo‘ladi. Ba’zida bu rolni bloklash bajaradi. Masalan,  $A$  muxim rolni o‘ynayapti va  $B$  jarayonni paydo bo‘lishini kutayotgan bo‘lsa, u holda  $A$  jarayon ishlashni davom ettirishga imkon berish uchun uni tang sohasidan ushbu jarayonni chiqishiga imkon berib  $B$  jarayonga bajarish navbatini taqdim etadi. Lekin murakkab tarafi, odatda rejalashtiruvchi ma’lumotlarga bog‘liq bo‘lgan zarur axborotlarga ega bo‘lmaydi.

To‘rtinchidan, rejalashtiruvchi kiritish/chiqarishda to‘xtalishlar paydo bo‘lganda qaror qabul qilishi kerak bo‘ladi. Agar kiritish/chiqarish operatsiya yakunlangan bo‘lsa, kutishda bloklangan qandaydir jarayonda o‘z ishini yakunlab kiritish/chiqarish qurilmasida to‘xtalish bo‘lib o‘tgandan so‘ng, ishlashga tayyor bo‘lishi mumkin. Rejalashtiruvchi qandaydir jarayonni ishga tushirish bo‘yicha qaror qabul qilishi kerak bo‘ladi.

### ***Protsessorni rejalashtirish***

Protsessorni rejalashtirish tizimdagi jarayonlar orasida uning vaqtini taqsimlash hisoblanadi. Rejalashtirishdan maqsad -

multidasturlash yordamida erishiladigan protsessorning maksimal yuklanishini ta'minlashdan iborat. Istalgan jarayonning bajarilishiga CPU / K/Ch sikli – protsessoridan foydalanish va kiritish/chiqarishni kutish davrlarining navbatlashishi sifatida qarash mumkin.

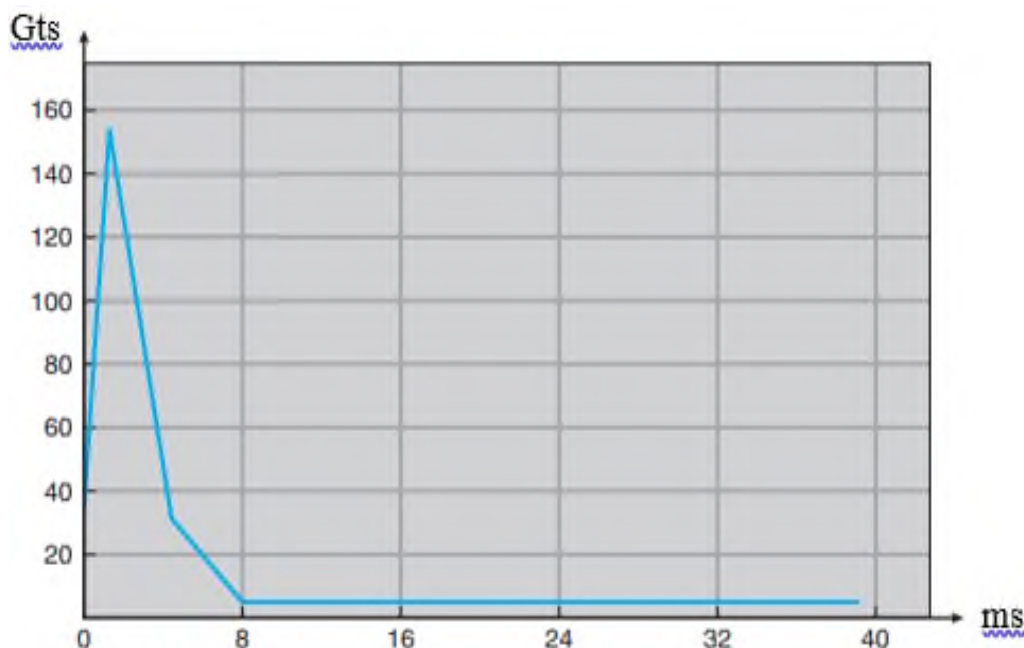


2.16- rasm. CPU va K/Ch sikllari o'zgaruvchan ketma-ketligi

Bir protsessorli tizimda bir vaqtning o'zida faqat bitta jarayon bajarilishi mumkin. Boshqa jarayonlar esa protsessorni bo'shashini kutishi kerak. Multidasturlashning vazifasi – protsessoridan maksimal darajada foydalanishdir. Ushbu turdagi rejalashtirish operatsion tizimning asosiy funksiyasidir. Deyarli barcha kompyuter resurslari foydalanishdan oldin rejalashtiriladi.

Protsessor, shubhasiz kompyuterning asosiy resurslaridan biridir. Shunday qilib, uni rejalashtirish operatsion tizimni loyihalashda muhim o'rin tutadi.

Protsessorni rejalashtirishning muvaffaqiyati jarayonlarning kuzatiladigan xususiyatlariga bog'liq: jarayonning bajarilishi protsessorni bajarish siklidan va K/Ch ni kutishdan iborat. Jarayonlar ushbu ikki holat orasida o'zgarib turadi. 2.17- rasmda operatsion tizimlardagi jarayonlarning haqiqiy holatini tahlil qilish asosida, protsessor faoliyati davrlarining taxminiy gistogrammasi ko'rsatilgan.



2.17- rasm. Protssessor faoliyati davrlarining taxminiy gistogrammasi

Diagramma shuni ko'rsatadiki, faoliyat davri qanchalik qisqa bo'lsa, bunday davrlarning chastotasi shunchalik yuqori va aksincha, faoliyat davrlarining chastotasi ularning davomiyligiga teskari proporsionaldir.

### ***Protssessorni rejalashtiruvchi***

Rejalashtiruvchi - xotiraga yuklangan va bajarishga tayyor bir nechta jarayonlardan birini tanlaydigan va ulardan biri uchun protssessorni ajratadigan OT komponenti hisoblanadi.

Rejalashtirish bo'yicha yechimlar quyidagi hollarda qabul qilinishi mumkin, agar jarayon:

1. Bajarilish holatidan kutish holatiga qayta ulanadi (masalan, jarayon kiritish/chiqarish qurilmasi ishini tugashini kutsa).
2. Bajarilish holatidan bajarishga tayyorlik holatiga o'tishga qayta ulanadi (masalan, uzilish sodir bo'lganda).
3. Kutish holatidan tayyorlik holatiga qayta ulanadi (masalan, kiritish/chiqarish hodisasi tugashi bilan).
4. Yakunlanadi.

1- va 4- turlardagi rejalashtirish jarayonni uzmasdan rejalashtirish (non-preemptive) atamasi bilan belgilanadi. 2- va 3- turlardagi rejalashtirish jarayonni uzish bilan rejalashtirish (preemptive) atamasi bilan belgilanadi.

Protsessor menejeri 2 quyi menejerdan tashkil topadi:

1. Vazifani rejalashtiruvchi;
2. Jarayonni rejalashtiruvchi.

### *1. Vazifani rejalashtiruvchi*

Vazifani rejalashtiruvchi yuqori darajadagi rejalashtiruvchi hisoblanadi va uning funksiyalari quyidagicha:

- ❖ Vazifalarni kirish navbatidan tanlab oladi;
- ❖ Ularni xarakteristikasiga qarab jarayonlar navbatiga kiritadi;
- ❖ Maqsad: vazifalarni tizim resurslaridan maksimal foydalanish tartibida joylashtirish;
- ❖ Resurslarni doimiy band holatda saqlashni tashkillashtiradi.
- ❖ Kiritish/chiqarish va hisoblash o'rtasidagi o'zaro muvozanatni ta'minlaydi.

### *2. Jarayonni rejalashtiruvchi*

Jarayonni rejalashtiruvchi past darajadagi rejalashtiruvchi hisoblanadi va uning funksiyalari quyidagicha:

- ❖ Jarayonlarni bajarilishi uchun protsessorni taqsimlaydi;
- ❖ Protsessor resurslarini qachon va qancha muddatga olishini belgilaydi;
- ❖ Uzilishlarni qayta ishlashni hal qiladi;
- ❖ Qachon jarayon to'xtatilishini va qayta tiklanib protsessordan foydalanishini tashkil qiladi.

### ***Vazifa va jarayon holatlari***

Vazifani rejalashtiruvchi yangi (new) va tugatish (finished) holatlariga, jarayonlarni rejalashtiruvchi esa tayyorlik (ready), bajarilish (running) va kutish (waiting) holatlarini boshqaradi. Jarayon operatsion tizimdan vazifa rejalashtiruvchisiga uzatilganida uning (jarayonning) holati har doim yangi (new) sifatida o'rnatiladi. Barcha jarayonlar dastlab vazifani rejalashtiruvchiga kelib tushadi. Qachonki jarayon vazifa rejalashtiruvchisidan jarayon rejalashtiruvchisiga o'tkazilganda uning holati tayyorlik (ready) holatiga o'zgartiriladi. Agar protsessor ishlash uchun tayyor bo'lsa (bo'sh bo'lsa) barcha kelayotgan jarayonlarni ko'zdan kechiradi va ma'lum algoritmlar asosida ularni tanlaydi va xotirani bo'sh deb hisoblab, jarayonni bajarishni boshlaydi. Jarayonning holati bajarilayotgan (running) ga

o'zgartiriladi. Oldindan belgilangan vaqtdan so'ng jarayon to'xtatiladi va boshqa bir jarayon (bajarilish uchun) protsessorni egallaydi. To'xtatilgan jarayonning holati jarayon rejalashtiruvchisi tomonidan tayyorlilik (ready) holatiga o'zgartiriladi. Jarayonlarning bunday almashtirilishi oldindan rejalashtirish siyosati deb nomlanadi. Agar jarayon ishga tushirilganda foydalanuvchi yoki boshqa jarayon tomonidan kiritish/chiqarish amali bajarilishini kutsa, u holda jarayon kutish (waiting) holatiga o'tkaziladi. Kiritish/chiqarish jarayoni yakunlanganligi haqida xabar kiritish/chiqarish qurilmasidan jarayonni rejalashtiruvchiga yetkazilganda jarayonning holati tayyorlilik (ready) holatiga o'zgartiriladi. Nihoyat, jarayon yakunlanganida yoki xatolik yuz berganida jarayon (oldindan) tugatiladi va uning holati tugatishga (finished) o'tkaziladi. Odatda, jarayon holatining o'zgarishlari jarayonni rejalashtiruvchi tomonidan amalga oshiriladi, va vazifani rejalashtiruvchiga bu o'zgarishlar haqida axborot beriladi. Shundan so'ng vazifani rejalashtiruvchi holatni tugatishga (finished) o'zgartiradi.

### ***Protsessor dispetcheri***

Protsessor dispetcheri protsessorni rejalashtiruvchi tanlagan jarayonga beradigan OT komponenti hisoblanadi. Dispetcher quyidagi amallar ketma-ketligini bajaradi:

- ❖ Kontekstni (tarkibni) qayta ulaydi;
- ❖ Protsessorni foydalanuvchi rejimiga qayta ulaydi;
- ❖ Foydalanuvchi dasturini qayta yuklash uchun uni mos manzil bo'yicha o'tishini bajaradi.

Dispetcherning yashirin aktivligi (dispatch latency) - bitta jarayonni to'xtatish va boshqa jarayonni boshlash uchun dispetcherga talab qilinadigan vaqt hisoblanadi. Ma'lumki, tizim bu vaqtni minimallashtirishga intilishi kerak, lekin rejalashtirish mezonlari to'plami murakkabroq.

### ***Rejalashtirish mezonlari***

Tizim u yoki bu darajada hisobga olishi kerak bo'lgan protsessorning beshta asosiy rejalashtirish mezonlari mavjud.

**Protsessordan foydalanish** (CPU utilization) - maksimal bo'lishi mumkin bo'lgan vaqt davrida uni bandlik rejimida saqlash

hisoblanadi. Optimallashtirish mezon: bu ko'rsatkichni maksimallashtirish.

**Tizimning o'tkazish qobiliyati** (throughput) - vaqt birligi ichida o'zining bajarilishini tugatadigan jarayonlar soni (o'rtacha) hisoblanadi.

**Jarayonga ishlov berish vaqti** (turnaround time) - qandaydir jarayonni bajarilishi uchun zarur bo'ladigan vaqt hisoblanadi. Optimallashtirish mezon: bu ko'rsatkichni minimallashtirish.

**Kutish vaqti** (waiting time) - jarayon bajarilishga tayyor jarayonlar navbatida kutadigan vaqt hisoblanadi.

**Javob vaqti** (response time) - interfaol tizimda vazifani bajarilish vaqti eng yaxshi mezon bo'lmasligi mumkin.

Istalgan optimallashtirishdagi kabi strategiyaga bog'liq bo'lmagan holda, barcha mezonlarni bir vaqtda qoniqtirish mumkin emas. Quyida turli rejalashtirish algoritmlarini (yoki strategiyalarini) ko'rib chiqamiz va ko'rsatilgan mezonlarning optimalligiga erishish nuqtai nazaridan ularning afzalliklari va kamchiliklarini ko'rib chiqamiz.

## 2.5. Jarayonlarni rejalashtirish algoritmlari

### *First-Come-First-Served (FCFS) algoritmi*

First-Come-First-Served (kelish tartibida xizmat ko'rsatish, ya'ni, birinchi kelganga birinchi xizmat ko'rsatish (FIFO) kabi bir xil) – algoritmi eng oddiy rejalashtirish algoritmi bo'lib, bunda protsessorning resurslari jarayonlarga ular iste'mol qiladigan resurslarga, xususan, jarayonning bajarilishi uchun talab qilinadigan u bildirgan vaqtga bog'liq bo'lmagan holda tizimga kelishi (kirishi) tartibida taqdim etiladi. Bu va boshqa algoritmlarni ko'rib chiqishda jarayonlarning nomlari va ularning qandaydir vaqt birliklarida ifodalanadigan bajarilish vaqt diapazonlarini Gant diagrammalaridan (Gantt charts) foydalanib aniqlaymiz.

Quyidagi misolni ko'rib chiqamiz. J1, J2 va J3 jarayonlar quyidagi aktivliklar davrlari bilan ko'rsatilgan tartibda tizimga kiritilgan bo'lsin:



2.3- jadval

Jarayon	Aktivlik davri
J1	24
J2	3
J3	3

U holda ularni rejalashtirish uchun FCFS algoritmidan foydalanishda protsessorni birinchi bo‘lib uzoq bo‘lishiga qaramasdan, birinchi jarayonni oladi. Bu holda protsessorni jarayonlar orasida taqsimlanishi 2.18- rasmda tasvirlangan.



2.18- rasm. FCFS algoritmi bo‘yicha rejalashtirish sxemasi (1- misol)

Shunday qilib, kutish vaqti  $J1 = 0$ ;  $J2 = 24$ ;  $J3 = 27$  bo‘ladi.

O‘rtacha kutish vaqti:  $(0 + 24 + 27)/3 = 17$

Agar jarayonlar tartibi boshqacha - J2, J3, J1 bo‘lsa (tizimga oxirgi kiritilgan jarayon – eng uzoq), u holda ularni rejalashtirish natijasi mutlaqo boshqacha bo‘ladi (2.19- rasm).



2.19- rasm. FCFS algoritmi bo‘yicha rejalashtirish sxemasi (2- misol)

Bu holda jarayonlarni kutish vaqti:  $J1 = 6$ ;  $J2 = 0$ ;  $J3 = 3$ .

O‘rtacha kutish vaqti:  $(6 + 0 + 3)/3 = 3$

Bu natija oldingi natijaga qaraganda ancha yaxshi. Birinchi misol namoyish etgan natija samarasi (convoy effect) – qisqa jarayon

uzoq jarayondan keyin xizmat ko'rsatiladigan hollarda jarayonlarni o'rtacha kutish vaqtini ortishi deyiladi.

### *Shortest Job First (SJF) algoritmi*

Shortest Job First (SJF, dastlab eng qisqa vazifani bajarish) algoritmi protsessorni rejalashtirish algoritmi bo'lib, bunda protsessor birinchi navbatda tizimdagi mavjud jarayonlardan eng qisqasiga beriladi. Bu holda har bir jarayon bilan uning navbatdagi aktivlik davri davomiyligi bog'lanadi. Bu davomiylilik eng qisqa jarayonga birinchi xizmat ko'rsatilishi uchun ishlatiladi. Bu algoritmni qo'llanishining ikkita sxemalari bo'lishi mumkin:

1. Jarayonlarni uzmasdan – jarayonga protsessor berilayotgan vaqtda uning vaqt kvanti tugamasdan jarayon uzilmasligi kerak.

2. Jarayonlarni uzish bilan – agar aktivlik vaqti aktiv jarayonning qolgan vaqtdan kichik bo'lgan yangi jarayon kelsa, aktiv jarayonni to'xtatish. Bu sxema Shortest-Remaining-Time-First (SRTF – dastlab eng qisqa vaqt) nomi bilan ma'lum.

Ko'rish qiyin emaski, SJF algoritmi u berilgan jarayonlar to'plami uchun minimal o'rtacha kutish vaqtini ta'minlashi mazmunida optimal bo'ladi. Jarayonlarni uzmasdan SJF algoritmining qo'llanishiga misolni ko'rib chiqamiz. Jarayonlar to'plami, tizimda ularning paydo bo'lishi vaqtlari va ularning aktivligi vaqtlari quyidagicha:

2.4- jadval

Jarayon	Paydo bo'lish vaqti	Aktivlik vaqti
J1	0.0	7
J2	2.0	4
J3	4.0	1
J4	5.0	4

Jarayonlarni uzmasdan SJF algoritmi bo'yicha jarayonlarni rejalashtirish sxemasi 2.20- rasmda keltirilgan.



2.20- rasm. Jarayonlarni uzmasdan SJF algoritmi bo'yicha jarayonlarni rejalashtirish sxemasi

Bu holda o‘rtacha kutish vaqti =  $(0 + 6 + 3 + 7)/4 = 4$ . Endi o‘sha jarayonlarga uzilishli SJF algoritmini qo‘llaymiz va o‘rtacha kutish vaqti qanday o‘zgarishini tahlil qilamiz. Algoritmning qo‘llanishi natijasi 2.21- rasmda tasvirlangan.



2.21- rasm. Jarayonlar uzilishli SJF algoritmi bo‘yicha jarayonlarni rejalashtirish sxemasi

Bu holda tizimga qisqaroq jarayon tushishi momentida jarayonning uzilishi prinsipi bir necha marta qo‘llanadi: 2 momentda 1- jarayon uziladi va qisqaroq 2- jarayon bajarila boshlanadi, 4 momentda 2- jarayon uziladi va qisqaroq 3- jarayon bajarila boshlanadi.

Diagrammadan ko‘rinib turibdiki, jarayonlarning uzilishi prinsipining qo‘llanishi tufayli protsessoridagi jarayonning uzluksiz bajarilishi davrlari yonma-yon bo‘lishi va boshqa jarayonlarni bajarilishi davrlarini bilan o‘rin almashishi mumkin.

Bu holda o‘rtacha kutish vaqti =  $(9 + 1 + 0 + 2)/4 = 3$ , ya‘ni kutilganidek, u jarayonlarni uzilishi prinsipi qo‘llanilmasligiga qaraganda kichik bo‘ldi.

### ***Ustuvorliklar bo‘yicha rejalashtirish***

Bu algoritmda har bir jarayon bilan uning ustuvorligi (butun son) bog‘lanadi. Protsessor eng katta ustuvorlikli jarayonga beriladi (kichik son yuqoriroq ustuvorlikni bildiradi, ya‘ni jarayonning eng yuqori ustuvorligini 1 ga teng deb olamiz).

Bu algoritm oldingi algoritm kabi uzilishli va uzilishsiz variatlarga ega. Shu bilan birga, SJF algoritmiga ustuvorliklar bo‘yicha rejalashtirish sifatida qarash mumkin, unda navbatdagi aktivlik vaqti ustuvorlik hisoblanadi. Ustuvorliklar bo‘yicha rejalashtirishda "ochlik" (starvation) muammosi – past ustuvorlikli jarayonlar hech qachon bajarilmasligi va cheksiz kutadigan vaziyatlar vujudga keladi. Operatsion tizimlarda bu muammoni yechishning

an'anaviy usuli jarayonning ortishini hisobga olish (aging) hisoblanadi. Vaqt o'tishi bilan jarayonning ustuvorligi tizim orqali oshiriladi.

### **Round Robin (RR) algoritmi**

Round Robin (RR, halqali tizim) algoritmi bu barcha jarayonlarga navbat bo'yicha bir xil vaqt kvantlarini berish hisoblanadi. Algoritmning nomi AQShdagi ommaviy qarta o'yinidan kelib chiqadi. Bu algoritmda har bir jarayon protsessor vaqtining uncha katta bo'lmagan kvanti – odatda 10-100 millisekundni oladi. Bu vaqt tugagandan keyin jarayon uziladi va tayyor jarayonlarni oxiriga joylashtiriladi.

Agar bajarilishga tayyor jarayonlar navbatida  $n$  jarayonlar va vaqt kvanti  $q$  ga teng bo'lsa, u holda har bir jarayon  $1/n$  protsessor vaqtini eng kattasi  $q$  birlikdan bir marta qismlab oladi. Hech bir jarayon  $(n-1)q$  vaqt birligidan ortiq kutmaydi.

Bu algoritmning unumdorligi  $q$  koeffitsientga bog'liq:

❖ agar  $q$  yuqori bo'lsa, u holda algoritm FCFS algoritmgaga deyarli ekvivalent;

❖ agar  $q$  past bo'lsa, u holda  $q$  tarkibiy qayta ulanish vaqtidan katta bo'lishi kerak, aks holda bitta jarayondan boshqa jarayonga qayta ulanishga ustama sarflar o'ta katta bo'ladi.

RR algoritmini qo'llanilishi misolini ko'rib chiqamiz. Tizimda aktivlik vaqtli quyidagi jarayonlar mavjud bo'lsin:

2.5- jadval

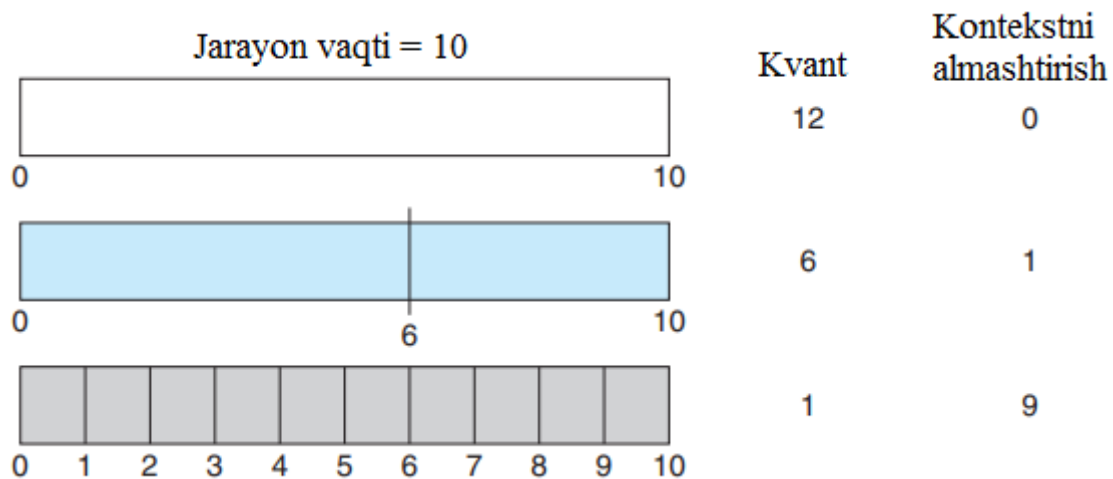
Jarayon	Aktivlik vaqti
J1	53
J2	17
J3	68
J4	24

$q = 20$  vaqt kvantili RR algoritmi bo'yicha protsessorni rejalashtirish sxemasi 2.22- rasmda keltirilgan.



2.22- rasm. RR algoritmini qo‘llanishiga misol ( $q = 20$ )

Odatda RR algoritmi SJF algoritmga qaraganda yomon aylanish vaqtiga ega (chunki har bir jarayon vaqt kvantlari boshqa jarayonlarga beriladigan vaqtda navbatdagi vaqt kvantini kutishi kerak bo‘ladi), lekin yaxshi javob vaqtiga ega. 2.23- rasmda kontekstni almashtirishlar sonining vaqt kvantiga bog‘liqligi ko‘rsatilgan: kvant qancha kichik bo‘lsa, kontekstni almashtirishlar soni shuncha ko‘p bo‘ladi.



2.23- rasm. Protsessor kvant vaqti va kontekstni almashtirish vaqti

### ***Ko‘p darajali navbat***

Binobarin, tizimdagi jarayonlar turli o‘ziga xosliklarga (masalan, paketli va interaktiv) ega bo‘lishi mumkin, amalda operatsion tizimlarda bajarilishga tayyor jarayonlar navbati ikkita navbatlarga bo‘linadi:

- ❖ asosiy (interaktiv jarayonlar);
- ❖ fon (paketli jarayonlar).

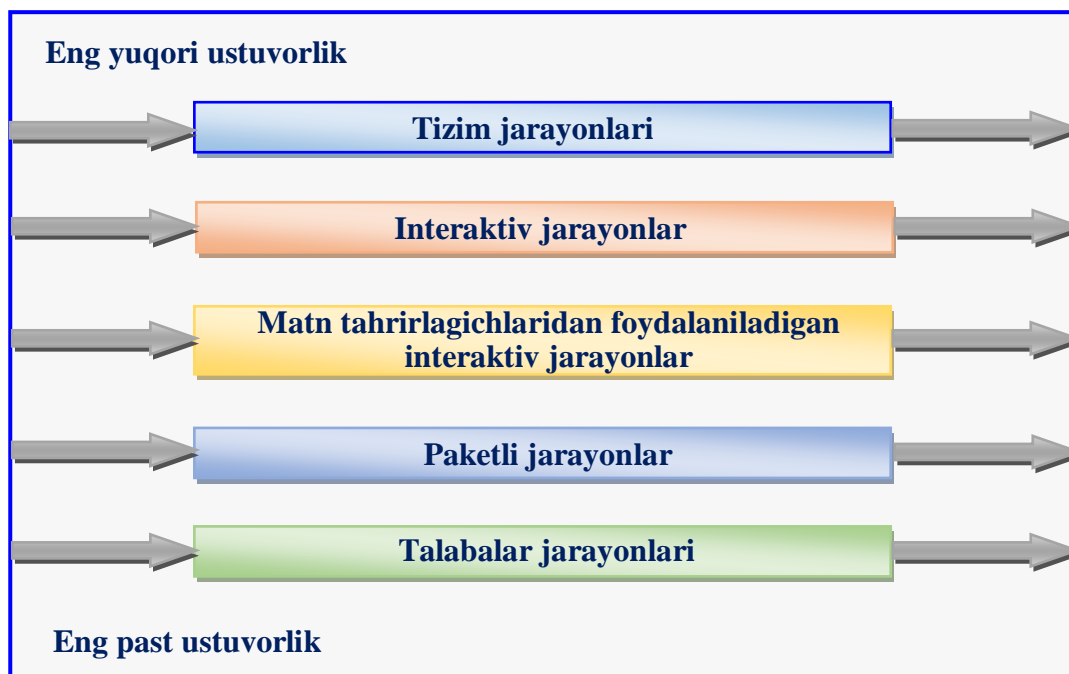
Har bir navbat o‘z rejalashtirish algoritmiga ega bo‘ladi. Asosiy navbat RR, fon navbat FCFS rejalashtirish algoritmiga ega bo‘ladi. Bu aralash algoritmda navbatlar orasidagi rejalashtirish, ya’ni u yoki bu

navbatdan jarayonlarni tanlash algoritmi zarur bo‘ladi. Navbatlar orasidagi rejalashtirish quyidagi turlarga bo‘linadi:

❖ Qayd etilgan ustuvorlikli – asosiy navbatdan, keyin fon navbatdan barcha jarayonlarga xizmat ko‘rsatish. Bunda “och qolish” ehtimolligi mavjud.

❖ Vaqt oralig‘ini ajratish – har bir navbat qandaydir protsessor vaqt oralig‘ini oladi, u jarayonlar orasida taqsimlanishi mumkin, masalan, 80% asosiy navbatdagi RRga va 20% fon navbatdagi FCFSga taqsimlanishi mumkin.

2.24- rasmda jarayonlarni rejalashtirish uchun ko‘p darajali navbat tuzilmasiga real misol keltirilgan.



2.24- rasm. Ko‘p darajali navbatni rejalashtirishga misol

Eng yuqori ustuvorlikka tizim jarayonlari ega, keyin interaktiv jarayonlar, undan past ustuvorlikka esa matn tahrirlagichlari chaqiriladigan interaktiv jarayonlarga ega (ular foydalanuvchilarning sekin ishlashi tufayli sezilarli katta vaqtni egallaydi), keyin paketli va nihoyat talabalar jarayonlari keladi.

Real vaziyat shunday, lekin muallif talabalar jarayonlarini “kamsitilishini” to‘g‘ri hisoblamaydi. Aynan ularga tizim jarayonlaridan keyingi ustuvorlikni, masalan, diplom ishlarini himoya qilishdan oldingi davrda berish kerak bo‘ladi.

## 2.6. Berkliklar, Berkliklarni aniqlash va bartaraf etish usullari

### *Berkliklar muammosi*

**Berklik (deadlock)** bu bloklangan jarayonlar to'plami bo'lib, ulardan har biri qandaydir resursga ega va bu to'plamdan qandaydir boshqa jarayon ega bo'lgan resursni kutadi. Berklikka oddiy misolni semaforalar yordamida modellashtirish oson.

Tizimda ikkita J1 va J2 jarayonlar murojaat qiladigan ikkita tashqi A va B qurilmalar bo'lsin. Semafor sinxronlashtirish maqsadida tashqi qurilmalarning har biri bilan bog'langan, ularni ham A va B bilan belgilaymiz. Semaforlar oldindan ochiq. Jarayonlardan har biriga har ikkala qurilmalar zarur bo'lsin, lekin ular qurilmalarga qarama-qarshi tartibda murojaat qiladi:

J1: kutish (A); kutish (B)

J2: kutish (B); kutish (A).

Bu holda berklik o'z o'rniga ega bo'ladi. J1 jarayon A semaforini yopish va birinchi qurilmani bloklash bilan ikkinchi qurilma bilan bog'langan B semaforini qachon ochilishini hech qachon mo'ljallay olmaydi, chunki uni J2 jarayon yorishga ulgurgan. Shunga o'xshash, J2 jarayon A semaforining ochilishini mo'ljallay olmaydi.

### *Tizim modeli*

Bunday vaziyatlarni tavsiflash va tadqiq qilish uchun tizimning rasman modelini umumiy ko'rinishda kiritamiz. Model yordamida jarayonlarning resurslari so'rovlar haqida, jarayonlarning resurslarga haqiqatda egaligi haqida va resurslarning bo'shashi haqidagi ma'lumotlarni beramiz. Tizimda  $m$  turlardagi resurslar (masalan, protsessor, xotira, kiritish/chiqarish qurilmalari) bo'lsin. Tizimda resurslar turlarini  $R_1, R_2, \dots, R_m$  bilan belgilaymiz. Har bir  $R_i$  resurs turi  $W_i$  nusxalarga ega bo'lsin. Har bir jarayon quyidagi usullardan biri orqali resursdan foydalanishi mumkin:

- ❖ so'rov (request);
- ❖ foydalanish (use);
- ❖ bo'shatish (release).

Berklikagar quyidagitar tashartlar bir vaqtda bajarilsa, vujudgakelishim mumkin:

1. O'zaro inkor qilish: har bir vaqt momentida faqat bitta jarayon resursga ulanishni olishi mumkin;



2. Saqlash va kutish: bitta resursni saqlayotgan jarayon boshqa jarayonlar ega bo'lgan boshqa resurslarni olishni kutadi;

3. Uzilishlarning bo'lmasligi: jarayon o'zining ishlashini tugatganidan keyingina resursni bo'shatishi mumkin.

4. Siklli kutish:  $J_0$  jarayon  $J_1$  jarayon ega bo'lgan resursni kutadigan,  $J_1$  jarayon  $J_2$  jarayon ega bo'lgan resursni kutadigan.....  $J_n$  jarayon  $J_1$  jarayon ega bo'lgan resursni kutadigan  $\{J_0, J_1, \dots, J_n\}$  to'plam mavjud.

### ***Resurslarni taqsimlash grafi***

Ko'rib chiqishga  $V$  balandliklar to'plami va  $E$  yoylar to'plamidan iborat resurslarni taqsimlash grafini kiritamiz.  $V$  ikkita turlardagi – balandliklar-jarayonlar va balandliklar-resurslarga bo'linadi. Boshqacha aytganda,  $V$  tizimdagi barcha jarayonlar  $J = \{J_1, J_2, \dots, J_n\}$  turdagi balandliklar to'plami va  $R = \{R_1, R_2, \dots, R_m\}$  turdagi balandliklar to'plamiga bo'linadi.

Yoylarning quyidagi ikkita turlarini kiritamiz:

❖ “so'rov” turdagi yoy (request edge) –  $J_i \rightarrow R_j$  turdagi yo'naltirilgan yoy.

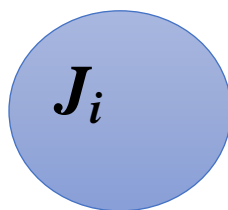
❖ “tayinlash” turdagi yoy (assignment edge) –  $J_i \leftarrow R_j$  turdagi yo'naltirilgan yoy.

Yoylarning turli yo'nalishlarining ma'nosi quyidagicha. Agar jarayon qanday resursga da'vogarlik qilsa, u holda yoy balandlik-jarayondan balandlik-resursga o'tkaziladi.

Aniq resurs birligi qandaydir jarayonga ajratilsa, u holda yoy bu belgiga tegishli bo'ladi va balandlik-resursdan jarayonning balandligiga o'tkaziladi. Kiritiladigan graf va uning grafining o'ziga xos xususiyatlarini aniqlashtiramiz.

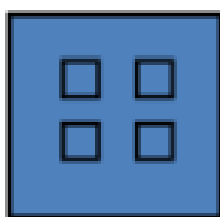
Zamonaviy atamada bu turdagi graf – zahiralangan graf (reserved graph) deyiladi. Uning balandlik-jarayoni oddiy ko'rinishga ega bo'ladi,  $R_j$  resursga mos keladigan balandlik-resurs esa  $W_j$  nimbalandliklardan tashkil topadi, ulardan har biri aniq bir resurs birligini belgilaydi. Zahiralangan graflar nazariyasida bunday balandliklar superbalandliklar (super-vertices) deyiladi.

Shunday qilib, so'rov yoyi umuman balandlik-jarayondan balandlik-resursga boradi, tayinlash yoyi esa balandlik-resursning mos nimbalandligidan balandlik-jarayonga boradi. Balandlik-jarayonga misol 2.25- rasmda keltirilgan.

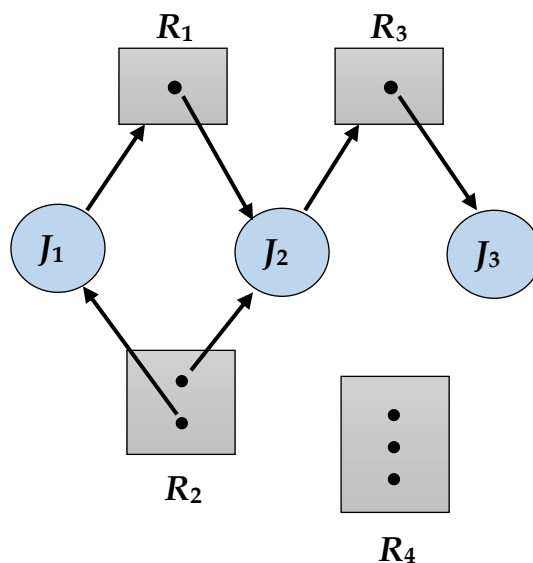


2.25- rasm. Resurslarni taqsimlash grafidagi balandlik-jarayonga misol

To'rtta nusxali superbalandlik-resursga misol 2.26-rasmda keltirilgan. Resursning har bir nusxasiga o'z nimbalandligi mos keladi. Resurslarni taqsimlash grafiga misol 2.27- rasmda keltirilgan.



2.26- rasm. To'rtta nusxali superbalandlik-resursga misol



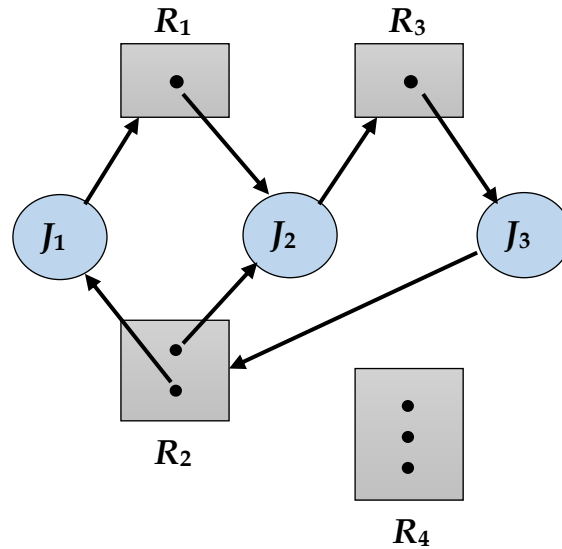
2.27- rasm. Resurslarni taqsimlash grafiga misol

Bu graf uchta jarayonlar va to'rtta resurslari turlariga ega bo'lgan tizimni aks ettiradi: 1- va 3- turlardagi resurslar bittadan nusxaga ega, 2- turdagi resurs ikkita nusxaga ega, 4- turdagi resurs uchta nusxaga ega. 1- jarayon 2- jarayon bilan band bo'lgan 1- resursga da'vogarlik qiladi. 2- jarayon 3- jarayon bilan band bo'lgan 3- resursga da'vogarlik qiladi. 2- resursning ikkita birliklari 1- va 2-

jarayonlarga berilgan.4- resurs taqsimlanmagan (barcha uchta birliklar bo‘sh).

**Resurslarni taqsimlash grafi bo‘yicha berklilarni qidirish**

Ma’lumki, bunday grafdasikl berklilning borligini bildiradi.



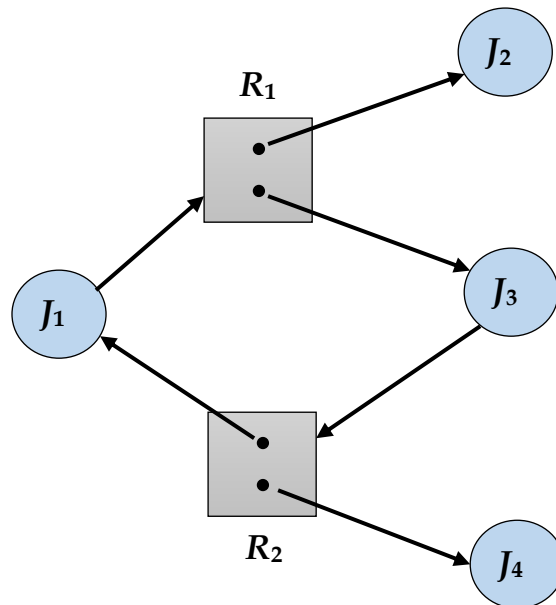
2.28- rasm. Berklikli resurslarni taqsimlash grafiga misol

2.28- rasmda berklikli resurslarni taqsimlash grafiga misol keltirilgan. 1, 2 va 3- jarayonlar orasidagi siklli kutish vaziyati mavjud. 1- jarayon, 2- jarayon ega bo‘lgan resursga da’vogarlik qiladi. 2- jarayon, 3- jarayon ega bo‘lgan resursga da’vogarlik qiladi. 3- jarayon bitta birligi 1- jarayonga, ikkinchi birligi 2- jarayonga berilgan resursga da’vogarlik qiladi.

Lekin har doim ham resurslarni taqsimlash grafida siklning bo‘lishi berklilni borligini bildirmaydi.

2.29- rasmda siklli, lekin berklisiz resurslarni taqsimlash grafiga misol keltirilgan. Bu holda (2.29- rasm) to‘rtta jarayonlar va ikkita resurslar turlari mavjud bo‘ladi. Siklda 1- va 3- balandliklar-jarayonlar qatnashadi. Lekin har bir resursda ikkitadan birliklar borligi tufayli berklilning oldini olishga erishiladi. 1- resursni kutadigan 1- jarayon uni bu resursning bitta birligiga ega bo‘lgan va kutish sikliga kirmaydigan 2- jarayon (1- jarayon emas) tugagandan keyin olishi mumkin. Shunga o‘xshash, 2- resursga da’vogarlik qiladigan 3-

jarayon uni 4- jarayon (1- jarayon emas) bo‘shatganidan keyin olishi mumkin.



2.29- rasm. Siklli, lekin berkliksiz resurslarni taqsimlash grafiga misol

Shunday qilib, quyidagi mulohazani aytish mumkin. Agar resurslarni taqsimlash grafi sikllarga ega bo‘lmasa, u holda tizimda berkliklar mavjud emas. Agar resurslarni taqsimlash grafi sikllarga ega bo‘lsa, u holda quyidagi ikkita hollar bo‘lishi mumkin:

1. Agar har bir turdagi resurslar faqat bittadan bo‘lsa, u holda berklik o‘z o‘rniga ega bo‘ladi;
2. Agar resurslar bir necha nusxalarda bo‘lsa, u holda berklik bo‘lishi mumkin.

### ***Berkliklarga ishlov berish usullari***

Nazariy jihatdan quyidagi berkliklarga ishlov berish usullari bo‘lishi mumkin:

- ❖ Tizim hech qachon berklik holatiga kirmasligiga amin bo‘ling;
- ❖ Tizim berklik holatiga kirishi mumkinligini olish, lekin berklikdan keyin qayta tiklanish imkoniyatini ko‘zda tutish.

Afsuski, amalda ko‘plab OTlarda (shu jumladan, UNIXda) berkliklar bilan kurashishning uchinchi “usuli” ham ishlatiladi. Berkliklar muammosi inkor qilinadi, lekin OT mualliflari hech bir asoslarsiz tizimda berkliklar mumkin emasligiga da’vo qilishadi.

### ***Berkliklarning oldini olish***

Berkliklarning oldini olish qanday usullarining mumkinligini tahlil qilamiz. Asosiy g'oya jarayonlar tomonidan resurslarga so'rovlar usullarini cheklash hisoblanadi. Resurslarga ega bo'lishni o'zaro inkor qilish (berklikning birinchi sharti) imkoniyatini cheklash uchun u barcha resurslar uchun ham talab qilinmasligini ta'kidlash zarur.

Bo'linadigan resurslar (masalan, konstantlar, kodlar, fayllari massivlari) uchun u talab qilinmaydi. Saqlash va kutish imkoniyatini cheklash (berklikning ikkinchi sharti) uchun qandaydir resursni so'raydign jarayon boshqa hech qanday resurslarga ega bo'lmasligini talab qilish mumkin.

Muqobil variant barcha jarayonlar ularning haqiqiy bajarilishini boshlanishigacha barcha zarur resurslarga ega bo'lishi talabi hisoblanadi. Afsuski, har ikkala talablarning bajarilishi resurslardan foydalanishning yetishmasligi va "och qolish"ga (starvation) olib keladi.

Jarayon resursni har bir kutishida resurslarni qayta taqsimlash algoritmi ma'qulroq hisoblanadi. Agar jarayon qandaydir A resursga ega bo'lsa va unga darhol ajratilishi mumkin bo'lmagan boshqa B resursni so'rasa, u holda jarayon kutishi kerak. Bunda jarayon egallagan A resurs darhol bo'shatilishi kerak. A resurs jarayon kutayotgan resurslar ro'yxatiga kiritiladi.

Jarayon faqat agar unga bir vaqtda u ega bo'lgan barcha eski resurslar va u kutayotgan yangi resurslar ajratilishi mumkin bo'lsa, qayta tiklanishi mumkin.

Sikli kutish vaziyatining oldini olish uchun eng oddiy yechim barcha resurslar turlari raqamlari bo'yicha tartiblashtirishni kiritish va jarayon resurslarni faqat ularning raqamlarini ortib borishi tartibida so'rashi talabini kiritish hisoblanadi.

Amalda bunday yechimni qo'llanishi va qulay bo'lishi ehtimoli kam, chunki iste'mol qilinadigan va talab qilinadigan resurslar turlarining o'ziga xosligi barcha bo'lishi mumkin raqamlashlarga hech qanday bog'liq emas va istalgan raqamli resursga ehtiyoj zarurati bo'yicha vujudga kelishi mumkin.

## ***Berkliklardan qochish***

Berkliklardan qochish usullari tizim har bir jarayon tizimga kiritilishi momentidan boshlab jarayon va uning resurslarga ehtiyojlari haqidagi qo'shimcha tekshirilmagan ma'lumotlarga ega bo'lishini talab qiladi. Eng oddiy va foydali model har jarayon tizimga kiritilishida unga kerak bo'ladigan har bir turdagi resurslarning maksimal hajmini ko'rsatishini talab qiladi.

Bu yondashish hatto oldingi OTlarda ishlatilgan va topshiriqlar pasporti nomiga – jarayonning har bir turdagi resurslarga maksimal ehtiyojlari ro'yxatiga – operativ va tashqi xotira, bajarilish vaqti, chop etish varaqlari va boshqalarga ega.

Berkliklardan qochish algoritmi siklli kutish vaziyati hech qachon yuz bermasligiga ishonch hosil qilish uchun resurslarni taqsimlanishi holatini tahlil qilishi kerak. Resurslarni taqsimlanishi holati mumkin resurslar hajmi, taqsimlangan resurslar hajmi va jarayonlarning maksimal talablari sifatidatavsiflanadi.

### **Nazorat savollari**

1. Operatsion tizimda jarayon nima?
2. Jarayonlarni boshqarish nima uchun kerak?
3. Jarayon dastur hisoblagichi nima uchun kerak?
4. Jarayonlarni boshqarish bloki nima va u nima uchun kerak?
5. Jarayonlarni boshqarish blokining vazifasi nima?
6. Jarayonlarni boshqarish bloki o'z ichiga qanday ma'lumotlarni oladi?
7. Jarayon qanday holatlarda bo'lishi mumkin?
8. Yangi holat bo'yicha ma'lumot bering.
9. Bajarilishga tayyor jarayonlar holatiga ta'rif bering.
10. Bajarilish holatiga ta'rif bering.
11. Kutish holatiga ta'rif bering.
12. Tugatish holatiga ta'rif bering.
13. Tayyor jarayonlar navbati nima?
14. Rejalashtiruvchi nima va uning vazifasi?
15. Uzoq muddatli rejalashtiruvchi qanday vazifani bajaradi?
16. O'rta muddatli rejalashtiruvchi qanday vazifani bajaradi?
17. Qisqa muddatli rejalashtiruvchi qanday vazifani bajaradi?

18. Kontekst jarayoni nima?
19. Kontekstni almashtirish nima?
20. Oqim nima?
21. Oqimning jarayondan farqini tushuntiring.
22. Oqimni afzalliklarini keltirib o‘ting.
23. Oqim turlari va bir-biridan farqlari?
24. Foydalanuvchi darajasidagi oqimlarni tushuntiring.
25. Yadro darajasidagi oqimlarni tushuntiring.
26. Ko‘p oqimli model turlarini keltiring.
27. Many to many modelini tushuntiring.
28. Many to one modelini tushuntiring.
29. One to one modelini tushuntiring.
30. Jarayonlarni dispetcherlashtirish nima?
31. Protsessorni rejalashtiruvchining vazifasi nima?
32. Protsessor manejerini tushuntiring.
33. Jarayonni rejalashtiruvchining vazifasi nima?
34. Vazifa va jarayon holatlari.
35. Protsessor dispetcherining vazifasini tushuntiring.
36. Rejalashtirish mezonlari nechta?
37. FCFS algoritmini tushuntiring.
38. SJF algoritmini tushuntiring.
39. Ustuvorliklar bo‘yicha dispetcherlashtirish algoritmini tushuntiring.
40. RR algoritmini tushuntiring.
41. Ko‘p darajali navbat algoritmini tushuntiring.
42. Berkliklar nima?
43. Berkliklar vujudga kelishi mumkin bo‘lgan shartlarni keltiring.
44. Resurslarni taqsimlash grafini tushuntiring.
45. Berkliklarga ishlov berish usullarini keltiring.
46. Berkliklarni oldini olishni tushuntiring.
47. Berkliklardan qochish usullari.

## **III BOB. XOTIRANI TASHKIL ETISH VA BOSHQARISH USULLARI**

### **3.1. Kompyuter xotirasi turlari va ularning tashkil etilishi**

Xotira aniq boshqaruvni talab qiladigan juda muhim resurs. Asosiy xotirani boshqarish juda muhimdir. Aslida, tarixiy nuqtai nazardan, butun tizimning ishlashi to'g'ridan-to'g'ri ikkita narsaga bog'liq edi: ishlov berishda qancha xotira mavjudligi va u qanday optimallashtirilganligida. Xotira ierarxiasini boshqaradigan operatsion tizimning qismi xotira menejeri yoki menejer deb ataladi. U xotirani samarali boshqarish uchun mo'ljallangan va xotiraning qaysi qismlari ishlatilishini kuzatishi, jarayonlar uchun kerak bo'lgan xotirani ajratishi va jarayonlar o'z ishini tugatgandan so'ng xotirani bo'shatishi kerak.

Ushbu bobda bir nechta turli xil xotiralarni boshqarish modellari ko'rib chiqiladi, juda oddiylaridan juda murakkablarigacha. Bundan tashqari virtual xotira tushunchasi masalalari ham ko'rib chiqilgan.

Xotira menejeri (shuningdek, operativ xotira yoki RAM, yoki asosiy xotira sifatida ham tanilgan) va xotirani taqsimlash sxemalarining to'rtta turi mavjud: bir foydalanuvchili tizimlar, belgilangan qismlar, dinamik qismlar va ko'chiriladigan dinamik qismlar. Hozirgi kundagi zamonaviy operatsion tizimlarda virtual xotira tushunchasi mavjud.

Xotira – buko'p dasturli operatsion tizim tomonidan ehtiyotkorlik bilan boshqarishni talab qiladigan muhim resursdir. Xotira ostida sifatida bu yerda tezkor (operativ) xotira tushuniladi. Tashqi xotira (saqlash qurilmasi) deb nomlanadigan qattiq diskning xotirasidan farqli o'laroq, ma'lumotlarni saqlash uchun operativ xotira doimiy quvvat manbai talab qiladi. Dastlabki operatsion tizimlarda xotirani boshqarish dasturni va uning ma'lumotlarini tashqi diskdan (perfokarta, magnit lenta yoki magnit disk) xotiraga yuklanar edi. Ko'p dasturlashni paydo bo'lishi bilan, operatsion tizim bir vaqtning o'zida bir nechta ishga tushuriladigan dasturlar o'rtasida mavjud xotirani taqsimlash bilan bog'liq yangi vazifalarga duch keldi.



Multidasturli tizimda xotirani boshqarish uchun OT funksiyalari quyidagilardan iborat:

- ❖ bo'sh va band xotirani kuzatish;
- ❖ jarayonlarga xotirani taqsimlash va jarayonlar tugaganda xotirani bo'shatish;
- ❖ asosiy xotira o'lchamlari undagi barcha jarayonlarni joylashtirish uchun yetarli bo'lmaganda jarayon kodlari va ma'lumotlarini operativ xotiradan diskka ko'chirish (to'liq yoki qisman), va qachonki unda bo'sh joy paydo bo'lganda ularni RAM ga qaytarish;
- ❖ fizik xotiraning ma'lum bir sohasiga dastur manzillarini o'rnatish;
- ❖ xotirani himoya qilish - boshqa jarayonga taqsimlangan xotiraga ma'lumotlarni yozishni yoki u yerdan o'qishni taqiqlash. Ushbu funksiya qoida tariqasida operatsion tizim dasturiy modullari tomonidan qurilma vositalari bilan yaqin hamkorlikda amalga oshiriladi.

### ***Kompyuter xotirasining fizik tuzilishi (tashkil etilishi)***

Kompyuterning xotira qurilmasi ikki xil turga: asosiy (bosh xotira, tezkor xotira, fizik xotira) va ikkilamchi (doimiy xotira, ichki xotira) xotiraga bo'linadi. Asosiy xotira bir baytli tartiblangan yacheyka massiviga ega bo'lib, har bir yacheyka o'zining manziliga (raqamiga) ega. Protsessor buyruqlarni asosiy xotiradan oladi, qayta ishlaydi va bajaradi. Buyruqlarni bajarishda asosiy xotiraning bir nechta yacheykalariga murojaat qilishga to'g'ri keladi. Odatda asosiy xotira yarim o'tkazgichli texnologiya asosida tayyorlanadi, shuning uchun xotiradagi ma'lumotlar elektr manbasidan uzilgandan so'ng o'chib ketadi.

Ikkilamchi xotira (bu asosan qattiq disklardir) bu chiziqli birlik manzilga ega bo'lgan joy va ularni ketma-ket joylashgan baytlar tashkil qiladi. Ikkilamchi xotiraning tezkor xotiradan farqi shundaki, u alohida energiyaga, katta hajmga, va samarali foydalanish imkoniyatiga ega.

3.1-rasmdagi ko'rsatilgan sxemaga yana bir nechta oraliq satxlarni qo'shish mumkin.

Har xil ko‘rinishdagi xotiralar ierarxiyaga, murojaat vaqti kamayib borishi, narxini oshishi va sig‘imi oshishi tarzida birlashishi mumkin.



3.1- rasm. Xotira ierarxiyasi

Ko‘p bosqichli sxemalar quyidagicha ishlatiladi. Ma’lumotlar odatda xotiraning yuqori satxlaridan qidiriladi, agar u yerdan topilmasa, ma’lumotlar katta raqamli satxlarda ham saqlanadi. Shuning uchun, u keyingi satxdan qidira boshlaydi. Agar kerakli ma’lumotni topsa, uni yuqoriroq satxga o‘tkazadi.

### *Manzil turlari*

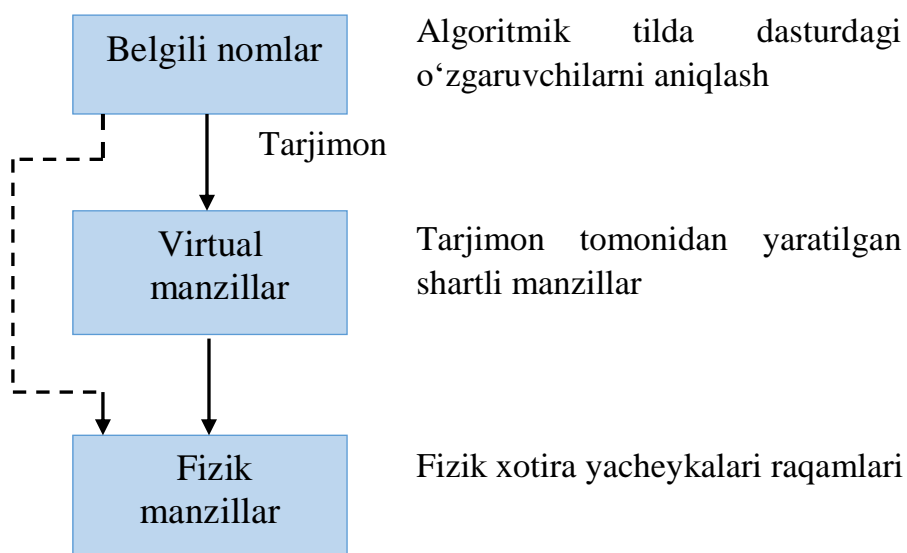
O‘zgaruvchilar va buyruqlarni aniqlash uchun dastur hayot siklining turli bosqichlarida belgili nomlar (belgilar), virtual manzillar va fizik manzillar ishlatiladi (3.2- rasm).

**Belgili nomlar** – foydalanuvchi tomonidan dasturni algoritmik tilda yoki assemblerda yozishda belgilanadi.

**Virtual manzillar** (matematik yoki mantiqiy) – dasturni mashina tiliga tarjima qiladigan tarjimon tomonidan yaratiladi. Tarjima paytida dastur tezkor xotiraning qayeriga yuklanishi umuman ma’lum emas, chunki tarjimon odatda o‘zgaruvchilar va buyruqlarga shartli virtual manzillarni tayinlaydi, odatda dasturning boshlang‘ich manzili nol manzili bo‘ladi.

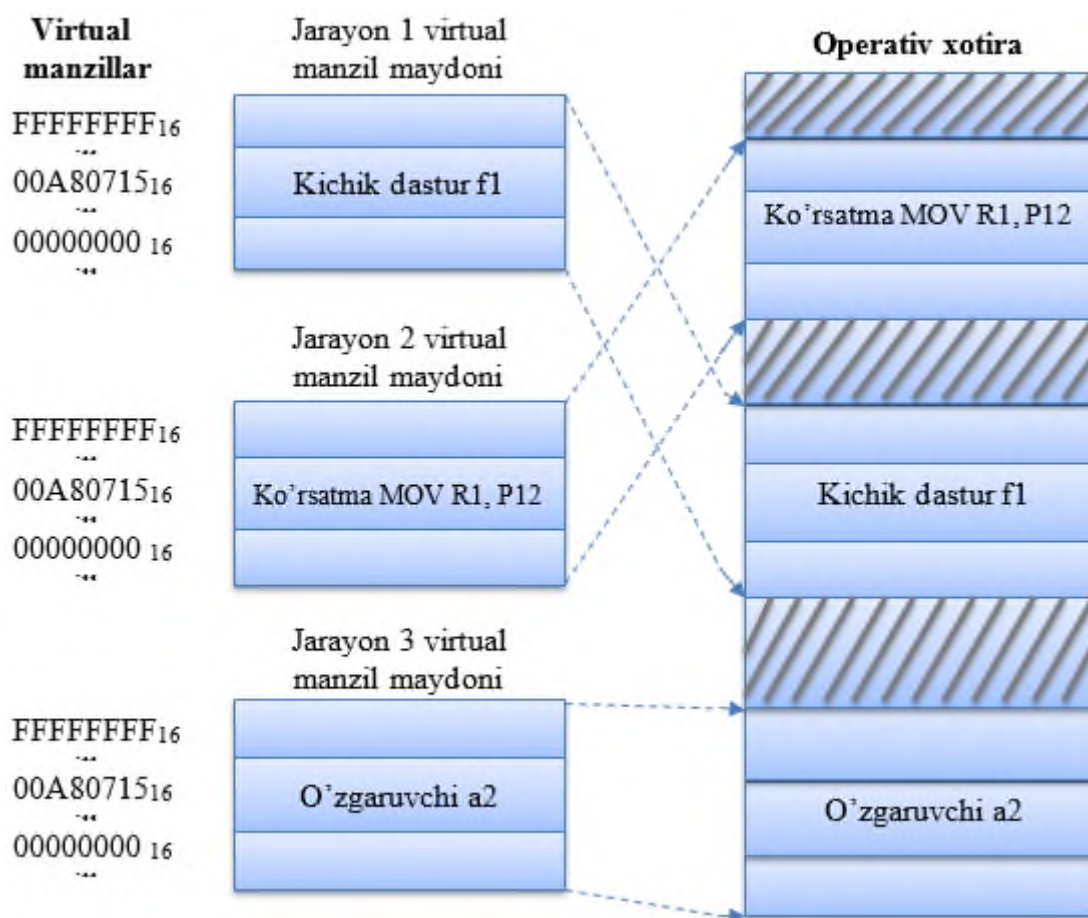
**Fizik manzillar** – o‘zgaruvchilar va buyruqlar joylashgan yoki joylashishi kerak bo‘lgan xotira xujayralari raqamlariga mos keladi.

Barcha jarayonlar uchun mumkin bo‘lgan virtual maydon manzillari diapazoni bir xil. Masalan, 32-razryadli virtual manzillardan foydalanganda, bu oraliq  $00000000_{16}$  va  $FFFFFFFF_{16}$  chegaralari bilan belgilanadi. Dastur tomonidan yaratilgan barcha virtual manzillarning to‘plami virtual manzil maydoni deb nomlanadi. Ushbu virtual manzillarga mos keladigan barcha fizik manzillarning to‘plami fizik manzillar maydoni deb nomlanadi. Virtualdan fizik manzilgacha ish vaqtini xaritalash, qurilma vositasi bo‘lgan xotirani boshqarish bloki (MMU) tomonidan amalga oshiriladi. MMU virtual manzilni fizik manzilga tarjima qilish uchun quyidagi mexanizmdan foydalanadi. Baza registridagi qiymat foydalanuvchi jarayoni natijasida hosil bo‘lgan har bir manzilga qo‘shiladi, ular xotiraga yuborilganida hisobga olinadi. Masalan, agar bazaviy registr qiymati 10000 bo‘lsa, foydalanuvchi 100 manzil manzilidan foydalanishga urinsa, dinamik ravishda 10100 manziliga joylashtiriladi. Foydalanuvchi dasturi virtual manzillar bilan shug‘ullanadi, u hech qachon haqiqiy fizik manzillarni ko‘rmaydi.



3.2- rasm. Manzil turlari

Shu bilan birga, har bir jarayon o‘zining virtual manzil maydoniga ega - tarjimon har bir dasturning parametrlari va kodlariga virtual manzillarni mustaqil ravishda tayinlaydi (3.3- rasm).



3.3- rasm. Bir nechta dasturlarning virtual manzil maydonlari

O'zgaruvchining virtual manzillari va turli xil jarayonlar ko'rsatmalarining bir-biriga to'g'ri kelishi nizolarga olib kelmaydi, chunki ushbu o'zgaruvchilar bir vaqtning o'zida xotirada bo'lganida, operatsion tizim ularni turli xil fizik manzillarga joylashtiradi. Turli xil operatsion tizimlar virtual manzillar maydonini tuzishning turli usullaridan foydalanadilar. Ular chiziqli va segmentli. Ba'zi bir operatsion tizimlarda, fizik xotira singari, jarayonning virtual manzillar maydoni doimiy ravishda virtual manzillarning ketma-ketligi sifatida taqdim etiladi. Manzil maydonining bunday tuzilishi chiziqli deb nomlanadi. Boshqa operatsion tizimlarda virtual manzillar maydoni segmentlar (yoki bo'limlar, yoki maydonlar yoki boshqa atamalar) deb nomlangan qismlarga bo'linadi.

### ***Fizik va mantiqiy xotira***

Fizik xotira. Ma'lum bo'lishicha, bu usulda boshqarishni tashkil etish xotira satxlariga kirishni va aloqa chastotasini kamaytiradi. Bu

yerda muhim rolni, chegaralangan vaqt davomida, xotira manzillarining kichik bo'lagi bilan ishlash xossasi o'ynaydi. Protessor keshi, qurilmalarning bir qismi hisoblanadi, shuning uchun operatsion tizimning xotira menejeri, asosan ma'lumotlarni kompyuterning asosiy va ichki xotira qismiga taqsimlash bilan shug'ullanadi. Bazi sxemalarda tezkor va ichki xotira o'rtasidagi oqimni dasturchi boshqaradi. Ammo bu bog'lanish dasturchi vaqtini yo'qotadi, shu sababli bu ishni OT ga yuklashga harakat qilinadi.

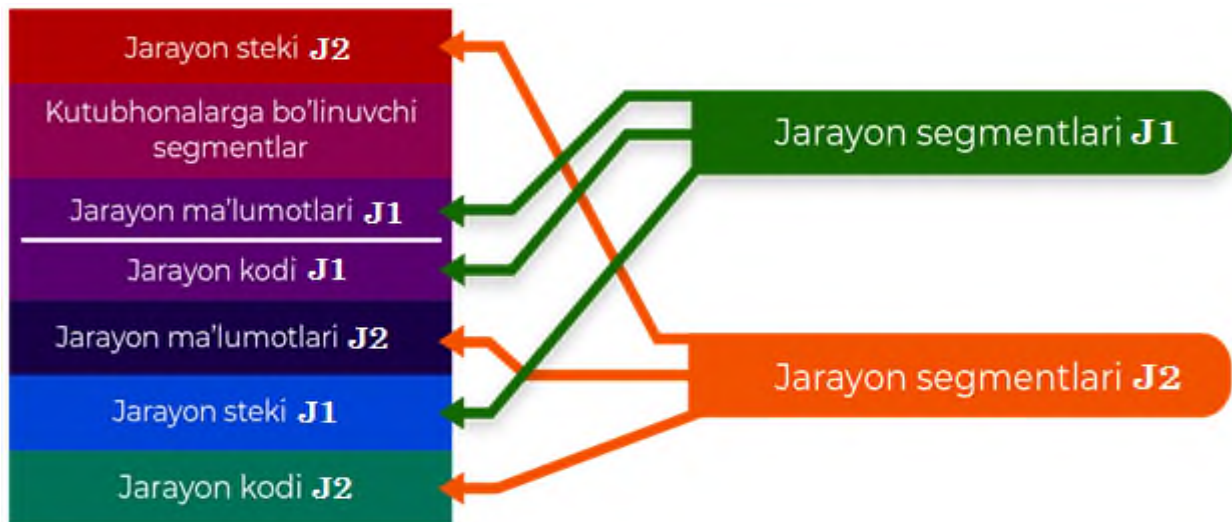
Fizik xotirada ma'lumotlarni real joylashishini ko'rsatuvchi asosiy xotiradagi manzillar - fizik manzillar deb ataladi. Dastur ishlaydigan fizik manzillar to'plami, fizik manzillar maydoni deb ataladi.

Mantiqiy (logical) xotira.Xotirani, yacheykalar chiziqli to'plami ko'rinishida tashkil etish, dasturchining dastur va ma'lumotlar saqlanishi ko'rinishi haqidagi tasavvuri bilan mos kelmaydi. Bazan jarayon tarkibiga kiruvchi hamma modullar xotirada ketma-ket joylashadi va chiziqli manzillar maydonini tashkil qiladi. Biroq ko'pincha modullar xotiraning turli joylarida joylashtiriladi va turlicha foydalaniladi.

Xotirani bosharish sxemasida, foydalanuvchining bunday tassavuriga mos keladigan ma'lumot va dasturlarni saqlash, segmentatsiya deyiladi. Segment-xotiraning aniq ko'rsatilgan qismi bo'lib, uning ichki qismida chiziqli manzillarni qo'llab-quvvatlaydi. Segment protsedura, massiv, stek yoki skalyar miqdorlardan tashkil topgan bo'ladi, lekin, odatda aralash turdagi ma'lumotlardan iborat bo'lmaydi. Boshida segmentlar dastur kodi fragmentlarini (matn redaktori, trigonometrik kutubxona va x.k.) jarayonlar bilan umumlashtirish zaruriyatidan kelib chiqqan bo'lishi kerak, chunki ularsiz har bir jarayon o'zining manzil maydonida ma'lumotlarning yana bir nusxasini saqlashiga to'g'ri kelar edi. Xotiraning, tizim bir nechta jarayonning ma'lumotlarini aks ettiradigan alohida qismlari bo'lib ular segmentlar deb nom oldi.

Xotira shunday qilib, chiziqli ko'rinishdan ikki o'lchamli ko'rinishga keldi. Manzil ikki komponentdan iborat bo'lib, ular: segment raqami va segment ichidagi joylashgan o'rnidir. Keyinchalik, jarayonning turli komponentalarini (dastur kodi, ma'lumotlar, stek va x.k.) turli segmentlarda joylashtirish qulay bo'lib qoldi. Yana shu narsa aniq bo'lib qoldiki, aniq segment ishini, unga segmentda

saqlanadigan ma'lumotlar ustida bajarilishi ruhsat berilgan operatsiyalar, masalan, murojaat xuquqi va operatsiyalar turi kabi atributlar qiymatini berib, nazorat qilish mumkin bo'lib qoldi. Jarayon segmentlarining kompyuter xotirasida joylashishi 3.4- rasmida ko'rsatilgan.



3.4- rasm. Jarayon segmentlarining kompyuter xotirasida joylashishi

Bazi, jarayonni manzil manzilini tasvirlaydigan segmentlar 3.4- rasmida ko'rsatilgan.

Aksariyat zamonaviy operatsion tizimlar xotirani segmentli boshqaruv xususiyatiga ega. OT larning bazi arxitekturalarida (masalan, Intel) segmentlash qurilmalar tomonidan qo'llanadi. Jarayon murojaat qiladigan manzillar, operativ xotirada mavjud bo'lgan real manzillardan shu tarzda farq qiladi. Har bir aniq holatda dastur foydalanadigan manzil, har xil usullar yordamida tasvirlanishi mumkin. Masalan, manzil, berilgan matnda odatda belgili bo'ladi. Kompilyator bu belgili manzil va o'zgaradigan manzillarni bog'laydi (masalan, n bayt modul boshidan). Dastur generatsiyalagan bunday manzil odatda mantiqiy manzil (virtual xotirali tizimlarda u ko'pincha virtual xotira) deb nomlanadi. Barcha mantiqiy manzillar to'plami mantiqiy (virtual) manzillar maydoni deb ataladi.

### ***Manzillar bog'lanishi***

Demak, mantiqiy va fizik manzillar maydonlari, tashkil etilishi va o'lchami bo'yicha bir biriga mos emas. Mantiqiy manzillar

maydonning maksimal o'lchami odatda protsessorning razryadi bilan aniqlanadi (masalan  $2^{32}$ ), va zamonaviy tizimlarda fizik manzillar maydonining hajmidan ko'zga ko'rinarli darajada yuqori bo'ladi. Shunday ekan, protsessor va operatsion tizim asosiy xotirada joylashgan dasturni dastur kodiga, real fizik manzilga tayangan holda yo'lni aks ettirishi kerak. Bunday ko'rinishda manzillarni tasvirlash manzillarni translatsiyasi yoki manzillarni bog'lash deb nomlanadi.

Mantiqiy manzilning fizik manzil bilan bog'lanishi dastur operatorining bajarilishigacha yoki bajarilish vaqtida amalga oshirilishi shart. Bunday holda, ko'rsatmalarni va ma'lumotlarni xotiraga bog'lash quyidagi qadamlar bo'yicha amalga oshiriladi.

❖ **Kompilyatsiya bosqichi.** Kompilatsiya bosqichida siz jarayonning xotirada joylashish joyini bilsangiz, unda mutloq (absolyut) kodni yaratishingiz mumkin. Masalan, agar foydalanuvchi jarayoni doimiy ravishda R joyida boshlanishini bilsangiz, u holda yaratilgan kompilyator kodi o'sha joyda boshlanadi va u yerdan davom etadi. Agar bir muncha vaqt o'tgach, dastlabki joylashuv o'zgarsa, ushbu kodni qayta yozishingiz kerak bo'ladi.

❖ **Yuklash bosqichi.** Agar kompilatsiya bosqichida dasturlarni joylashtirish to'g'risida axborot mavjud bo'lmasa, u holda kompilyator o'zgaruvchan kodni yaratishi kerak. Bu holatda yakuniy bog'lanish yuklanish vaqtigacha xotirada saqlanadi. Agar dastlabki manzil o'zgarsa, o'zgarishlarni hisobga olib foydalanuvchi kodini qayta yuklash kerak bo'ladi.

❖ **Bajarilish bosqichi.** Agar jarayonni bajarilish vaqtida bir xotira segmentidan boshqa xotira segmentiga o'tkazish mumkin bo'lsa, u holda bajarilish muddati tugaguncha bog'lanish kechiktirilishi kerak. Bu yerda mavjud maxsus qurilmalardan foydalangan ma'qul, masalan, register joyini o'zgartirish. Ko'plab zamonaviy operatsion tizimlar bajarilish bosqichida manzil translatsiyasidan foydalanadi, buning uchun maxsus qurilma vositalaridan foydalaniladi.

### **3.2. Xotirani boshqarish va undan umumiy foydalanish mexanizmi**

#### ***Xotiraning boshqaruv tizimi funksiyasi***

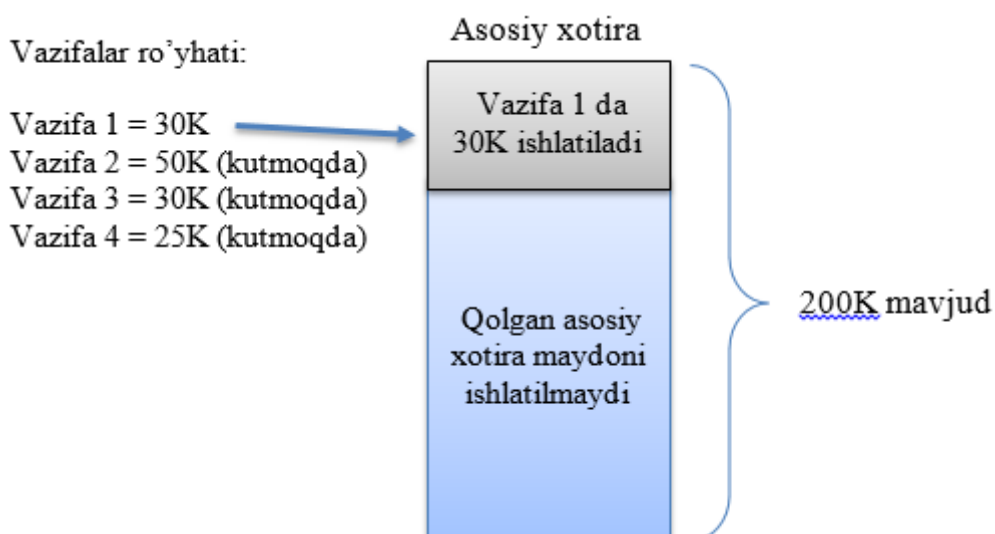
Xotiradan samarali foydalanishni ta'minlash uchun operatsion tizim quyidagi funksiyalarni bajarishi lozim:

- ❖ Fizik xotirani aniq bir sohasida jarayon manzillar to'plamini aks ettirish;
- ❖ Qarama-qarshi jarayonlar o'rtasida xotirani taqsimlash;
- ❖ Jarayonlar manzillar maydoniga ruhsatni boshqarish;
- ❖ Operativ xotirada joy qolmaganda, tashqi xotiraga jarayonlarni (qisman yoki to'liq) yuklash;
- ❖ Bo'sh va band xotirani hisobga olish.

### ***Bir foydalanuvchili sxema***

Dastlabki operatsion tizimlarda xotirani boshqarishning eng oddiy usullari qo'llanilgan. Boshida foydalanuvchining har bir jarayoni asosiy xotiraga ko'chirilishi kerak bo'lgan, xotira uzluksiz maydonini band qilgan, tizim esa qo'shimcha foydalanuvchi jarayonlarga bir vaqtning o'zida asosiy xotirada joylashib turgunicha xizmat ko'rsatadi.

Xotirani taqsimlashning birinchi sxemasi: 3.5-rasmda ko'rsatilgandek, har bir qayta ishlangan dastur to'liq xotiraga yuklandi va unga zarur bo'lgan xotira maydonini ajratdi. Bu yerda kalit so'zlar yaxlitlik va doimiylikdir. Agar dastur juda katta bo'lsa va mavjud xotira maydoniga mos kelmasa (xotira maydoni o'lchami yetarli bo'lmasa), uni bajarib bo'lmaydi. Va dastlabki kompyuterlar juda katta bo'lishiga qaramay, ular juda kam xotiraga ega edilar.





3.5- rasm. Bitta dastur bir vaqtning o'zida xotirada saqlanadi.  
Xotiraning qolgan qismi ishlatilmaydi

Bu barcha kompyuterlar uchun sezilarli cheklovni namoyish etadi – ular faqat cheklangan xotiraga ega va agar dastur xotira maydoniga mos kelmasa asosiy xotiraning hajmini oshirish kerak yoki dasturni o'zgartirish (kichraytirish) kerak. Odatda, uni kichiklashtirish yoki dastur segmentlarini (dasturga bo'laklarni) to'ldirishga imkon beradigan usullardan foydalanish orqali o'zgartiriladi. (Qoplash usuli (overlay) - bu dastur segmentlarini ikkinchi darajali xotiradan asosiy xotiraga bajarish uchun o'tkazishdir, shunda ikki yoki undan ortiq segmentlar bir xil xotira maydonlarini egallab turgan navbatni egallab olishlari mumkin.) Overlay - bu tizimda o'rnatilganidan ko'ra ko'proq xotirani egallaydigan dasturlarni yaratishga imkon beradigan dasturlash usulidir. Har bir foydalanuvchiga har bir vazifa uchun mavjud bo'lgan asosiy xotiradan foydalanish huquqi beriladi va vazifalar ketma-ket bajariladi. Xotirani taqsimlash uchun operatsion tizim oddiy algoritmdan foydalanadi (muammoni hal qilish uchun qadamma-qadam protsedura):

Vazifani bir foydalanuvchi tizimiga yuklash algoritmi

1. Dastur xotirasini asosiy registrda saqlash (xotirani himoyasi uchun)
2. Dastur hisoblagichini birinchi xotira yacheyasining manziliga o'rnatish (u dastur tomonidan ishlatiladigan xotira miqdorini kuzatadi)
3. Dasturning birinchi ko'rsatmasini o'qish
4. Dastur hisoblagichini ko'rsatmadagi baytlar soniga ko'paytirish
5. So'nggi ko'rsatma berildimi?  
agar ha bo'lsa, dasturni yuklashni to'xtatish  
agar yo'q bo'lsa, unda 6-bosqichdan davom etish
6. Dastur hisoblagichi xotira hajmidan kattaroqmi?  
agar ha bo'lsa, dasturni yuklashni to'xtatish  
agar yo'q bo'lsa, unda 7-bosqich bilan davom etish
7. Ko'rsatmani xotiraga yuklash
8. Keyingi dastur ko'rsatmalarini o'qish
9. 4-bosqichga o'tish.

E'tibor bering, operatsion tizimning "Xotira menejeri" tomonidan bajarilgan ishlarning hajmi minimal, funksiyalarni bajarish kodi sodda va mantiq juda oddiy.

Faqat ikkita qurilma vositasi kerak bo'ladi: asosiy manzilni saqlash uchun registr va xotirada o'qilayotgan paytda dastur hajmini kuzatib borish uchun akkumulyator (y ham register).

Dastur xotiraga to'liq yuklanganidan so'ng, u odatiy tarzda yoki operatsion tizimning aralashuvi bilan tugatilmaguncha o'sha yerda qoladi.

Ushbu turdagi xotirani taqsimlash sxemasining asosiy muammolaridan biri shundaki, u ko'p dasturlash yoki tarmoqni qo'llab-quvvatlamaydi, u bir vaqtning o'zida faqat bitta vazifani bajara oladi. Bu usul 1940- va 1950- yillardagi tijorat kompyuterlarida qo'llanilgan. Shuning uchun, 1950- yillarning ohiri va 1960- yillarning boshlarida xotirani boshqarishning yangi sxemasi talab qilindi, va mustaqil operatsiyalarni takrorlash orqali kompyuter tizimi resurslaridan foydalanishga imkon beradigan qismlardan foydalanildi.

Xotirani boshqarishning ushbu sxemalari bugungi operatsion tizimlar tomonidan kam ishlatiladi, ammo o'rganish juda muhimdir, chunki har biri xotira boshqaruvini rivojlantirishga yordam beradigan asosiy tushunchalarni taqdim etadi. Kompyuter tizimi bosh asosiy masalasi dasturni boshqarishdir. Xotiraning alohida roli protsessor dastur ko'rsatmalarini ular xotirada bo'lgan taqdirdagina bajarishi mumkinligi bilan izohlanadi. Dasturlar va ularning murojaat qiladigan ma'lumotlari, bajarilish jarayonida operativ xotirada (hech bo'lmasa qisman) joylashgan bo'lishi shart. Operatsion tizimga, xotirani, foydalanuvchi jarayonlari va OT komponentalari orasida taqsimlashga to'g'ri keladi. Operatsion tizimning bu faoliyati xotirani boshqarish deyiladi. Shunday qilib, xotira sinchiklab boshqarishni talab etadigan resursdir. Yaqin kunlarga xotira eng qimmat resurs hisoblangan. Operatsion tizimning, xotirani boshqaradigan qismi, xotira menejeri deyiladi. Xotira amaliy dastur modullari va operatsion tizimning modullari o'rtasida ham taqsimlanadi.

### **3.3. Xotirani taqsimlash, qayta taqsimlash va jamlash usullari**

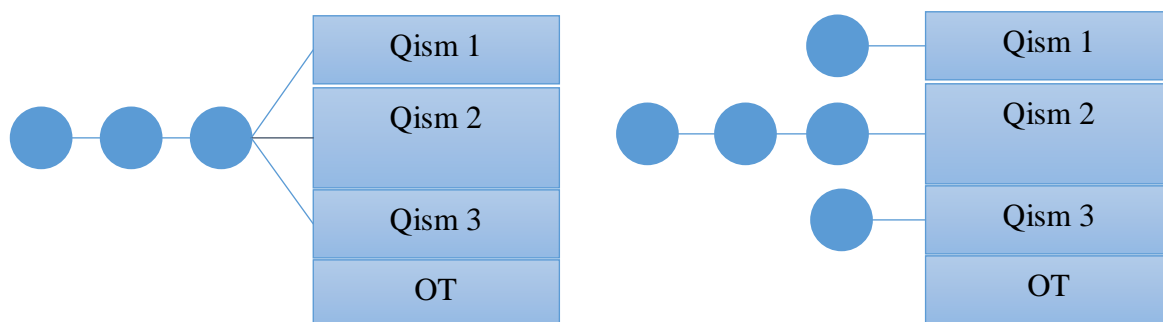
#### ***Belgilangan (statik) qismlar***

Tezkor xotirani boshqarishning eng oddiy yo‘li uni oldindan (ishga tushurish bosqichida yoki tizim yuklanishi vaqtida) bir qancha belgilangan (statik) o‘lchamdagi qismlarga bo‘lishdan iboratdir. Kelib tushayotgan jarayonlar u yoki bu qismga joylashtiriladi. Shu sababli fizik manzillar maydonining shartli bo‘linishi yuzaga keladi. Jarayonning mantiqiy va fizik manzillari bog‘lanishi uni aniq bir qismga yuklash vaqtida yoki bazan kompilatsiya vaqtida yuzaga keladi.

Har bir qism o‘zining jarayonlar navbatiga ega, yoki hamma qismlar uchun jarayonlar global navbati mavjud bo‘lishi mumkin. Bu sxema IBM OS/360 (MFT), DES RSX-11 va shunga yaqin boshqa tizimlarda qo‘llanilgan. Xotirani boshqarish tizimi jarayonni hajmini baholaydi, unga mos keluvchi qismni tanlaydi, jarayonni bu qismga yuklaydi va manzillarni sozlaydi.

3.6- rasmda belgilangan qisimli sxemalar ko‘rsatilgan: (a) navbati umumiy bo‘lgan jarayonlar, (b) alohida navbatli jarayonlar.

Bu sxemaning kamchiligi ko‘rinib turibdiki, bir vaqtda bajariladigan jarayonlar soni qismlar soni bilan cheklangan. Boshqa muhim kamchiligi shundan iboratki, taklif qilinayotgan sxema, ichki fragmentlashdan, yani jarayonga ajratilgan, ammo ishlatilmagan xotira qismini yo‘qotish bilan qattiq zararlanadi. Fragmentatsiya, jarayon o‘ziga ajratilgan qismni to‘liq band qilmasligi yoki bazi qismlar, bajariladigan foydalanuvchi dasturlari uchun kichik bo‘lganligidan kelib chiqadi.



3.6- rasm. Overlayli (qoplangan) tuzilish

Jarayon mantiqiy manzillar maydoni hajmi, unga ajratilgan qism hajmidan katta (yoki eng katta hajmdan ham katta) bo‘lgan holatlarda, ba‘zan overlay nomli (yoki qoplanadigan tuzilishli) tashkil etadigan texnikadan foydalaniladi.

Asosiy g'oya – faqat ayni vaqtda kerak bo'lgan dastur ko'rsatmalarini xotirada saqlab turishdir. Overlay tuzilishning tavsifini yozish uchun odatda maxsus sodda (overlay description language) tildan foydalaniladi.

Asosiy xotirada qo'llanilgan belgilangan qismlar (statik qismlar deb ham ataladi) multidasturlashga imkon beradigan birinchi urinishdir – har bir vazifa uchun bitta qism. Har bir qismning o'lchami tizim ishga tushirilganda belgilanganligi sababli, har bir qism faqat kompyuter tizimi o'chirilgan, qayta konfiguratsiya qilingan va qayta ishga tushirilganida qayta tuzilishi mumkin edi. Shunday qilib, tizimni ishga tushirgandan so'ng, qism o'lchamlari o'zgarishsiz qoldi. Ushbu sxemada muhim omil paydo bo'ldi: vazifa xotirasi maydonini himoya qilish. Qismga vazifa berilgandan so'ng, boshqa vazifalarga unga tasodifan yoki qasddan kirishga ruhsat berilmagan. Qismlarga kirishning bu usuli bir foydalanuvchili tizimga qaraganda har bir qismni himoya qilishni nazarda tutadi. Chunki bir foydalanuvchilik tizimda istalgan vaqtda faqat bitta ish asosiy xotirada bo'lgan, shuning uchun operatsion tizimning asosiy xotirada joylashgan qismi himoya qilinishi kerak edi. Shu bilan birga, belgilangan qismni taqsimlash sxemalari uchun asosiy xotirada mavjud bo'lgan har bir qism uchun himoya majburiy edi. Odatda bu kompyuter qurilmalari va operatsion tizimning birgalikdagi javobgarligi edi. Vazifalarni xotirada saqlash uchun ishlatiladigan algoritm bir foydalanuvchi tizimida ishlatilgandan ko'ra bir necha marotaba ko'proq qadamlarni talab qiladi, chunki vazifa hajmi to'liq mos kelishiga ishonch hosil qilish uchun bo'lim hajmiga mos kelishi kerak. Keyin, yetarlicha kattalikdagi blok aniqlanganda, uning mavjudligini tekshirish uchun qismning holatini tekshirish kerak. Har bir qism faqat bitta dastur tomonidan ishlatilishi mumkin. Har bir qismning o'lchamini kompyuter operatori oldindan belgilab qo'ygan, shuning uchun tizimni qayta ishga tushurmasdan hajmni o'zgartirish mumkin emas.

Vazifani belgilangan qismlarga yuklash algoritmi

1. Vazifa talab qilgan xotira hajmini aniqlash
2. Agar vazifa hajmi  $>$  qism hajmidan katta bo'lsa

Unda, vazifani rad etadi

Operatorga kerakli xabarni chop etadi

Navbatdagi vazifani bajarish uchun 1-qadamga o'tish

Aks holda

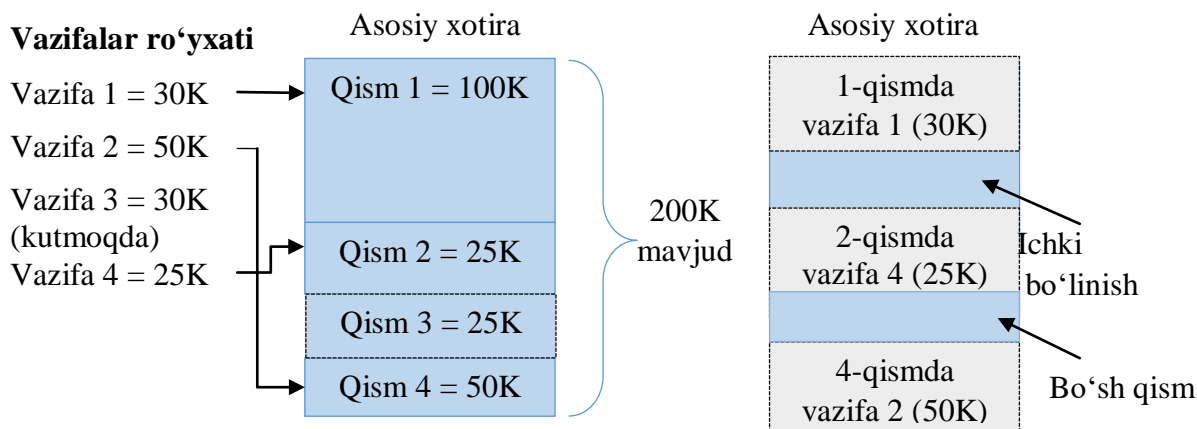
- 3-qadamdan davom etadi
3. Hisoblagichni 1 ga o‘rnatish
  4. Hisoblagich o‘rnatilganda xotiradagi bo‘limlar soni katta yoki teng bo‘lganda bajarish  
 Agar vazifa hajmi xotira qismi hajmidan (hisoblagich) katta bo‘lsa  
 Unda hisoblagich = hisoblagich + 1  
 Aks holda  
 Agar xotira qismi hajmi (hisoblagich) = “bo‘sh” bo‘lsa  
 Unda xotira qismiga (hisoblagich) vazifani yuklash  
 Xotira qismi holatini (hisoblagich) “band” ga o‘zgartirish  
 Aks holda  
 Hisoblagich = hisoblagich + 1  
 Bajarishni tugatish
  5. Hozirgi vaqtda bo‘sh qismlar mavjud bo‘lmasa, vazifani kutish navbatiga qo‘yish
  6. Navbatda turgan vazifani bajarish uchun 1-qadamga o‘tish.

Ushbu taqsimlash sxemasi bir foydalanuvchi tizimiga qaraganda ancha moslashuvchan, chunki u bir vaqtning o‘zida bir nechta dasturlarni xotirada saqlashga imkon beradi. Biroq, bu hali ham dasturni doimiy ravishda va bajarilishning boshidan oxirigacha xotirada saqlashni talab qiladi. Vazifa uchun xotira maydonini taqsimlash uchun operatsion tizimning xotira menejeri har bir xotira qismining o‘lchamini, uning manzili, kirish cheklovlari va tizimning joriy holatini (bo‘sh yoki band) ko‘rsatadigan jadvalni (3.1- jadval) saqlashi kerak.

3.1- jadval

Qism hajmi	Xotira manzili	Ruxsat	Qism holati
100 K	200 K	Vazifa 1	Band
25 K	300 K	Vazifa 4	Band
25 K	325 K		Bo‘sh
50 K	350 K	Vazifa 2	Band

Har bir vazifa bajarilib tugatilgach, uning xotira qismining holati band holatdan bo‘shga o‘zgaradi, va ushbu qismga kiruvchi boshqa vazifani kiritish mumkin.



3.7- rasm. Asosiy xotiradan qismlarga ajratilgan holda foydalanish

Asosiy xotiradan qismlarga ajratilgan holda foydalanish 3.1- jadvalda keltirilgan. Vazifa 1 mavjud 100 Kbayt xotiraning atigi 30 Kbaytini oladi, vazifa 3 qismlarni bo‘shlashini kutishi kerak, hatto 1- qismda 70K bo‘sh joy mavjud bo‘lsa ham. Vazifalarga “kerakli hajmdagi birinchi mavjud qism” asosida bo‘sh joy ajratilgan.

Agar tizimda bajariladigan barcha vazifalar bir xil o‘lchamli bo‘lsa yoki o‘lchamlari oldindan ma’lum bo‘lsa va konfiguratsiyaga qarab o‘zgarmasa, belgilangan qismlar sxemasi yaxshi ishlaydi. Ideal holda, bu tizimda yaqin soatlarda, kunlarda yoki haftalarda bajariladigan barcha vazifalar to‘g‘risida aniq ma’lumot talab qiladi. Ammo, agar operator kelajakni aniq taxmin qila olmasa, qismlarning o‘lchamlari tasodifiy ravishda belgilanadi va ularning o‘lchami kiruvchi vazifalar uchun juda kichik yoki juda katta bo‘lishi mumkin. Qismning o‘lchamlari juda kichik bo‘lsa, jiddiy oqibatlarga olib keladi. Boshqa tomondan, agar qismlar juda katta bo‘lsa, xotira behuda sarflanadi. Agar vazifa to‘liq qismni egallamasa, qismda foydalanilmagan xotira bo‘sh qoladi, uni boshqa vazifaga o‘tkazish mumkin emas, chunki har bir qism bir vaqtning o‘zida bitta vazifaga beriladi. Bu ajralmas birlikdir. 3.7- rasmda shunday vaziyatlardan biri ko‘rsatilgan. Ruxsat etilgan qismlardan qisman foydalanish va bir vaqtning o‘zida qism ichida foydalanilmagan bo‘shliqlarni yaratishning bu hodisasi ichki bo‘linish (fragmentatsiya) deb ataladi

va belgilangan qismlar uchun xotirani taqsimlash sxemasining asosiy kamchiligi hisoblanadi.

## ***Dinamik qismlar***

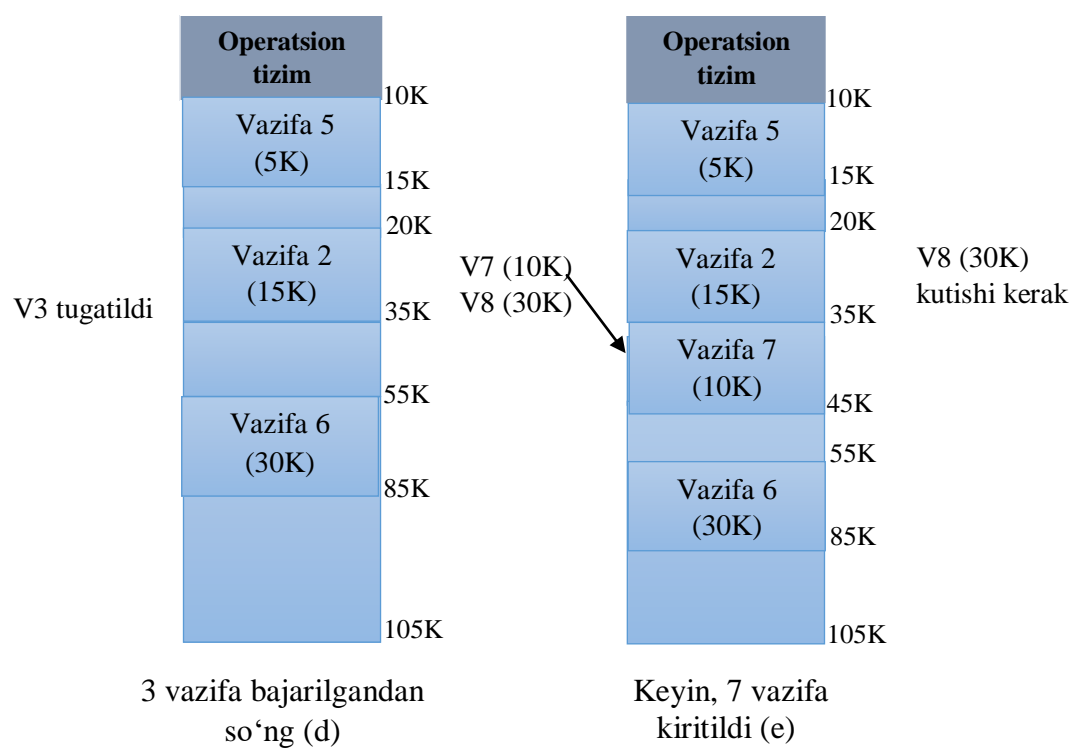
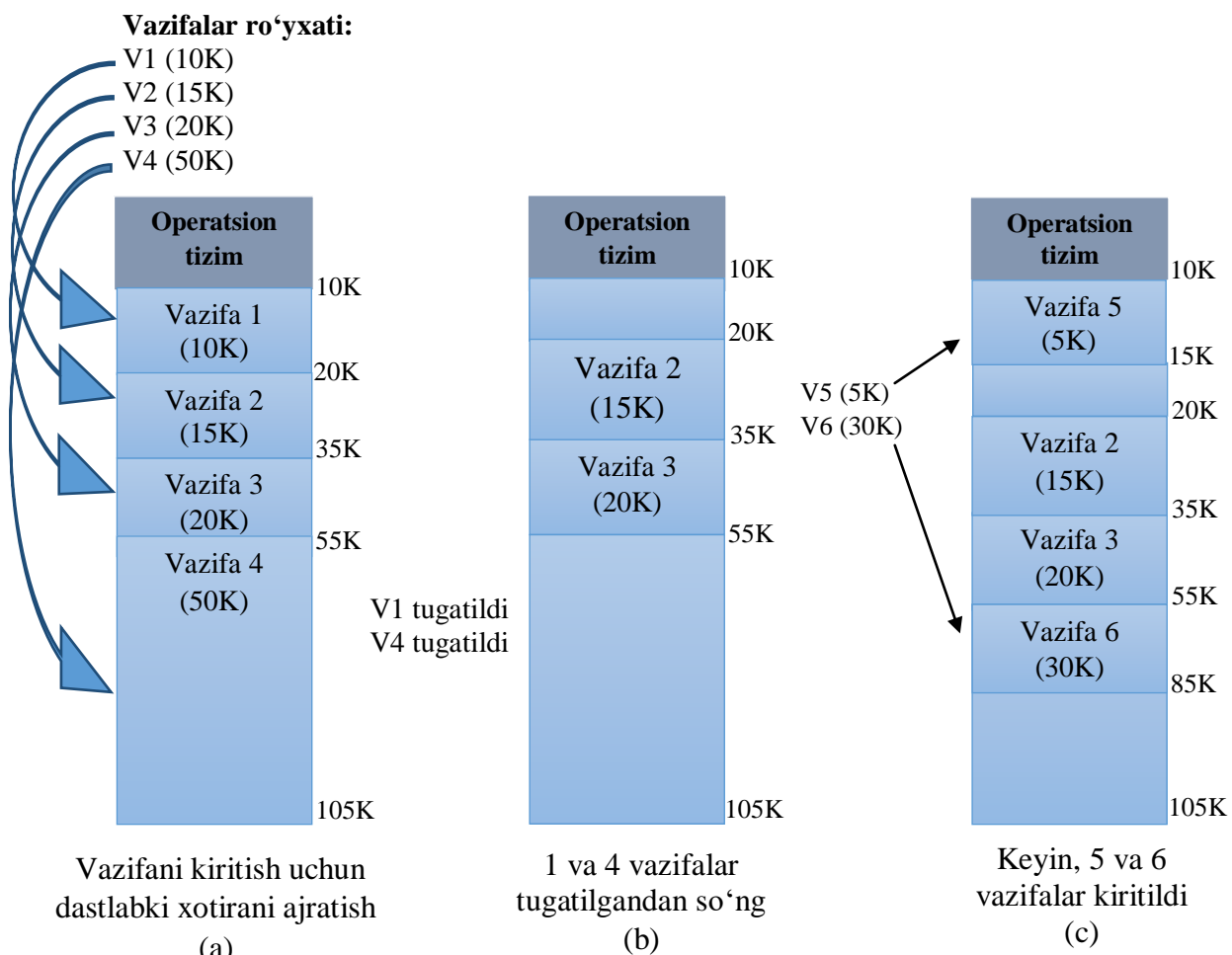
Dinamik qismlarda mavjud xotira qo'shni bloklarda saqlanadi, lekin vazifalarga yuklanish vaqtida qancha xotira maydoni kerak bo'lsa, shuncha xotira maydoni taqdim qilinadi. Bu belgilangan qismlarga nisbatan sezilarli yaxshilanish bo'lsada, bu muammoni to'liq hal qilmaydi. 3.8- rasmda ko'rsatilgandek, dinamik qismlarni taqsimlash sxemasi birinchi vazifalarni yuklashda xotiradan to'liq foydalanadi.

Ammo tizimga yangi vazifalar kiritilganda, vazifaning kattaligi bo'shagan xotirani hajmiga to'g'ri kelmasa, ular bo'sh joyga ustuvorlik (prioritet) tartibida joylashtiriladi. 3.8- rasmda "birinchi kelganga - birinchi xizmat" ustuvorligi ko'rsatilgan. Shunday qilib, keyingi xotirani taqsimlash, taqsimlangan xotira bloklari o'rtasida bo'sh xotiraning bo'laklarini (fragmentlarini) yaratadi. Ushbu muammo tashqi bo'linish deb nomlanadi va ichki bo'linish singari, xotirani yo'qotishga imkon beradi. 3.8- rasmda (e) uchta bo'sh qism mavjud 5 KB, 10 KB va 20 KB jami 35 KB, 8- vazifani bajarish uchun atigi 30 KB kerak bo'ladi. Biroq, ular bir-biriga qo'shni emasligi va vazifalar doimiy ravishda yuklanganligi sababli, ushbu sxema 8- vazifani kutishga majbur qiladi. Asosiy xotirani taqsimlashda dinamik qismlardan foydalanish. Quyidagi 5 ta rasmda (a-e) asosiy xotiraga qayta ishlash uchun 8 ta vazifa yuborildi va "birinchi kelganga - birinchi xizmat" algoritmi asosida xotira maydoni ajratilgan. Hatto qismlar orasida yetarli bo'sh xotira bo'lsa ham, 8 vazifa (e) kutishi kerak.

### ***Xotirani taqsimlashning umumiy vazifasi va uni hal qilish strategiyalari***

Umuman olganda, operatsion tizimlarda bir nechta qo'shni hududlarda qo'shni xotirani taqsimlash qo'llanilishi mumkin. Bo'sh maydon – bu bo'sh xotiraning qo'shni blokidir. Bo'sh maydonlar tasodifiy ravishda xotiradan tarqalishi mumkin. Jarayonni yuklashda, uni joylashtirish uchun yetarlicha katta bo'lgan har qanday bo'sh qo'shni hududdan xotira taqdim etiladi. Shu bilan birga operatsion tizim bo'sh xotira maydonlari ro'yhati va band xotira maydonlari ro'yhatini saqlaydi. Ushbu maydonlarning barchasi tasodifiy xotirada joylashgan bo'lishi mumkin va turli uzunliklarga ega.





3.8- rasm. Qismlarni dinamik taqsimlashda asosiy xotiradan foydalanish

Xotirani taqsimlashning umumiy vazifasi mavjud boʻsh xotira maydonlarini roʻyxatlash va ularni qilingan maydonlar roʻyxatlash. Ushbu muammoni hal qilish uchun quyidagi algoritmlar (strategiyalar) qoʻllaniladi: birinchi mos usul (First-Fit algorithm), eng yaxshi moslash usuli (Best-Fit algorithm) va eng yomon moslash usuli (Worst-Fit algorithm).

### ***Birinchi nisbatan optimal taqsimlash (Best-Fit Versus First-Fit Allocation)***

Birinchi (First-Fit Allocation) mos keladigan taqsimlash sxemasidan foydalanib, 1-vazifa birinchi mavjud boʻsh joyni talab qiladi va egallaydi. Operatsion tizim mos qismni qidirmaydi, lekin yetarli hajmga ega boʻlgan eng yaqin joylashgan xotira qismiga ishni taqsimlaydi.

Ushbu qismlar birinchi mos keladigan xotirani ajratish (birinchi qism talablariga javob beradi) yoki eng yaxshi (yoʻqotishlarning eng kam miqdori, talablarga javob beradigan eng kichik qism) xotira taqsimoti (Best-Fit Allocation) asosida ajratilishi mumkin. Ikkala sxemada ham xotira menejeri boʻsh yoki foydalanilgan qismlarning (boʻsh/band) xotira roʻyxatlarini hajmiga yoki joylashishiga qarab tashkil qiladi. Eng yaxshi mos keladigan taqsimlash usuli boʻsh/band boʻlgan roʻyxatlarni kichikdan kattasiga qarab tartiblaydi. Birinchi mos usul (first-fit method) xotira maydonlari tomonidan tashkil etilgan boʻsh/band boʻlmagan roʻyxatlarni, past darajali xotiradan yuqori darajali xotiraga qadar saqlaydi. Ularning har biri maʼlum bir taqsimlash sxemasining ehtiyojlariga qarab oʻz afzalliklariga ega - eng yaxshi taqsimlash usuli odatda xotira maydonidan unumli foydalanishni taʼminlaydi, birinchi mos taqsimlash algoritmi tezroq taqsimlashni amalga oshiradi.

Birinchi mos keladigan sxemadan foydalanib, 1- vazifa birinchi boʻsh joyni talab qiladi. 2- vazifa, birinchi qismni talab qiladi, joylashishi uchun yetarlicha katta, ammo 3- vazifani bajarish uchun yetarlicha katta boʻlgan oxirgi blokni talab qiladi. Shuning uchun, 3- vazifa (yulduzcha bilan belgilangan) 75 Mb foydalanilmagan xotira maydoni (ichki qism) mavjud boʻlsa ham, katta blok paydo boʻlguncha kutishi kerak. Eʼtibor bering, xotira roʻyxati xotira maydoniga qarab tartiblangan.

Vazifa ro'yxati:

Vazifa raqami	So'ralgan xotira
V1	10K
V2	20K
V3	30K*
V4	10K

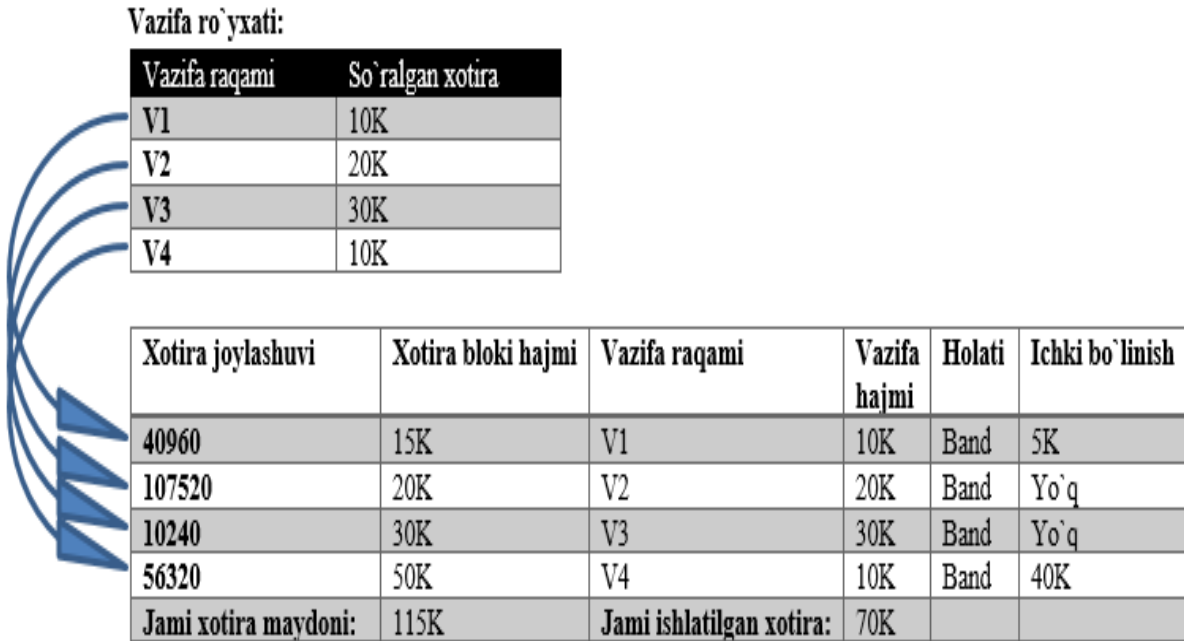
Xotira joylashuvi	Xotira bloki hajmi	Vazifa raqami	Vazifa hajmi	Holati	Ichki bo'linish
10240	30K	V1	10K	Band	20K
40960	15K	V4	20K	Band	5K
56320	50K	V2	30K	Band	30K
107520	20K			Bo'sh	
<b>Jami xotira maydoni:</b>	<b>115K</b>	<b>Jami ishlatilgan xotira:</b>	<b>40K</b>		

### 3.9- rasm. Birinchi mos keladigan sxemadan foydalanish

Xotiradan foydalanish hajmi oshirildi, ammo xotirani taqsimlash jarayoni ko'proq vaqtni talab etadi. Bundan tashqari, ichki bo'linish kamaygan bo'lsa ham, u to'liq yo'q qilinmadi.

Birinchi mos keladigan algoritm, xotira menejeri uchun ikkita ro'yxatni saqlaydi, birinchisi bo'sh xotira bloklari va ikkinchisi band qilingan xotira bloklari. Operatsiya har bir vazifa hajmini har bir xotira blokining o'lchami bilan mos keladigan yetarlicha katta blok topilmaguncha taqqoslaydigan oddiy sikldan iborat. Keyin vazifa ushbu xotira blokida saqlanadi va xotira menejeri keyingi navbatni kirish navbatidan olish uchun sikldan chiqadi. Agar butun ro'yxat behuda qidirilsa, u holda vazifa kutish navbati ichiga joylashtiriladi. Keyin xotira menejeri keyingi vazifani tanlaydi va jarayonni takrorlaydi.

Eng yaxshi taqsimlash sxemasi mos keladi. 1-vazifa 2 va 3-vazifalar kabi eng yaqin bo'sh qismda taqsimlanadi. 4-vazifa eng mos bo'lmasa ham, bo'sh bo'lgan yagona qismga taqsimlangan. Ushbu sxemada barcha to'rtta vazifa kutishsiz qayta ishlanadi. Yodda tuting, xotira ro'yxati xotira hajmiga qarab tartiblangan. Ushbu sxema yordamida xotiradan yanada samarali foydalaniladi, ammo uni amalga oshirish sekinroq hisoblanadi.



3.10- rasm. Eng mos keladigan sxemadan foydalanish

Eng mos va birinchi mos keladigan algoritmlar juda farq qiladi. Birinchi usul qanday amalga oshiriladi:

First-Fit Algoritmi

1. Hisoblagichni 1 ga o'rnatish
2. Bajarishda, hisoblagich  $\leq$  xotiradagi bloklar soni  
 Agar vazifa hajmi  $>$  xotira hajmi (hisoblagich) bo'lsa  
 Unda, hisoblagich = hisoblagich + 1  
 Xotiraga (hisoblagich) vazifani yuklash  
 Bo'sh/band xotira ro'yhatlarini sozlash  
 4-bosqichga o'tish  
 Tugatish
3. Vazifani kutish navbatiga qo'yish
4. Keyingi vazifaga o'tish.

3.2-jadvalda 200 bo'sh maydonni bloklash so'rovi faqat xotira menejeriga berilgan (maydonlar so'zlar, baytlar yoki tizim boshqaradigan boshqa birlik bo'lishi mumkin). Birinchi mos keladigan algoritmdan foydalanib va ro'yxatning yuqorisidan boshlab, xotira menejeri 6785 manzilida joylashgan vazifani bajarishi uchun yetarlicha katta bo'lgan birinchi xotira blokini topadi. Keyin, vazifa 6785 maydondan boshlanib, keyingi 200 bo'sh maydonni egallaydi. Keyingi qadam, bo'sh xotira bloki hozirda 6985 (6785 emas, balki

avvalgi kabi) maydonda joylashganligini va unda faqat 400 ta bo'sh maydon mavjudligini (oldingidek 600 emas) belgilash uchun bo'sh ro'yxatni o'rnatishdir.

### 3.2- jadval

So'rovdan oldin		So'rovdan keyin	
Boshlang'ich manzil	Xotira bloki hajmi	Boshlang'ich manzil	Xotira bloki hajmi
4075	105	4075	105
5225	5	5225	5
6785	600	*6985	400
7560	20	7560	20
7600	205	7600	205
10250	4050	10250	4050
15125	230	15125	230
24500	1000	24500	1000

Eng yaxshi moslashtirish algoritmi biroz murakkabroq, chunki maqsad vazifa uchun mos keladigan eng kichik xotira blokini topishdir:

#### Best-Fit Algoritmi

1. xotira blokini (0) = 99999 ishga tushirish
2. boshlang'ich xotira yo'qotilishi = xotira bloki (0) – vazifa hajmini hisoblash

3. Indeksni = 0 ga sozlash

4. Hisoblagichni 1 ga o'rnatish

5. Hisoblagich < = xotiradagi bloklar soni bo'lsa bajarish

Agar vazifa hajmi > xotira hajmi (hisoblagich) bo'lsa

Unda, hisoblagich = hisoblagich + 1

Keyin

Xotiradagi yo'qotish = xotira hajmi (hisoblagich) – vazifa hajmi

Agar boshlang'ich xotira yo'qotishlari > xotira yo'qotishlari

Unda, indeks = hisoblagich

boshlang'ich xotira yo'qotishlari = xotira yo'qotishlari

hisoblagich = hisoblagich + 1

Bajarishni tugatish

6. Agar indeks = 0 bo'lsa

Unda, vazifani kutish navbatiga qo'yish

Keyin

xotira maydoniga (pastki indeks) vazifani yuklash  
bo'sh/band xotira ro'yxatlarini o'rnatish  
7. Keyingi vazifaga o'tish.

Eng yaxshi moslangan algoritm bilan bog'liq muammolardan biri shundaki, tanlovni amalga oshirishdan oldin, butun jadvalni qidirishi kerak, chunki xotira bloklari fizik xotirada joylashgan joyiga qarab ketma-ket saqlanadi (3.10-rasmda ko'rsatilganidek, xotira qismlari hajmiga qarab emas). Tizim xotira qismining o'lchamlari oshib boradigan tartibda ro'yxatni doimiy ravishda tiklash algoritmini ishlab chiqishi mumkin, ammo bu qo'shimcha xarajatlarni keltirib chiqaradi va uzoq muddatda qayta ishlash vaqtidan unumli foydalanishi mumkin emas. Eng yaxshi mos keladigan algoritm faqat bo'sh xotira bloklari ro'yxati bilan tasvirlangan.

3.3- jadval

So'rovdan oldin		So'rovdan keyin	
Boshlang'ich manzil	Xotira bloki hajmi	Boshlang'ich manzil	Xotira bloki hajmi
4075	105	4075	105
5225	5	5225	5
6785	600	6785	600
7560	20	7560	20
7600	205	*7800	5
10250	4050	10250	4050
15125	230	15125	230
24500	1000	24500	1000

Ushbu jadvalda eng yaxshi mos keladigan algoritmdan foydalangan holda, so'rovdan oldingi va keyingi har bir xotira blokining holatini ko'rsatadi. Jadvalda 200 ta bo'sh joyni bloklash to'g'risidagi so'rov endigina xotira menejeriga jo'natildi. Eng yaxshi mos keladigan algoritmdan foydalanib, ro'yxatning yuqorisidan boshlab, xotira menejeri barcha ro'yxatni qidiradi va 7600 manzilidan boshlab xotira blokini topadi, bu vazifani bajarish uchun yetarlicha katta bo'lgan eng kichik blokdir. Xotira menejeri eng yaxshi mos keladigan algoritmdan foydalanib, ro'yxatning yuqori qismidan boshlab, butun ro'yxatni qidiradi va 7600 manzilidan boshlanadigan xotira blokini topadi, bu vazifani bajarish uchun yetarlicha katta bo'lgan eng kichik blokdir. Ushbu blokni tanlash bo'sh joyni bexuda

sarflashni kamaytiradi (faqatgina 5K bo'sh joy yo'qotiladi, bu to'rtta alternativ blokga qaraganda kamroq). Keyin vazifa 7600 manzildan boshlanib, keyingi 200 ta manzilni egallaydi.

Eng yomon moslash usuli: ro'yxatdan eng mos keladigan eng katta maydonni tanlashdir. Nima uchun eng katta? Bo'linishning oldini olish uchun (bo'linish muammosi ushbu ma'ruzada batafsil ko'rib chiqiladi). Birinchi va ikkinchi strategiyalarni qo'llash quyidagi nuqtai nazardan yaxshiroq: bajarilish tezligi va ishlatilgan xotiraning minimal hajmi bo'yicha. Biroq, ulardan foydalanish bo'linishni keltirib chiqarishi mumkin.

### **3.4. Xotirani tartiblash va kengaytirish, virtual xotira**

#### ***Bo'linish (fragmentatsiya)***

Bo'linish (fragmentatsiya) - bu xotiraning kichik o'lchamdagi, qo'shni bo'lmagan maydonlarga bo'linishidir. Bo'linishni (fragmentatsiyani) ikkita turi mavjud: tashqi bo'linish va ichki bo'linish.

#### Tashqi bo'linish

Xotiraning umumiy maydoni so'rovni qondirish yoki undagi jarayonni o'tkazish uchun yetarli, ammo u o'zaro bog'liq bo'lmaydi, shuning uchun undan foydalanib bo'lmaydi.

#### Ichki bo'linish

Jarayonga belgilangan xotira bloki kattaroqdir. Xotiraning bir qismi ishlatilmay qoladi, chunki undan boshqa jarayon foydalana olmaydi. Quyidagi diagrammada bo'linish qanday qilib xotira yo'qolishiga olib kelishi va siqish texnikasidan foydalanib, bo'lingan xotiradan ko'proq bo'sh xotirani yaratish uchun foydalanish mumkinligi ko'rsatilgan.

Ichki bo'linish xotirani taqsimlashning aniq algoritmini qo'llagan tizim natijasida yuzaga kelishi mumkin, so'rovga javoban xotira talab qilinganiga nisbatan biroz kattaroq ajratiladi.



3.11- rasm. Siqishdan oldingi xotiradagi bo‘linish



3.12- rasm. Siqishdan keyingi xotira

Masalan, kattaligi ikkita sahifaga qadar bo‘lgan. Xotirani sahifali tashkil etish ushbu ma’ruzada keyinchalik batafsil muhokama qilinadi.

### ***O‘zgaruvchan qismli sxemalar***

Bundan oldingi ko‘rib chiqilgan xotirani taqsimlash sxemalarida (belgilangan qismlar va dinamik qismlar) mavjud muammolarni (bo‘linishlarni oldini olish uchun) yechimi o‘zgaruvchan dinamik qismlarni ishlab chiqish edi.

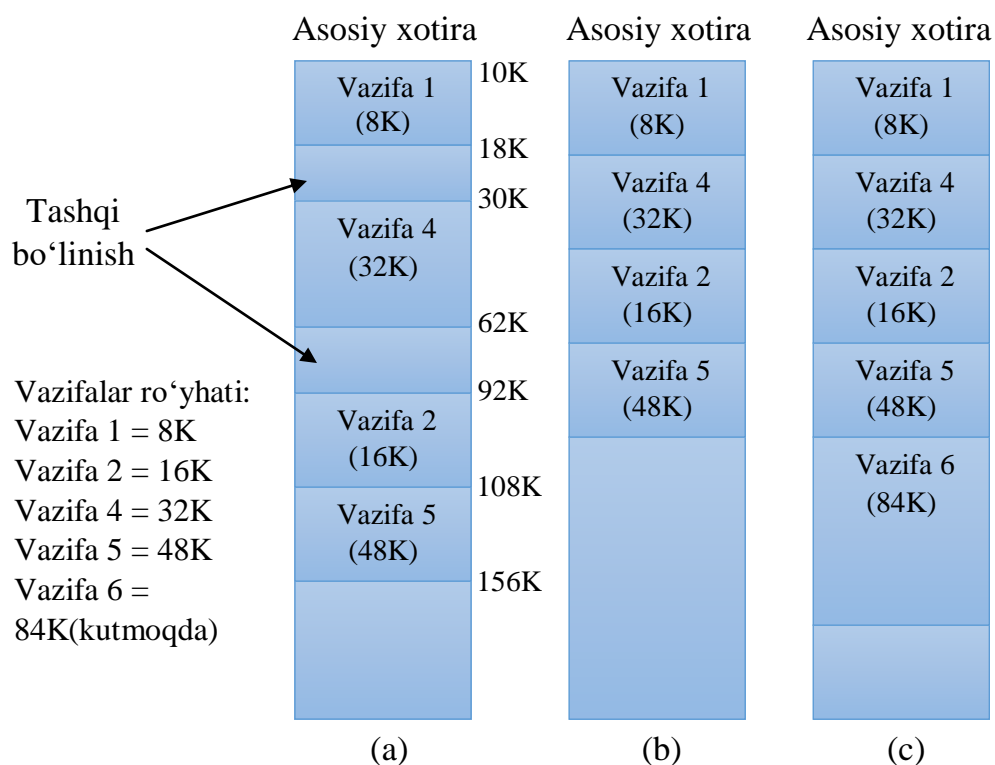
Qoida bo‘yicha svoping (almashtirish) tizimi belgilangan qismlarga asoslanishi mumkin. Ammo dinamik taqsimlash yoki o‘zgaruvchi qismli sxemalar samarali hisoblanadi. Chunki ular hamma jarayonlar to‘liq ravishda xotirada joylashganda, yani svoping bo‘lmagan hollarda qo‘llaniladi.

Bu holda, boshida xotira butunlay bo‘sh va oldindan qismlarga bo‘lingan bo‘lmaydi. Yangi ishga tushirilayotgan vazifaga qat’iy ravishda kerakli xotiraning o‘zi ajratiladi (undan ko‘p emas). Jarayon chiqarilgandan so‘ng, xotira vaqtincha bo‘shatiladi. Bir qancha vaqt o‘tgandan so‘ng xotira turli o‘lchamdagi o‘zgaruvchan sonli qismlardan iborat bo‘lib qoladi. Yonma-yon bo‘lgan bo‘sh joylar birlashtirilishi mumkin.

Ushbu xotirani taqsimlash sxemasi yordamida xotira menejeri barcha bo‘sh bloklarni bir joyga to‘plash va bir xotira blokini yetarlicha katta qilish (kirishni kutayotgan ba’zi yoki barcha



vazifalarni joylashtirish uchun) va ularni siqish uchun dasturlarni o‘zgartiradi. Xotirani siqish operatsion tizim tomonidan xotira maydonining bo‘lingan (fragmentatsiya) qismlarini tiklash (defragmentatsiya) uchun amalga oshiriladi. Lekin bu oson ish emas. Birinchidan, xotiradagi har bir dastur bir-biriga qo‘shni bo‘lishi uchun ko‘chirilishi kerak, so‘ngra har bir manzil va har bir dasturdagi manzilga bog‘lanish, dasturning xotirada yangi joylashishini hisobga olgan holda sozlanishi kerak.



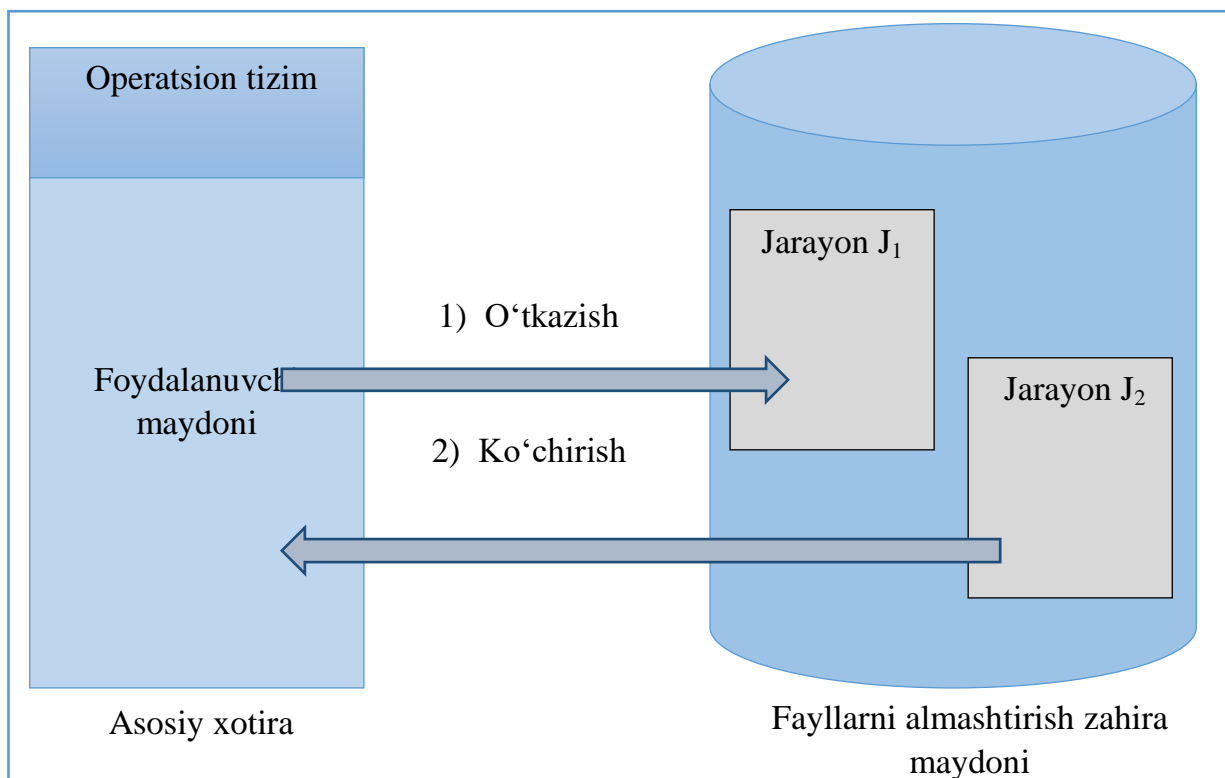
3.13- rasm. O‘zgaruvchan qisimli sxemaga misol

Dastlabki 10 Kb xotirani egallaydigan, operatsion tizimni xotirani siqishdan oldingi va keyingi uchta rasmi. 6-vazifa uchun 84 Kb kerak bo‘lsa, (a) tashqi bo‘linishdagi dastlabki xotira tuzilishi 96 Kbni tashkil qiladi. Siqishdan so‘ng (b) darhol tashqi bo‘linish bartaraf etildi, bu esa 6-vazifa uchun joy yaratishga imkon berdi, yuklangandan keyin (c) da ko‘rsatilgan.

### ***Almashtirish (swapping)***

Tizimni qayta ishlash jarayonida foydalanuvchi jarayoni turli xil holatlarda bo‘lishi mumkin. Xususan, vaqtni taqsimlash rejimida amalga oshirilsa, masalan, foydalanuvchi terminal orqasida keyingi buyruqni ko‘rib chiqsa yoki o‘z dasturining manba kodini o‘zgartirsa,

jarayon bir muncha vaqt harakatsiz bo‘lishi mumkin. Bunday hollarda, jarayon operatsion tizim tomonidan diskka ko‘chirilishi mumkin, chunki u egallagan xotira ayni paytda boshqa faol jarayon uchun zarurdir. Almashtirish (swapping) - bu operatsion tizimning faol bo‘lmagan jarayon tasvirini diskka o‘tkazish (yozish) yoki faol jarayonni asosiy xotiraga ko‘chirish (o‘qish) uchun harakatlardir. Bunday harakatlarni bajarish zarurati asosiy xotiraning yetishmasligidan kelib chiqadi. Almashtirish (swapping) jarayoni quyidagi 3.14- rasmda keltirilgan.



3.14- rasm. Almashtirish jarayoni

Zahira maydoni (backing store) - bu operatsion tizim tomonidan ko‘chiriladigan jarayonlar tasvirilarini (fayllarni o‘tkazish) saqlash uchun foydalaniladigan disk xotirasi maydoni. Zahira maydoni iloji boricha samarali tashkil etiladi: xotiradagi barcha jarayon tasvirilariga to‘g‘ridan-to‘g‘ri kirishni ta‘minlaydi (masalan, jarayon raqami jadvali orqali). O‘tkazish va ko‘chirish algoritmining eng mashhur versiyasi bu: ustuvorliklarga asoslanib o‘tkazish va ko‘chirish; yuqori ustuvorlikga ega jarayonlar bajariladi, past ustuvorlikga ega jarayonlar diskka o‘tkaziladi.

O‘tkazishga sarflanadigan eng katta vaqt - bu ma‘lumotlarni uzatish harajatidir: to‘liq jarayon tasviri katta hajmdagi xotirani

egallashi mumkin. Umumiy o'tkazish vaqti o'tkaziladigan ma'lumotlarning hajmiga mutanosibdir. Keng tarqalgan operatsion tizimlarda - UNIX, Linux, Windows va boshqalar – o'tkazish va ko'chirish (swapping) bo'yicha har xil strategiyalar amalga oshiriladi. Almashtirish jarayoni odatda ishlashga ta'sir qilsada, parallel ravishda bir nechta katta jarayonlarni boshqarishga yordam beradi, shuning uchun almashtirish ham xotirani siqish usuli sifatida tanilgan. O'zgartirish jarayoni uchun talab qilinadigan umumiy vaqt butun jarayonni ikkinchi darajali diskka ko'chirish uchun zarur bo'lgan vaqtni, so'ngra jarayonni asosiy xotirani tiklash uchun zarur bo'lgan vaqtni o'z ichiga oladi. Aytaylik, foydalanuvchi jarayoni 2048 KB hajmda va almashish amalga oshiriladigan standart qattiq diskda ma'lumot uzatish tezligi sekundiga 1 MB ni tashkil qiladi. Xotiraga yoki 1000K hajmdagi jarayonning haqiqiy uzatilishi 2000 millisekundga to'g'ri keladi. Endi, kirish va chiqish vaqtini hisobga olgan holda, jarayon asosiy xotirani tiklash uchun raqobatlashganda 4000 millisekund va boshqa qo'shimcha xarajatlarni talab qiladi.

### ***Sahifali xotira***

Yuqorida tavsiflangan sxemalarda xotiradan samarali foydalanilmaydi, shuning uchun ham xotirani taqsimlashning zamonaviy sxemalarida jarayonni operativ xotirada uzluksiz blok sifatida joylashtirish ko'zda tutilmagan. Zamonaviy operatsion tizimlarda, vazifani asosiy xotiraga yuklashdan oldin u sahifalar (Pages) deb nomlangan qismlarga bo'linadi.

Sahifali xotira - bu xotirani boshqarish usuli bo'lib, unda jarayonning manzil maydoni bir xil hajmdagi sahifalar deb ataladigan bloklarga bo'linadi (hajmi 2 daraja, 512 dan 8192 baytgacha). Jarayon hajmi sahifalar soni bilan o'lchanadi. Xuddi shu tarzda, asosiy xotira doimiy o'lchamdagi (fizik) xotiraning kichik bloklariga bo'linadi, ular kadrlar deb nomlanadi va asosiy xotirani optimallashtirish va tashqi qismlarga bo'lmaslik uchun kadr hajmi sahifa o'lchamiga teng bo'lib qoladi.

Xotirani sahifali tashkil etishda eng oddiy va eng keng tarqalgan usul (yoki paging), xotiraning ham mantiqiy manzilli maydoni, ham fizik maydonini bir xil o'lchamdagi sahifa va bloklar to'plami ko'rinishida tashkil etishdir. Bunda mantiqiy sahifalar (page) yuzaga keladi va ularga mos fizik xotira birliklari - fizik sahifalar yoki sahifa

kadrlari deb ataladi (page frames). Sahifalar (va sahifa kadrlari) odatda 2 sonini darajasidan iborat bo'lgan belgilangan qat'iy uzunlikka egadir va ular bir-biri bilan kesishmaydi. Har bir kadr ma'lumotlarning bir sahifasini o'z ichiga oladi xotirani bunday tashkil etishda tashqi bo'linish (fragmentatsiya) bo'lmaydi va ichki bo'linishdan kelib chiqadigan yo'qotish faqat oxirgi sahifalardan kelib chiqadigan yo'qotish bilan chegaralanadi.

Sahifali tizimda mantiqiy manzil - tartiblangan juftlikdan (p, d) iborat, bu yerda p virtual xotira sahifasi tartib raqami, d esa bu sahifa doirasidagi element o'rnini bildiradi. Manzil manzilini sahifalarga bo'lish xisoblash tizimi tomonidan, dasturchi aralashmagan holda amalga oshiriladi. Shuning uchun ham, manzil, operatsion tizim nuqtai nazaridagina ikki o'lchamlidir, dasturchi nuqtai nazaridan esa jarayon manzili chiziqli hisoblanadi.

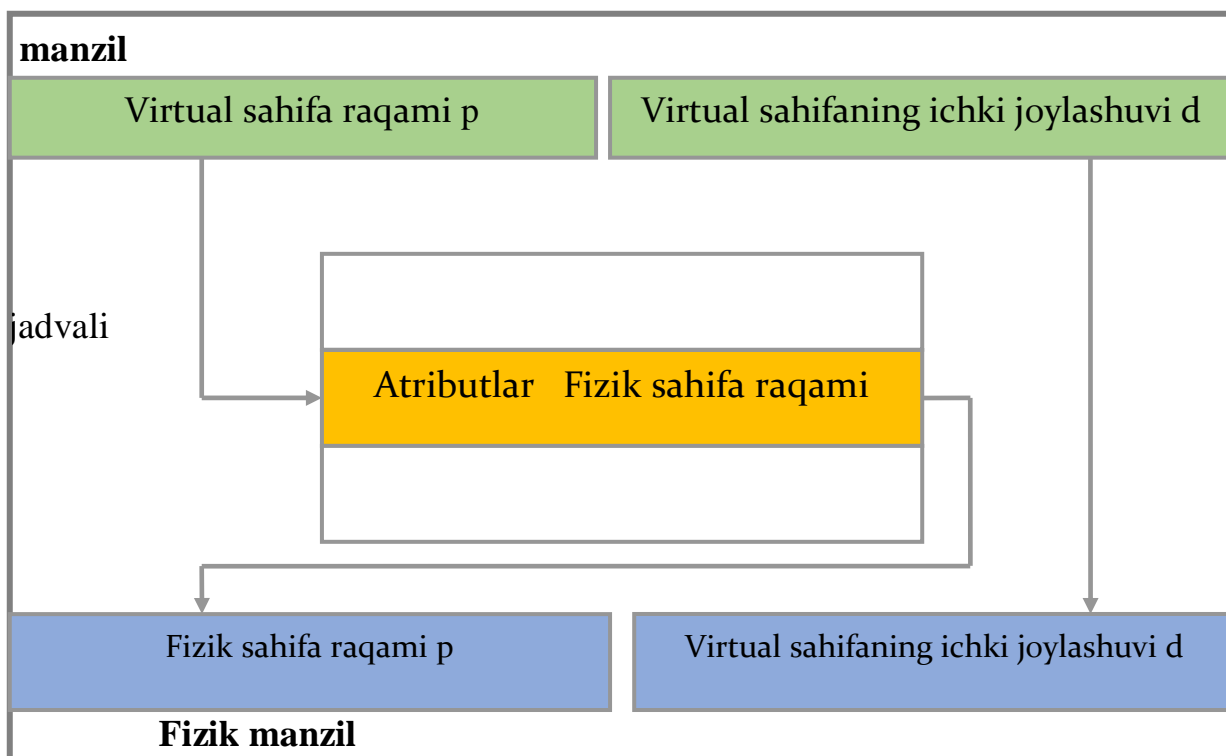
Yuqorida keltirilgan sxema, jarayonlarni to'liq joylashtirish uchun kadrlarning uzluksiz sohasi yetarli bo'lmagan hollarda ham, jarayonni yuklash imkonini beradi. Ammo, bu sxemada manzilni translyatsiyalash uchun bitta asos registri yetarli emas. Mantiqiy manzillarni fizik manzillarda aks ettirish, mantiqiy sahifalarni fizik sahifalarda aks ettirishga keltiriladi va operativ xotirada saqlanadigan sahifalar jadvalidan iborat bo'ladi. Ba'zida, sahifalar jadvali – jadval ko'rinishidagi chiziqli bo'lakli funksiya ham deyiladi. Mantiqiy manzilning bog'lanishi 3.15-rasmda ko'rsatilgan.

Bunda bajariladigan jarayon  $v = (p, d)$  mantiqiy manzilga murojaat qiladi, va aks ettirish mexanizmi sahifa tartib raqami  $r$  ni sahifalar jadvalidan qidiradi, bu sahifa  $r^*$  sahifa kadrida joylashganligini aniqlaydi va real manzil  $r^*$  ni  $d$  ga aylantiradi.

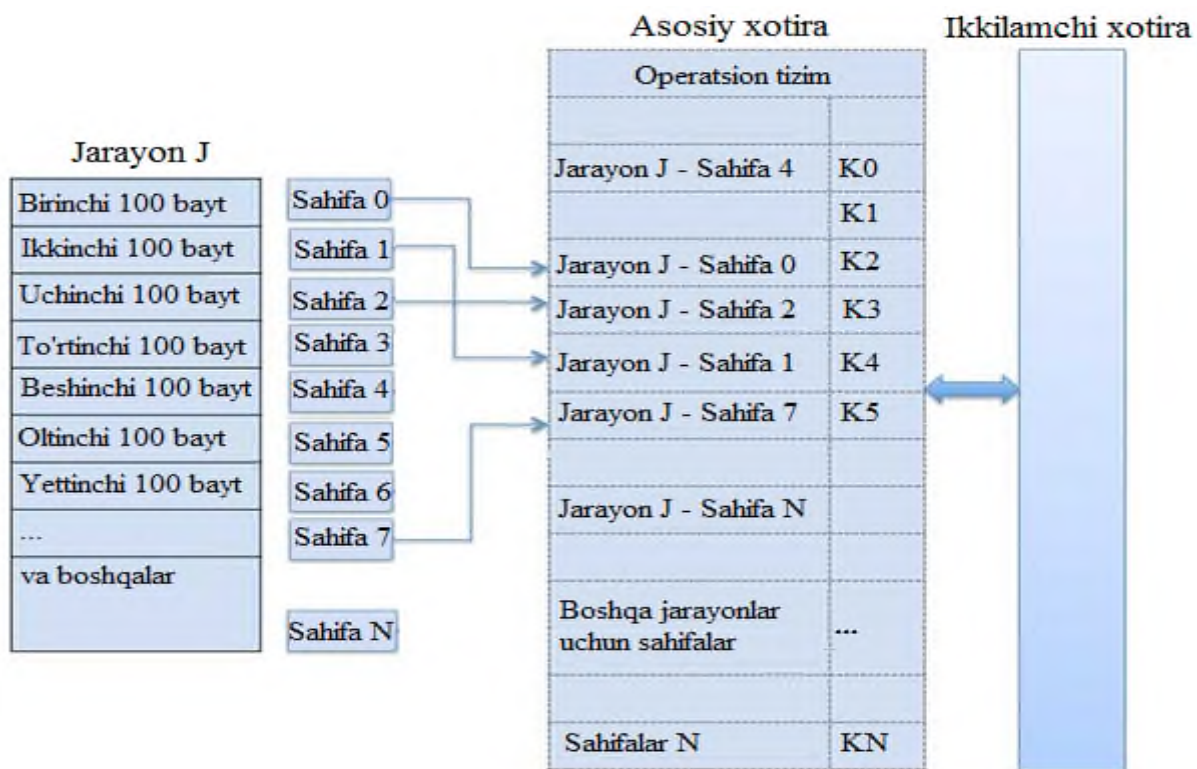
Xotirani sahifali tashkil etish (Paging) deyarli barcha operatsion tizimlarda ishlatiladigan eng keng tarqalgan xotirani boshqarish algoritmidir.

Sahifalar jadvali (page table) protsessorning maxsus registrida manzillashtiriladi va kadrlar raqamini mantiqiy manzil bo'yicha aniqlashga yordam beradi.

Bu asosiy masaladan tashqari sahifalar jadvali qatorida yozilgan atributlar yordamidan aniq sahifaga murojaat nazorati va uni himoyasini tashkil etish mumkin.



3.15- rasm. Xotirani sahifali tashkil etishda mantiqiy va fizik manzillarning bog‘lanishi



3.16- rasm. Sahifani tashkil qilishga misol

### ***Manzillar translatsiyasi***

Sahifa manzili mantiqiy manzil deb ataladi va sahifa raqami va sahifani almashtirish bilan ifodalanadi.

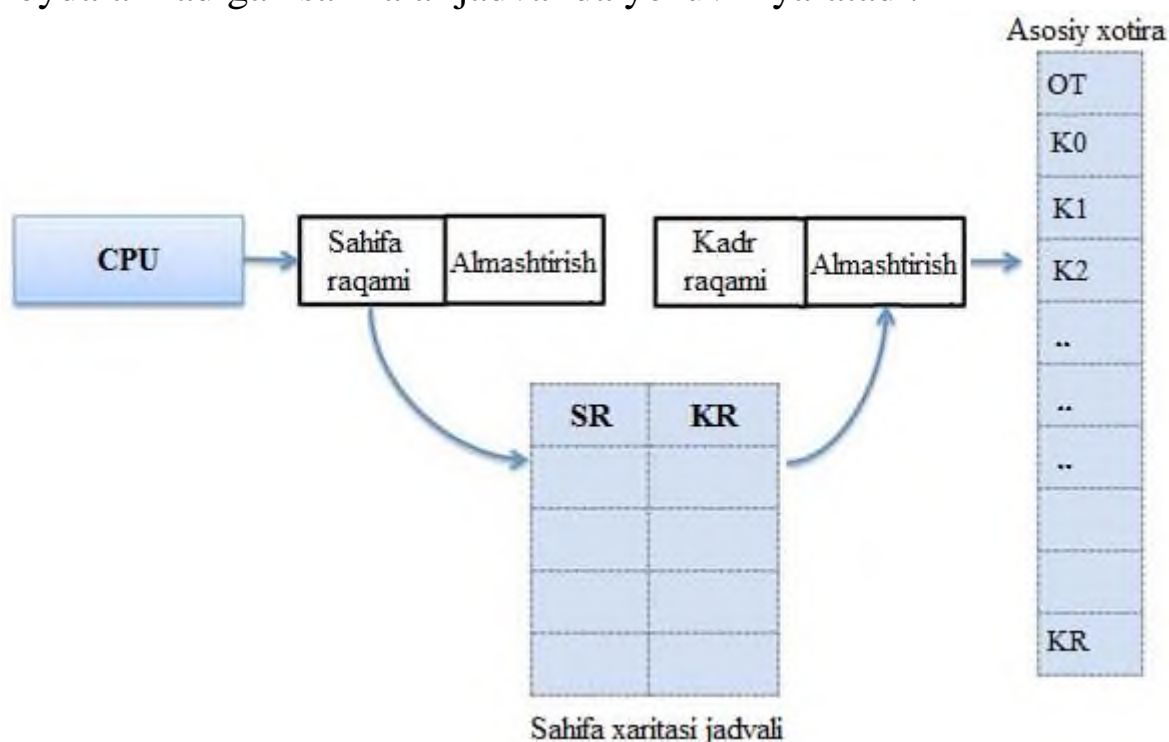
Mantiqiy manzil = sahifa raqami + sahifani almashtirish

Kadr manzili fizik manzil deb ataladi va kadr raqami va sahifani almashtirish bilan ifodalanadi.

Fizik manzil = kadr raqami + sahifani almashtirish

Sahifalar xaritasi jadvali deb nomlangan ma'lumotlar tuzilishi protsessor sahifasi va fizik xotiradagi kadr o'rtasidagi munosabatni kuzatish uchun ishlatiladi.

Tizim har qanday sahifaga kadr ajratganda, u mantiqiy manzilni fizik manzilga aylantiradi va dasturni bajarish paytida foydalaniladigan sahifalar jadvalida yozuvni yaratadi.



3.17- rasm. Sahifa va kadrni almashtirish sxemasi

Jarayonni bajarish kerak bo'lganda, uning tegishli sahifalari har qanday mavjud bo'lgan xotira kadrlariga yuklanadi. Aytaylik, sizda 8Kb dastur mavjud, ammo sizning xotirangiz belgilangan vaqtda atigi 5Kb sig'ira oladi, shunda sahifa kontseptsiyasi paydo bo'ladi. Kompyuter operativ xotirasida joy qolmaganida, operatsion tizim bo'sh yoki keraksiz sahifalarni ikkinchi darajali xotiraga ko'chiradi,

boshqa operatsiyalar uchun operativ xotirani bo‘shatadi va dastur tomonidan kerak bo‘lganda ularni qaytarib beradi. Ushbu jarayon dasturning to‘liq bajarilishi mobaynida davom etadi, unda operatsion tizim bo‘sh sahifalarni asosiy xotiradan olib tashlaydi va ularni ikkinchi darajali xotiraga yozadi va dastur tomonidan talab qilinganda ularni qaytarib beradi.

### ***Sahifali xotira afzalliklari va kamchiliklari***

Almashtirishning afzalliklari va kamchiliklari ro‘yxati quyidagicha:

❖ Sahifali xotira tashqi bo‘linishni kamaytiradi, ammo baribir ichki bo‘linishdan aziyat chekmoqda.

❖ Sahifali xotira oson amalga oshiriladi va xotirani boshqarishning samarali usuli hisoblanadi.

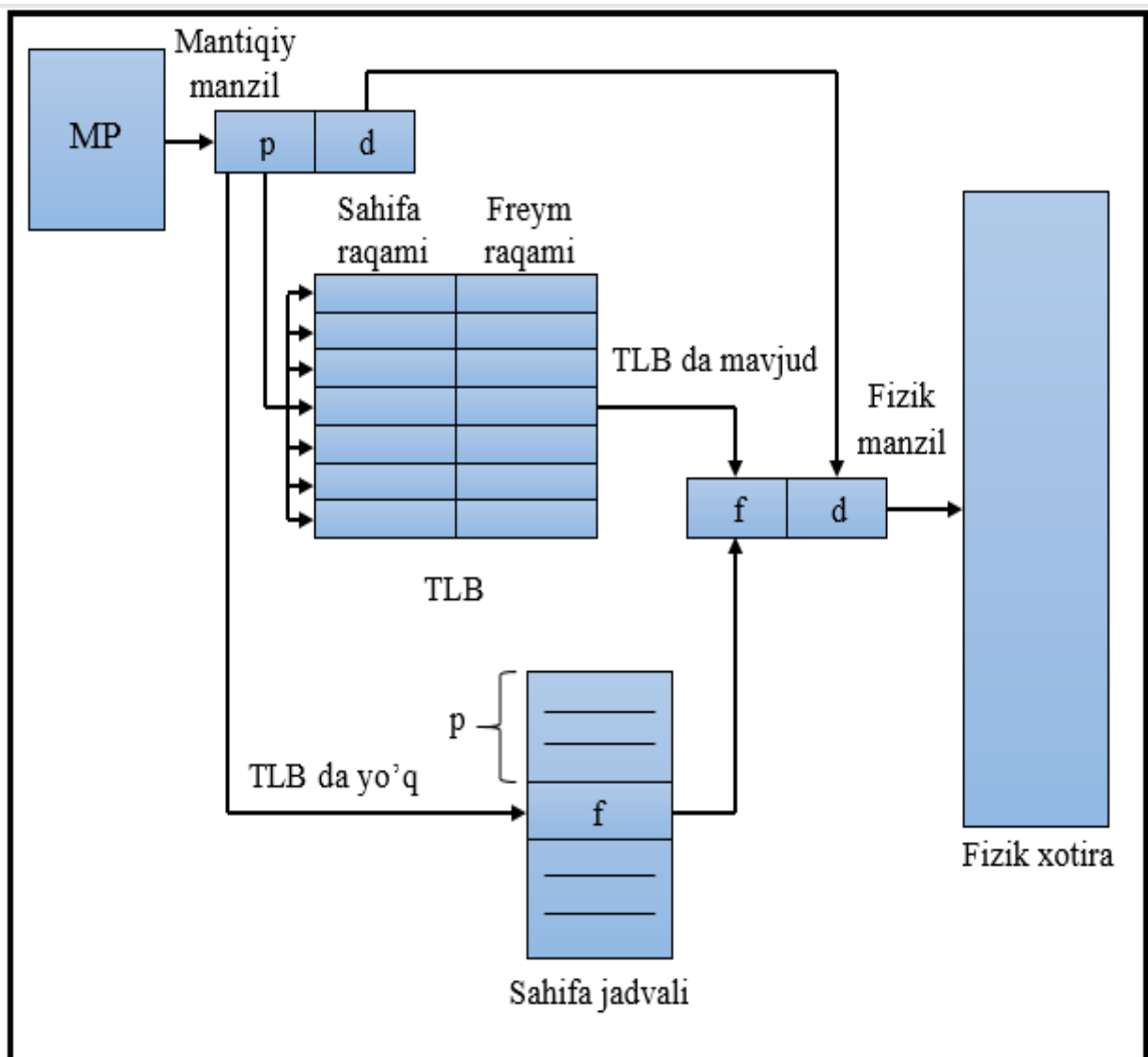
❖ Bir xil o‘lchamdagi sahifalar va kadrlar tufayli, almashish juda oson.

❖ Sahifalar jadvali qo‘shimcha xotira maydonini talab qiladi, shuning uchun u operativ xotirasi kichik bo‘lgan tizim uchun mos kelmasligi mumkin.

### ***Sahifalar jadvallarini amalga oshirish***

Assotsiativ (kesh) xotiradan foydalanish. Sahifalar jadvali – bu fizik xotiraning uzluksiz qismi. Tizimda sahifalar jadvaliga ko‘rsatuvchi va uning uzunligini saqlovchi sahifalar jadvalining bazaviy registri (page table base register – PTBR) mavjud. Shunday qilib sahifalar ko‘rinishda tashkil etilganida xotiradan foydalanishning barcha hollarida xotiraga amalda bir marta emas, balki ikki marta murojaat etish talab etiladi – birinchisi sahifalar jadvaliga, ikkinchisi bevosita ma’lumotlar yoki buyruqga. Bu holatda xotiraning boshqarishni soddaroq usullariga nisbatan ma’lum kamchiligi va sahifali tashkil etishni samarasizligidir. Yorliqli arxitekturali tizimlarda, masalan “Elbrus”, sahifalar jadvali registrida (foydalanuvchining sahifalar jadvali registri – FSJR) sahifalar jadvalining deskriptori mavjud va unda uning manzilidan tashqari uzunligi ham bor. Ikki murojaat muammosi sahifalarning assotsiativ xotirasi (cache), yoki boshqacha nomlanganda translyatsiya manzillari buferi (translation lookaside buffer – TLB)ni kiritish yo‘li bilan hal etiladi. Assotsiativ xotira mazmunan – bu (sahifa raqami, kadr raqami)

turdagi juftlarning assotsiativ ro‘yxati. Uning ishlash tezligi asosiy xotira va registrlarga nisbatan anchagina yuqoriroq. Assotsiativ xotiradan foydalanish bilan manzillar translyatsiyasi sxemasi o‘zgaradi: agar mantiqiy manzildagi sahifaning raqami assotsiativ xotirada topilsa, u holda uning elementidan kadrning tegishli raqami chiqariladi. Agarda sahifaning raqami assotsiativ xotirada bo‘lmasa, u oddiy tarzda sahifalar jadvalidan olinadi, lekin assotsiativ xotiraga kiritiladi. Shunday qilib assotsiativ xotirada eng ko‘p foydalaniladigan sahifalar to‘g‘risida axborot yig‘ilib boriladi. TLB dan foydalanilgan holda manzillarni translyatsiya qilish modifikatsiyalangan sxemasi 3.18- rasmda tasvirlangan.



3.18- rasm. Assotsiativ xotiradan foydalalanib manzillarni translyatsiya qilish sxemasi



TLB dan foydalanilgan holda xotiraga murojaat etishning o'rtacha vaqtini baholash.

Taxmin qilaylik – assotsiativ qidiruv o'rtacha vaqtning  $\alpha$  birligini talab etadi. Xotira siklini 1 deb qabul qilamiz. Yana bir ko'rsatkichni kiritamiz – 0 dan 1 gacha (hit ratio), u sahifaning raqami necha marta (o'rtacha) TLB da topiladi – sahifa raqamining assotsiativ xotirada bo'lishining empirik ehtimolligi. Murojaat vaqtining matematik kutilishini hisoblaymiz - Effective Access Time (EAT). Sahifa raqami TLB da topilmaslik ehtimolligi 1 ga teng.

### ***Xotiraning himoyalanihi***

Sahifali tashkil etish yordamida manzillashda balki mantiqiy manzil noto'g'ri shakllangan va uning sahifa raqami protsessorning mantiqiy xotirasi chegarasidan chiqib ketadi. Noto'g'ri manzillashdan himoyalanih sahifalar jadvalining har bir elementida qo'shimcha bitni valid-invalid saqlash va tekshirish yo'li bilan amalga oshirilishi mumkin. Valid qiymati shu raqamli sahifa protsessorning mantiqiy xotirasiga tegishliligini ko'rsatadi, invalid qiymati – nimadir to'g'ri emasligini bildiradi. Valid-invalid biti yordamida protsessor xotirasini himoyalashni tashkil etish 3.19-rasmda tasvirlangan.

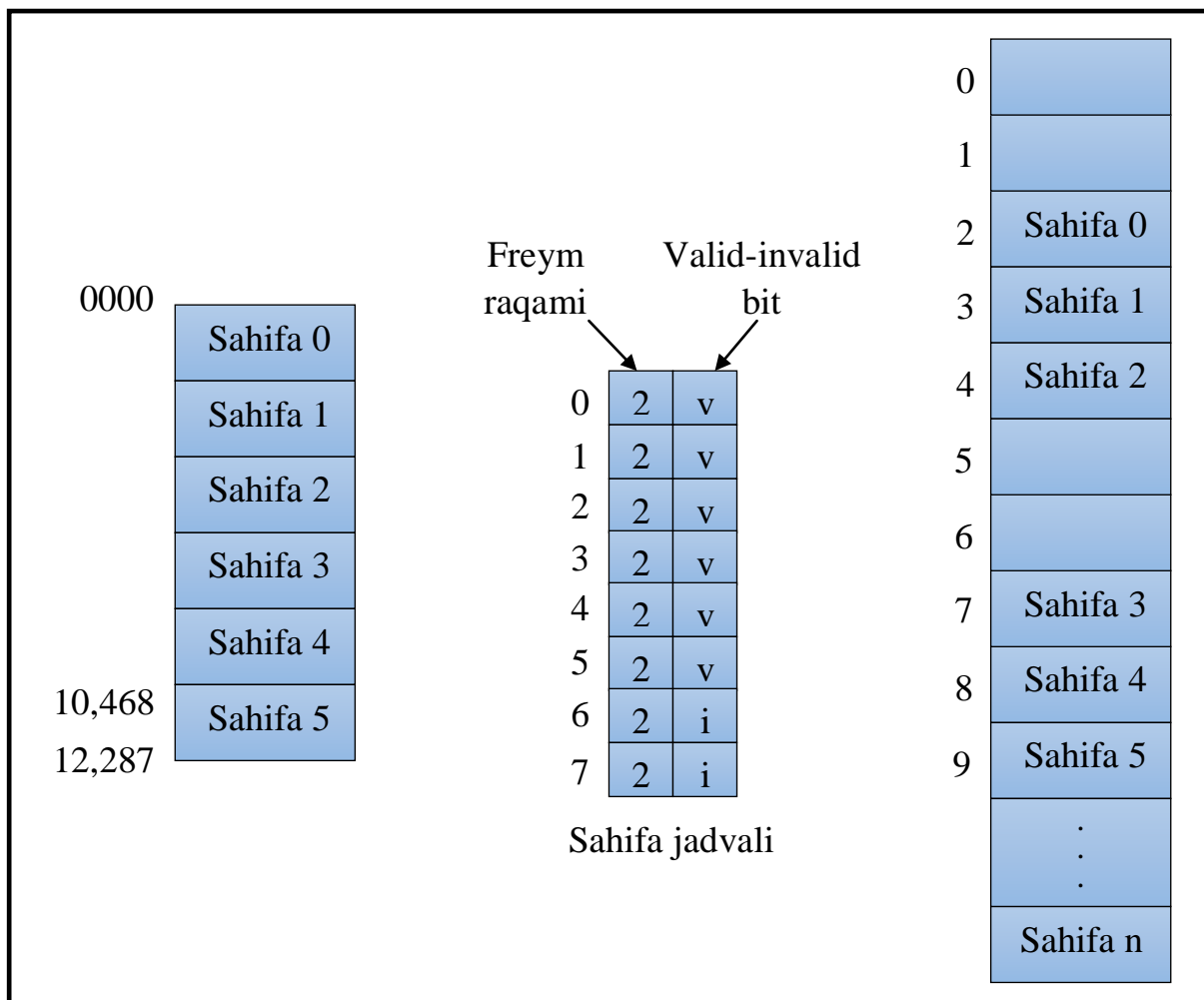
Misolda jarayon 0 dan 5 gacha bo'lgan raqamlar bilan belgilangan oltita mantiqiy sahifalarga ega. "6" va "7" elementlar jarayonning mantiqiy sahifalariga mos emas, shuning uchun ularda valid-invalid bitlar invalid qiymatida o'rnatilgan. Shuning uchun mantiqiy manzil bo'yicha "6" yoki "7" raqamli sahifalarga murojaat qilishga bo'lgan urunishida noto'g'ri manzillash bo'yicha to'xtalish ro'y beradi.

### ***Qo'shni xotirani taqsimlash***

Xotirani taqsimlashning eng oddiy va eng keng tarqalgan algoritmi qo'shni xotirani taqsimlashdir - bitta qo'shni xotira maydonida foydalanuvchi jarayonlari uchun xotirani taqsimlash.

Asosiy xotira bir-biriga qarab "o'sadigan" ikkita qo'shni qismlarga bo'linadi: OTning doimiy qismi va uzilish vektori - pastki manzillarda, foydalanuvchi jarayonlari – manzil bo'yicha. Har bir jarayon uchun ko'chirish registri unga ajratilgan xotira maydonining boshlanishini, chegara registri mantiqiy manzillar oralig'ining

uzunligini o'z ichiga oladi. Har bir mantiqiy manzil chegara registri tarkibidan kichik bo'lishi kerak.



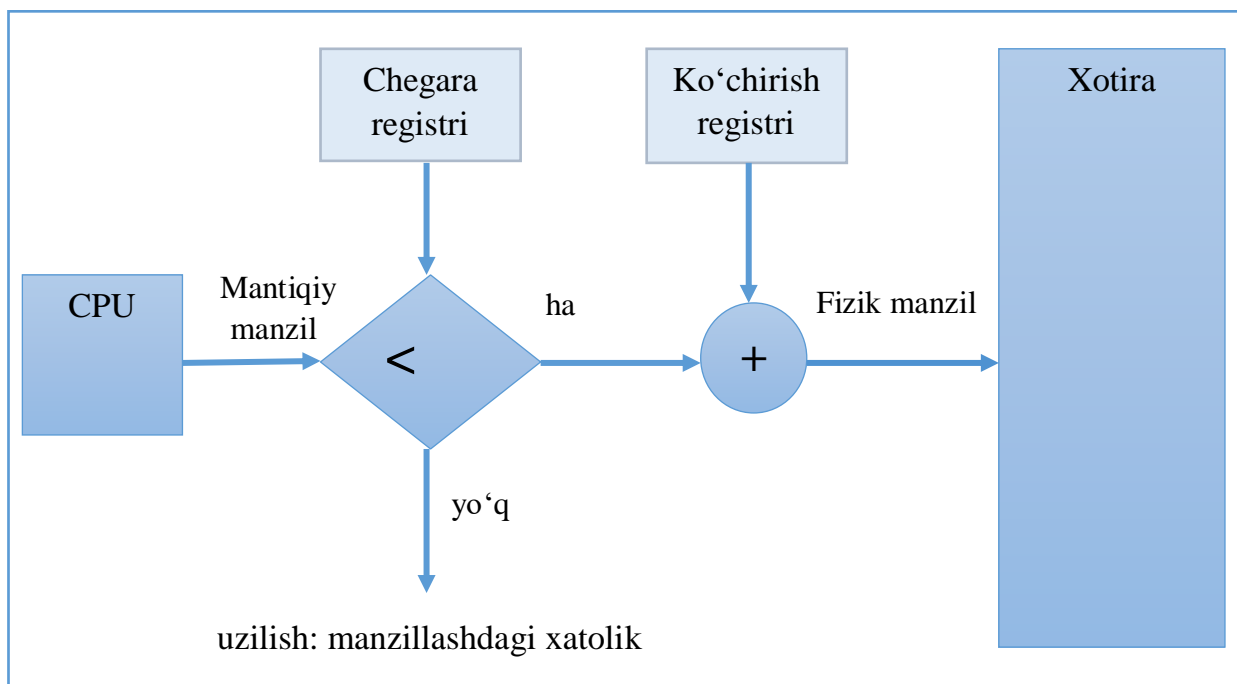
3.19- rasm. Valid-invalid biti yordamida protsessor xotirasini himoyalash

Fizik manzil qurilma tomonidan mantiqiy manzil va ko'chirish registrining qiymati yig'indisi sifatida hisoblanadi. Ko'chirish va chegara registerlari apparat bilan qo'llab-quvvatlash manzillar sxemasi quyidagi 3.20- rasmda keltirilgan.

### ***Segmentli xotira (Segmentation)***

Segmentlash - bu xotirani boshqarish usuli bo'lib, unda har bir vazifa turli o'lchamdagi bir nechta segmentlarga bo'linadi, har bir modul uchun tegishli funksiyalarni bajaradigan qismlar mavjud. Segmentlar sahifalardan farqli ravishda o'zgaruvchan o'lchamga ega bo'lishadi. Har bir segment aslida har xil mantiqiy dastur manzili

maydoni bilan farqlanadi. Jarayonni bajarish kerak bo‘lganda, uning har bir segmenti mavjud xotiraning qo‘shni blokiga yuklangan bo‘lsa ham, uning tegishli segmentatsiyasi o‘zaro bog‘liq bo‘lmagan xotiraga yuklanadi. Segmentli xotirani boshqarish sahifali xotira bilan juda o‘xshash, ammo bu yerda segmentlar uzunligi bo‘yicha o‘zgaruvchan bo‘lib, almashtirish sahifalari esa belgilangan o‘lchamga ega. Dastur segmenti dasturning asosiy funksiyasini, yordamchi funksiyalarni, ma’lumotlar tuzilishini va boshqalarni o‘z ichiga oladi. Segmentlashga misol 3.21- rasmda keltirilgan.

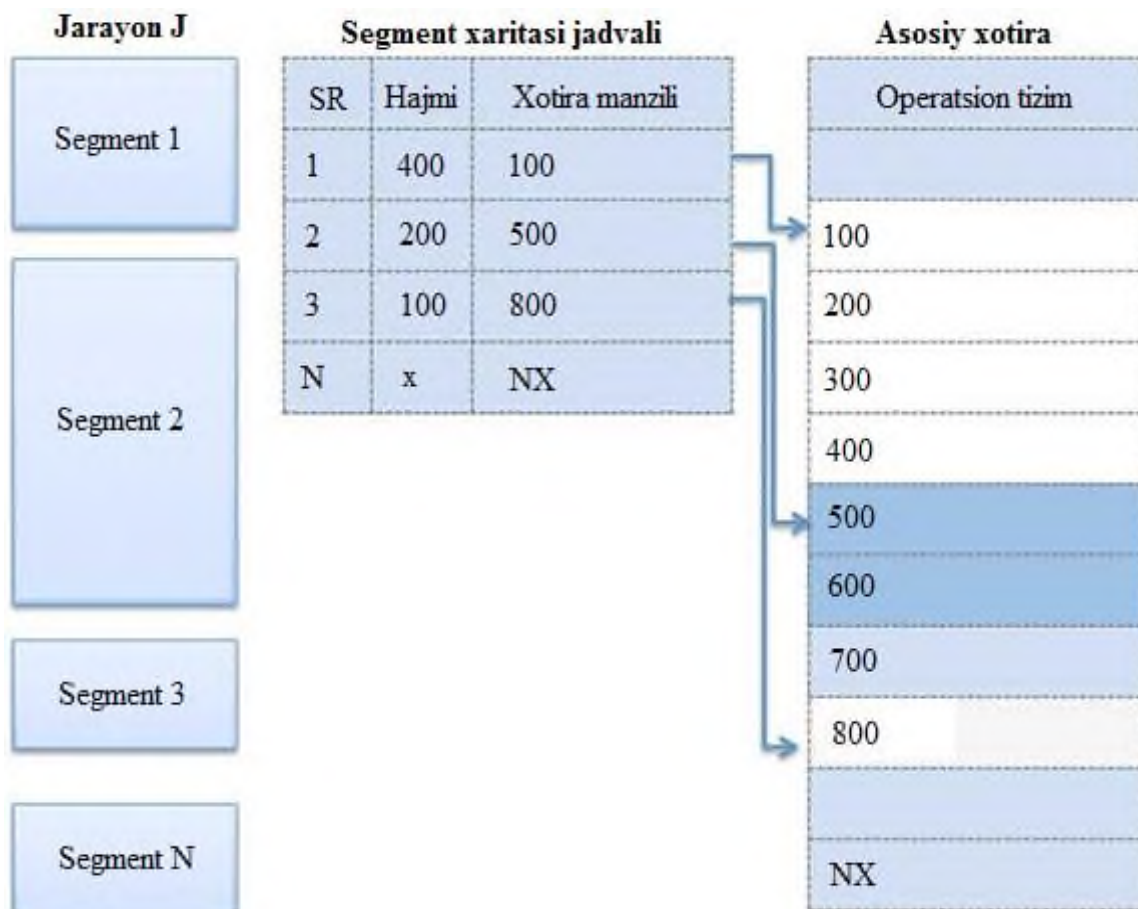


3.20- rasm. Qo‘shni xotirani taqsimlash sxemasi

Operatsion tizim har bir jarayon uchun segment xaritasi jadvalini va bo‘sh xotira bloklari ro‘yxatini, shuningdek segment raqamlarini, ularning o‘lchamlarini va asosiy xotiradagi tegishli xotira maydonlarini o‘z ichiga oladi. Har bir segment uchun jadvalda segmentning boshlang‘ich manzili va segment uzunligi saqlanadi. Xotira manzili segment va almashtirish ma’lumotlarini o‘z ichiga oladi.

Har bir segment – 0 dan boshlanadigan manzillarning to‘g‘ri chiziqli ketma-ketligidir. Segmentning maksimal o‘lchami protsessorning razryadi bilan aniqlanadi (bunda 32 razryadlida manzillashda  $2^{32}$  bayt yoki 4 Gb). Segment manzili dinamik ravishda

o'zgarishi mumkin (masalan, stek segmenti). Segmentlar jadvalining elementida segment boshlanishi fizik manzilidan tashqari odatda segment uzunligi ham beriladi.



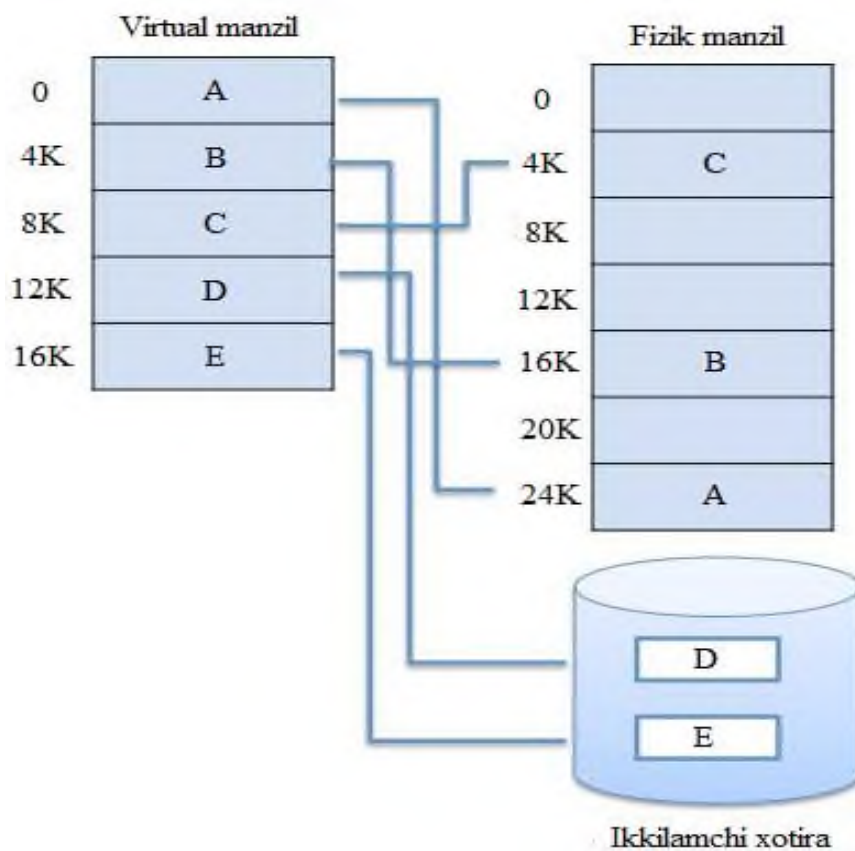
3.21- rasm. Segmentlashga misol

### ***Virtual xotira (Virtual memory)***

Kompyuter tizimda o'rnatilgan hajmdan ko'proq xotiraga murojaat qilishi mumkin. Ushbu qo'shimcha xotira aslida virtual xotira deb nomlanadi va u kompyuterning operativ xotirasini taqlid qilish uchun o'rnatilgan qattiq diskning bir qismi hisoblanadi. Ushbu sxemaning asosiy ko'zga tashlanadigan afzalligi shundaki, dasturlar fizik xotiradan kattaroq bo'lishi mumkin. Virtual xotira ikki maqsadda xizmat qiladi. Birinchidan, bu bizga disk yordamida fizik xotiradan foydalanishni kengaytirishga imkon beradi. Ikkinchidan, bu bizga xotirani himoya qilishga imkon beradi, chunki har bir virtual manzil fizik manzilga tarjima qilinadi.

Dasturni asosiy xotiraga to'liq yuklash talab qilinmaydigan holatlar quyidagicha:

- ❖ Foydalanuvchi tomonidan yozilgan xatolar bilan ishlash protseduralari ma'lumotlar yoki hisob-kitoblarda xatolik yuz bergan taqdirdagina qo'llaniladi;
- ❖ Dasturning belgilangan parametrlari va funksiyalari kamdan-kam ishlatilishi mumkin;
- ❖ Aslida katta bo'lmagan hajmdagi jadval amalda ishlatilgan bo'lsa ham, ko'p jadvallarga belgilangan miqdordagi manzil maydoni belgilanadi;
- ❖ Har bir foydalanuvchi dasturini xotiraga o'rnatish yoki almashtirish uchun kamroq kiritish/chiqarish operatsiyalari talab qilinadi;
- ❖ Dastur endi mavjud fizik xotira miqdori bilan cheklanmaydi;
- ❖ Har bir foydalanuvchi dasturi kamroq fizik xotirani egallashi mumkin, shu bilan birga ko'proq dasturlar bir vaqtning o'zida ishga tushirilishi, protsessordan foydalanish va o'tkazish qobiliyati mos ravishda oshadi.



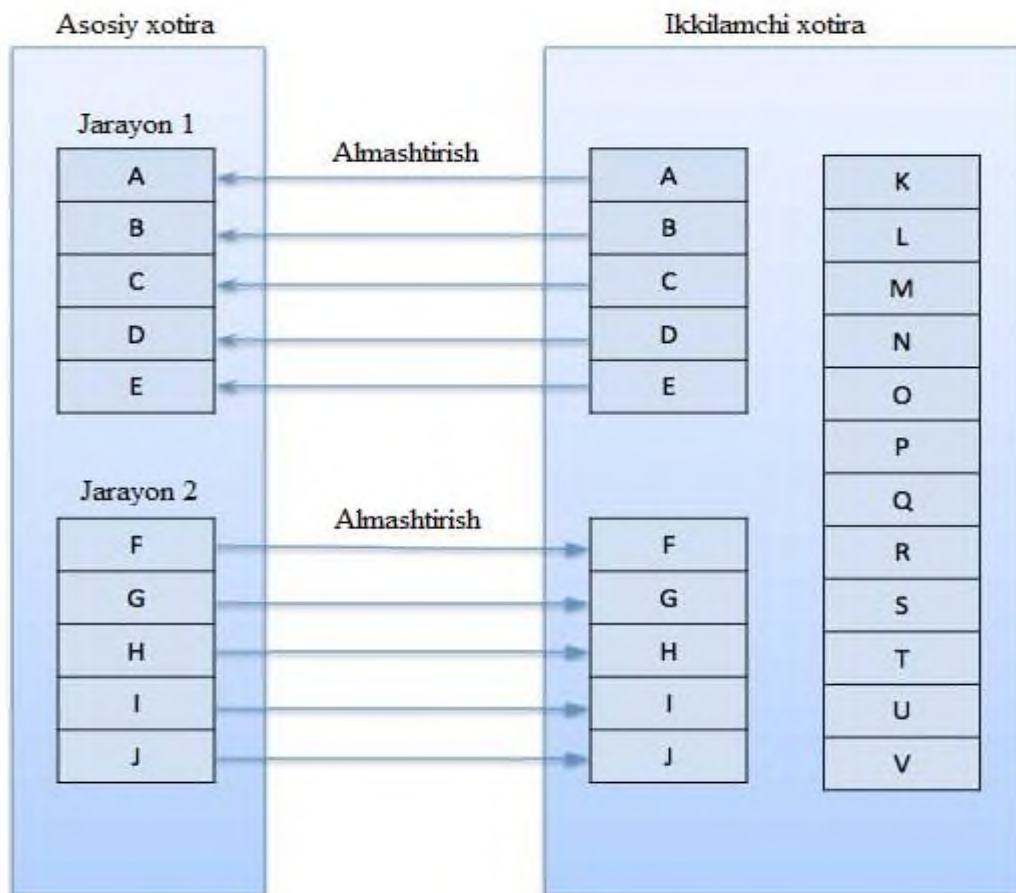
3.22- rasm. Virtual xotira tizimi sxemasi

Umumiy foydalanish uchun mo'ljallangan zamonaviy mikroprotsessorlar, xotirani boshqarish qurilmasi yoki xotirani boshqarish bloki (MMU), qurilma tarkibiga kiritilgan. Xotirani boshqarish blokining vazifasi virtual manzillarni fizik manzillarga tarjima qilishdir. Asosiy misol quyidagi 3.22- rasmda keltirilgan.

Virtual xotira odatda talabni belgilash orqali amalga oshiriladi. U segmentlash yordamida ham amalga oshirilishi mumkin.

### ***Sahifani almashtirish (Demand paging)***

Sahifalarni almashtirish tizimi - almashtirish va sahifali tizimga juda o'xshash. Jarayonlar ikkilamchi xotirada joylashadi va sahifalar oldindan emas, balki talabga ko'ra yuklanadi. Kontekstni almashtirish sodir bo'lganda, operatsion tizim eski dasturning biron bir sahifasini diskka yoki yangi dasturning biron bir sahifasini asosiy xotiraga ko'chirmaydi. Buning o'rniga u shunchaki birinchi sahifani yuklaganidan so'ng yangi dasturni ishga tushiradi va ular bog'langan dastur sahifalarini oladi.



3.23- rasm. Sahifani almashtirish tizimiga misol

### Afzalliklari

- ❖ Katta virtual xotira;
- ❖ Xotiradan yanada samaraliroq foydalanish;
- ❖ Ko‘p dasturlash darajasida chegara yo‘q;

### Kamchiliklari

❖ Sahifali boshqarishning oddiy usuliga qaraganda sahifalar soni va sahifani qayta ishlash uchun protsessor yuklanishi hajmi ko‘proq.

### *Sahifani almashtirish algoritmi*

Sahifani almashtirish algoritmlari – bu usul operatsion tizim yordamida qaysi xotira sahifalarini o‘zgartirish, diskka yozish kerakligini hal qiladi. Sahifada xatolik yuz berganda va sahifani taqsimlash uchun ishlatib bo‘lmaydigan holatlarda, agar sahifalar mavjud bo‘lmasa yoki bo‘sh sahifalar soni talab qilinadigan miqdordan kam bo‘lsa, sahifani almashtirish har safar sodir bo‘ladi. O‘zgartirish uchun tanlangan va yuklanmagan sahifaga yana murojaat qilinganida, u diskdan ma’lumotlarni o‘qishi kerak va bu K/Ch tugallanishini talab qiladi. Ushbu jarayon sahifani almashtirish algoritmining sifatini aniqlaydi: sahifalarni joylashtirish uchun kutish vaqti qanchalik qisqa bo‘lsa, algoritm shunchalik yaxshi bo‘ladi. Sahifani almashtirish algoritmi qurilma tomonidan taqdim etilgan sahifalarga kirish to‘g‘risidagi cheklangan ma’lumotlarga qaraydi va sahifalarni o‘tkazib yuborishni kamaytirish uchun qaysi sahifalarni almashtirish kerakligini tanlashga harakat qiladi, uni dastlabki saqlash xarajatlari va algoritmning protsessor vaqti bilan taqqoslaydi. Sahifani almashtirish algoritmlari juda ko‘p. Bulardan eng ko‘p qo‘llaniladiganlari FIFO va LRU algoritmlari hisoblanadi.

### **Nazorat savollari**

1. Xotirani boshqarish nima uchun kerak?
2. Xotira menejeri qanday vazifalarni bajaradi?
3. Xotira qurilmasi turlari va bir-biridan farqlari?
4. Xotirani boshqarish uchun operatsion tizim funksiyalarini keltiring.
5. Manzil turlari va bir-biridan farqlarini tushuntiring.
6. Fizik xotira nima?

7. Mantiqiy xotira nima?
8. Manzillar bog‘lanishini tushuntiring.
9. Bir foydalanuvchilik xotira sxemasini tushuntiring.
10. Belgilangan qismlarni tushuntiring.
11. Dinamik qismlarni tushuntiring.
12. First-fit algoritmini tushuntiring.
13. Best-fit algoritmini tushuntiring.
14. First-fit algoritmining Best-fit algoritmidan farqini tushuntiring.
15. O‘zgaruvchan qisimli sxemalarni tushuntiring.
16. Sahifali xotirani tushuntiring.
17. Almashtirish (swapping) nima uchun kerak?
18. Sahifali xotirani tushuntiring.
19. Qo‘shni xotirani taqsimlashni tushuntiring.
20. Segmentli xotirani tushuntiring.
21. Virtual xotira nima?
22. Sahifani almashtirish nima uchun kerak?
23. Sahifani almashtirish algoritmlarini keltiring.
24. Segmentli va sahifali xotiraning bir-biridan farqi?
25. Sahifali xotiraning afzalliklari va kamchiliklari.
26. Best-fit algoritmi afzalliklari.



## **IVBOB. KOMPYUTER TASHQI QURILMALARINI BOSHQARISH**

Operatsion tizim jarayonlar, xotira, fayllarniboshqarishdan tashqari kompyuterga ulangan barcha kiritish/chiqarish (K/Ch) qurilmalarini boshqarishni ham ta'minlashi kerak. OTning asosiy vazifalaridan biri dasturlar va kompyuter yordamchi (periferik) qurilmalari o'rtasida ma'lumotlar almashinuvini ta'minlashdir. Aslida, ushbu vazifani bajarish uchun operatsion tizimlarning prototiplari bo'lib xizmat qiladigan birinchi tizim dasturlari ishlab chiqilgan. Zamonaviy OT larda yordamchi qurilmalar bilan ma'lumot almashish funksiyalari kiritish/chiqarish quyi tizimi tomonidan amalga oshiriladi. Ushbu quyi tizimning mijozlari nafaqat foydalanuvchilar va ilovalar, balki tizim ma'lumotlarini olishlari yoki ularni namoyish qilishlari kerak bo'lgan OTning ba'zi komponentlari, masalan, jarayonni boshqarish quyi tizimi, faol jarayonni o'zgartirganda, to'xtatilgan jarayonning kontekstini diskka yozishi va diskdan ishga tushirilgan jarayonning kontekstini o'qishi kerak. Kiritish/chiqarish qurilmalarining asosiy komponentlari tashqi qurilmalarni va fayl tizimini boshqaradigan drayverlardir.

### **4.1. Kompyuter tizimlarining kiritish/chiqarish qurilmalari**

Kiritish/chiqarish amallarini bajaradigan tashqi qurilmalarni uch guruhga ajratish mumkin:

- ❖ foydalanuvchi bilan ishlaydigan qurilmalar. Kompyuter bilan bog'lanish uchun ishlatiladi. Bu guruhga printerlar, displeylar, klaviaturalar, manipulyatorlar (sichqoncha, yo'lakbol, joystik) va boshqalar kiradi;

- ❖ kompyuter bilan ishlaydigan qurilmalar. Elektron qurilmalar bilan bog'lanish uchun foydalaniladi. Ular qatoriga diskli qurilmalar va magnit tasmali qurilmalar, sensorlar, kontrollerlar, o'zgartiruvchilar kiradi;

- ❖ kommunikatsiyalar. Masofadagi qurilmalar bilan bog'lanish uchun foydalaniladi. Bu guruhga raqamli liniyalarning modemlari va adapterlari kiradi. Ushbu qurilmalarni shuningdek, foydalanuvchi va

kompyuter bilan ishlaydigan qurilmalarning kombinatsiyasidan tashkil topgan qurilmalar deb ham ataladi.

Boshqa xususiyati bo'yicha kiritish/chiqarish qurilmalarni blokli va belgili turlariga ajratish mumkin.

Blokli qurilmalar har birining o'z manzili bo'lgan, belgilangan uzunlikdagi bloklarda axborotni saqlovchi qurilmalarni o'z ichiga oladi. Blok hajmi odatda 512 dan 65536 baytgacha o'zgaradi. Blokli qurilmaning muhim xususiyatlaridan biri, har bir blok qolgan bloklardan mustaqil o'qish va yozish mumkinligidir. Eng keng tarqalgan blokli qurilmalar – bu qattiq disklar, Blu-ray diskleri va USB flesh xotira.

Agar e'tibor bilan qaralsa qurilmalar o'rtasida chegara blok bo'yicha manzillanadi, va qurilma qa'tiy belgilanmagan to'g'ridan-to'g'ri manzillash mumkin bo'lmagan alohida qurilmani ko'rish mumkin. Disk blokli manzillash qurilmasiga asoslangan, chunki diskni o'quvchi qurilma joriy holatiga bog'liqlikdan tashqari har doim uni belgilangan silindrga o'tkazish, so'ng kerakli yo'lakka alohida blokni yozish yoki o'qish mumkin bo'ladi. Magnitli tasma (magnitafon) - diskni zahira nusxalarini saqlash uchun ishlatiladi. Tasmada blok ketma-ketlikda saqlanadi. Agar magnitafon, bir nechta buyruqlarni o'qishni buyursa, undan tasmani qaytadan o'rash talab etiladi va jarayon hali beri yakunlovchi blokni berguncha ma'lumot o'qish boshlanadi. Bu operatsiya diskda bloklarni izlashga o'xshaydi, u sezilarli darajada katta vaqtni oladi.

Kiritish/chiqarish qurilmasini boshqa turi - bu belgili qurilma. Belgili qurilma hech qanday blokli tuzilishiz oqim belgili taqdim etiladi yoki qabul qilinadi. Belgili qurilma manzillanmaydi va izlash operatsiyasini bajarmaydi. Printer, tarmoq interfeysi (kartasi), sichqoncha va diskga o'xshash bo'lmagan boshqa qurilmalarni belgili qurilma sifatida ko'rib chiqish mumkin.

Bunday sxemani toifalarga ajratib bo'lmaydi. Ayrim qurilmalar hech bir toifaga mos kelmaydi. Masalan, soatlar blokli manzillashga kirmaydi. Shuningdek ular shakllanmaydi va belgili oqimlarni qabul qilmaydi. Ularning barchasida qa'tiy belgilangan vaqt momentida to'xtalish sodir bo'ladi. Blokli va belgili qurilmalarning barcha qismlari yetarlicha ko'p narsalarni o'z ichiga olgan va kiritish/chiqarish qurilmasidan mustaqil operatsion tizim dasturiy ta'minoti uchun asosiy ishlarni amalga oshirishga imkon beradigan

qulay vosita hisoblanadi. Fayl tizimi mavhum blokli qurilma bilan muloqot qiladi, qurilmaga bog‘liq qismi dasturiy ta‘minotning past pog‘onasida - qurilma drayverida qoladi.

Kiritish/chiqarish qurilmasi ishlash tezligi oralig‘i juda keng. Turli sinflarga tegishli va har bir sinf doirasidagi kiritish/chiqarish qurilmalari o‘rtasida sezilarli darajada farqlanishlar mavjudligini ta’kidlash joiz.

Bu farqlanish ma’lumot uzatish tezliklari bilan qurilmaga xizmat ko‘rsatish sifatini ta’minlashga imkon beradigan dasturni sezilarli darajada murakkablashtiradi. Yildan yilga ko‘plab kiritish/chiqarish qurilmalari ishlash tezligi o‘sib bormoqda. Bu farqlanishlar quyidagi xarakteristikalariga taalluqli:

- ❖ Ma’lumotlarni uzatish tezligi (bir nechta qatorga farqlanadi);
- ❖ Qo‘llash. Qurilmani qo‘llab-quvvatlovchi har bir amal dasturiy ta‘minot va operatsion tizimning algoritmiga ta’sir ko‘rsatadi (masalan, fayllar yoki virtual xotiraning sahifalarini saqlash uchun foydalaniladigan disk, turli dasturiy ta‘minotni talab etadi);

4.1- jadval

<b>Qurilma</b>	<b>Ma’lumot almashish tezligi</b>
Klaviatura	10 bayt/s
Sichqoncha	100 bayt/s
Modem 56K	7 Kbayt/s
Raqamli kamera	3,5 Mbayt/s
Blu-ray disk 4x	18 Mbayt/s
802.11n standartli simsiz tarmoq	37,5 Mbayt/s
USB 2.0	60 Mbayt/s
FireWire 800	100 Mbayt/s
Gigabit Ethernet standart tarmog‘i	125 Mbayt/s
SATA 3 disk	600 Mbayt/s
USB 3.0	625 Mbayt/s
SCSI Ultra 5 disk	640 Mbayt/s
Bir yo‘lakli shina PCIe 3.0	985 Mbayt/s
Shina Thunderbolt 2	2,5 Gbayt/s
SONET OC-768 tarmog‘i	5 Gbayt/s

❖ Boshqaruvning murakkabligi. Printer uchun nisbatan sodda interfeys kerak, diskga esa – ancha murakkabroq. Bu farqlarning OT ga ta'siri kiritish/chiqarish kontrollerlarni murakkablashi bilan bartaraf etiladi;

❖ Ma'lumotlar uzatish birligi. Ma'lumotlar bloklar yoki bayt yoki belgilar oqimlari bilan uzatilishi mumkin;

❖ Ma'lumotlarni taqdim etish. Turli qurilmalar ma'lumotlarni kodlashning, shu jumladan, belgilarni kodlash va juftlikni nazorat qilish, turli sxemalaridan foydalanadi;

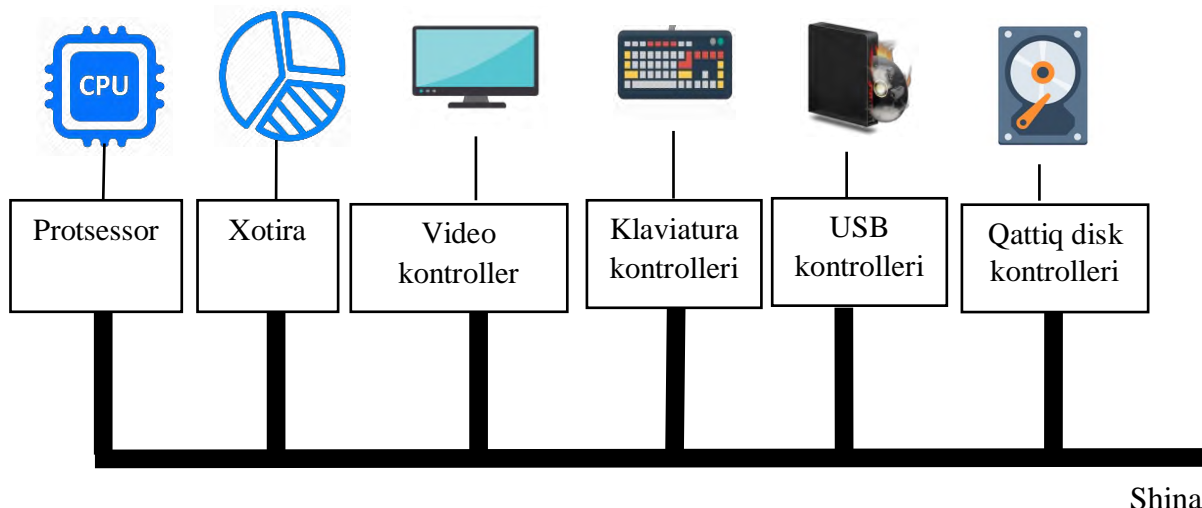
❖ Xatolik shartlari. Xatoliklar tabiati, ular haqida xabar berish usuli, ularning asoratlari va ehtimolli javoblari bir qurilmadan boshqa qurilmaga o'tganida keskin farqlanadi.

Ba'zi eng keng tarqalgan qurilmalarning ma'lumot uzatish tezligi 4.1- jadvalda keltirilgan.

### ***Kontroller qurilmasi***

Kiritish/chiqarish qurilmasi mexanik va elektr komponentlardan iborat. Bu komponentlar ko'p holatlarda maksimal model va modelni umumlashtirish maqsadida mantiqiy ajratilishi mumkin. Odatda u shaxsiy kompyuterlarda kengayuvchan slotga o'rnatiladigan ma'lum bir ko'rinishli plata shaklida bo'ladi. Mexanik komponentlar - qurilmaning o'zi. Odatda kontroller platasi asosiy qurilmaning o'ziga kabel orqali ulanishi mumkin bo'lgan qismlar bilan jihozlangan bo'ladi. Ko'plab kontrollerlar ikki, to'rt, xatto sakkizta qurilmalarni boshqarish imkoniyatiga ega. Agar kontroller va qurilma o'rtasida interfeys standart bo'lsa, unda belgilangan rasmiy standartlar ANSI, IEEE yoki ISO, yoxud alohida qurilma va kontrollerlar ishini soddalashtirish uchun chiqarilgan standartlar mavjud. Ko'plab kompaniyalar SATA, SCSI, USB, Thunderbolt yoki FireWire (IEEE 1394) bilan mos keladigan qattiq disklarni ishlab chiqaradilar.

Ko'plab kichik kompyuterlar yagona shina modeli bo'yicha tashkil qilingan qurilmalar bilan birga ishlaydi (4.1-rasm). Ko'plab mashina, meynfreymlar (dastlabki superkompyuterlar) *kiritish/chiqarish kanali* deb ataladigan ixtisoslashtirilgan kompyuterlarga kiritish/chiqarish xizmatini ko'rsatish uchun bir nechta shinali modelni ishlatadi. Bunday tashkil qilish asosiy protsessorda yuklamani pasaytirishga imkon beradi.



4.1- rasm. Protsessor, xotira, kiritish/chiqarish va boshqa kontrollerlarni ulanish modeli

Qurilma va kontroller o'rtasidagi interfeys qisman past pog'ona interfeysi hisoblanadi. Masalan, biron bir qattiq disk sektor hajmi 512 bayt bo'lgan yo'lakga 2000000 sektor uchun formatlanishi mumkin. Darhaqiqat, kontroller sektor sarlavhasidan (kirish qismi) boshlanadigan ketma-ket bit oqimlarini oladi, keyin sektorda mavjud bo'lgan 4096 bitni kuzatadi va *xatolikni to'g'irlash kodi* (Error-Correcting Code) deb nomlangan nazorat summasi ohitiga yetadi. Sektor sarlavhasi formatlanayotgan (shakl berilayotganda) vaqtda diskga yoziladi. Boshqaruv qurilmasining vazifasi ketma-ket oqimni baytlar blokiga aylantirish va kerak bo'lganda xatoliklarni tuzatishdir. U silindr va sektor raqami, sektor hajmi, sinxronizatsiyalash axboroti va boshqalarni o'z ichiga oladi. Blok bayti odatda kontroller buferida bitdan bitlarni to'playdi. So'ng blok nazorat summasini tekshiradi va agar u sektor sarlovhasida ko'rsatilgan bilan mos tushsa, uni xatosiz deb e'lon qilgandan so'ng, uni tezkor xotiraga ko'chirishi mumkin.

Monitor kontrolleri (videokontroller) past pog'onadagi qurilmalar kabi bitli ketma-ketlikda ishlaydi. U xotira baytlari hisoblaniladi va o'z ichiga signalni tasvirlash va shakllantirish belgilarini oladi, elektron trubka nurlarini modulyatsiyalashda ishlatiladi, uni ekranda rasmini chiqaradi. Bundan tashqari, videokontroller signalni shakllantiradi, elektr nurlarni vertikal va gorizontal joylashishini boshqaradi. Bu signallar suyuq kristall ekranlarda alohida piksellarda ko'rsatiladi va ularni yorug'roq ko'rsatadi.

Kontrollerning ayrim qurilmalari asosan, disk bosqichma-bosqich to'xtalishi juda qiyin. Masalan, zamonaviy diskni o'quvchi kontroller qurilmalari bir necha megabaytli ichki xotiraga ega. Kontrollerni o'qish operatsiyasini bajarish natijasida kerakli silindr ko'rsatilgandan so'ng ma'lumotlarni hisoblash va saqlash amalga oshiriladi (sektorga kirishga ruhsat berishni kutib o'tirmaydi). Bunday keshlash ma'lumotlar so'rovi ketma-ketligida samaraliroq bo'ladi. Talab qilinayotgan ma'lumotni olingandan so'ng kontroller sektorlar ketma-ketligida keshlashni davom ettirishi mumkin bo'ladi. Shunga o'xshash mexanizmlar diskga murojaat qilmasdan o'qish, ko'plab so'rovlariga xizmat ko'rsatishga imkon beradi.

### ***Kiritish/chiqarishni xotirada tasvirlanishi***

Har bir kontrollerda bir nechta registrlar bor, ularning yordami bilan markaziy protsessorga murojaat qilish mumkin. Ushbu registrlarga yozish orqali operatsion tizim qurilmaga ma'lumotlarni taqdim etish, ma'lumotlarni olish, yoqish, o'chirish yoki boshqa amallarni bajarishni buyurishi mumkin. Ushbu registrlardagi ma'lumotlarni o'qib, operatsion tizim qurilmaning joriy holati, yangi buyruqni qabul qilishga tayyor yoki yo'qligini bilib olishi mumkin.

Ko'plab qurilmalarda boshqaruv registrlaridan tashqari, operatsion tizim ma'lumotlarni o'qishi va yozishi uchun ma'lumotlar buferi mavjud. Masalan, kompyuterlarning ekranda piksellarni ko'rsatadigan eng keng tarqalgan usuli bu video xotira. Buni video xotira o'zining buferida (ma'lumotlar tasviri yoziladigan operatsion tizim dasturi) taqdim etadi. Protsessor boshqaruv registri va qurilma ma'lumotlar buferi bilan birgalikda ishlashining ikki xil yo'li mavjud. Birinchisi har bir registr kiritish/chiqarish portiga - 8 yoki 16 razryadli sonlarni berish taklif etiladi. IN REG, PORT kabi maxsus K/Ch buyruqlaridan foydalanib, markaziy protsessor PORT boshqaruv registridan ma'lumotlarni o'qiy oladi va natijani REG registrida saqlaydi. Xuddi shunday, OUT PORT, REG buyrug'i yordamida markaziy protsessor o'zining REG registrini PORT boshqaruv registriga yozishi mumkin.

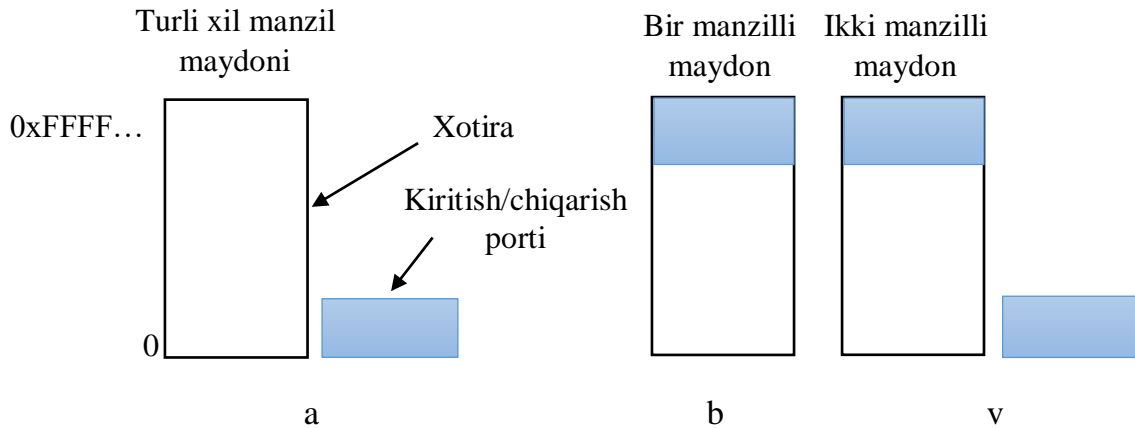
Masalan:

IN REG, PORT

Boshqaruv registr PORT qurilmasida protsessor REG registri quyidagi buyruqlar yozuvini o'z ichiga oladi:

## OUT PORT, REG

Ko‘plab birinchi kompyuterlar, meynfreymlar (ko‘proq IBM 360 va uning barcha eski versiyalari) shunday ko‘rinishda ishlagan. Bu holatda xotira va kiritish/chiqarish sohasi 4.2a- rasmda ko‘rsatilgandek farqli manzil maydonlariga ega bo‘ladi.



4.2- rasm. Xotira va kiritish/chiqarish maydoni bo‘linishi variantlari: a) xotira va kiritish/chiqarish maydoni qismlari; b) kiritish/chiqarishni xotirada tasvirlanishi; v) aralash variant.

Boshqa kompyuterlarda kiritish/chiqarish registrlari odatiy xotira manzil maydonidan foydalanadi (4.2b-rasm). Bunday tashkillash-tirilish kiritish/chiqarishni xotirada tasvirlanishi deb ataladi. U birinchi bo‘lib kichik kompyuter (mini kompyuter) PDP-11da foydalanilgan. Har bir boshqaruv registri xotira bilan bog‘lanmagan xotira noyob manzilini beradi. 4.2v-rasmda boshqaruv registri uchun alohida kiritish/chiqarish porti va ma‘lumotlar buferi xotirasida tasvirlashda foydalaniladigan aralash sxema keltirilgan. Qurilma ma‘lumotlar buferi 640 Kbayt dan 1 Mbaytgacha zahiralangan manzil diapozonidan, va kiritish/chiqarish porti birinchi 64 Kbayt joyini oladigan Pentium protsessorlariga asoslangan tizimda qo‘llaniladigan arxitekturaga o‘xshaydi.

Protsessor barcha holatlarda uni manzilini shina liniyasi manziliga kiritadi, so‘ng boshqaruv liniyasi bo‘yicha o‘qish signalini beradi. Xotiraga murojaat qilishdan kiritish/chiqarish maydoniga murojaat qilishning farqi ikkinchi liniya signalini talab qilishida. Xotiraga murojaat qilinganda xotira so‘rovga javob qaytarishi kerak. Kiritish/chiqarish muhitiga murojaat qilinganda qurilma javob

qaytaradi. Har bir xotira moduli va har bir kiritish/chiqarish qurilmasi manzillari solishtiriladi, xizmat ko'rsatiladigan manzil diapozonida shinaga yozadigan yagona manzil muhiti mavjud (4.2b-rasm). Agar diapazon oralig'ida bo'lsa qurilma so'rovga javob qaytaradi.

### ***To'xtalishlar***

Kontroller registri bir yoki bir necha bitlarni o'z ichiga oladi. Ularni chiqarish operatsiyasi yakunlanganligini tekshirish va aniqlash va yangi ma'lumotlarni qurilmaga kiritish mumkin. Davr protsessorlarda bajariladi va tekshiriluvchi bit *so'rov* yoki *faol kutish* deb ataladigan ma'lumotlarni uzatish yoki qabul qilish qurilmasi tayyorlilik holatida bo'ladi. Ma'lumotlarni uzatish va qabul qilishga tashqi qurilma tayyorligini kutish sababli bir topshiriqli tizimlarda uncha katta bo'lmagan ajratilgan kirishga ruhsat berilgan faol kutish ancha vaqtni talab qiladi.

Bitli holatda qo'shimcha ravishda ko'plab kontrollerlar yozish va o'qish uchun registr tayyorligi to'g'risida protsessorga xabar berishga yordam beradigan to'xtalishlardan foydalanadi. Kiritish/chiqarish kontekstida ko'plab interfeysli qurilmalar ma'lumotlar tayyorligini yoki operatsiyalar yakunlanganligini bildiruvchi chiqarishni generatsiyalaydi, biroq tizimli shina belgilangan liniyasi ishlaydi. To'xtalishni chaqirish operatsiyasi yakunlanishi natijasida protsessor to'xtatiladi va to'xtalishlarni qayta ishlash protsedurasi ishga tushadi. To'xtalishlarni qayta ishlash protsedurasi kiritish/chiqarish yakunlanganligi to'g'risida operatsion tizimga axborot beradi. Shundan so'ng operatsion tizim bitlar holatini tekshirishi va takroran uzatishda yoki ma'lumotni olingandagi xatoliklarni yo'q qilishi mumkin.

Kontrollerga kiradigan to'xtalishlar soni chegaralangan. Masalan, Pentium shaxsiy kompyuterlarida kiritish/chiqarish qurilmasiga faqat 15 ta to'xtalishni kiritishga ruhsat berilgan. Kompyuter ona platasiga o'rnatilgan ayrim kontrollerlar eskirgan. Kontroller to'xtalish signallariga muvofiq o'rnatilgan birlashtiruvchi platalari qismlariga qo'yiladi va ba'zida qurilma yozuvchi yoki qabul qilgich vosita orqali yoqiladi. Agar foydalanuvchi yangi kartani o'rnatasa, u mavjud qurilmalar bilan muammo chiqarmasligi uchun to'xtalishni o'rnatishni foydalanuvchining o'zi amalga oshirishi kerak bo'ladi. Ko'plab foydalanuvchilar buni noto'g'ri bajaradilar, tizimni



yuklash bosqichida to‘xtalish liniyasini qurilmaga mustaqil sozlovchi BIOS tufayli, avtomatik sozlash mexanizmi (Plug and Play) paydo bo‘ldi.

## **4.2. Operatsion tizimning qurilmalarni boshqarish vazifalari**

Multidasturli OTning kiritish/chiqarish kichik tizimi (Input/Output Subsystem) orqali kompyuterni tashqi qurilmalar bilan ma’lumotlar almashinuvida qator umumiy vazifalarni bajarishi kerak, shularning eng muhimlari quyidagilardan iborat:

- ❖ Kiritish/chiqarish qurilmalari va protsessorni parallel ravishda ishlashini tashkil etish;
- ❖ almashinuv tezliklarini muvofiqlashtirish va ma’lumotlarni keshlash;
- ❖ qurilmalar va ma’lumotlarni jarayonlar o‘rtasida taqsimlash;
- ❖ qurilmalarga qulay dasturiy interfeyslarni taqdim etish;
- ❖ tizimga yangi drayverni oson ulash imkoniyati shakllangan holda drayverlar keng spektrini ishlashini ta’minlash;
- ❖ drayverlarni dinamik tarzda yuklash va tizimdan chiqarish;
- ❖ sinxron va asinxron kiritish/chiqarish amallar ishlashini ta’minlash.

Ushbu barcha vazifalar ushbu ma’ruzada keyinchalik batafsil ko‘rib chiqiladi.

Shaxsiy kompyuterlar uchun kiritish/chiqarish vazifalari uch xil usulda bajarilishi mumkin:

1. Dasturlanadigan kiritish/chiqarish yordamida. Bu holda protsessorga kiritish/chiqarish bilan bog‘liq buyruq uchratilsa u bu buyruqni kiritish/chiqarish kontrolleriga tegishli buyruqlarni yuborish yo‘li bilan bajaradi. Bu qurilma tegishli amalni bajaradi, keyin esa kiritish/chiqarish holati registrlarida tegishli bitlarni o‘rnatadi va boshqa hech qanday signallarini, shu jumladan to‘xtatish signallarini, yubormaydi. Kiritish/chiqarish amalni yakunlanganligini tekshirish maqsadida protsessor kiritish/chiqarish modulning holatini muntazam ravishda tekshirib boradi. Shunday qilib protsessor kiritish/chiqarish amallarni, shu jumladan qurilma holatini aniqlash, o‘qish-yozish buyruqlarini jo‘natish va ma’lumotlarni uzatishni bevosita boshqarib boradi. Protsessor kiritish/chiqarish kontrolleriga kerakli buyruqlarni jo‘natadi va joriy jarayonni kiritish/chiqarish amalini yakunlanishini

kutish holatiga o'tkazadi. Mazkur usulni kamchiligi – bu kiritish/chiqarishni boshqarish bilan bog'liq jarayon vaqtining katta yo'qotishlaridir.

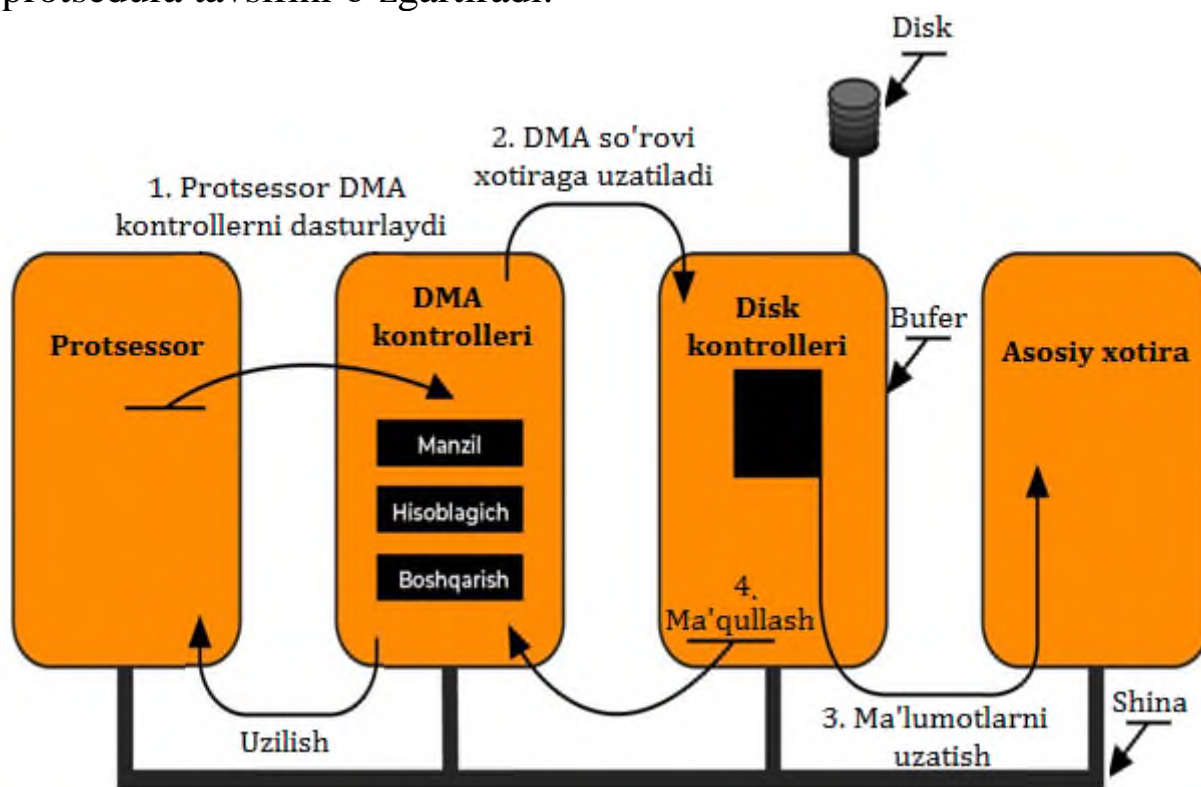
2. To'xtatishlar bilan boshqariladigan kiritish/chiqarish. Protsessor kiritish/chiqarish kontrolleriga kerakli buyruqlarni jo'natadi va joriy jarayonni bajarishni davom etadi, agar kiritish/chiqarish amallar bajarishni kutishda zarurat bo'lmasa. Aks holda joriy jarayon kiritish/chiqarish yakunlanganligi to'g'risida uzilish signali olingunga qadar vaqtinchalik to'xtatiladi, protsessor esa boshqa jarayonni bajarishga o'tkaziladi. To'xtatishlar mavjudligini protsessor bajarilayotgan buyruqlarni har bir siklining oxirida tekshiradi. Bunday kiritish/chiqarish dasturlanadigan kiritish/chiqarishga nisbatan anchagina samaraliroq, chunki bunda protsessorning behuda bekor turishi bilan bog'liq keraksiz kutishga yo'l qo'yilmaydi. Lekin bu holda ham kiritish/chiqarish baribir protsessorning katta miqdorda vaqtini iste'mol qiladi, chunki xotiradan kiritish/chiqarish moduli (kontroller)ga uzatiladigan har bir so'z yoki teskari amal albatta protsessor orqali o'tishi kerak.

3. Xotiraga to'g'ridan-to'g'ri kirish (DMA) kontrolleri yordamida. Xotirada kiritish/chiqarish qurilmasi tasvirlanishini qo'llab-quvvatlashni mustaqil bo'lishi uchun markaziy protsessorga u bilan ma'lumotlar almashish uchun kontroller qurilmasini manzillash kerak bo'ladi. Protsessor bayt asosida kontrollerdan ma'lumotlarni so'rashi mumkin. Agar ma'lumotlarni katta blokli qurilmadan qabul qilish talab etilsa (masalan, disk), vaqtning katta qismi uni ishga tushirishga sarflanadi. Ushbu sabab bo'yicha xotira bilan birga ishlash uchun *xotiraga to'g'ridan-to'g'ri kirish* (Direct Memory Access, DMA) deb ataladigan usul ishlatiladi, ko'plab kompyuterlar bunday kontrollerlarga ega (4.3-rasm). Ba'zida DMAkontrollerlar boshqa kontrollerlarga qo'yiladi (masalan diskni o'quvchi kontrollerlar), lekin har bir qurilmaga talab qilinadi. Tizimlar ko'pincha yagona DMAkontrolleriga ega bo'ladilar, ya'ni odatda ona plataga joylashtiriladi. U parallel ravishda ko'plab kiritish/chiqarish qurilmalarida ma'lumotlar almashishni boshqaradi.

DMA kontroller o'qish va yozish uchun protsessorga kirishga ruhsati bor bir nechta registrlarga ega: manzil registri, baytlarni hisoblagich va qator boshqaruv registrlari. Kiritish/chiqarish portidan so'nggi foydalanadigan bir davrda jo'natuvchi ma'lumotlar (bayt yoki

soʻz) va bayt sonlari almashadi, maʼlumotlar almashishni yoʻnaltiradi (oʻqiydi yoki yozadi).

DMA funksiyasini aniqlashtirib olish uchun dastlab xotiraga toʻgʻridan-toʻgʻri kirish mavjud boʻlmagan diskli oʻqishni amalga oshirilishini koʻrib chiqamiz. Dastlab kontroller bitdan bitgacha ketma-ketlikda kontroller ichki buferi koʻrsatilmagunga qadar blok (bir yoki bir nechta sektor) deb hisoblaniladi. Soʻng nazorat summasi hisoblaniladi va mavjud xatolar tekshiriladi. Hisoblash, siklik, baytli yoki soʻzli koʻrinishda amalga oshiriladi. Bayt (soʻz) hisoblangandan soʻng u asosiy xotirada saqlanadi, xotira manzili koʻpayadi, hisoblagich qolgan elementlarni dekrementlaydi. Hisoblagich nolga teng boʻlganda sikl toʻxtatiladi. Xotiraga toʻgʻridan-toʻgʻri kirish protsedura tavsifini oʻzgartiradi.



4.3-rasm DMA-kontrolleri ishlashi

Dastlab protsessor DMAkontrollerni dasturlaydi, uning registriga yozuvlarni yozadi (qayerga joʻnatilishi kontrollerda koʻrsatilgan (4.3-rasmda 1-qadam). Soʻng kontrollerga oʻzining ichki buferida diskdagi maʼlumotlarni hisoblash buyrugʻini joʻnatadi va nazorat summasini tekshiradi. DMA maʼlumotlari kontrollerda paydo boʻlgandan soʻng ishlashni boshlashi mumkin boʻladi.

DMAkontroller ma'lumotlarni ko'chirishni boshlaydi, o'qish so'rovini shina bo'yicha disk kontrolleriga jo'natadi (2-qadam). Bu so'rov o'qishga odatiy so'rov bo'ladi, chunki diskni kontroller markaziy protsessoridan yoki DMAkontrolleridan jo'natilganligini bilmaydi. Xotira manzili odatda manzilli shinada joylashgan bo'ladi, o'zining ichki buferidan keyingi so'zlarni qayerga jo'natish kerakligi to'g'risida boxabar bo'ladi. Xotirada yozish shinaning yana bir standart sikli bo'ladi. (3-qadam). Yozish tugagandan keyin disk kontrolleri shina bo'yicha DMAkontrollerini qo'llab-quvvatlaydigan signalni jo'natadi (4-qadam). So'ng DMAkontroller xotirada foydalanilgan manzilni inkrementlaydi va hisoblagich bayt qiymatini dekrementlaydi. Shundan so'ng hisoblagich nolga teng bo'lgunga qadar 2-4- qadamlar takrorlanadi. DMAkontrollerida ko'chirish (nusxa olish) sikli tugashi bo'yicha protsessor to'xtatiladi. Operatsion tizimga xotiradan diskli blokni ko'chirish kerak emas. U allaqachon u yerda joylashgan bo'ladi. Kontrollerga ichki xotira ikki sabab tufayli zarur. Birinchisi ichki buferlash hisobiga disk kontrolleri xotiraga ma'lumotlar jo'natilish boshlanishigacha nazorat summasini tekshira olishi mumkin bo'ladi. Agar qiymatlar mos kelmasa, xatolik to'g'risida signal shakllantiriladi va ma'lumotlarni uzatish amalga oshirilmaydi.

Ikkinchidan diskni o'qish operatsiyasi boshlanganda bitlarni kontroller qabul qilish va qilmasligiga bog'liq bo'lmagan holda doimiy tezlik bilan kelib tushishi boshlanadi. Agar disk kontrolleri bu ma'lumotlarni xotiraga to'g'ridan-to'g'ri yozishga harakat qilsa, uni tizimli shina bo'yicha amalga oshiradi. Agar navbatdagi so'zni uzatishda shina biron bir boshqa qurilma bilan band bo'lsa disk kontrolleri kutishiga to'g'ri keladi. Agar keyingi so'z kontroller kechikishlarni saqlashini amalga oshirishga ulgurishiga nisbatan diskda avval kelgan bo'lsa kontroller avvalgi so'zni yo'qotadi yoki uni yana bir bor xotiralaydi (esga oladi). Agar shina intensiv foydalanilsa kontroller bir nechta so'zlarni shu zahoti saqlashi kerak bo'ladi va ko'plab xizmat ishlarini bajaradi. Xotiraga to'g'ridan-to'g'ri kirish operatsiyasi boshlanmagunga qadar shina ichki buferi mavjud bo'lishi zarur emas. Natijada diskli qurilma soddaroq bo'ladi, chunki vaqtinchalik parametrlarni xotiraga to'g'ridan-to'g'ri kirishi qiyin bo'lmaydi.

## ***Protsessor va kiritish/chiqarish qurilmalarini parallel ravishda ishlashi***

Kiritish/chiqarish qurilmalari va protsessorni parallel ravishda ishlashi mazkur qurilmaga tegishli kontroller va drayverni o‘zaro harakatda bo‘lish yo‘li bilan tashkil etiladi. Kontroller muntazam ravishda drayverdan qurilmaga chiqariladigan ma’lumotlar, hamda, bu ma’lumotlarga nisbatan qanday amallar bajarilish kerakligini bildiruvchi (masalan, ekranning ma’lum qismiga matn ko‘rinishda ma’lumotni chiqarish yoki diskning aniq sektoriga yozish) boshqaruv ko‘rsatmalarini qabul qiladi. Kontroller boshqaruvi ostida qurilma o‘z amallarini ma’lum bir vaqt davrida mustaqil tarzda, markaziy protsessor va OT tomonidan e’tibor talab qilmagan holda, bajarishi mumkin. Bu vaqt ko‘p omillarga bog‘liq – chiqarilayotgan ma’lumotlarning hajmi, qurilmaning ishlash tezligi va hokazo. Xatto eng oddiy, sodda funksiyalarni bajaradigan kontroller odatda protsessor tomonidan navbatdagi ko‘rsatmani olganidan keyin mazkur funksiyani bajarish uchun bir muncha ko‘p vaqt sarflaydi. Bunday xulosa murakkab kontrollerlarga nisbatan ham o‘rinli, chunki barcha kiritish/chiqarish qurilmaning, hatto eng katta tezlik bilan ishlaydigan, ishlash tezligi odatda protsessorning tezligidan sezilarli darajada past. Demak protsessor tomonidan buyruqlar berilishi o‘rtasidagi davrlarda kontrollerlar o‘z amallarini OT ga bog‘lanmagan holda bajaradi. OT ning kiritish/chiqarish kichik tizimidan “faqat” tegishli kontrollerlardan har bir drayverning unga keluvchi to‘xtatish signallarga javob berishning ma’qul vaqtini ta’minlagan holda, kompyuter tarkibiga kiritilgan barcha kiritish/chiqarish qurilmalarning drayverlarini ishga tushurish va vaqtinchalik to‘xtatish talab etiladi. Bunda protsessorning kiritish/chiqarishni ta’minlashga sarflaydigan vaqti foydalanuvchilar jarayonlari foydasiga minimallashtirilgan bo‘lishi lozim. Mazkur vazifa real vaqtda ishlaydigan tizimlarda jarayonlarni rejalashtirish/dispatcherlashtirishning klassik topshirig‘i deb hisoblanadi va odatda to‘xtatish bo‘yicha ko‘p darajali ustuvorlik sxema asosida hal etiladi. Javob berishning ma’qul darajasini ta’minlash uchun barcha drayverlar (yoki drayverlarning bir qismi) mazkur qurilmaga ruxsat berilgan javob berish vaqti va tegishli drayverni bajarish uchun jarayon sarflaydigan vaqtga ko‘ra bir nechta

ustuvor darajalar bo'yicha taqsimlanadi. Ustuvorlik sxemasini amalga oshirish uchun, odatda OT ning to'xtatishlar dispetcheri ishlatiladi.

### ***Almashinuv tezliklarini muvofiqlashtirish va ma'lumotlarni keshlash***

Ma'lumotlar almashinuvida doimo tezliklarni muvofiqlashtirish masalasi paydo bo'ladi. Masalan, agar bir foydalanuvchi jarayon ayrim ma'lumotlarni ishlab chiqib, ularni operativ xotira orqali boshqa foydalanuvchi jarayonga uzatayotgan bo'lsa, bu vaziyatga umuman olganda ma'lumotlarni generatsiya qilish va ularni o'qish tezligi bir-biriga mos kelmaydi. Amaliy jarayonlar o'rtasida ma'lumotlar almashinuvi paytida tezlikni muvofiqlashtirish odatda operativ xotiradagi ma'lumotlarni buferlash va jarayonlarni buferdan foydalanishini sinxronlash hisobiga erishiladi.

Shunga o'xshash yo'l bilan kiritish/chiqarish kichik tizimda almashinuv tezliklarini muvofiqlashtirish muammosi hal etiladi. Bunda operativ xotirada ma'lumotlarni bufferlash usulidan keng foydalaniladi. Kiritish/chiqarishni yuqori tezligini ta'minlash birinchi navbatdagi vazifa deb hisoblangan ixtisoslashgan operatsion tizimlarda (real vaqtda boshqarish, tarmoq fayl xizmatlari va hokazo) operativ xotiraning asosiy qismi amaliy dasturlar kodlariga emas, balki ma'lumotlar buferlariga ajratiladi. Lekin faqat kiritish/chiqarish kichik tizimning operativ xotirasi asosida bufferlash yetarli emas, chunki jarayonlar ishlov talab etiladigan ma'lumotlarni joylashtiradigan operativ xotira bilan almashinuvning tezligi va tashqi qurilmaning ishlash tezligi o'rtasidagi tafovut operativ xotirani vaqtinchalik bufer sifatida foydalanish uchun ko'pincha xaddan tashqari juda katta bo'ladi; uning hajmi shunchaki (oddiygina) yetmasligi mumkin. Katta hajmdagi ma'lumotlar almashinuvi zaruratida kiritish/chiqarish tizimlarda spulingdan foydalaniladi; bunda ma'lumotlar diskning alohidagi, diskli fayl yoki spul-fayl deb nomlanuvchi, qismida bufferlanadi. Spul-fayl qo'llanishining yaqqol misoli deb ma'lumotlarni printerga chiqarishni tashkil etish hisoblanadi. Chop etiladigan hujjatlarning bir nechta o'n megabayt hajmda bo'lishi tez-tez uchraydigan hol, shuning uchun ularni vaqtinchalik saqlash uchun (har bir hujjatni chop etish bir necha daqiqadan boshlab o'n daqiqalargacha vaqt talab etadi) operativ xotiraning hajmi shubhasiz yetarli emas. Bu muammoning boshqa

yechimi – bu tashqi qurilmalar kontrollerlarida katta hajmli bufer xotirasidan foydalanish. Bunday yondashuv ma'lumotlarni diskka joylashtirish almashinuvni haddan tashqari sekinlashtirgan (yoki ma'lumotlar diskning o'ziga chiqarilgan) hollarda ayniqsa foydali. Masalan, grafik displeylarning kontrollerlarida hajmi operativ xotira bilan taqqoslanadigan bufer xotira qo'llaniladi, va bu grafik shakllarni ekranga chiqarishni sezilarli darajada tezlashtiradi. Kiritish/chiqarishda ma'lumotlarni buferlash nafaqat protsessor va tashqi qurilmaning ishlash tezliklarini muvofilashtiradi, balki boshqa masalani hal etadi, ya'ni ma'lumotlarni keshlash hisobiga kiritish/chiqarish real amallari sonini qisqartiradi. Diskli kesh deyarli barcha operatsion tizimlarni kiritish/chiqarish kichik tizimlarining ajralmas qismi bo'lib, saqlanayotgan ma'lumotlardan foydalanish vaqtini sezilarli darajada qisqartiradi.

### ***Qurilma va ma'lumotlarni ajratish***

Kiritish/chiqarish qurilmalari jarayonlarga ham yakka (monopol), ham ajratilgan tarzda foydalanish uchun berilishi mumkin. Bunda OT jarayonlarni hisoblash tizimining boshqa resurslardan foydalanish usullaridan kontrollerlarga ham foydalanishni ta'minlashi kerak, ya'ni jarayon bajariluvchi nomidan foydalanuvchi yoki foydalanuvchilar guruhining qurilma ustidan u yoki bu amalni bajarishga huquqlarini tekshirish. OT faqat ma'lum qurilmadan foydalanish emas, balki bu qurilma saqlaydigan ma'lumotlarning ayrim qismlaridan foydalanish ustidan nazorat qilishi mumkin. Diskli fayllar va kataloglardan foydalanishni nazorat qilish muhim bo'lgan qurilmaning yaqqol misolidir. Ohirgi vaqtlarda qurilmadan umumiy holda birgalikda foydalanish rejimi topshirig'i zaruriy shart bo'lib qolgan. Bitta qurilma vaqtning turli davrlarida ham ajratilgan, ham yakka rejimlarda ishlashi mumkin. Shunga qaramasdan faqat bitta rejimda ishlash xos bo'lgan qurilmalar mavjud, masalan ketma-ket portlar va alifbo-raqamli terminallar ko'pincha yakka rejimda foydalaniladi, disklar esa – birgalikda foydalanish rejimida. Birgalikda foydalanilgan hollarda, umumiy unumdorlikni oshirish maqsadida OT turli jarayonlar uchun kiritish/chiqarish amallar ketma-ketligini optimallashtirishi kerak. Masalan, bir nechta jarayonlarni disk bilan ma'lumotlar almashinuvida amallar ketma-ketligi shunday tartibga solish mumkinki, bunda kallaklarni unumsiz xarakatlanish vaqt sarfi

sezilarli kamayadi (bunda ayrim jarayonlar uchun kiritish/chiqarish amallari ma'lum darajada sekinlashishi mumkin). Qurilmaning jarayonlar o'rtasida taqsimlanishida jarayonlar ma'lumotlarini bo'lish zarurati paydo bo'lishi mumkin. Odatda bunday ehtiyoj ketma-ketligi qurilmalardan birgalikda foydalanish paytida vujudga keladi, chunki bunday qurilmalar to'g'ridan-to'g'ri foydalaniladigan qurilmalardan farqli ravishda manzillanmaydi. Bunday qurilmaning yaqqol vakili – bu printer. Bunday qurilmalar uchun chiqarish ketma-ketligi topshirig'i tashkil etiladi, bunda har bir vazifa ma'lumotlarning bo'lib bo'lmaydigan bir qismi, masalan, chop etishga mo'ljallangan hujjat, shaklida bo'ladi. Vazifalar ketma-ketligini saqlash uchun spul-fayl ishlatiladi, u printer va operativ xotiraning tezliklarini muvofiqlashtiradi, hamda, ma'lumotlarni mantiqiy qismlarga bo'linishini tashkil etishga imkon beradi. Ma'lumotlarni spul-fayldagi o'zining qismlariga joylashtirib, jarayonlar ularni printerga bir vaqtda chiqarishi mumkin.

### **4.3. Ketma-ket yozishga asoslangan axborot saqlash qurilmalari**

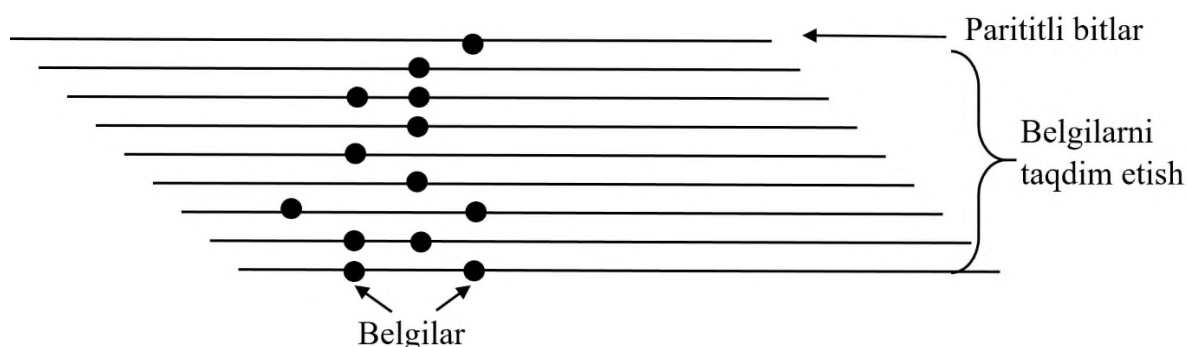
Magnit lenta dastlabki kompyuter tizimlarida odatiy ikkilamchi xotira sifatida ishlab chiqilgan va ketma-ket saqlanadigan yozuvlarni taqdim etadi. Ushbu yozuvlarning uzunligi odatda dastur tomonidan belgilangan va har bir yozuv lentadagi o'rniga qarab aniqlanishi mumkin. Shuning uchun, bitta yozuvga kirish uchun lenta o'rnatilishi va boshidan kerakli ma'lumot yozilgan pozitsiya topilgunga qadar tez yo'naltirilishi kerak. Bu kerakli ma'lumotni topish uchun ko'p vaqt talab qiladigan jarayon bo'lishi mumkin.

Bu qancha vaqt davom etishi mumkinligini bilish uchun 2400 fut (731.5 metr) uzunlikdagi lenta g'altakdan foydalanadigan gipotetik kompyuter tizimini ko'rib chiqaylik (4.4- rasmga qarang). Ma'lumotlar lenta uzunligini bo'ylab o'tuvchi to'qqiz parallel yo'laklarning sakkiztasida qayd etiladi. Rasmning yuqori qismida ko'rsatilgan to'qqizinchi yo'lak muntazam xatolarni tekshirish uchun ishlatiladigan parititli bitga ega.

Bir dyuymga yozilishi mumkin bo'lgan belgilar soni lentaning zichligi bilan aniqlanadi, masalan, dyuym uchun 1600 bayt (bpi). Masalan, agar siz har birida 160 belgidan iborat yozuvlar bo'lsa va



ularni zichligi 1600 bpi bo'lgan lentada saqlasangiz, nazariy jihatdan siz 10 dona yozuvni lentaning bir dyuymida saqlashingiz mumkin.



4.4- rasm. To'qqiztayo'laklimagnitlenta

Biroq, amalda, busizyozuvlarniqandaysaqlashniqarorqilganingizgabog'liq: alohida-alohidayokiguruhlanganbloklarga. Agaryozuvlaralohidasaqlansa, harbiroyuzvni boshlashvatugashjoylariniko'rsatishuchunbo'shjoyajratilishikerak. Agaryozuvlarbloklardasaqlanadiganbo'lsa, uholdabutunblokoldidanvaundankeyinbo'shjoyqoldiriladi, lekinalohidayozuvlarblokichidaketma-ketsaqlanadi.

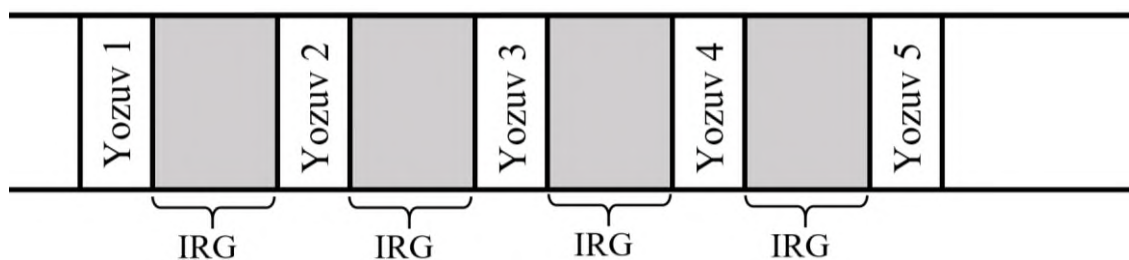
Alohidayokiblokdasaqlasho'rtasidagifarqnitushunishuchun, keling, magnittasmadao'qishvayozishmexanizminiko'ribchiqaylik. Magnitlenta faqat yozuv gaki rish zaruratitug'ilgandao'qish/yozishkallagi ostidasiljiydi; boshqavaqtlardato'xtabturadi. Shundayqilib, lentato'xtaydi, o'qiydivayuqoritezlikdaharakatqiladiyokito'xtaydi, yozadivayanaboshlaydivahokazo.

Yozuvlarxuddishundaytartibdayozilaredi.

Lentanito'xtatishuchunvaqtvabo'shjoykerak, shuninguchunharbiroyuzvorasidabo'shjoyqoldiriladi.

Buyozuvlarniajratishoralig'i (IRG-interrecordgap) ajratilganyozuvlarningo'lchamidanqat'iynazar, uzunligi 1/2 dyuymnitashkilqiladi. Shuninguchun, agar 10 tayo zuvalohidasaqlansa, harbiroyozuvlorasidato'qqizta 1/2 dyuymli yozuvlarniajratishoralig'ibo'ladi. Ushbu misolda biz har bir yozuv atigi 1/10 dyuym deb taxmin qilamiz. 4.5- rasmda 1 dyuymli ma'lumotlarni saqlash uchun 5,5 dyuymli lenta kerak, bu saqlash vositasidan foydalanishning unchalik samarali usuli emas. Yozuvlarnilenta gazyozib olishdan oldin ularni bloklarga guruhlash muqobi

lusulhisoblanadi. Bubliklashdebnomlanadivafaylyaratilganda amalga oshiriladi.



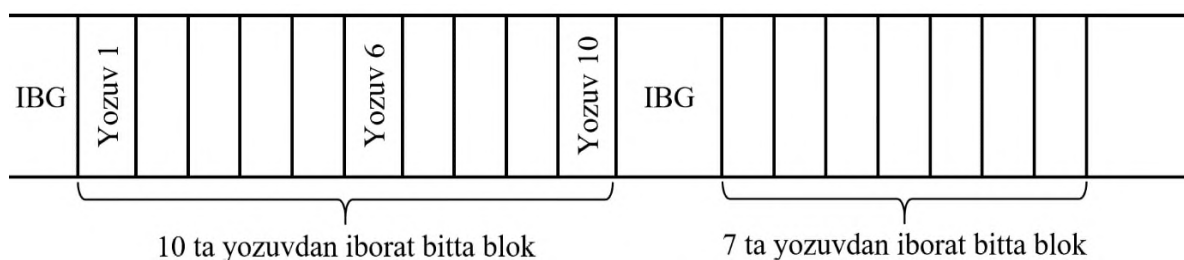
4.5- rasm. Magnitlentadayozuvlarniajratishoralig‘i (IRG)

Blokdaagi yozuvlar soni odatda dastur tomonidan belgilanadi va u uzatish tezligi (ips-inches per second) deb nomlanib, lenta zichligi bo‘lgan lentani aylantirish tezligiga (bpi-bytes per inch) transport tezligini ko‘paytirishdan hosil boladi.

$$\text{uzatish tezligi} = \text{lenta zichligi} * \text{transport tezligi}$$

Aytaylik, bizning gipotetik tizimimizda transport tezligi sekundiga 200 dyuymni tashkil qiladi. Shu sababli, har bir dyuymdagi 1600 bayt (bpi) holatida bir soniyada jami 320000 bayt o‘tkazilishi mumkin, shuning uchun nazariy jihatdan blokning optimal hajmi 320000 baytni tashkil qiladi. Ammo bu usul butun blokni buferga o‘qishni talab qiladi, shuning uchun bufer kamida blokdan kattaroq bo‘lishi kerak.

4.6- rasmda qayd etilganidek, blokirovka oralig‘i (IBG-interblock gap) uzun-ligi 1/2 dyuymga teng, ammo har bir 10 ta yozuv blokidagi ma‘lumotlar hozirda atigi 1 dyuym lentada saqlanadi - shuning uchun biz bor-yo‘g‘i 1,5 dyuymdan foydalandik (4.5- rasmda ko‘rsatilgan 5.5 dyuymo‘rniga) va biz atigi 1/2 dyuymli lentani (4,5 dyuym o‘rniga) isrof qildik.



4.6- rasm. Magnit lentaga yozilgan bloklar

Bloklash ikki xil afzalliklarga ega:

- Faqatgina READ buyrug'i butun blokni harakatlantirishi mumkin va bir nechta mantiqiy yozuvlarni o'z ichiga olgan fizik yozuvni asosiy xotiraga ko'chirishi mumkinligi sababli kamroq kirish/kiritish operatsiyalari kerak bo'ladi.

- Fizik yozuvlar hajmi bo'shliqdan kattaroq bo'lganligi hisobiga kamroq joy isrof qilinadi.

Bloklashning ikkita kamchiliklari taqqoslash bilan yumshoq ko'rinadi:

- Blokirovka qilish, blokirovkadan chiqarish va yozishda davom etish uchun qo'shimcha dasturlar va dasturiy ta'minot kerak.

- Agar sizga bitta mantiqiy yozuv kerak bo'lsa, blokni to'liq o'qishingiz kerakligi uchun bufer maydoni isrof bo'lishi mumkin.

Blokka yoki magnit lentadagi yozuvga kirish uchun qancha vaqt talab etiladi? Bu yozuv qayerda joylashganiga bog'liq, ammo biz ba'zi umumiy hisob-kitoblarni amalga oshirishimiz mumkin. Masalan, bizning 2400 futli (731.5 metr) g'altakning lenta tashish tezligi sekundiga 200 dyuym (ips) bo'lganida taxminan 2,5 daqiqada to'xtamasdan o'qilishi mumkin. Shuning uchun, lentadagi so'nggi yozuvga kirish uchun 2,5 daqiqa vaqt kerak bo'ladi. O'rtacha hisobda yozuvga kirish uchun 1,25 daqiqa vaqt ketadi. Bir yozuvga ketma-ket kirish uchun lentani ishga tushurish va to'xtatish uchun sarflangan - 0,003 sekund yoki 3 millisekund vaqt oralig'i talab etiladi.

4.2- jadval

<b>Qiymatlar</b>	<b>Kirish vaqti</b>
Maksimum kirish vaqti	2.5 minut
O'rtacha kirish vaqti	1.25 minut
Minimum kirish vaqti	3 sekund

4.2- jadvaldanki, kirishvaqtlariturlichabo'lishimumkin.

Bumagnittasmasiniodatdagiikkilamchixotirauchunyomonroqmuhitgaa ylantiradi, bundantashqarijudayuqoriketma-

ketlikfaolligibo'lganfayllarbundanmustasno, ya'nidasturdavomida90-100 foizyozuvlarga ketma-ketkirish talabqilinadigan fayllarbundan mustasno.

2400 fut (731.5 metr) vatezligisekundiga 200 dyuym (ips) bo'lgan magnit lentachun yozuvlarga kirish vaqtlari 4.2-jadvaldako'rsatilgan.

#### **4.4. To'g'ridan to'g'ri yozishga asoslangan axborot saqlash qurilmalari**

To'g'ridan to'g'ri kirishni saqlash qurilmalari (DASD) – bu belgilangan joyga to'g'ridan-to'g'ri o'qish yoki yozish imkoniyatiga ega qurilmalardir. DASDlarni uch toifaga bo'lish mumkin: magnit disklar, optik disklar va flesh-xotira. DASDga kirish o'rtasidagi vaqt farqi magnit lentani ishlatishda unchalik katta bo'lmasa ham, ma'lum bir yozuvning joylashuvi unga kirish uchun qancha vaqt kerak bo'lishiga bevosita ta'sir qiladi.

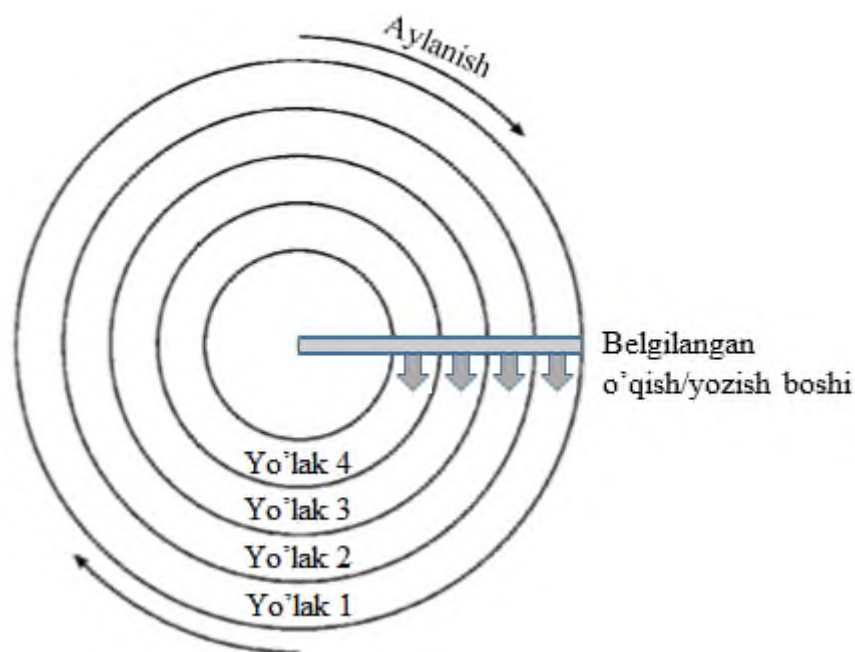
##### ***Belgilangan boshli magnit disk qurilmasi***

Belgilangan boshli magnit disk odatda magnit plyonka bilan qoplangan, katta CD yoki DVDga o'xshaydi, odatda formatlanadigan, har ikki tomoni ham konsentrik aylanalardan iborat. Har bir aylana - bu yo'lak. Ma'lumotlar har bir yo'lakka uning ustida joylashgan belgilangan o'qish/yoziq boshi orqali ketma-ket yoziladi. Belgilangan boshli disklar 4.7-rasmda ko'rsatilgan. Shuningdek ular juda tezkor, harakatlanuvchi boshli magnit disk larga qaraganda tezroq hisoblanadi. Harakatlanuvchi boshli magnit disk qurilmasini keyingi bo'limda ko'rib chiqamiz. Lekin, uning asosiy kamchiligi - bu harakatlanadigan boshli disk larga nisbatan narxining yuqoriligi va xotira hajmining kichikligi (kamayishi) hisoblanadi (chunki o'qish/yoziq boshlarining kengligini ta'minlash uchun yo'laklar bir-biridan uzoqroq joylashishi kerak). Ushbu qurilmalar tezlik katta ahamiyatga ega bo'lgan hollarda, masalan, kosmik parvozda yoki samalyotlarda qo'llaniladi.

##### ***Harakatlanuvchi boshli magnit disk qurilmasi***

Kompyuterning qattiq disk lari kabi harakatlanadigan boshli magnit disk larda har bir diskning har bir yuzasida harakatlanuvchi

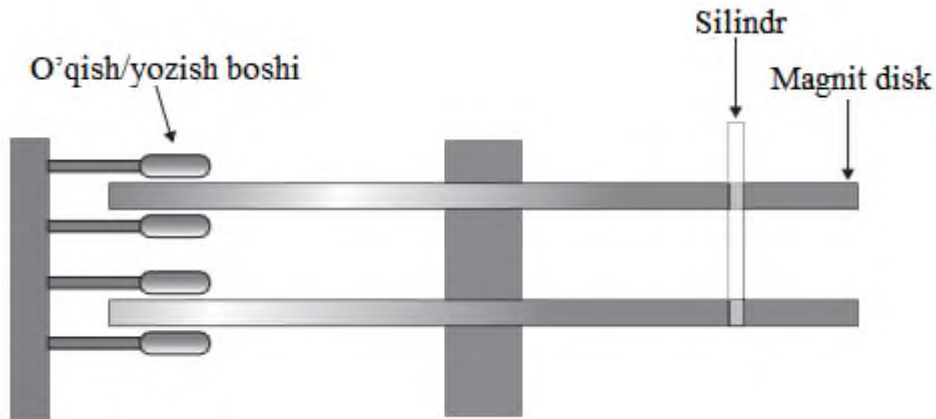
bitta o'qish/yoziq boshi mavjud. Disklar alohida disk bo'lishi yoki magnet plastinkalar to'plami bo'lgan disk paketining bir qismi bo'lishi mumkin. 4.8- rasmda odatiy disk to'plami ko'rsatilgan - umumiy markaziy dastakga (shpindelga) o'rnatilgan bir nechta plastinkalar, har bir juft disk o'rtasida o'qish/yoziq boshlarini harakatlantirish uchun yetarli bo'sh joy ajratilgan. 4.8- rasmda ko'rsatilgandek, har bir disk (dastakda yuqorida va pastda joylashgan-lardan tashqari) ikkita yoziq sirtiga (yuzaga) ega va har bir sirt ma'lum miqdordagi konsentrik yo'laklar bilan formatlanadi, ularga ma'lumotlar yoziqadi. Yo'laklar soni ishlab chiqaruvchidan ishlab chiqaruvchiga farq qiladi, ammo qoida tariqasida yuqori quvvatli qattiq diskda minglab yoki undan ko'p yo'laklar mavjud. Har bir sirdagi har bir yo'lak raqamlangan: 0 yo'lak har bir sirdagi tashqi konsentrik aylanani belgilaydi; eng yuqori raqamli yo'lak markazda joylashgan.



4.7- rasm. To'rtta o'qish/yoziq boshli, belgilangan boshli magnet disk

4.9- rasmda ko'rsatilgan dastak har bir juft sirt o'rtasida ikkita o'qish/yoziq boshini harakatlantiradi: biri yuqoridagi sirt uchun va ikkinchisi pastdagi sirt uchun. Dastak barcha boshlarni birgalikda harakatlantiradi, shuning uchun agar bitta bosh 36 yo'lakda bo'lsa, boshqacha aytganda unda barcha boshlar ham 36 yo'lakda joylashgan bo'ladi. Lekin, o'zlarining virtual silindrni yaratib o'zlarining tegishli

sirtlarida joylashadi. Ba'zi qiziqarli savollar tug'iladi: birinchi yuzada bir qator yozuvlarni yozish samaraliroq bo'ladimi va bu sirt to'lganidan keyin ikkinchi sirtda, so'ngra uchinchi sirtda yozishni davom ettirish mumkinmi? Yoki yozishni davom ettirish uchun boshning ichkarisiga o'tishdan oldin har bir sirtning har bir yo'lagini to'ldirish yaxshiroqmi?



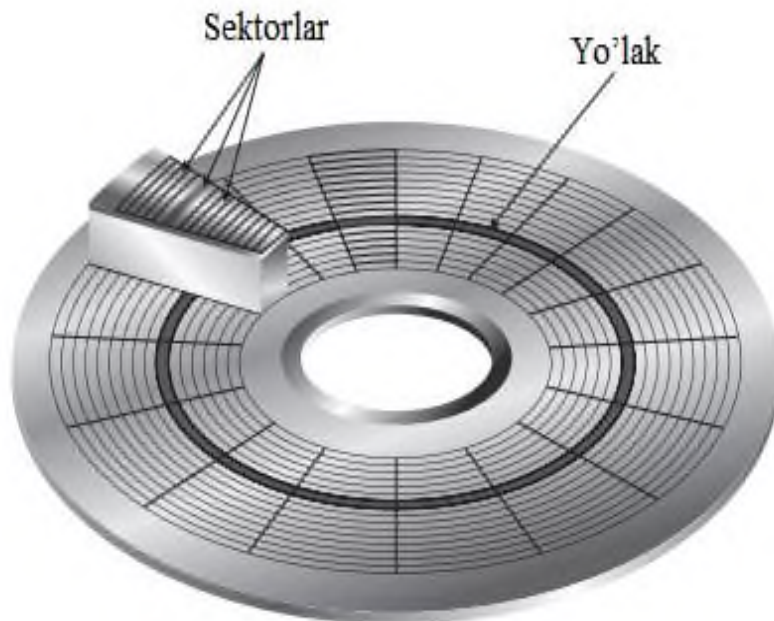
4.8- rasm. Disk magnit plastinalar to'plamidir. O'qish/yozish boshlari har bir juft sirt o'rtasida harakatlanadi va barcha boshlar bir-biriga birlashtirilib, dastak tomonidan harakatga keltiriladi



4.9- rasm. SHK standart qattiq diski, dastak har bir juft sirt o'rtasida ikkita o'qish/yozish boshini harakatlantiradi

Disk to'plamini yo'lakma-yo'lak to'ldirishdan ko'ra sirtan sirtni to'ldirish sekinroq va bu bizni qimmatli tushunchaga olib keladi. Agar biz barcha 0 yo'laklar sirtlarini to'ldirsak, biz virtual ma'lumotlar silindrini olamiz. Yo'laklar qancha bo'lsa, shuncha

silindr mavjud va silindrlar disklar to‘plami kabi uzun. Istalgan har qanday yozuvga kirish uchun, tizimga uchta narsa kerak: silindr raqami, dastak o‘qish/yozish boshini harakatlantirishi uchun; uning yuzasi soni, shuning uchun o‘qish/yozish uchun mos keladigan bosh faollashtirilgan; va 4.10-rasmda ko‘rsatilgandek uning sektor raqami, shuning uchun o‘qish/yozish boshi o‘qish yoki yozishni boshlash vaqtini biladi.



4.10- rasm. Magnit diskdagi bloklar har xil o‘lchamga ega: markazga yaqinrog‘i kamroq va markazdan uzoqrog‘i ko‘proq. Ushbu farqni qoplash uchun disk doimiy burchak tezligida (CAV) aylanadi. Ba’zi optik disklar bir necha qatlamlarda o‘qish va yozish imkoniyatiga ega, bu xotira hajmini sezilarli darajada oshiradi.

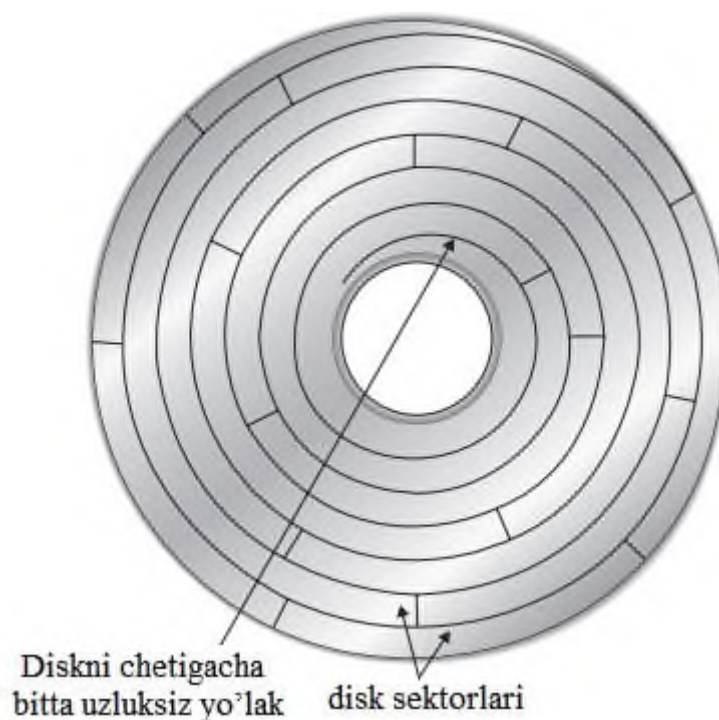
Bitta aniqlik: biz ushbu munozarada sirt atamasini ishlatdik, chunki bu konsepsiyalarni tushunishni osonlashtiradi. Biroq, oddiy adabiyotlarda "yo‘lak" atamasi odatda sirtni va konsentrik yo‘lakni aniqlash uchun ishlatiladi. Shuning uchun bizning sirt/yo‘lakdan foydalanishimiz boshqa ba’zi matnlarda ishlatiladigan yo‘lak yoki bosh atamasi bilan bir xil.

### ***Optik disk qurilmasi***

Optik diskarning paydo bo‘lishi lazer texnologiyalari sohasidagi o‘zgarishlar tufayli mumkin bo‘ldi. Optik disk va magnit disk

o'rtasidagi farqlar orasida disk yo'laklari va sektorlarining dizayni mavjud. Sektorlarning konsentrik yo'laklaridan tashkil topgan magnit disk doimiy tezlikda aylanadi - bu doimiy burchak tezligi (CAV - constant angular velocity) deb nomlanadi. Diskning tashqi tomonidagi sektorlar, ichki tomonidagi o'qish/yozish boshlariga qaraganda tezroq aylanishi sababli, tashqi sektorlar disk markaziga yaqin joylashgan sektorlarga qaraganda ancha katta. Ushbu format saqlash joyini isrof qiladi, ammo ma'lumotlarni olish tezligini maksimal darajada oshiradi.

Boshqa tomondan, optik disk 4.11-rasmda ko'rsatilgandek, markazdan disk chetigacha cho'zilgan bir xil o'lchamdagi sektorlarning bitta spiral yo'lakidan iborat. Ushbu alohida yo'lak ham sektorlarga ega, ammo diskdagi joylashuvidan qat'iy nazar barcha sektorlar bir xil hajmda bo'ladi.



4.11- rasm. Optik diskda sektorlar (hamma sektorlar bu yerda ko'rsatilmagan) butun disk bo'ylab bir xil hajmga ega. Diskovod kompensatsiya qilish uchun tezlikni o'zgartiradi, lekin doimiy chiziqli tezlikda (CLV) aylanadi

Ushbu dizayn sizga optik diskka bir xil o'lchamdagi magnit disk bilan taqqoslaganda ko'proq sektorlarni va ko'proq ma'lumotlarni joylashtirish imkonini beradi. Diskovod sektorning diskdagi o'rnini



qoplash uchun diskning aylanish tezligini sozlaydi - bu doimiy chiziqli tezlik (CLV- constant linear velocity) deb nomlanadi. Shuning uchun, diskning markazida joylashgan sektorlarni o'qish uchun disk tezroq va tashqi chetiga yaqin sektorlarni o'qish uchun sekinroq aylanadi. Agar siz diskovodni amalda tinglayotgan bo'lsangiz, u ushbu sozlamalarni o'zgartirganda, u tezligini qanday o'zgartirayotganini eshitishingiz mumkin.

Optik disk diskovodi ishlashining eng muhim ikkita ko'rsatkichi ma'lumot uzatish tezligi va o'rtacha kirish vaqti hisoblanadi. Ma'lumot uzatish tezligi sekundiga megabaytlarda o'lchanadi va diskdan katta hajmdagi ma'lumotlarni o'qish mumkin bo'lgan tezlikni anglatadi. Ushbu omil ketma-ket kirishni talab qiladigan dasturlar, masalan, audio va videoni ishga tushurish uchun juda muhimdir.

Masalan, tezkor ma'lumot uzatish tezligiga ega DVD, videofaylni sekinroq uzatish tezligiga ega bo'lgan qurilmaga qaraganda, yozilgan video segmentini ishga tushurishda kamroq kadrlar qisqarishiga olib keladi. Bu yanada yumshoqroq tasvirni yaratadi.

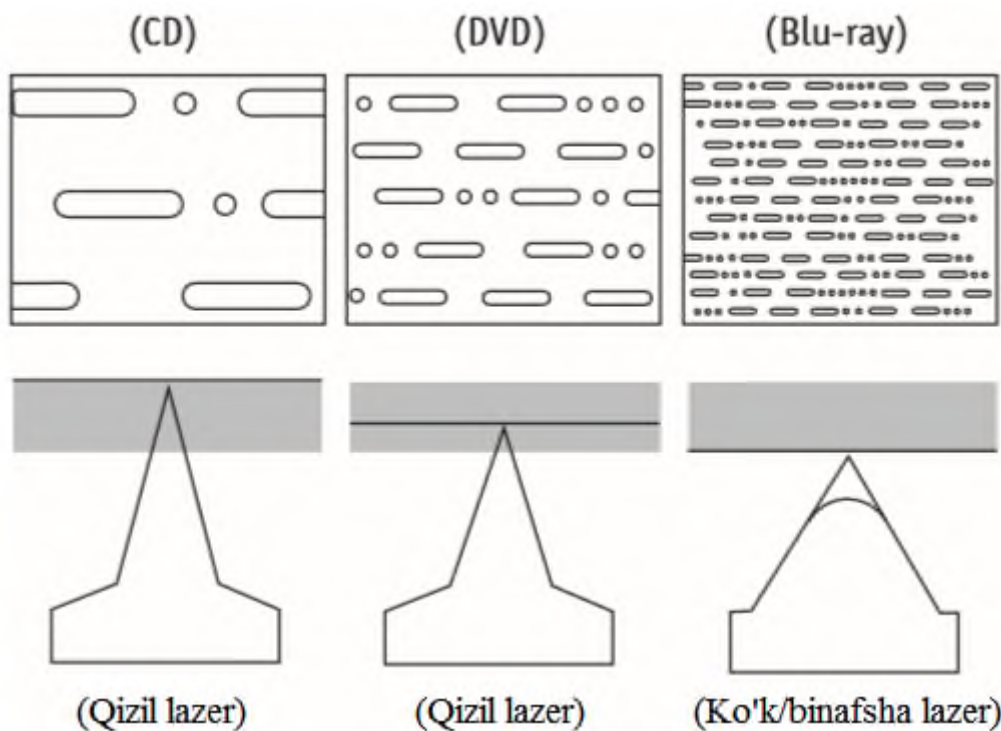
Ammo ketma-ket saqlanmagan ma'lumotlarni olish uchun diskovodga kirish vaqti muhimroq bo'lishi mumkin. Boshni diskdagi muayyan joyga o'tkazish uchun talab qilinadigan o'rtacha vaqtni ko'rsatadigan kirish vaqti millisekundlarda (ms) ko'rsatilgan. Eng tezkor qurilmalar eng kichik o'rtacha kirish vaqtiga ega, bu tasodifiy ma'lumotlarni qidirishda eng muhim omil hisoblanadi, masalan, ma'lumotlar bazasida. Shuning uchun, tezkor ma'lumot uzatish tezligi, diskga ketma-ket kirish uchun, masalan, videoni ishga tushurishda muhimdir, ammo tezkor kirish vaqti diskda keng tarqalgan ma'lumotlarni olishda juda muhimdir.

Optik diskovodlarning uchinchi muhim xususiyati kesh hajmi. Garchi bu tezlikni o'lchash bo'lmasa ham, kesh hajmi qabul qilinadigan ish samaradorligiga sezilarli ta'sir qiladi. Qurilma keshi bufer vazifasini bajaradi, ma'lumot bloklarini diskga uzatadi, foydalanuvchi yaqinda olingan ba'zi ma'lumotlarni qayta o'qishni xohlashini kutadi, agar ma'lumot keshda qolsa tezda bajarilishi mumkin. Ba'zi hollarda, kesh diskdagi keyingi ma'lumot blokini qidirib, oldindan o'qish uchun bufer vazifasini bajarishi mumkin. Oldindan o'qish keshlari uzluksiz ma'lumotlar oqimi mavjud bo'lganda, multimediyani ishga tushirish uchun eng foydali bo'lishi

mumkin. Biroq, ular tezda to'lishi sababli, oldindan o'qish keshlari ma'lumotlar bazasini yoki elektron kitoblarni ko'rib chiqishda foydali bo'lishi mumkin. Bunday hollarda, kesh foydalanuvchi joriy ma'lumotni o'qiyotganda tiklanishni amalga oshiradi.

Disklarning tarqatgichi va hajmiga qarab optik disk tizimlarining bir necha turlari mavjud: 4.12- rasmda ko'rsatilgandek CD, DVD va Blu-ray.

Ma'lumotlarni optik diskka joylashtirish uchun yuqori intensivlikdagi lazer nurlari diskdagi chuqurchalarni kuydiradi. 0 larni ifodalovchi bu *chuqurchalar*, kuydirilayotgan tekis maydonlardan farqli o'laroq, 1 larni ifodalaydigan maydonlar *yerlar* deb nomlanadi. Birinchi sektorlar diskning o'rtasida joylashgan bo'lib, lazer har bir sektorni navbat bilan o'qiydi. Agar diskda bir nechta qatlam bo'lsa, ikkinchi qatlamni o'qish uchun tashqi lazer kursi tashqi chetidan ichki qismga o'tganda, lazer kursi ikkinchi qatlamni o'qish uchun o'zgartiradi.



4.12- rasm. Optik diskni o'qish va ularga yozish texnologiyasi

Izoh: CD o'qish qurilmasi texnologiyasi (chapda) disk pastki qatlamining pastki qismiga yozish uchun qizil lazerdan foydalanadi. DVDlar (o'rtada) kichikroq qizil lazerdan foydalanib diskning pastki

qismiga yozib oladi, shunda yo‘laklarni oldindan belgilash mumkin. Blu-ray (o‘ngda) zichroq yo‘laklar bilan bir necha qatlamlarga yozib olish uchun ingichka ko‘k lazerdan foydalanadi (blu-raydisc.com saytidan).

### ***CD va DVD texnologiyasi***

CD yoki DVD pleyerda ma’lumotlar past quvvatli qizil lazer yordamida fokuslanib, ma’lumotlar qayta o‘qiladi, u diskning himoya qatlamidan ma’lumot yoziladigan CD-disk yo‘lakiga (yoki DVD-yo‘lakga) o‘tadi. Yerdagi yengil hodisa fotodetektorda aks etadi va chuqurdagi yorug‘lik hodisasi tarqab ketadi va so‘riladi. Keyin fotodetektor yorug‘lik intensivligini 1 va 0 raqamli signalga aylantiradi. CD va DVD yozadigan diskovodlar faqat o‘qish uchun mo‘ljallangan disk pleyerlariga qaraganda qimmatroq disklarni boshqarish qurilmalarini talab qiladi, chunki ular har bir tashuvchiga xos bo‘lgan yozish mexanizmlarini o‘z ichiga olishi kerak. Masalan, kompakt-disk bir necha qatlamlardan iborat bo‘lib, ularda oltinni aks ettiruvchi qatlam va bo‘yoq qatlami mavjud bo‘lib, u ma’lumotlarni yozib olish uchun ishlatiladi. Yozib olish boshi ma’lumotlarni yozish uchun kuchli lazer nuridan foydalanadi. Lazer nurining energiyasi so‘rilganida va unga yozilganidan keyin unio‘chirib bo‘lmaydi, bo‘yoq ustiga doimiy belgi qo‘yiladi. U o‘qilganda, bo‘yoqdagi izning mavjudligi lazer nurining tarqalishiga olib keladi va yorug‘lik o‘qilgan joyga qaytarilmaydi. Biroq, bo‘yoq izlari yo‘q bo‘lganda, oltin qatlam yorug‘likni o‘qilgan tomonga qaytaradi. Bu chuqurlar va yerlarni o‘qish jarayoniga o‘xshaydi.

Yoziladigan kompakt diskni (CD-R) yaratish uchun ishlatiladigan dasturiy ta’minot ISO 9096 kabi standart formatdan foydalanadi, bu avtomatik ravishda xatolarni tekshiradi va har bir faylning joylashishini kuzatish uchun ishlatiladigan tarkibiy jadvalni yaratadi. Ma’lumotni yozish uchun lazer nurlari diskni isitadi va uning holatini kristalldan amorf holatga o‘zgartiradi. Ma’lumotlar kam quvvatli lazer nuri yordamida o‘qilganda, amorf holat o‘qni sezmaydigan yorug‘likni tarqatadi. Bu 0 deb talqin qilinadi va chuqurlarni o‘qiyotganda nima sodir bo‘lishiga o‘xshash. Boshqa tomondan, yorug‘lik kristalli mintaqalarga kirganda, yorug‘lik o‘qilgan boshga qaytariladi va bu o‘qish paytida sodir bo‘ladigan narsaga o‘xshaydi va 1 deb talqin qilinadi. Ma’lumotni yo‘q qilish

uchun CD-RW drayveri qotishmani zaiflashtirish va uni asl kristal holatiga qaytarish uchun yetarlicha chuqurlarni isitish uchun past energiyali nurni ishlatadi. DVD-disklar kompakt-disklar bilan bir xil dizayn va hajmga ega bo'lsa-da, ular ko'proq ma'lumotlarni saqlashlari mumkin. Ikki qatlamli bir tomonlama DVD 13 ta kompakt-diskga teng bo'lishi mumkin; uning qizil lazeri CD ning qizil lazeriga qaraganda qisqa to'lqin uzunligi bilan kichik chuqurchalarni hosil qiladi. Siqish texnologiyasining afzalliklari (keyingi bobda muhokama qilinadi) MPEG video-kompressiyasi kabi yuqori sig'imli DVD-larga qo'shilganda, bir tomonlama ikki qavatli DVD-disk 8,6 GB hajmga ega bo'lishi mumkin, bu kengaytirilgan audio bilan ikki soatlik film uchun yetarlidir.

### ***Blu-ray Disk texnologiyasi***

Blu-ray disk DVD yoki CD bilan bir xil fizik hajmga ega, ammo ma'lumotlarni o'qish va yozish uchun ishlatiladigan lazer texnologiyasi butunlay boshqacha. 4.12-rasmda ko'rsatilgandek, Blu-ray diskidagi chuqurlar (har biri 1 tadan) kichikroq va yo'laklar DVD yoki CD-ga qaraganda ancha zichroq. Eski CD va DVD-lar bilan ishlash uchun Blu-ray mahsulotlarini orqaga qarab moslashtirish mumkin bo'lsa-da, ko'k-binafsharang Blu-ray lazer (405 nm), qizil CD/DVD lazeridan (650 nm) ko'ra qisqa to'lqin uzunligiga ega. Bu sizga ma'lumotlarni yanada zichroq yig'ish va kichikroq hajmda saqlash imkonini beradi. Bunga qo'shimcha ravishda, ko'k-binafsha rangli lazer diskdagi ingichka qatlamga yozishi mumkin, bu bir-birining ustiga bir nechta qatlamlarni yozib olish imkonini beradi va bitta diskda saqlanishi mumkin bo'lgan ma'lumotlar miqdorini sezilarli darajada oshiradi. Disk formati yuqori aniqlikdagi videoning tijorat istiqbollarini kengaytirish va katta hajmdagi ma'lumotlarni, ayniqsa Java dasturlash tilida o'yinlar va interfaol dasturlarni saqlash uchun yaratilgan. Blu-ray pleyerlari menyular va foydalanuvchilarning o'zaro aloqalari uchun Java dasturlarini bajaradi. Bu yozish vaqtida har bir Blu-ray diskida, unga o'xshash DVDga (ikki qatlamli disk uchun 8,5 Gb) nisbatan ko'proq ma'lumot bo'lishi mumkin (ikki qatlamli disk uchun 50 Gb). (Pioneer Electronics kompaniyasi yangi 20 qatlamli diskleri 500 Gb ma'lumotni sig'dira oladi. CD va DVD singari, Blu-ray disklarining bir necha formati mavjud: faqat o'qish

uchun (BD-ROM), yozish uchun (BD-R) va qayta yoziladigan (BD-RE).

### ***Flesh xotira***

Flash xotira bu faqat elektr yordamida o‘chiriladigan, dasturlanadigan doimiy xotira (EEPROM). Bu operativ xotirani emulyatsiya qiladigan, operativ xotiradan farqli o‘laroq, hatto quvvat manbai bo‘lmasa ham, ma’lumotlarni ishonchli saqlaydi. Tarixan, flesh-xotira asosan kompyuterlar uchun boshlang‘ich ma’lumotlarini saqlash uchun ishlatilgan, ammo hozirda uyali telefonlar, mobil qurilmalar, musiqa pleyerlari va boshqalar uchun ma’lumotlarni saqlash uchun ishlatiladi. Flash xotira hodisalardan foydalanadi (Fowler-Nordheim tunneli deb nomlanadi). Elektronlarni suzuvchi zatvor tranzistorlari orqali yuborish uchun ishlatiladi, ular elektr o‘chirilganidan keyin ham qoladi. Flash xotira foydalanuvchilarga ma’lumotlarni saqlashga imkon beradi. U turli xil konfiguratsiyalarda, jumladan ixcham flesh-kartalarda, smart-kartalarda va xotira kartalarida sotiladi va ular ko‘pincha kompyuterga USB port orqali ulanadi. Flash xotira nomini o‘z ma’lumotlarni o‘chirish uchun ishlatiladigan texnikadan oldi. Ma’lumotni yozish uchun elektr zaryad suzuvchi zatvor deb nomlangan bitta tranzistor orqali, so‘ngra metall oksidi qatlami orqali va ikkinchi tranzistor orqali boshqariladigan zatvor orqali yuboriladi, u yerda zaryad yacheykada o‘chirilguncha saqlanadi. Barcha qiymatlarni tiklash uchun, butun xaritada miltillovchi deb nomlangan kuchli elektr maydoni qo‘llaniladi. Biroq, flesh-xotira buzilmaydi. Ikkita cheklov mavjud: bitlarni faqat katta hajmli xotira blokiga yoritish orqali o‘chirish mumkin va har bir o‘chirish bilan blok kamroq barqaror bo‘ladi. Vaqt o‘tishi bilan (10,000-1,000,000,000 ishlatilgandan keyin), flesh xotira endi ma’lumotlarni ishonchli saqlamaydi.

## **4.5. Qurilmalarni boshqarish drayveri va dasturiy ta’minoti**

### ***Qurilmalar dasturiy interfeysi***

O‘z ichiga kiritish/chiqarish protseduralarni qamrab oluvchi amaliy dasturlar ishlab chiqishni osonlashtirish uchun OT xususiyatidan qat’iy nazar, barcha qurilmalarda kiritish/chiqarish bazaviy amallarning umumiy to‘plamidan foydalanish imkoniyatini

beruvchi periferik qurilmalar va ilovalar o'rtasidagi ekranlashtirilgan mantiqiy interfeysni qo'llab-quvvatlashi kerak. Bunday interfeysning asosi sifatida deyarli barcha operatsion tizimlar kiritish/chiqarish qurilmaning faylli modelini qo'llab-quvvatlaydi. Bu yerda biz yana bir marta virtualizatsiyaning mahsuldor konsepsiyasi bilan to'qnashamiz. Real kiritish/chiqarish qurilmalarning barcha ko'p sonli turlarini operatsion tizim bitta virtual turdagi qurilma bilan almashtiradi. Barcha virtual qurilmalar yagona tarzda ishlaydi, maxsus yoki virtual fayllar deb nomlanadigan ko'rinishda shakllanadi. Har bir kiritish/chiqarish qurilmasiga alohida maxsus fayl bog'lanadi. U bu qurilmani amaliy jarayonlarga va operatsion tizimning qolgan qismiga baytlarning stukturalanmagan to'plami sifatida taqdim etadi. Natijada real qurilmalar uchun kiritish/chiqarishning murakkab protseduralarini yozishni o'rniga, dasturchi endi oddiygina mazkur qurilmalar bilan bog'langan maxsus fayllardan o'qish va maxsus fayllarga yozish amallardan foydalanish mumkin. Fayl-qurilma modelining jozibadorligi uning barcha turdagi qurilmalar uchun soddaligi va universalligidadir. Lekin ko'p hollarda, masalan, grafik ma'lumotni displey yoki printeriga chiqarish, tarmoq almashinuvi amallarni dasturlash va boshqalar, bu model juda sodda va amaliyotchi dasturchini ushbu qurilmalar uchun kiritish/chiqarishni dasturlashning odatdagi vazifalaridan ozod qilmaydi. Shuning uchun ayrim turdagi qurilmalar uchun OT ularning xususiyatlarini aks ettiruvchi yanada rivojlangan interfeysni qo'llab-quvvatlaydi.

### ***Qurilma drayverlari***

Kompyuterga ulangan har bir kiritish/chiqarish qurilmasini boshqarish uchun maxsus dasturlar talab etiladi.

Bu dasturlar qurilma drayveri deyiladi, ko'pincha qurilma ishlab chiqaruvchisi tomonidan yaratiladi, qurilma bilan birgalikda kompakt diskda tarqatiladi. Har bir operatsion tizim uchun maxsus drayverlar talab qilinar ekan, odatda ishlab chiqaruvchilar bir nechta keng tarqalgan operatsion tizimlar uchun drayverlarni joylashtiradilar.

Har bir drayver bir turdagi qurilmaga yoki bir biriga o'xshash ko'plab qurilmalarga kiradilar. Masalan, bitta sichqoncha drayveriga qaramasdan tizim bir nechta turdagi sichqonchalarni qo'llab-quvvatlaydi. Disk o'quvchi qurilma drayverlari bir necha turdagi

diskli, turli hajmli va tezlikni qo'llab-quvvatlashi, shuningdek kompaktdiskli imkoniyatli bo'lishi mumkin.

Drayver qurilma apparat qismiga kirish imkoniyatini beradi, unda kontroller registri mavjud, o'nta operatsion tizim yadrosida integratsiyalanadi. Bunday usul yuqori samaradorlikni ta'minlaydi. Lekin past ishonchlikga ega bo'ladi, sababi, istalgan qurilma drayveridagi xato butun bir tizimni ishini izdan chiqaradi.

Operatsion tizim nuqtai nazaridan qaralganda drayverlar blokli (masalan, diskli) va belgili (masalan, klaviatura va printerlar) qurilmalari uchun bo'ladi. Ko'plab operatsion tizim kompyuterga tegishli bo'lgan barcha blokli va barcha belgili qurilmalarni qo'llab-quvvatlashi kerak bo'lgan ikkita standart interfeys aniqlaniladi. Interfeyslar o'zining ishini bajarilishini drayver orqali ta'minlaydigan operatsion tizimlarda chaqiriladigan protseduralar majmuini o'z ichiga oladi.

Drayverning vazifasi yuqori pogona apparatdan mustaqil dastur mavhum so'rovlarini qabul qilish va so'rov bajarilganligi to'g'risida xabar berish. Odatiy so'rov diskdan drayverga kelib tushadi - berilgan ma'lumotlar bloki deb hisoblaniladi. Shu sababli agar so'rovni uzatish vaqtida harakatsiz bo'lsa, u darrov ishlashni boshlaydi. Agar drayver band bo'lsa, so'rov odatda navbatga qo'yiladi va unga imkoniyat bo'lganda xizmat ko'rsatiladi.

Kiritish/chiqarish so'roviga xizmat ko'rsatishda birinchi qadam uzatilgan parametrlarni to'g'riligini tekshirish va xatolarni to'g'irlash bo'ladi. Agar so'rov to'g'ri bo'lsa keyingi qadam mavhum taqdim etilgan shakldan aniq bir shaklga o'tkaziladi.

Shundan so'ng belgilangan buyruqlarni bajarishda drayver kontroller registri orqali uning qurilmasiga uzatishni boshlaydi. Sodda kontrollerlar bitta buyruq bo'yicha bir marta qabul qiladi, nisbatan murakkab buyruqlar ro'yxati bilan bog'liqlikni qo'llab-quvvatlash operatsion tizimning ishtirokisiz bajariladi.

Barcha buyruqlar uzatilgandan so'ng holat ikki senariydan bittaga o'tadi. Qurilma drayverlari ko'p holatlarda uning uchun belgilangan ish bajarilmagunga qadar kutadilar, shu sababli qurilmadan to'xtalishlar kelib tushgunga qadar bloklash holatida bo'ladi. Boshqa bir variantda operatsiya kutishsiz yakunlanadi va drayverni bloklashga zarurat bo'lmaydi.

Agar drayver bloklansa unda to'xtalish bo'lganda blokdan ozod etiladi. Boshqa bir holatda drayver bloklanmaydi, umuman bloklanmaydi.

Drayver operatsiyasini yakunlash bo'yicha operatsiya xatosiz o'tganligiga ishonch hosil qilishi kerak. Agar bari to'g'ri bo'lsa drayverga qurilma dasturiy ta'minotidan mustaqil ravishda ma'lumotlarni uzatish mumkin bo'ladi (faqat o'qilgan bloklar). Va nihoyat drayver operatsiya yakunlanganligi to'g'risida ma'lumot uchun chaqiruvchi dastur ayrim axborotlariga murojaat qilinadi. Agar navbatda boshqa so'rov turgan bo'lsa, ulardan biri tanlaniladi va ishga tushiriladi, boshqa so'z bilan aytganda drayver keyingi so'rovni kutishga bloklanadi.

### ***Drayverlarning keng spektrini qo'llab-quvvatlash***

Drayver ikki turdagi interfeyslarni qo'llab-quvvatlashi kerak:

❖ OT yadrosi modullari bilan (kiritish/chiqarish kichik tizimi, tizim so'rovlari, jarayon va xotirani boshqaruvchi kichik tizimlar modullari va hokazo) "drayver-yadro" interfeysi (Driver Kernel Interface, DKI);

❖ tashqi qurilmalar kontrollerlari bilan "drayver-qurilma" interfeysi (Driver Device Interface, DDI).

"Drayver-yadro" interfeysi barcha hollarda standartlashtirilgan bo'lishi kerak, "drayver-qurilma" interfeysni esa kiritish/chiqarish kichik tizimi drayverga kontrollerning apparaturasi bilan bevosita o'zaro harakat qilishga ruxsat bermagan holda va bu amallarni mustaqil bajarganda standartlashtirish kerak bo'ladi. Drayverni apparaturadan ekranlashtirish juda foydali funksiya hisoblanadi, chunki bu vaziyatda drayver apparatura platformasidan mustaqil bo'lib qoladi. Kiritish/chiqarish kichik tizimi ma'lum klassdagi qurilmalarga maxsus interfeys taqdim etib, bir necha turdagi interfeyslarni DKI/DDI qo'llab-quvvatlashi mumkin. Masalan, Windows NT oilasidagi OT larda tarmoq adapteri drayverlari uchun NDIS (Network Driver Interface Specification) standartli interfeys nazarda tutilgan, lekin tarmoq transport protokollarining drayverlari tarmoq dasturiy ta'minotning yuqori qatlamlari bilan TDI (Transport Driver Interface) interfeysi orqali o'zaro harakat qiladi. Operatsion tizim drayverlarini ishlab chiqish jarayonini qo'llab-quvvatlash uchun odatda tegishli instrumental vositalar (kutubxonalar, kompilyator va sozlovchilar) to'plamidan iborat DDK (Driver Development Kit) drayverlar to'plami ishlab chiqariladi.

### ***Drayverlarni dinamik yuklash va tizimdan chiqarish***



Mazkur OT tomonidan qo'llab-quvvatlashga imkoni bor yordamchi (periferik) qurilmalar doimo OT boshqaradigan aniq mashinada o'rnatiladigan qurilmalar to'plamidan sezilarli darajada kengroq bo'ladi. Shuning uchun OT qimmatli xususiyati, ya'ni operativ xotiraga kerakli drayverni dinamik ravishda yuklash (OT to'xtamagan holda) va mazkur qurilmani qo'llab-quvvatlash ehtiyoji qolmagandan keyin uni tizimdan chiqarish, xotira tizimi maydonini sezilarli darajada tejashi mumkin. Drayverlarni dinamik yuklashni qo'llab-quvvatlash deyarli barcha zamonaviy universal operatsion tizimlarga qo'yilgan talab hisoblanadi. Kompyuterning tashqi qurilmalarning joriy konfiguratsiyasi o'zgargan holda drayverlarni dinamik yuklashning muqobil yo'li deb, kerakli drayverlar to'plami joylashgan yadro kodini qayta kompilyatsiya qilish hisoblanadi. Bunda yadroning barcha qismlari o'rtasida statik bog'lanishlar o'rniga dinamik bog'lanishlar yaratiladi. Masalan, shunday yo'l bilan mazkur muammo Unix operatsion tizimining avvalgi versiyalarda hal etilgan. Yadro va drayverlar o'rtasidagi statik bog'lanishlar OT tuzilmasini soddalashtiradi, lekin bu yondashuv operatsion tizim modullari boshlang'ich kodlarini mavjudligini talab qiladi; ulardan foydalanish esa ko'pincha mustasno (Unix/Linux ning notijorat versiyalari uchun) emas, balki qoidadir. Shuningdek, bu variantda operatsion tizimning avvalgi ishlab turgan versiyasini to'xtatish va uni yangi bilan almashtirish kerak, OT ishlashidagi tanaffuslar ayrim hollarda yo'l qo'yilmasligi mumkin.

#### **4.6. Ma'lumotlarni kiritish/chiqarish menejeri va dasturiy ta'minoti**

##### ***Fayl tizimlarni qo'llab-quvvatlash***

Disklar periferik qurilmalarining alohida turi hisoblanadi, chunki aynan ularda ham foydalanuvchi, ham tizim ma'lumotlarining aksariyat qismi saqlanadi. Disklardagi ma'lumotlar fayl tizimlarda joylashtiriladi, va fayl tizimi xususiyatlari asosan OT o'zining xususiyatlari bilan aniqlanadi. Fayl tizimining ommalash-ganligi ko'pincha uni "o'zining" operatsion tizimidan boshqa operatsion tizimlarga ko'chirilishiga olib keladi, masalan FAT fayl tizimi boshida MS-DOS da, keiynchalik esa OS/2, MS Windows tizimlarida va Unix

tizimining ko'p variantlarida qo'llanilgan. Shuning uchun kiritish/chiqarish kichik tizimiga bir nechta ommaviy fayl tizimlarini qo'llab-quvvatlash xuddi periferik qurilmalarining keng spektrini qo'llab-quvvatlashdek muhim. Kiritish/chiqarish kichik tizimi arxitekturasini uning tarkibiga osongina, kodlarni qayta yozishga zarurat bo'lmasdan, yangi fayl tizimlarni kiritish imkoniyati ham muhim hisoblanadi. Odatda operatsion tizimda mazkur vazifani yechishga javob beruvchi dasturiy ta'minotining maxsus qatlami bo'ladi, masalan, Unix operatsion tizimi versiyalarida VFS (Virtual File System) kodi asosida System V Release 4 qatlami.

### ***Kiritish/chiqarish menejeri***

Kiritish/chiqarish kichik tizimida tashqi qurilmalarning xususiyatlari va vertikal kichik tizimlarni tashkil etuvchi modullar bilan bir qatorda universal turdagi modullar ham mavjud. Bu modullar kiritish/chiqarish kichik tizimining barcha boshqa komponentlarining o'zaro ishlashini va foydalanuvchi jarayonlari hamda OT ning boshqa kichik tizimlari bilan o'zaro munosabatlarini tashkil etadi. Xuddi qurilmalarni boshqarish funksiyalariga o'xshab, bu tashkillashtiruvchi funksiyalar barcha darajalar bo'yicha taqsimlanadi va kiritish/chiqarish menejeri nomli qobiqni tashkil etadi. Kiritish/chiqarish menejerning vazifalari to'rtta interfeysni qo'llab-quvvatlashdan iborat:

- ❖ Foydalanuvchi kiritish/chiqarish interfeysi;
- ❖ Kiritish/chiqarish qurilmalari bilan interfeys;
- ❖ OT ning boshqa kichik tizimlari bilan interfeys;
- ❖ Kiritish/chiqarish kichik tizimning qismlari bilan ichki interfeys.

Menejerning yuqori qatlamini foydalanuvchi jarayonlardan tizimli ko'rinishda keladigan kiritish/chiqarishga so'rovlarni qabul qiladigan va ularni ma'lum qurilmalarga javob beruvchi modul va drayverlarga qayta manzillaydigan hamda kiritish/chiqarish amallarining natijalarini jarayonlarga qaytaradigan OT ning modullari tashkil etadi. Shunday qilib bu qatlam kiritish/chiqarish foydalanuvchi interfeysni qo'llab-quvvatlaydi va amaliyotchi dasturchilarga tashqi qurilmalar hamda ularda joylashgan ma'lumotlar bilan ishlashda maksimal qulayliklar yaratadi. Menejerning pastki qatlami drayverlarni kompyuterni apparat platformasining xususiyatlaridan

(kiritish/chiqarish shinalari, uzilishlar tizimi va boshqalar) ekranlashtirish yo‘li bilan tashqi qurilmalar kontrollerlari bilan bevosita o‘zaro harakatni amalga oshiradi. Bu qatlam kiritish/chiqarish shinalarning manzillari va formatidan mustaqil ravishda ma’lum umumiyashtirilgan shaklda drayverlardan kontrollerlar registrlari bilan ma’lumotlar almashinuviga so‘rovlar qabul qiladi, va keyin bu so‘rovlarni apparatli platformaga bog‘liq formatga aylantiradi. Avval ko‘rib chiqilgan to‘htatishlar dispetcheri kiritish/chiqarish menejning tarkibiga kirishi mumkin yoki yadroning alohida moduli shaklida bo‘lishi mumkin. Oxirgi vaziyatda kiritish/chiqarish menejeri to‘xtatishlar dispetcheri uchun dispetcherga so‘rov manbai to‘g‘risida umumlashtirilgan axborotlarni uzatib, uzilish so‘rovlarini boshlang‘ich ishlovini bajaradi. Menejning yana bir funksiyasi – bu kiritish/chiqarish kichik tizimi modullarini OT ning jarayonlarni boshqarish, virtual xotira va boshqa kichik tizimlari singari boshqa kichik tizimlari modullari bilan o‘zaro harakatini tashkillashtirish. Kiritish/chiqarish menejning muhim funksiyasi kiritish/chiqarish kichik tizimni qismlarining o‘zaro harakati uchun ma’lum qulay muhitni yaratishdan iborat. Buning uchun kiritish/chiqarish menejeriga kiritish/chiqarish modullari o‘zaro harakatini ma’lum standart ichki interfeysini qo‘llab-quvvatlash kiritiladi, hamda menejerga drayverlar ishlashida eng ko‘p foydalaniladigan funksiyalarni bajarish yuklatiladi. Bu funksiyalar drayver ayrim tez-tez uchraydigan amallarni bajarilishini chaqirish imkoniyatini beradigan tizimli protseduralar ko‘rinishda shakllantiriladi. Misollar tariqasida kontroller registrlari bilan almashinuvchi amallari, kiritish/chiqarish ma’lumotlarni oraliq saqlash buferlarni olib borish, bir nechta drayverlar ishlashini sinxronlashtirish, ma’lumotlarni foydalanuvchi fazodan tizimning fazosiga nusxalashni keltirish mumkin. Bunday yondashuv yangi drayverlar va faylli tizimlarni ishlab chiqish va ularni OT tarkibiga joylashtirishni sezilarli darajada yengillashtiradi. Kiritish/chiqarish menejeri vazifasini Unix operatsion tizimining ko‘p versiyalarida mavjud bo‘lgan STREAMS muhiti bajaradi. Boshqa misol tariqasida Windows NT oilasi OT larning kiritish/chiqarish menejerini keltirish mumkin. U modullar o‘rtasidagi o‘zaro harakatni kiritish/chiqarish so‘rovlar paketlari (I/O Request Packet, IRP) yordamida tashkillashtiradi. Tizimli chaqiruv protsedurasidan so‘rov olgach

menejer IRP ni shakllantiradi va uni kerakli drayverga uzatadi. Soʻralgan amal bajarilganidan keyin drayver javobni yana bir IRP paketi koʻrinishida menejerga qaytarida, u esa, oʻz navbatida zarurat boʻlgan holda shu IRP paketni boshqa drayverga uzatishi mumkin. Menejer drayverlarga oʻzaro bogʻlanishlar (biddings) berishga imkon yaratadi, va oʻzaro aloqalar toʻgʻrisidagi axborot asosida IRP paketlarni uzatishni amalga oshiriladi. Shuningdek, Windows NT oilasi OT ning menejeri tizimni toʻxtatmagan holda drayverlarni dinamik yuklash va chiqarishni qoʻllab-quvvatlaydi. Standart ichki modullararo interfeysning mavjudligi kiritish/chiqarish kichik tizimining chidamliligini oshiradi va kengayuvchanligini yaxshilaydi, ammo uning ishlash tezligini biroz sekinlashtirishi mumkin, chunki har bir qatlam va qismlarga boʻlinish boshqaruvni toʻgʻridan-toʻgʻri uzatishlar bilan mustahkamlashga nisbatan qoʻshimcha amallarga olib keladi.

### ***Kiritish/chiqarish dasturiy taʼminoti***

Kiritish/chiqarish dasturiy taʼminotini ishlab chiqish konsepsiyasi qurilmadan mustaqilligi bilan maʼlum. Bu konsepsiya aniq bir qurilmani koʻrsatmasdan turib istalgan kiritish/chiqarish qurilmasiga kirishga ruhsat olish yoʻllari yozilgan dastur imkoniyatlarini bildiradi. Kiruvchi fayldagi maʼlumotlarni oʻquvchi dastur disket, qattiq disk yoki kompakt-diskdagi fayllar bilan bir xil ishlay olishi kerak. Aynan nimaga dasturda hech qanday oʻzgarishlarsiz, masalan, quyidagiga oʻxshash buyruqlarni bajarish uchun kerak

sort <input >output

Bu buyruq kiritish qurilmalari sifati - yumshoq disk, IDEdisk, SCSIdisk yoki klaviaturalar bilan ishlay olishi kerak. Chiqarish qurilmasi sifatida xam ekran, diskdagi istalgan fayl yoki printer bilan bir xil ishlay olishi kerak. Asosiy muammo operatsion tizim yechishi kerak boʻlgan ushbu qurilmalar bilan bogʻlanishlarning farqli ekanligidadir. Fayl nomi yoki qurilma sodda matn qatoridan yoki butun qiymat va fizik qurilmaga bogʻliq boʻlmagan koʻrinishga ega boʻlishi kerak.

Kiritish/chiqarish qurilmasining boshqa bir muhim jihatlari xatolarni qayta ishlashida. Xatolar apparaturaga yaqin koʻrinishda qayta ishlanishi mumkin. Agar kontroller oʻqish jarayonida xatolikni

aniqlasa, u bu xatoni o'zi o'zgartirish imkoniyatlari asosida xarakat qilib ko'rishi kerak. Agar buni bajarishga uni kuchi yetmasa, u holda qurilma drayverda qayta ishlanishi kerak. Ko'pincha xatolar vaqtinchalik bo'lishi mumkin. Masalan, o'quvchi moslamadagi changni chiqishi tufayli xato o'qishi. Bunday xatolar blokni takroran o'qish jarayonida sodir bo'lmaydi. Faqat agar past pog'onada muammo paydo bo'lsa bu haqda yuqori pog'onaga axborot beradi. Ko'p holatlarda xatodan so'ng qayta tiklangandan yuqori pog'ona uchun aniq bo'lgan past pog'onada qilish afzal ko'riladi, unda mavjud uzilishlar haqida yuqori pog'ona xabardor bo'lmasligi ham mumkin bo'ladi.

Yana bir kalit so'z - ma'lumotlarni uzatish usuli: asinxrona (to'xtalishlarni boshqarish) teskari sinxronli (bloklovchi). Fizik pog'onada kiritish/chiqarish ko'plab operatsiyalari asinxronli (markaziy protsessor ma'lumotlar uzatishni boshlaydi va to'xtalish bo'lmagunga qadar unga e'tibor bermaydi) bo'ladi. Foydalanuvchi dasturlarini kiritish/chiqarish operatsiyalarini qo'llaydigan (chaqiriq tizimiga murojaat qilingandan so'ng receive dasturi buferda ma'lumotlar paydo bo'lmagunga qadar avtomatik to'xtatadi) dasturni yozish sezilarli darajada oson. Kiritish/chiqarish operatsiyasi uchun asinxronli, foydalanuvchi dasturi uchun esa operatsion tizimda ishlaydigan bloklovchi bo'ladi.

Kiritish/chiqarish dasturiy ta'minoti haqida gapirganda *buferlashga* e'tibor berish kerak bo'ladi. Ma'lumotlar ko'pincha qurilmadan kelib tushadi, u oxirgi nuqtasiga darrov saqlashning imkoniyati yo'q. Masalan, paket tarmoq orqali kelganda, operatsion tizim bu paket nimaga kerakligini o'rganib chiqmaguncha uni qayerga joylashtirishni bilmaydi. Bundan tashqari, real vaqtda ishlaydigan boshqa ko'plab qurilmalar uchun ma'lumotlarni kelib tushish muddatini ko'rsatuvchi parametrlar muhim hisoblaniladi (masalan, raqamli ovozni qayta tiklovchi qurilmalar), shu sababli qabul qilib olingan ma'lumotlar buferni to'ldirish tezligiga bog'liq bo'lmagan, ovoz chiqaruvchi buferdan chiqarib olishda tezlik uchun avvaldan chiquvchi buferda joylashtiriladi. Buferlashda sezilarli miqdorda ma'lumotlarni ko'chirishi nazarda tutiladi. Ko'pincha kiritish/chiqarish operatsiyasini samarodoligini pasaytiruvchi omil sifatida ko'riladi.

Bu tushuncha soʻngida ajratish qurilmasi va qismlarga boʻlish qurilmalari boʻladi. Diskga oʻxshash baʼzi kiritish/chiqarish qurilmalari koʻp sonli foydalanuvchilar bilan bir vaqtda ishlay olishi mumkin. Shu sababli bir nechta foydalanuvchilar bitta diskda bitta faylni ochganda muammo paydo boʻlmasligi kerak. Magnit tasmali tarqatgich kabi boshqa qurilmalar ushbu qurilmada u yakunlanmagunga qadar bir foydalanuvchilik boʻladi. Agar bir vaqtning oʻzida ikki va undan ortiq foydalanuvchilar bitta tasmaga aralash bloklarni yozishsa, yaxshi narsa olinmaydi. Ajratilgan (monopol foydalaniladigan) qurilmasi birgalikda bloklash kabi muammolarni keltirib chiqaradi. Operatsion tizim qismlarga ajratish va ajratilgan qurilmalarini boshqarish va turli xil muammolarni mustaqil yecha olishi kerak.

Bu muammolarni hal qilish uchun kiritish/chiqarish dasturiy taʼminotini toʻrtta pogʻonaga ajratiladi:

- ❖ Toʻxtalishlarni qayta ishlovchi (past pogʻona);
- ❖ Qurilma drayveri;
- ❖ Apparatura kodidan mustaqil operatsion tizimlar;
- ❖ Foydalanuvchi dasturlari (yuqori pogʻona).

### ***Toʻxtalishlarni qayta ishlovchi***

Baʼzida kiritish/chiqarish dasturlari koʻplab kiritish/chiqarish operatsiyalarida toʻxtalishlardan qochish uchun foydali boʻladi. Toʻxtalish berkitilgan boʻlishi kerak. Drayver down protsedura semaforasida, oʻzgaruvchan holat wait protsedurasiga, xabar yoki shunga oʻxshash receive protsedurasida bajarib oʻzini oʻzi bloklashi mumkin.

Toʻxtalish sodir boʻlganda toʻxtalishlarni qayta ishlovchi ishga tushadi. Uni yakunida u drayverni blokirovkadan chiqaradi. Ayrim holatlarda semaforada **up** protsedurasi orqali amalga oshiriladi. Boshqa holatlarda toʻxtalishlarni qayta ishlovchi holatni oʻzgatirishli signal monitorini chaqiradi. Yoki bloklangan drayverga xabar joʻnatadi. Istalgan holatda ham natija bir xil boʻladi - drayver blokdan chiqariladi va ishini davom ettiradi. Bu sxema xususiy holatlarda, stek va hisoblagich buyruqlari olganda drayverlarda juda yaxshi ishlaydi.

### ***Qurilmadan mustaqil kiritish/chiqarish dasturiy taʼminoti***

Ba'zi bir dasturiy ta'minot qismlari aniq bir qurilma bilan birga ishlash uchun mo'ljallangan, uning qismi nisbatan qurilmadan mustaqil. Drayver va qurilmadan mustaqil dasturiy ta'minot o'rtasidagi aniq chegara tizimdan o'tadi, ba'zi bir funksiyalar qurilmadan mustaqil holda amalga oshirilishi mumkin. Keyingi funksiya odatda qurilmadan mustaqil dasturiy ta'minotni amalga oshiradi:

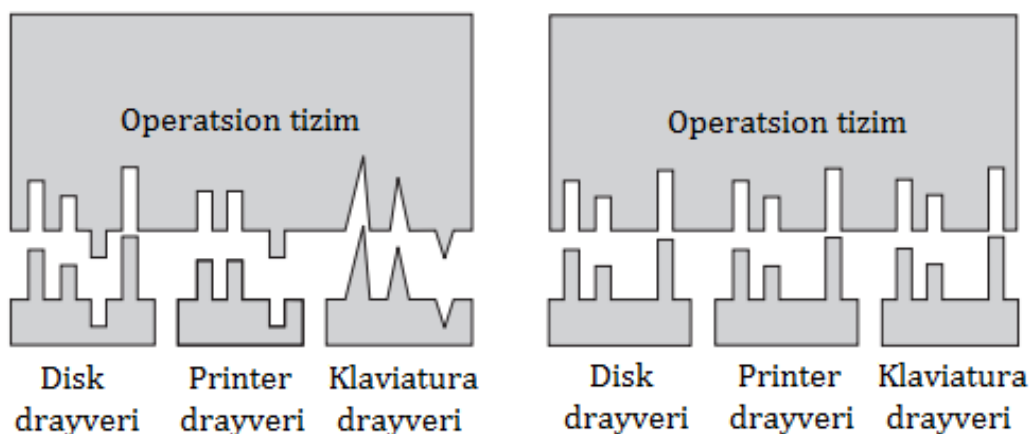
- ❖ qurilma drayveri uchun yagona ko'rinishli interfeyslar;
- ❖ buferlash;
- ❖ xatoliklar to'g'risida xabar;
- ❖ ajratilgan qurilmani egallash va ozod etish;
- ❖ blok hajmiga bog'liq bo'lmagan apparat ta'minoti.

Qurilmadan mustaqil dasturiy ta'minotning bosh maqsadi – barcha qurilmalar uchun umumiy kiritish/chiqarish funksiyasini bajarish va foydalanuvchi darajasida dasturlar uchun yagona ko'rinishli interfeyslarni taqdim etish.

### ***Qurilma drayveri uchun yagona ko'rinishli interfeyslar***

Operatsion tizimdagi asosiy vazifasi – kiritish/chiqarish va drayver qurilmalarini taqdim etishni nisbatan bir xil shaklga keltirish. Agar disklar, printerlar, monitorlar, klaviaturalar va boshqa qurilmalar har xil interfeysga ega bo'lib, har biri yangi qurilma kompyuterga ulanganda operatsion tizimdan modifikatsiyasini talab qiladi. 4.13a-rasmda sxema ko'rinishida holat, ya'ni har bir qurilma drayveri operatsion tizim bilan shaxsiy interfeysiga ega bo'lgani, 4.13b-rasmda barcha drayverlar bir xil interfeysga ega bo'lgani tasvirlangan.

Standart interfeyslar yangi qurilma drayverlari tizimga kirishi soddalashtirilgan. Standartga rioya qilish drayver muallifi qaysi bir funksiya joriy qilinganligini va yadroda qanday chaqiriq joylashganligini biladi.



4.13- rasm. a-rasmda drayverlar turli xil interfeysga ega; b-rasmda barcha drayverlar bir xil interfeysga ega

Amaliyotda farqlanadi, biroq ularning soni ko‘p emas, va turli xil turlar o‘rtasida ko‘plab umumiylik mavjud.

Xatto blokli va belgili qurilmalar kamdan kam bir xil funksiyani bajaradi.

Interfeysni bir xillashtirishning muhim omillaridan biri – kiritish/chiqarish qurilmalarini nomlash usuli. Apparatdan mustaqil bo‘lgan dastur drayverlariga tegishli qurilmalariga belgili nomlar berish. Masalan, UNIX va MINIX 3 da qurilma nomi maxsus fayl indeksini ko‘rsatuvchi bir ma’noli /dev/disk0, mos bo‘lgan drayver esa qurilmani bosh raqami bo‘yicha aniqlaniladi.

### ***Buferslash***

Buferslash blokli va belgili qurilmalar uchun juda muhim hisoblanadi. Apparat ta’minoti blokli qurilmadan odatda katta bloklarni o‘qish va yozishni talab qiladi. Biroq foydalanuvchi dasturlari uchun bunday chegara yo‘q, ular istalgan hajmda axborotni uzatishga haqli bo‘ladi. Shu sababli agar foydalanuvchi faqat blokning yarmini jo‘natsa, odatda operatsion tizim diskga bu ma’lumotlarni darrov yozmaydi, blokning qolgan qismini jo‘natilishini kutadi. Belgili qurilmalarda foydalanuvchi ularni qabul qilib olish holatidagi qurilmaga nisbatan tezroq ma’lumotlarni jo‘natadi, bunday ko‘rinishda bu yerda buferlash zarur. Ma’lumotlar kelib tushishi bundan mustasno emas, masalan, klaviaturadan oldinga o‘tib ketishi mumkin va bu holat buffersiz bo‘lmaydi.

### ***Xatoliklar to‘g‘risida xabar***



Xatoliklar to'g'risidagi xabar kiritish/chiqarish qurilmasi kontekstining hech qayerida sodir bo'lmaydi. operatsion tizim ularni qayta ishlanishini yuqori darajada kuchaytirishni davom ettirishi kerak. Ko'plab xatolar aniq bir qurilma uchun xususiyatli bo'ladi va drayverda qayta ishlanishi kerak, chunki uni to'g'irlashni faqat u biladi (masalan, harakatlarni takrorlash, xatoliklarni rad etish yoki tizimni rad etish).

O'ziga xos xatoliklar – disk blokiga kirishga ruhsat bermaydi yoki buzadi. Disk drayveri va agar yuqorida turuvchi dasturga axborot berishni uddalay olmasa, bir necha marotaba takroran o'qishga harakat qilinadi. Bu vaqtda xatolikni qayta ishlash apparatdan mustaqil holda bo'ladi. Agar xatolik foydalanuvchi dasturini o'qish joyida bo'lsa, chaqiriqlarni bajaruvchi dastur ma'lumotlarini jo'natish yetarli bo'ladi. Agar tizim tuzilishini muhim qismlarini o'qish imkoniyati bo'lmasa, xatolik to'g'risida axborotni tizimga uzatadi va o'z ishini yakunlaydi.

### ***Ajratilgan qurilmalarni band qilish va ozod etish (bo'shatish)***

Ayrim qurilmalar masalan, CD-RW uzatgich vaqtning har bir onida jarayonni to'liq egallanishini nazarda tutadi. Operatsion tizim bunday qurilma bilan foydalanish so'rovini ko'rib chiqishi va ularni so'ralayotgan qurilmaga kirishga bog'liq bo'lgan so'rovni bajarilishini qabul qilishi yoki rad javobini berishi kerak. Bunday so'rovlarni qayta ishlash sodda usuli maxsus fayllar bilan o'zaro bog'lanish bo'yicha open chaqiriq tizimini joriy qilish bilan yakunlanadi. Agar qurilmaga kirishga ruhsat bo'lmasa, open chaqirig'i muvaffaqiyatsiz yakunlanadi. Close chaqiriq tizimiga murojaat etilib qurilma bo'shatiladi.

### ***Blok hajmiga bog'liq bo'lmagan apparat ta'minoti***

Turli xil disklarda turli hajmli sektorlar bo'lishi mumkin. Qurilmadan mustaqil dasturiy ta'minot yuqori pog'onadan bu faktlarni yashirishi kerak va yagona ko'rinishli hajmli blok sifatida taqdim etishi kerak, masalan, bitta mantiqiy asosda bir nechta fizik segmentlarni birlashtirilishi. Shu sababli nisbatan yuqori pog'ona fizik sektor hajmiga bog'liq bo'lmagan bitta mantiqiy blok hajmida faqat mavhum qurilmaga ega bo'ladi.

Ayrim belgili qurilmalar o'zining baytli ma'lumotlarini taqdim etadilar (masalan, modemlar), unda boshqalar uni kattaroq miqdorda ajratadilar (tarmoq interfeyslari). Bular shuningdek yashirilishi bilan farq qilishi mumkin.

### ***Foydalanuvchi muhitida kiritish/chiqarish dasturiy ta'minoti***

Kiritish/chiqarish qurilmasini katta bir qismi operatsion tizimga tegishli bo'lsa uning oz qismi kutubxona, foydalanuvchi dasturlarini komponovkalash yoki hatto yadroda ishlaydigan butun bir dasturni o'z ichiga oladi. Chaqiriq tizimlari kiritish/chiqarish chaqiriq tizimlarini o'z ichiga oladi. Odatda kutubxona protseduralaridan iborat bo'ladi.

Masalan: `count = write (fdf buffer, nbytes);`

Agar C – dastur bunday chaqiriqni o'z ichiga olsa, kutubxona write protsedurasi dasturlar bilan butun bir ko'rinishga keltiriladi, bunday ko'rinishda dastur bajarilish vaqtida xotiraga yuklanadigan ikkilik kodni ko'rsatadi. Bunday kutubxonali protseduralar to'plami shubhasiz kiritish/chiqarish qurilmasini qismi bo'ladi. Kutubxona protseduralari kamdan kam holatlarda kiritish/chiqarish operatsiyalari shaklida bajariladi. Masalan, `print f` C-protsedurasi matn qatori kirishida qabul qilinadi va bir nechta qo'llash imkoniyati bo'lgan undan ASCII-qatorlarini yaratadi, shundan so'ng bevosita chiqarish uchun write chaqiriq tizimini ishlatadi. Ba'zi foydalanuvchi muhiti kiritish/chiqarish dasturiy ta'minoti kutubxonali protseduradan iborat bo'ladi. Boshqa bir muhim toifa – bu spuling tizimi. *Spuling* (buferlash) o'zida ko'p topshiriqli tizimlarda ajratilgan qurilmalar bilan ishlash yo'llarini taqdim etadi. Spulingdan foydalanadigan qurilmaning o'ziga xosligi printer bo'ladi. Printerda har bir foydalanuvchiga maxsus belgili fayl ochishga ruhsat beradi. Biroq jarayon uni ochishda joriy bir necha soatda printeriga murojaat etmaydi bu vaqtda hech bir jarayon hech narsani chop eta olmaydi. Bu bilan birga *domen* deb ataladigan maxsus jarayon va *spuling katalogi* deb ataladigan maxsus katalogni yaratadi. Faylni chop etish uchun dastlab jarayon spuling katalogida joylashgan, chop etish uchun mo'ljallangan maxsus faylni yaratadi. Bu faylni maxsus fayl printeriga foydalanishga ruhsat berilgan yagona jarayonni *domen* chop etadi. Bunday ko'rinishda, muammo qandaydir jarayon printerni uzoq vaqt davomida egallab tursa foydalanuvchining to'g'ridan-to'g'ri kirishidan maxsus fayllarni himoyalashni hal qilish bo'ladi. Spuling

faqat printer uchun foydalanilmaydi. Masalan, domen tarkibiga kiradigan elektron pochta dasturiy ta'minotida. Jo'natilishi kerak bo'lgan xat spuling katoligada joylashadi. So'ng domen elektron pochta ochib ko'radi, uni qayerga jo'natish kerakligini aniqlaydi va jo'natadi. Vaqtning istalgan momentida kirishga ruhsat olinmaganlikni olish ehtimolligi mavjud bo'ladi, bu holatda domen spuling katalogida xatni qoldiradi va keyinroq takroran uzatishga harakat qilish to'g'risida axborotni qayd etib qo'yadi. Demon shuningdek, kechikish to'g'risidagi xabarni jo'natishi mumkin, agar xat joriy bir necha soatda yoki kunda jo'natilmasa – xatni jo'natish imkoniyati yo'qligi to'g'risida axborotni jo'natadi.

Masalan, foydalanuvchi dasturi operatsion tizimda ishga tushirilgan chaqiriqlarni qayta ishlash uchun fayldagi blokni o'qishga harakat qiladi. Qurilmadan mustaqil dasturiy ta'minot bu blokni keshdan qidiradi. Agar talab qilinayotgan blok bo'lsa, u apparaturaga murojaat qilish va diskdan bu blokni qabul qilish uchun qurilma drayverini chaqiradi. Jarayon diskni o'qish operatsiyalari yakunlangunga qadar bloklanadi.

Disk operatsiyani yakunlaganda, apparat uziladi. To'xtalishlarni qayta ishlash nima sodir bo'lganligini aniqlash uchun ishga tushadi, unda qaysi mexanizm e'tibor talab etilayotganligi aniqlaniladi. So'ng u mexanizm holati to'g'risida axborotni oladi va kiritish/chiqarish so'rovini qayta ishlashni yakunlash uchun "vaqtinchalik to'xtab turgan" jarayonni faollashtiradi, va foydalanuvchi rejimi ishlashni davom ettirishga topshiriladi.

### ***O'zaro birgalikda bloklash***

Kompyuter tizimlarida vaqtning har bir onida faqat bitta jarayonda foydalanish mumkin bo'lgan resurlar mavjud. Misol sifatida printerlarni magnitli lenta tarqatgichlar va tizimning ichki jadval elementlarini keltirish mumkin. Printeriga bir vaqtning o'zida ma'lumot uzatadigan ikkita jarayon mavjud, belgilar to'plamini chop etishga berish mantiqsiz. Bitta va shu fayl tizimi jadval elementidan foydalanadigan ikkita jarayon mavjud bo'lishi fayl tuzilishini barbod bo'lishiga olib keladi. Shu sababli barcha operatsion tizimlar dasturi va apparat resurslariga ohirgi chora sifatida vaqtinchalik kirishni jarayonlarga taqdim etishi mumkin.

Amaliy jarayon ko‘pincha bir emas bir necha resurslarga kira olishi zarur. Faraz qilamiz, masalan ikkala jarayonning har biri kompakt – diskdan hujjat nusxasini yozib olishni istashi mumkin. *A* jarayon skanerdan foydalanishga ruhsat so‘raydi va uni oladi. *B* jarayon boshqa dastur bilan dasturlanganligi sababli dastlab kompakt diskga yozish uchun qurilmaga murojaat qiladi. So‘ng *A* jarayon kompakt diskga yozish uchun qurilmaga murojaat qiladi, lekin bu qurilma *B* jarayon bilan band qilinganligi sababli so‘rovga rad javobi beriladi. Afsuski, kompakt – diskga yozish uchun qurilmani ozod qilishda *B* skanerdan so‘raydi. Jarayonlar bloklanganligini ko‘rsatadi. Bunday holat o‘zaro birgalikda bloklash yoki berklik deyiladi.

### ***Resurslar***

Tizim jarayon qurilma, fayl va boshqalarga kirish qoidalarini taqdim etganda o‘zaro birgalikda bloklashni ko‘rishimiz mumkin. O‘zaro birgalikda bloklash to‘g‘risida umumiy bir xulosaga kelib obyektga kirishni resurs deb atashimiz mumkin. Resurs qurilma (masalan, magnitli tasmali tarqatgich) yoki axborot bir qismi (ma‘lumotlar omborida berkitilgan yozuv) bo‘lishi mumkin. Kompyuterda murojaat qilish mumkin bo‘lgan turli hajmli resurslar mavjud. Bundan tashqari tizimda biron bir resursning bir nechta nusxalarini ko‘rishimiz mumkin. Agar tizimda *bir xil* deb ataladigan bir nechta o‘zaro almashadigan resurs nusxalarini bor, unda resurslarga murojaat qilishga javob nusxalardan kirishga ruhsat etilgan istalgan birini taqdim etishi mumkin. Resurs – vaqtning istalgan onida faqat bitta jarayondan foydalanish qoidasi.

Resurslar ikki xil bo‘ladi: *yuklanadigan* va *yuklanmaydigan*. Yuklanadigan resurs jarayonning egasidan olish zararsiz. Bunday resurslarga xotirani kiritish mumkin. Masalan, foydalanuvchi xotirasi 64 Mbayt li tizimga bitta printer va har biri nimanidir chop etishni istayotgan 64 Kbaytli ikkita jarayon bor. *A* jarayon so‘raydi va printerni oladi, so‘ng chop etish uchun ma‘lumotlarni hisoblashni boshlaydi. Yana hisoblashni tugamasdan, o‘zining kvant vaqtini oshiradi va o‘rin almashtirish maydonida diskga yuklaydi.

Shundan so‘ng *B* jarayon ishlaydi va bekordan bekorga printerga murojaat qilishga harakat qiladi. Natijada berklikli vaziyatga tushib qolinadi, *A* jarayon printerni egallab olganligi sababli, *B* jarayon xotirani band qiladi va ulardan hech biri resursiz o‘z ishini davom

ettira olmaydi. *B* jarayonni xotiraga yuklash ta'qiqlanmagan, uni o'rin almashtirish maydonida xotiraga o'tkaziladi va *A* jarayonni xotiradan diskga yuklanadi. Shundan so'ng *A* jarayon hisoblashni yakunlashi, chop etishni bajarishi va so'ng printerni bo'shatishi mumkin. O'zaro birgalikda bloklanish sodir bo'lmaydi.

Yuklanmaydigan resurs – hisoblash natijasida yo'q qilib bo'lmaydigan joriy egasidan olib bo'lmaydigan resurs. Agar kompakt – diskga yozish vaqtida yozish uchun qurilmadan jarayonni to'satdan tortib olsa va uni boshqa jarayonga uzatsa, unda kompakt – disk ishdan chiqish natijasini olamiz. Kompakt diskni yozish uchun qurilma vaqtinchalik resurs ixtiyoriy vaqtda yuklanmaydi.

O'zaro birgalikda bloklash yuklanmaydigan resurs hisoblanadi. Berklikli holatda, ya'ni resurs ko'rinishiga qarama-qarshi bo'lganda, odatda bir jarayondan boshqasiga resurslarni taqsimlashga ruhsat beradi. Shu sababli biz e'tiborimizni yuklanmaydigan resurslarga qaratamiz.

Resursdan foydalanish uchun zarur hodisalar ketma-ketligi mavhum ko'rinishi quyidagi shaklda tasvirlanadi:

- ❖ Resursni so'rash;
- ❖ Resursdan foydalanish;
- ❖ Resursga qaytish.

Agar resurs ruhsat etilmagan bo'lsa, uning jarayoni so'rovni kutadi. Ba'zi operatsion tizimlarda avtomatik bloklaydigan jarayon resursiga murojaat muvaffaqiyatsiz bo'lsa va shundan so'ng qayta yangilansa resurs kirishga ruhsat etilgan bo'lib qoladi. Boshqa tizimlarda resurs so'rovi va rad etishda xatolik kodiga murojaat etilsa, unda chaqiriluvchi jarayon biroz kutib turishi va harakatni takroran bajarishi mumkin.

### ***O'zaro birgalikda bloklanish shartlari***

Quyida o'zaro bir-birini bloklashning 4 ta sharti keltirilgan.:

❖ **Bir-birini chiqarib tashlash.** Joriy vaqtdagi har bir resurs bitta jarayonga beriladi yoki ruhsat etiladi.

❖ **Ushlab turish yoki kutish.** Avvalgi olingan resurslarni ushlab turish vaqtida jarayon yangi resurslarni so'rashga haqli bo'ladi.

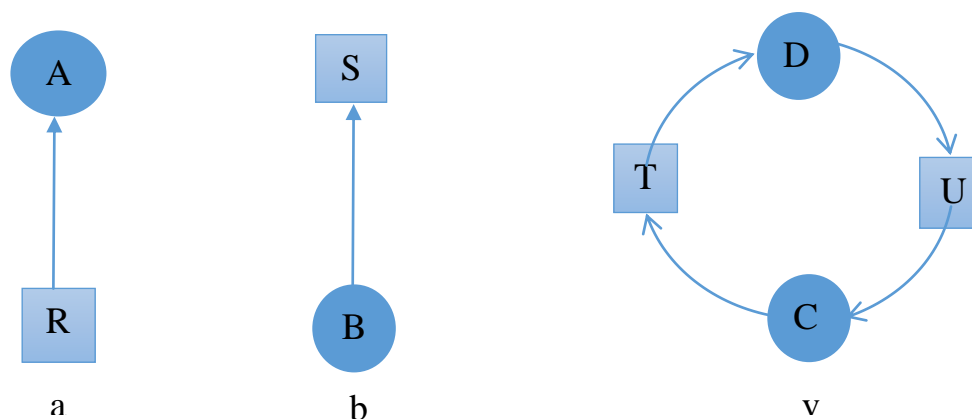
❖ **Resurslarni majburiy ravishda bo'shatish.** Jarayonning avval qabul qilib olgan resurslarini olib qo'yish mumkin emas. Jarayon egasi o'zi resurslarni ozod etishi kerak.

❖ **Siklik kutish.** Ikki va undan ortiq jarayonlarning siklik ketma-ketligi mavjud bo‘lishi kerak, ularning har biri, ya’ni resursga kirishga ruhsat kutayotganlar.

Bir-birini bloklash sodir bo‘lganda yuqoridagi to‘rtta shartni bajarilishi kerak. Agar hech bo‘lmaganda ulardan biri bajarilmasa, beriklikli holat bo‘lmaydi.

### ***Bir – birini bloklashni modellashtirish***

Graflar yo‘nalishidan foydalanib yuqoridagi to‘rtta bir-birini bloklash shartini modellashtirish mumkin. Graflar ikki xil tugunga ega: kichkina doirada ko‘rsatilgan jarayonlar va kvadratda tasvirlangan resurslar. Resurs tugunidan jarayon tuguniga yo‘naltirilgan resurs avval jarayon tomonidan so‘ralganligini, olinganligini va joriy vaqtda jarayonda foydalanyotganligini bildiradi. 4.14a- rasmda *R* resurs joriy vaqtda *A* jarayonga berilgan.



4.14- rasm. Jarayonlarni bo‘linishi: a – resurs band; b – resurs so‘rovi; v – bir-birini bloklash

Jarayondan resursga yo‘nalgan rebro ayni vaqtda jarayon bloklangan va ushbu resursga kirishni kutish holatida ekanligini bildiradi. 4.14b-rasmda *B* jarayon *S* resursni kutayapti. 4.14v-rasmda bir birini bloklash tasvirlangan. *S* jarayon *D* jarayon ushlab turgan *T* resursni kutayapti. *D* jarayon *T* resursni bo‘shatish niyati yo‘q, chunki u *C* jarayondan foydalanilayotgan *U* jarayonni kutayapti. Ikkala jarayon cheksiz vaqtgacha kutish holatida bo‘ladi.

Graf sikli siklik ravishda jarayon va resurslarni o‘z ichiga olganini bir birini bloklash mavjud ekanligi bildiradi (tizimda har bir

ko‘rinishga bitta resurs bo‘lish taklif qilinadi). Bu misolda sikl *C-T-D-U-C* ketma-ketlikda bo‘ladi.

Bir biriga qarshi kurashishi uchun amaliyotda to‘rtta strategiya ishlatiladi:

❖ Muammolarni to‘liq rad etish. Agar siz muammoni rad etmasangiz, u sizni rad etishi mumkin.

❖ Aniqlash va bartaraf etish. Bir birini bloklash holati paydo bo‘lishga ruhsat beriladi, so‘ng u aniqlaniladi va muammoni hal qilish uchun choralar ko‘riladi.

❖ Bir birini bloklash to‘rtta shartidan birontasi bajarilmaslikka yo‘l qo‘ymaslik uchun oldini olish.

❖ Berklik holatidan dinamik qochishda resurslarni to‘g‘ri taqsimlash.

### **Nazorat savollari**

1. Qurilmalarni boshqarish nima uchun kerak?
2. Qurilmalar turlarini keltiring.
3. O‘zaro birgalikda bloklash nima va uning shartlarini tushuntiring.
4. Kiritish/chiqarish menejeri vazifalari.
5. Tarmoq qurilmalariga misol keltiring.
6. Kontroller qanday vazifani bajaradi?
7. Kiritish/chiqarishni xotirada tasvirlanishini tushuntiring.
8. Kiritish/chiqarish qurilmalarida to‘xtalishlarni tushuntiring.
9. Qurilmalarni boshqarish bo‘yicha operatsion tizimning vazifalari.
10. DMA kontrolleri nima va uning vazifasi?
11. Protessor va kiritish/chiqarish qurilmalarini parallel ishlashini tushuntiring.

## **VBOB. FAYL TIZIMI VA FAYLLARNI BOSHQARISH JARAYONI**

### **5.1. Operatsion tizimlar fayl tizimi, tuzilishi va tashkil etilishi**

#### ***Fayllarni boshqarish***

Operatsion tizimining muhim vazifalaridan biri foydalanuvchiga diskda saqlanayotgan ma'lumotlar bilan qulay ishlash vositalarini taqdim etish hisoblanadi. Buning uchun OT saqlangan ma'lumotlarning fizik tuzilishini foydalanuvchi uchun qulay bo'lgan mantiqiy model bilan almashtiradi - tizim kataloglari va fayllari ierarxik tashkil qilingan bo'ladi. Har qanday kompyuter dasturi ma'lumotlarni qabul qiladi, saqlaydi va namoyish qiladi. Ish paytida, jarayon cheklangan miqdordagi ma'lumotni o'z manzil maydonida saqlashi mumkin, chunki uning hajmi virtual manzil maydoni doirasi bilan cheklangan. Ba'zi amaliy dasturlar uchun, masalan, aviachiptalarni bron qilish tizimlari, bank tizimlari va boshqalar, faqat virtual manzillar maydoni yetarli bo'lmaydi. Bundan tashqari, jarayon tugaganidan so'ng, uning manzil maydonida saqlanadigan ma'lumotlar yo'qoladi. Shu bilan birga, bir qator dasturlar (masalan, ma'lumotlar bazalari) uchun uni uzoq vaqt, ba'zan esa abadiy saqlash kerak. Bunday dasturlar uchun jarayon tugaganidan keyin ma'lumotlarning yo'qolishiga yo'l qo'yilmaydi. Kompyuter, shuningdek, ishlamay qolganda, jarayonda uzilishlar bo'lganda ham ma'lumot saqlanishi kerak.

#### ***Fayl tizimining maqsadi va vazifalari***

Fayl - bu tashqi xotiraning nomlangan maydoni bo'lib, unga yozilishi va o'qilishi mumkin bo'lgan ma'lumotlar, shuningdek, ushbu maydonda saqlangan ma'lumotlar va OTga ushbu ma'lumotlarni boshqarishga imkon beradigan atributlar to'plami kiradi.

Fayl menejeri (shuningdek, fayllarni boshqarish tizimi deb ham ataladi) bu fayllarni yaratish, o'chirish, o'zgartirish va fayllarga kirishni boshqarish, shuningdek fayllar foydalanadigan resurslarni boshqarish uchun javob beradigan dasturiy ta'minotdir. Kompyuter tizimida fayl menejeri o'z fayllarini fayl nomi, uning ikkinchi darajali



saqlash qurilmasida joylashgan joyi va har bir fayl to'g'risidagi muhim ma'lumotlarni o'z ichiga olgan kataloglar yordamida kuzatib boradi. Fayl menejeri on-layn foydalanuvchilar uchun, buferlash operatsiyalari va interaktiv hisoblash uchun ma'lumotlar va ma'lumotlar kutubxonasini qo'llab-quvvatlaydi. Ushbu funksiyalar qurilma menejeri (Device Manager) bilan birgalikda amalga oshiriladi. Fayl menejerining vazifasi murakkab. U tizimning fizik komponentalari, uning axborot resurslari va siyosati, fayllarni saqlash va taqsimlash uchun javobgardir.

O'z majburiyatlarini bajarish uchun u quyidagi to'rtta vazifani bajarishi kerak:

1. Har bir fayl saqlanadigan joyni kuzatib borish;
2. Fayllarning qayerda va qanday saqlanishini aniqlaydi, mavjud xotira maydonlaridan samarali foydalanish va fayllarga samarali kirishni ta'minlaydigan siyosatdan foydalanish;
3. Foydalanuvchidan faylga kirish huquqi olib tashlanigandan keyin, har bir faylni belgilash va undan foydalanishni yozib olish;
4. Faylni saqlash qurilmasiga qaytarish kerak bo'lsa, belgilangan fayllarni bekor qilish va uning mavjudligini uni kutayotganlarga yetkazish.

Kompyuter tizimida fayllar menejeri o'z fayllari nomi, uning saqlash qurilmasida joylashgan joyi va har bir fayl to'g'risidagi muhim ma'lumotlarni o'z ichiga olgan kataloglar yordamida kuzatib boradi.

Masalan, fayl tizimi kutubxonaga o'xshaydi, Fayl menejeri xuddi shu to'rtta vazifani bajaradigan kutubxonachining rolini o'ynaydi:

1. Kutubxonachi to'plamdagi har bir buyumni kuzatib borish uchun katalogdan foydalanadi. Har bir yozuvda foydalanuvchiga kerakli kitoblarni topishga yordam beradigan telefon raqamlari va ma'lumotlar ko'rsatilgan.
2. Kutubxona to'plamdagi barcha buyumlarni, shu jumladan, katta hajmdagi kitoblar, jurnallar, DVD disklar, xaritalar va videolarni saqlash siyosatiga amal qiladi.
3. Qachonki u so'rala, buyum (kitoblar, jurnallar, disklar, videolar va boshqalar) javondan olinadi va ijarachini ismi jurnalga qayd etiladi.

4. Buyum qaytarib berilganda, kutubxonachi jurnalga tegishli yozuvni kiritadi va uni saqlaydi.

Fayldan foydalanishning asosiy maqsadlari quyida keltirilgan:

❖ Axborotlarni uzoq vaqt va ishonchli himoyalash. Resursga bog‘liq bo‘lmagan saqlash qurilmasi hisobiga uzoq vaqt saqlanadi, yuqori ishonchlilik esa faylga kirishni himoyalash vositalari va OT dasturiy kodining umumiy tashkil etilishi bilan belgilanadi, aksariyat hollarda qurilmaning ishdan chiqishi fayllarda saqlanadigan ma‘lumotni yo‘q qilmaydi.

❖ Axborotlardan birgalikda foydalanish. Foydalanuvchida fayllar bilan ishlashning qulay vositalari, shu jumladan, kataloglar-ma‘lumotnomalari, fayllarni guruhlariga birlashtirish va belgilar bo‘yicha fayllarni izlash vositalari, fayllarni yaratish, o‘zgartirish va o‘chirish uchun buyruqlar to‘plami bo‘lishi kerak. Fayl biron bir foydalanuvchi tomonidan yaratilgan bo‘lishi, boshqa biri undan foydalanishi mumkin, shu sababli faylni yaratuvchisi yoki ma‘mur boshqa foydalanuvchilarni unga kirish huquqini belgilashi mumkin.

**Fayl tizimi** (FT) – bu operatsion tizimning bir qismi va quyidagilarni o‘z ichiga oladi.

❖ Diskdagi barcha fayllar majmuini;

❖ Fayllarniboshqarishuchunfoydalaniladiganma‘lumotlartuzilis hito‘plami (masalan, diskdabo‘shvabandjoylarnitaqsimlashjadvali, fayldeskriptorlari, faylkatalogikabi);

❖ Faylni izlash, nomlash, yozish, o‘qish, o‘chirish va yaratish kabi fayl bilan bog‘liq turli xil operatsiyalarni amalga oshiruvchi dasturiy tizim vositalar to‘plami.

Fayl tizimi faylni taqdim etuvchi ba‘zi mavhum obyekt ishlarini bajarishda sodda operatsiyalar to‘plamidan iborat dasturlardan foydalanadi. Shu sababli dasturchilarga diskdagi ma‘lumotlarni joylashuvi, ma‘lumotlarni buferlash va boshqa uzoq vaqt xotirada saqlovchi qurilmada ma‘lumotlarni uzatish bilan bog‘liq past pog‘ona muammolarini to‘liq bilishi shart emas. Bu funksiyalarning barini fayl tizimi o‘z ichiga oladi. Fayl tizimi diskli xotirani taqsimlaydi, faylni nomlaydi, tashqi xotiradagi tegishli manzilda fayl nomini tasvirlaydi, ma‘lumotlarga kirishni ta‘minlaydi, qismlarni qo‘llab-quvvatlaydi, faylni himoya qiladi va qayta tiklaydi.

Bunday ko‘rinishda fayl tizimi oraliq qatlam rolini bajaradi, uzoq vaqt saqlanadigan ma‘lumotlarni fizik tashkil etilishining barcha

murakkabliklarini himoyalaydi va bu saqlanuvchiga nisbatan sodda mantiqiy modeli dasturni fayl yaratuvchisiga taqdim etadi, shuningdek unga murakkab fayllar uchun qulay bo'lgan buyruqlardan foydalanishni taqdim etadi.

FT tomonidan yechiladigan vazifalar umuman hisoblash jarayonini tashkil qilish usuliga bog'liq. FT ning eng sodda ko'rinishi bu bir foydalanuvchilik va bir dasturli OT lar, masalan MS-DOS. Bir foydalanuvchilik va bir dasturli FT ning asosiy funksiyasi quyidagi masalalarni hal etish hisoblanadi:

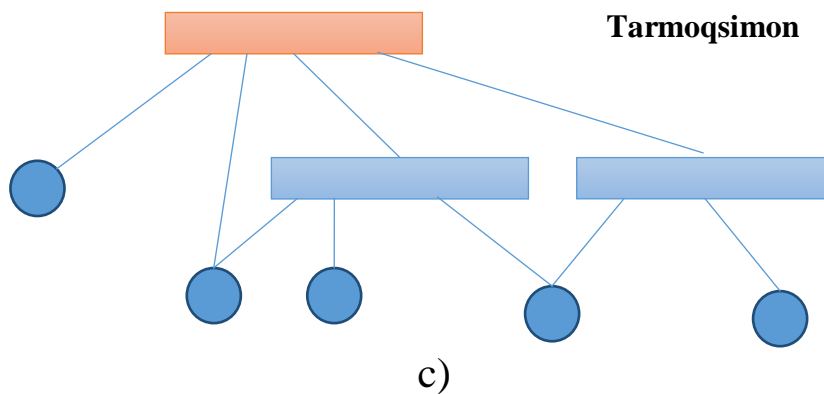
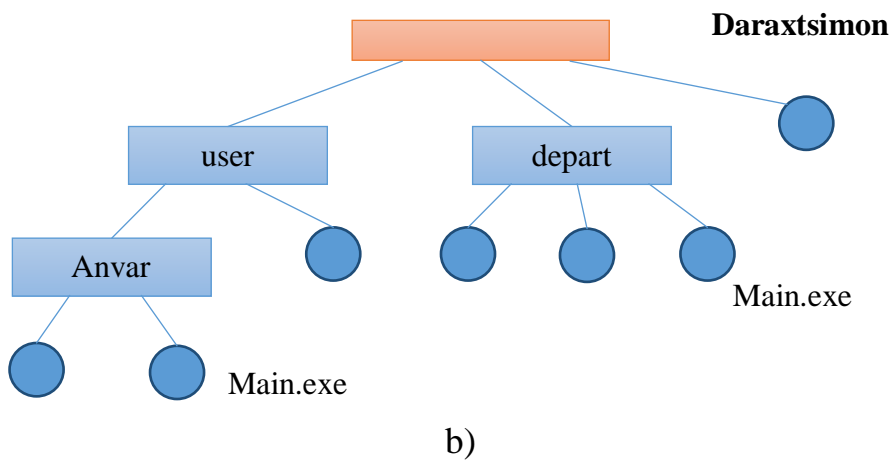
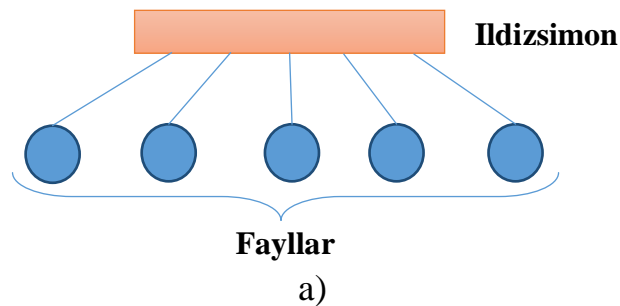
- ❖ Faylni nomlash;
- ❖ Ilovalar uchun dasturiy interfeys;
- ❖ Ma'lumotlar saqlaydigan fizik tashkil etuvchisida fayl tizimining mantiqiy modelini tasvirlash;
- ❖ Resursdagi to'xtalishlarga, qurilma va dasturiy vositalar xatoliklariga fayl tizimining bardoshlilik.

FT masalasi bir foydalanuvchi ishlashiga mo'ljallangan, lekin unga bir vaqtning o'zida bir nechta jarayonlarni ishga tushirishga imkon beradigan *bir foydalanuvchilik ko'pdasturli OT* operatsiyalarini murakkablashtiradi. Bu toifadagi birinchi OT OS/2 bo'lgan. Yangi topshiriqni qo'shish masalasini bajarishda faylga bir nechta jarayonlar kirishiga ruhsat berilishi kerak bo'lgan. Bu holatda fayl resurslarga ajratilgan, demak fayl tizimi ushbu resurslar bilan bog'liq barcha muammolarni hal etishi kerak bo'lgan. Bir foydalanuvchilik ko'pdasturli fayl tizimida faylni va uning qismlarini bloklash, quvib o'tishni oldini olish, berklikni rad etish, nusxalarni moslashtirish va boshqalarni ko'rib chiqish mumkin.

### ***Fayl tizimlarini ierarxik tuzilishi***

Foydalanuvchi belgili nom bo'yicha faylga murojaat etadi. Biroq inson xotirasida obyekt qiymatlarini saqlab qolish cheklangan, ya'ni foydalanuvchi nom bo'yicha murojaat qilishi mumkin. Nom joylarini ierarxik tashkil qilish bu chegaralarni kengaytirish imkoniyatini beradi. Aynan shu sababli ko'plab fayl tizimlari nisbatan quyi pog'ona katalogi nisbatan yuqori pog'ona katalogi tarkibiga kirishi mumkin bo'lgan pog'onalarni tashkil etish hisobiga ierarxik strukturaga ega bo'ladi, shu sababli ierarxik struktura xususiy holatlarda barcha fayllar bitta katalog ichiga kirganda bir pog'onali tashkil etilgan bo'ladi

(5.1a- rasm). Ierarxik katalogni tavsiflovchi graf daraxtli (5.1b- rasm) yoki tarmoqli (5.1v- rasm) bo‘lishi mumkin.



5.1- rasm. Fayl tizimi ierarxik tuzilishi

Katalog agar faqat bitta katalogga kirishga ruhsat berilgan bo‘lsa daraxtli va agar fayl bir nechta kataloglarda joylashgan bo‘lsa tarmoqli bo‘ladi.

Masalan, MS-DOS va Windows da kataloglar daraxtsimon, Unix da esa tarmoq ko‘rinishda ifodalanadi. Daraxtsimon strukturada har

bir fayl sahifalarda bo‘ladi. Katalog - ildiz katalog yoki ildiz (root) deb ataladigan yuqori pog‘onada turadi.

Bunday tashkil etishda foydalanuvchi barcha fayllar nomlarini eslab qolish zaruratidan ozod bo‘ladi, unga shu yoki boshqa fayllarni qaysi guruhda saqlanishi mumkinligini bilish yetarli bo‘ladi. Ya’ni uni kataloglar ketma-ketligini ko‘rib izlab topish uchun. Ierarxik struktura ko‘p foydalanuvchilik tizimlar uchun ham qulay: har bir foydalanuvchi o‘zining katalogi yoki katalog daraxti shoxlarida o‘zining fayllarini joylashtiradi.

### ***Faylga kirish mexanizmlari***

Operatsion tizimlarda fayllarga kirishning uchta usuli mavjud:

- ❖ Ketma-ket kirish;
- ❖ To‘g‘ridan-to‘g‘ri kirish;
- ❖ Indeksli ketma-ket kirish.

**Ketma-ket kirish** - bu fayllarga kirishning ma’lum bir ketma-ketlikda amalga oshirilishi, ya’ni fayldagi ma’lumotlar ketma-ket tartibda qayta ishlanadi. Misol uchun, kompilyator odatda ushbu usulda fayllarga kirishga ruhsat oladi.

**To‘g‘ridan-to‘g‘ri kirish** - har bir yozuv faylda o‘zining manzillariga ega bo‘lib, undan foydalanib fayllarga to‘g‘ridan-to‘g‘ri o‘qish yoki yozish uchun kirish mumkin. Yozuvlar faylda hech qanday ketma-ketlikda bo‘lishi shart emas va ular qo‘shni tashuvchi joylarda bo‘lmasligi kerak.

**Indeksli ketma-ket kirish** – ushbu mexanizm ham fayllarga ketma-ket kirishga asoslangan. Ushbu usul turli xil bloklar uchun ko‘rsatgichlarni o‘z ichiga olgan fayl uchun indeksni yaratadi. Fayldagi yozuvni topish uchun avval indeks ketma-ketlikda izlanadi, so‘ng uning ko‘rsatgichi yordamida faylga to‘g‘ridan-to‘g‘ri kirish amalga oshiriladi.

## **5.2. Fayllarni fizik tashkil etish, ularni o‘qish va yozish usullari**

### ***Fayl tizimlarini fizik tashkil etilishi***

Foydalanuvchi fayl tizimi to‘g‘risida kamdan kam holatda fayllar umumiy tartibda saqlaniladigan ierarxik tashkil qilingan ko‘plab axborot obyektlaridan iborat deb tasavvur qiladi. Faylning asosiy maqsadi baytlarni uzluksiz to‘plamidan tasvirlaydi, aslida esa

disk bo‘ylab qismlarga bo‘lingan bo‘ladi. Chunki u bu bo‘linishlar faylni mantiqiy tuzilishi bilan hech qanday bog‘lanmagan. Masalan, uning alohida mantiqiy yozuvi diskning son sanoqsiz sektorlarida joylashgan bo‘lishi mumkin. Real qurilmada faylni, katalogni va axborot tizimlarini joylashtirish faylni fizik tashkil etilishi bilan tavsiflanadi. Turli xil fayl tizimlari turli xil fizik tashkil etilishga ega bo‘ladi.

### ***Disklar, qismlar, sektorlar, klasterlar***

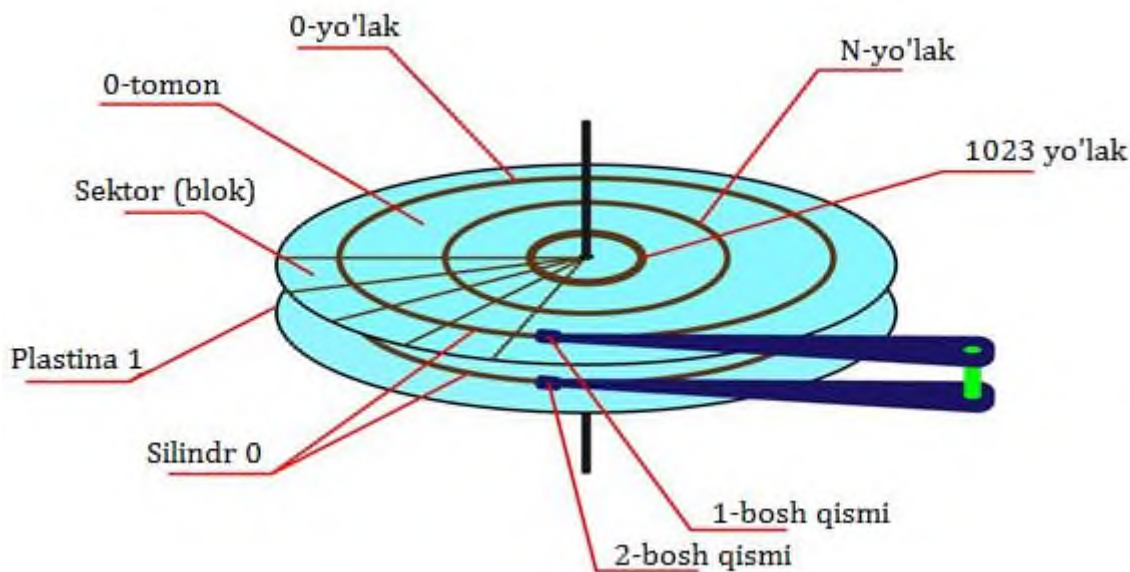
Fayllarni saqlashda zamonaviy hisoblash tizimlarida foydalaniladigan asosiy qurilma turi qattiq disk bo‘ladi. Diskli tarqatgich qattiq va yumshoq magnitli disklardagi ma’lumotlarni yozish va solishtirib o‘qish uchun mo‘ljallangan.

Qattiq disk har biri bitta yoki ikkita tomoni magnitli materilallar bilan berkitilgan bir yoki bir nechta shishali yoki metalli plastinkadan iborat bo‘ladi. Bunday ko‘rinishda, umumiy holatda disk plastina paketidan iborat bo‘ladi (5.2- rasm).

Har bir tomondagi har bir plastina ingichka bitta umumiy markazga ega halqa – ma’lumotlar saqlanadigan yo‘lak chizig‘i bo‘ladi. Yo‘laklar miqdori disk turiga bog‘liq bo‘ladi. Yo‘laklar raqami 0 dan boshlanib diskning markazidagi tashqi chiziqgacha boradi. Disk aylanganda kichkina bosh (ma’lumotlarni o‘qish) deb nomlangan element magnitli yo‘lak orqali ikkilik ma’lumotlarni o‘qiydi yoki ularni magnitli diskga yozadi. Ma’lumotlarni o‘qish ushbu yo‘lakga joylashtirilgan bo‘lishi mumkin. Ma’lumotlarni o‘qish diskning yuqorisida diskretli qadamlarni o‘zgartiradi, har bir qadam bitta qadamga siljishga to‘g‘ri keladi.

Diskga yozish ma’lumotlarni o‘qish usuli orqali yo‘lakning magnit xususiyatlarini o‘zgartirishi tufayli amalga oshiriladi. Ba’zi disklarda har bir yuza bo‘ylab bitta ma’lumotni o‘qishga (kichkina bosh) o‘zgaradi, boshqalarda esa har bir yo‘lakda ma’lumotlarni o‘qish asosida bo‘ladi. Birinchi holatda “kichkina boshdan” axborotlarni izlash uchun disk radiusi bo‘yicha o‘zgaradi. Barcha kichkina boshlar odatda yagona joyini o‘zgartiruvchi mexanizm va sinxron harakatlanish bilan mustahkamlangan bo‘ladi. Shu sababli “kichkina bosh” bitta yuza qism yo‘lagidagi ma’lumotni qayd etganda qolgan barcha “kichkina bosh” lar xuddi shu raqamli yo‘lakda to‘xtaydi. Har bir yo‘lak bitta yo‘lak boshqa biri bilan kichkina boshni

o'zgartirishni talab qilmaydigan alohida kichkina boshga ega bo'ladi va buning hisobiga ma'lumotlarni izlashga sarflanadigan vaqt tejaladi.



5.2- rasm. Qattiq disk qurilmasining sxemasi

Plastinaning yuqa qismidagi bitta radiusidagi yo'lak majmui silindr (cylinder) deb ataladi. Har bir yo'lak sektor yoki blok deb ataladigan fragmentlarga (qismlarga) bo'linadi. Barcha yo'laklar sektorlar soni bilan teng bo'ladi. Sektor ikkilik darajasi bilan ifodalanadigan aniq bir tizim hajmi uchun belgilangan bo'ladi. Ko'pincha sektorning hajmi 512 baytdan iborat bo'ladi. Turli xil radiusli yo'laklar bir xil sektor miqdoriga ega bo'ladi. Markazga yaqin yo'lakka nisbatan uzoqrog'iga yozish zichroq bo'ladi (markazga yaqin yo'laklarning hajmi kichikroq bo'ladi).

Kerakli sektorni diskdan izlashda unga sektorning barcha tashkil etuvchilarini berish kerak bo'ladi:

- ❖ silindr raqami;
- ❖ yuzasi raqami;
- ❖ yo'lak sektor raqami.

Sektor – tezkor xotira bilan disk qurilmasi ma'lumotlar almashadigan qisqa manzil. Bu dastur diskdan faqat bitta baytni o'qishni talab etsa, butun bir sektor o'qiladi va kerakli ma'lumotni tanlash uchun tizimga uzatishini bildiradi. Amaliy dastur sektorlarda emas baytlarda operatsiyani bajaradi, shu sababli talab etilayotgan ma'lumot sektori qisqa hajmini ko'rsatish shart bo'lmaydi, u holda so'rov zarur axborotlarni o'z ichiga olgan bir nechta sektorlarni va

talab qilinyotgan ortiqcha ma'lumotlarni o'z ichiga olgan bir yoki ikkita sektorni o'qishni o'z ichiga oladi.

Operatsion tizim disk bilan ishlaganda klaster deb ataladigan diskdagi yagona joydan foydalanadi. OT faylni yaratish yordamida uning uchun diskdan joy so'raydi. Masalan, agar fayl 2560 hajmli joyga ega bo'lsa, fayl tizimida klaster hajmi esa 1024 bilan belgilangan bo'lsa, u holda fayl diskni 3 ta klasterga ajratadi.

Klaster – diskga xotirani taqsimlash orqali fayl tizimi operatsiyalarini bajaradigan eng kam miqdorli disk joyi. Yo'laklar va sektorlar fizik yoki past pog'ona, diskni formatlash protseduralarini bajarish natijasida yaratiladi. Diskda belgilangan blok chegarasi uchun identifikatsiyalangan axborotlar yoziladi. Past pog'ona disk shakli ushbu disk foydalaniladigan operatsion tizimga bog'liq emas.

Aniq bir fayl tizimi uchun disk belgisi yuqori pog'ona yoki mantiqiy formatlash (shakllantirish) protseduralarida bajariladi. Yuqori pog'onada shakllantirish klaster o'lchamini begilab beradi va diskga fayl tizimi bilan ishlashda zarur bo'ladigan axborotlarni yozadi, ya'ni kirishga ruhsat berilgan yoki foydalanilmaydigan to'g'risida, fayl va kataloglarni ajratilish hududi chegarasi to'g'risida, hududlardagi shikastlanishlar to'g'risida axborotlar. Bundan tashqari operatsion tizimga yuklovchini, ya'ni kompyuter reestri yoki manba ulangandan so'ng operatsion tizimga initsializatsiyalash jarayonini boshlab beradigan uncha katta bo'lmagan dasturni yozadi.

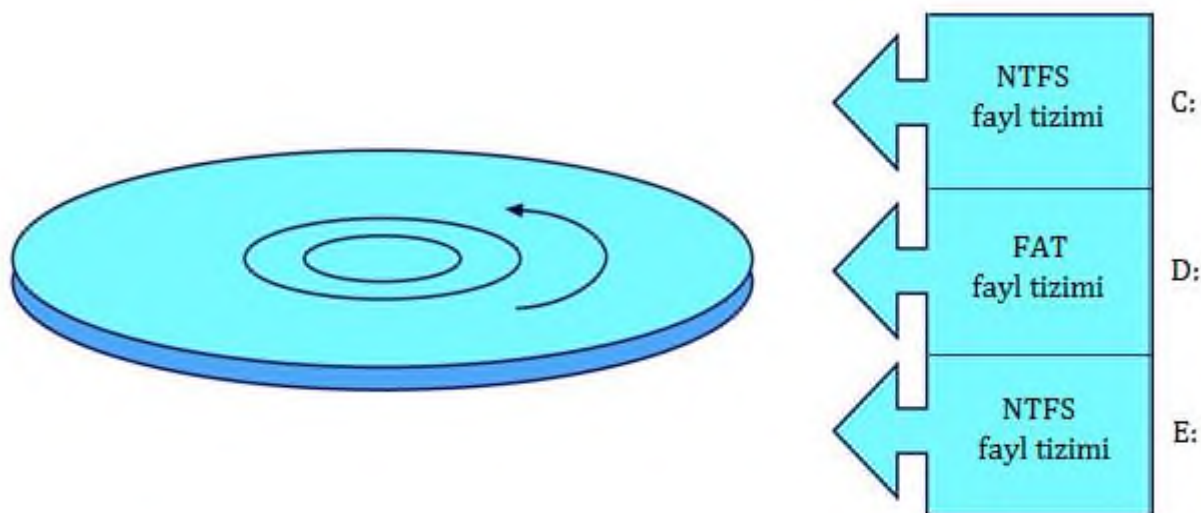
Ma'lum bir fayl tizimidagi diskni formatlashga nisbatan uni qismlarga ajratish mumkin. Qism - operatsion tizim foydalanuvchiga mantiqiy qurilma sifatida taqdim etadigan fizik diskning uzluksiz qismi.

Mantiqiy vosita aynan foydalanuvchi bilan birga ishlaydi, unga nomlari bo'yicha murojaat etadi, masalan, A, V, S, SYS va boshqalar. Turli turdagi operatsion tizimlar ularning barchasi uchun yagona qismlar to'g'risidagi tushunchadan foydalanadilar, lekin har bir OT turi uchun maxsuslashtirilgan uning mantiqiy vositasi asosida yaratadi. Har bir mantiqiy vosita faqat bitta fayl tizimini yaratishi mumkin.

Ushbu fizik diskdagi turli xil mantiqiy vositalar shu va turli xil turdagi fayl tizimlarida joylashadi. 5.3- rasmda ikkita fayl tizimi o'rnatilgan NTFS (C va E qismlar) va bitta FAT (D qism) fayl tizimi o'rnatilgan 3 qismga ajratilgan disk keltirilgan.



Operatsion tizim qismlarni turli xil holatlarini qo‘llab-quvvatlashi mumkin, asosan operatsion tizim modelini yuklash uchun foydalanilishi mumkin bo‘lgan qismlarni qayd etadigan va faqat fayl ma’lumotlarini saqlash va ilovalarni o‘rnatish mumkin bo‘lgan bo‘limlarni tasvirlaydi. Diskning bo‘limlaridan biri faol bo‘ladi. Aynan shu bo‘lim operatsion tizimni yuklovchi hisoblanadi.



5.3- rasm. Ikkita NTFS fayl tizimi va bitta FAT fayl tizimi o‘rnatilgan 3 qismga ajratilgan disk

### ***Faylni manzillash va fizik tashkil etish***

Fayl tizimining fizik tuzilishining muhim komponentlaridan biri faylning fizik tuzilishi bo‘ladi, ya’ni diskga faylni joylashtirish usuli mavjud.

Faylni fizik tashkil etishning asosiy samaradorlik mezonlari quyidagilar:

- ❖ ma’lumotlarga kirish tezligi;
- ❖ faylni axborot manzili hajmi;
- ❖ disk bo‘sh joyi qismi miqdori;
- ❖ faylning maksimal hajmi.

Uzluksiz joylashtirish – fizik tuzilishning sodda ko‘rinishi, ya’ni disk xotirasida uzluksiz ishtirok etishini ko‘rsatuvchi diskning klasterlar ketma-ketligini taqdim etuvchi fayl. Ushbu usulning asosiy afzalligi yuqori tezlikda kira olishi, bu fayl klasterini o‘qish va izlashga sarflanadigan vaqtni minimallashtiradi. Shuningdek axborot manzil hajmini minimallashtirish – fayl hajmi va birinchi raqamli klasterda saqlash yetarli. Ushbu fizik tuzilishning fayl miqdorining

maksimal imkoniyatlari chegaralanmagan. Biroq bu variantda qator kamchiliklarga ega, ya'ni uning mantiqiy soddaligiga qaramasdan amaliyotda qo'llashning murakkabligi hisoblanadi. Ahamiyat berilsa ushbu sxemani joriy qilish oson emas. Darhaqiqat, agar har bir o'zgarishda fayl o'zining hajmini oshirib borsa faylga ajratilgan joyning o'lchami qanday bo'ladi? Yana bir muammo qismlarga bo'linish (fragmentlash).

Faylni fizik tashkil qilishning keyingi usuli – disk xotirasi klasteri ro'yxati bilan bog'langan ko'rinishda joylashishi. Har bir klaster boshlanishida keyingi klasterni ko'rsatishni o'z ichiga oladi. Bu holatda axborot manzili minimallashtiriladi: faylni joylashtirib unga birinchi klaster raqamini berish mumkin bo'ladi. Avvalgi usuldan farqi har bir klaster biron bir fayl bilan klaster zanjiri bo'ylab birlashishi mumkin bo'ladi, shubhasiz, klaster darajasida fragmentlash bo'lmaydi. Fayl klasterlar miqdorini oshirib o'zining mavjudlik davrida o'lchamini o'zgartirishi mumkin. Kamchiligi ushbu fayl joyiga ixtiyoriy kirishni joriy qilish murakkab – faylni klaster tartib raqami bo'yicha beshinchisini o'qish uchun klaster zanjirida joylashgan dastlabki to'rtta klasterni o'qishi kerak bo'ladi. Bundan tashqari fayl ma'lumotlar miqdori ikkilik darajasiga teng bo'lmagan bitta klasterni o'z ichiga oladi, ko'plab dasturlar esa ikkilik darajasiga teng o'lchamda klaster ma'lumotlarini o'qiydi.

Keng tarqalgan qo'llash usullaridan masalan FAT fayl tizimida indeks ro'yxati bilan bog'langan ko'rinishda joylashtirilish bo'ladi. Ushbu usulda avvalgilariga ba'zi bir o'zgartirishlar kiritilgan. Bu yerda xam fayl klaster ro'yxati bilan bog'langan ko'rinishda xotirani ajratadi. Birinchi klaster raqami ushbu fayl xarakteristikalari saqlanadigan katalog yozuvini xotirlaydi. Qolgan manzil axborotlar fayl klasteridan alohida bo'ladi. Diskdagi har bir klaster ba'zi bir element – indeks bilan bog'langan bo'ladi. Indekslar diskning alohida bir qismida joylashadi – bu FAT (File Allocation Table) jadval MS-DOS da bitta klasterni band qiladi. Agar xotira bo'sh bo'lsa, barcha indekslar nol qiymatiga ega bo'ladi. Agar ayrim fayllarga berilgan bo'lsaba'zi klasterlar Nunda bu klasterda indeks ushbu faylning keyingi klasteri M raqamiga yoki bu klaster navbatdagi fayl uchun qabul qilingan maxsus belgiga teng bo'ladi. Faylni avvalgi klaster indeksi Nbelgisini qabul qilishi klasterga yangiddan berilganligini ko'rsatadi.

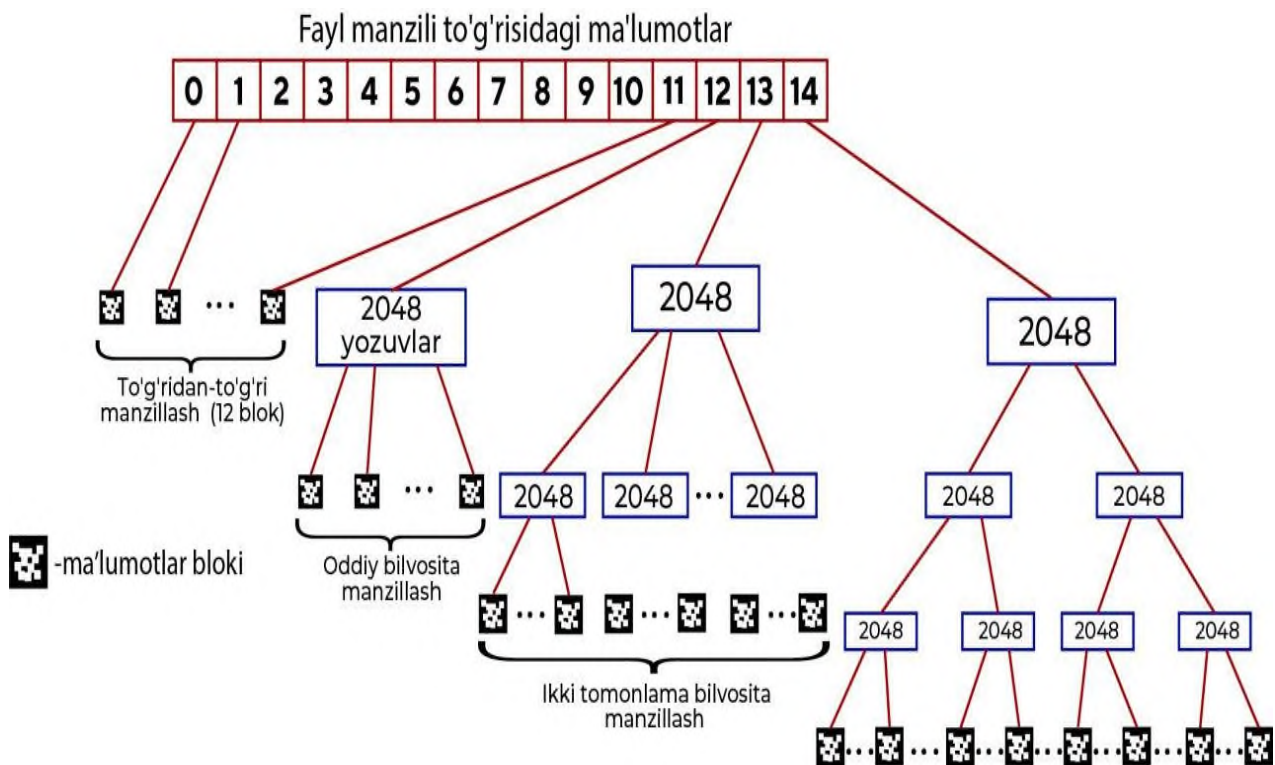
Bunday fizik tashkil qilinish avvalgi usullarning barcha afzalliklarini saqlab qoladi: minimal axborot manzili, fragmentatsiyaning yoʻqligi, oʻlchamning oʻzgarish muammolari mavjud emasligi. Bundan tashqari, ushbu usul qoʻshimcha afzalliklarga ega. Birinchidan fayl klasteriga ixtiyoriy kirish uchun indeks jadvalini oʻz ichiga olgan faqat disk sektorlarini oʻqish yetarli boʻladi, zanjir boʻyicha faylning kerakli klasterlar miqdorini hisoblaydi va kerakli klaster raqamini aniqlaydi. Ikkinchidan, ushbu fayl klasterini toʻliqligicha band qiladi, demak, hajm ikkining darajasiga teng boʻladi.

Faylni fizik joylashtirishning yana bir usuli ushbu faylni band qilgan klaster raqamlarini hisoblab chiqish boʻladi. Bu raqamlar toʻplami va fayl manzil xizmati boʻladi. Ushbu usulning kamchiliklari: manzil uzunligi fayl oʻlchamiga bogʻliqligi va katta fayllar uchun manzil uzunligi sezilarli darajada uzun boʻlishi mumkin. Afzalliklari fayl klasteriga ixtiyoriy kirish tezligining yuqoriligi. Bu yerda faylning ixtiyoriy klasteri manzilini izlash orqali koʻrsatuvchi zanjirni koʻrib chiqishni rad etib, manzilga toʻgʻridan-toʻgʻri kirish qoʻllaniladi. Ushbu usulda klaster darajasida fragmentlash mavjud emas.

Soʻnggi usul Unix OT ning anʼanaviy fayl tizimlari s5 va ufs da foydalaniladigan baʼzi bir modifikatsiyalar boʻladi. Axborot manzili hajmini qisqartirish uchun ufs fayl tizimida chetki qismlarni manzillashni bevosita qoʻshishga imkon beradigan kombinatsiyalashgan klasterini manzillash sxemasidan foydalaniladi. Fayl manzilini saqlash uchun 15 ta maydon ajratiladi, ularning har biri 4 baytdan iborat boʻladi (5.4- rasm). Agar fayl miqdori 12 ta klasterga teng yoki kichik boʻlsa, u holda bu klaster manzilning birinchi oʻn ikkita maydoniga oʻtkaziladi. Agar klaster 8 Kbayt oʻlchamga (ufs qoʻllab-quvvatlaydigan klasterning maksimal oʻlchami) ega boʻlsa, unda  $8192 * 12 = 98\ 304$  baytgacha faylni manzillash mumkin.

Agar fayl oʻlchami 12 klasterdan oshsa, u holda navbatdagi 13- maydon keyingi klaster raqamini emas, navbatdagi fayl klasteri raqami joylashishi mumkin boʻlgan klaster raqamini oʻz ichiga oladi. Bunday koʻrinishda 13- manzil elementi chekka qism manzili uchun foydalaniladi. 8 Kbayt oʻlchamli klaster 13- elementni koʻrsatib navbatdagi maʼlumotlar fayli klasterini 2048 raqamini oʻz ichiga olishi mumkin va fayl hajmi  $8192 * (12 + 2048) = 16\ 875\ 520$

baytgacha o‘shishi mumkin. Agar o‘lchami  $12 + 2048 = 2060$  klastergacha ortsa, u holda 14- maydondan foydalaniladi. Unda har biri ma’lumotlar faylining 2048 ta raqamini saqlaydigan 2048 ta klaster raqamini o‘z ichiga oladigan klaster raqami joylashadi. Bu yerda ikkilik bilvosita manzillashdan foydalaniladi. Uning yordami bilan  $8192 * (12 + 2048 + 2048^2) = 3,43766 * 10^{10}$  baytgacha fayldagi klasterlarni manzillash mumkin.



5.4- rasm. Fayl manzili

Va nihoyat agar fayl  $12 + 2048 + 2048^2 = 4\,196\,364$  klasterini o‘z ichiga olsa, u holda uchtali bilvosita manzillash uchun, so‘ngi 15- maydondan foydalaniladi, ya’ni faylga manzil berish imkoniga ega bo‘lish uchun quyidagi maksimal hajmdan foydalanadi.

$$8192 * (12 + 2048 + 2048^2 + 2048^3) = 7,0403 * 10^{13} \text{ bayt.}$$

Bunday ko‘rinishda, 8 kbayt klaster o‘lchami orqali ufs fayl tizimi 8 milliard klaster saqlanadigan 70 trillion bayt ma’lumotidan iborat faylni qo‘llab-quvvatlaydi. 5.4- rasmda ko‘rinib turibdiki maksimal darajada katta fayl to‘g‘risida axborot manzilini berish uchun manzil markaziy qismida 4 bayt (60 bayt) hamda bilvosita manzil qismida  $1 + (1 + 2048) + (1 + 2048 + 2048^2) = 4\,198\,403$  klasteri bo‘yicha 15 ta element talab etiladi. O‘lchamning kattaligiga

qaramasdan, bu miqdor manzillanyotgan ma'lumot hajmining 0,005 % atrofida bo'ladi.

### **5.3. Fayllar, ularning turlari va atributlari**

#### ***Fayl nomlari***

Barcha fayl turlari belgili nomga ega bo'ladi. Fayl tizimini ierarxik tashkil qilishda odatda quyidagi fayl nomlaridan foydalaniladi:

- ❖ Sodda belgili nom;
- ❖ To'liq belgili nom;
- ❖ Belgilinomgataa'luqliginiko'rsatuvchinomlar;
- ❖ Noyob nomlar (sonli identifikatorlar).

Sodda yoki qisqa belgili nomlar, bitta katalog qismida faylni identifikatsiyalaydi. Sodda nomlar dasturchi va foydalanuvchi fayllariga taqdim etiladi, shu sababli ular belgi ro'yhati va belgi nomlari OT da cheklangan holatda ishtirok etishi kerak. Yaqin vaqtlargacha bu oraliq qisqa bo'lgan. MS-DOS FAT fayl tizimida nom uzunligi bilan chegaralangan (8 ta belgi – nomning o'zi, 3 ta belgi – nomning kengaytmasi), Unix OT ning ko'plab versiyalari qo'llab-quvvatlaydigan s5 fayl tizimida esa oddiy belgili nomlar 14 ta belgidan ko'p bo'lmasligi kerak. Biroq foydalanuvchiga uzun nomlar bilan ishlash qulay, shu sababli ular shu yoki boshqa fayl haqida gapirilayotganligi tushunarli bo'lishi uchun faylga eslab qolish oson bo'lgan nomlar berishga imkon beradi. Shu bois zamonaviy fayl tizimlari, hamda takomillashtirilgan variantda uzun sodda belgili nomli fayllarni qo'llab-quvvatlovchi fayl tizimlari mavjud. Masalan, Windows oilasiga OT tarkibiga kiruvchi NTFS va FAT32 fayl tizimlarida, fayl nomi 255 ta belgigacha bo'lishi mumkin.

Sodda nomli fayl va kataloglarga misol:

- ❖ operatsion tizimlar.doc;
- ❖ project.exe;
- ❖ lecture1.doc;
- ❖ 1-ma'ruza.ppt.

Fayl tizimi ierarxiyasida turli xil fayl, ular turli xil kataloglarda joylashish sharti orqali sodda belgili nomli bir xil nomlarga ega bo'lishga ruhsat etiladi. Buyerdako'pfaylli – bittasisoddanomli xemaishlatiladi.

Bunday tizimda bir ma'noli identifikatsiya fayli uchun to'liq nom bilan nomlashdan foydalaniladi.

To'liq nomildan ushbu fayl gacha bo'lgan yo'l orqali soddabelgili barcha kataloglar zanjirini o'zida taqdim etadi.

Bunday ko'rinishda to'liq nom nomlarning tarkibidabo'ladi,

ya'ni soddanomlar OTajratgichidabir-biridanalohida qabulqiladi.

Ajratgich sifatida ko'pinchato'g'rivateskarichiziqishlatiladi,

shu sababli katalogdizini ko'rsatilmaydi. 6.1b-rasmda Main.exe soddanomiga ega ikkita fayl ko'rsatilgan, biroq ular /depart/Main.exe va /user/Anvar/Main.exe nomlari bilan farqlanadi.

Fayl va uning to'liq nomi o'rtasida daraxtsimon fayl tizimida bitta fayl – bitta to'liq nomga mos bir ma'noni bildiradi. Tarmoq tuzilishiga ega fayl tizimida fayl bir nechta kataloglar tarkibiga kirishi mumkin, demak bir nechta to'liq nomlarga ega bo'ladi, bu yerda bitta fayl – ko'plab to'liq nomga mos keladi. Ikkala holatda ham fayl to'liq nomi bir ma'noli identifikatsiyalaydi. Fayl nisbiy nomga ham tenglashtirilishi mumkin.

Fayl nisbiy nomi “joriy katalog” tushunchasi orqali aniqlaniladi. Fayl tizimi katalogida bittasi har bir vaqt onida har bir foydalanuvchi uchun joriy fayl bo'ladi, chunki bu katalogni OT buyrug'i asosida foydalanuvchining o'zi tanlaydi. Fayl tizimi faylli to'liq nomini shakllantirish uchun nisbiy nomga qo'shimcha sifatida keyinchalik foydalanish maqsadida joriy katalog nomini belgilaydi. Foydalanuvchi nisbiy nomi qo'llash orqali joriy katalogdan ushbu fayl gacha marshrut yo'li orqali katalogni fayl zanjir nomini identifikatsiyalaydi. Masalan, agar joriy katalog /user katalogi bo'lsa, unda faylning nisbiy nomi /user/Anvar/Main.exe quyidagi shaklda ifodalanadi: Anvar/Main.exe.

Belgili nomlar foydalanuvchi uchun qulay, lekin operatsion tizim uchun emas. OT o'zining ichki maqsadlari uchun faylga noyob nomlarni taqdim etadi. Ya'ni bitta fayl – bitta noyob nom kabi. Noyob nom foydalanuvchi fayliga yoki ilovasiga taqdim etilgan bir xil bir yoki bir nechta belgili nomlari mavjud bo'ladi. Noyob nomlar faqat operatsion tizimda foydalanishga mo'ljallanadi va o'zida identifikator sonlarini nomoyon qiladi. Bunday noyob nomli faylga Unix tizimida indeksli tavsiflovchi raqamini misol qilib keltirish mumkin

### ***Fayl turlari***

Fayl tizimi turli xil fayllarni qo'llab-quvvatlaydi va ular quyidagilar bo'lishi mumkin:

- ❖ Odatiy fayl;
- ❖ Fayllar – katalogi;
- ❖ Maxsus fayllar;
- ❖ Nomlanganalar (yoki konveyerlar);
- ❖ Xotirada akslanadigan fayllar.

**Odatiy fayllar**(yoki oddiy fayllar)– o'z ichiga foydalanuvchi yoki foydalanuvchi dasturi va tizimda ishlashi natijasida shakllangan ixtiyoriy xarakterli axborotlarni oladi. Ko'plab zamonaviy operatsion tizimlar (masalan, Unix, Windows) odatiy fayl tuzilishini cheklamaydi va nazorat qilmaydi. Odatiy fayllar u bilan birga ishlaydigan ilovalarni aniqlab oladi. Masalan, matn tahrirlagichi biron bir kod bilan taqdim etiladigan belgi qatorlaridan iborat mantli faylni yaratadi. Bu hujjat, dastur boshlang'ich matni yoki boshqalar bo'lishi mumkin. Matnli faylni ekranda o'qish va printerda chop etish mumkin. Ikkilik fayllari belgi kodlaridan foydalanmaydi, ular ko'pincha murakkab strukturaga ega bo'ladi, masalan dastur kodini bajarilishi yoki arxiv fayllari. Barcha operatsion tizimlar hech bo'lmaganda bitta fayl turini aniqlay olishi kerak. Ko'p foydalanuvchilik tizimli fayl tizimida yana bir masala – bir foydalanuvchi fayliga boshqa bir foydalanuvchini ruhsat etilmagan kirishidan himoyalash qo'shiladi. Fayl tizimi funksiyasi yana bir murakkab ta'rafi operatsion tizim tarmoq tarkibida ishlashi.

**Kataloglar** – fayllar to'plami, biron bir noformal ko'rinishi bo'yicha foydalanuvchi guruhlariga to'g'risida axborot ma'lumotnoma tizimini o'z ichiga oladi. Kataloglar fayl tizimini qo'llab-quvvatlaydigan tizim fayllari. Ko'plab OT lar katalogida istalgan turdagi fayllarni kiritish mumkin. Boshqa kataloglarda esa, izlash uchun qulay bo'lgan daraxtsimon strukturani shakllantirish hisobiga bo'ladi. Kataloglar fayl nomlari va fayllarni boshqarish uchun foydalanadigan xususiyatlar o'rtasidagi yozishmalarni o'rnatadilar. Ushbu xususiyatlar fayl turini, faylga ruxsatni, uni diskka joylashtirishni, hajmini, yaratilish sanasi va vaqtini va boshqalarni o'z ichiga oladi. Katalog bilan qolgan barcha munosabatlar oddiy fayllardagi kabi fayl tizimida ko'rib chiqiladi.

**Maxsus fayllar** – tashqi qurilmalar (terminallar, printerlar va boshqalar) va faylga kirish mexanizmi soddalashtirishda foydalaniladigan kiritish/chiqarish qurilmalari bilan bog'langan

fayllar. Maxsus fayllar faylni o‘qish yoki yozish buyruqlari orqali kiritish/chiqarish operatsiyalarini bajarishda foydalanuvchiga yordam beradi. Blokli maxsus fayllar disklarni modellashtirish uchun ishlatiladi.

**Nomlangan kanallar(yoki konveyerlar)** - bitta dasturning chiqish faylini boshqa dasturning kirish fayliga bog‘lashga imkon beradigan siklik buferlar.

**Xotirada akslanadigan fayllar** - belgilangan virtual manzilda jarayon manzili maydonida joylashtirilgan oddiy fayllar.

### ***Fayl atributlari***

“Fayl” tushunchasi o‘z ichiga faqat uning ma’lumotlarini saqlashdan tashqari atributlarni ham oladi. Atributlar – fayl xususiyatini tavsiflovchi axborot.

Fayl atributlari quyidagilar bo‘lishi mumkin:

- ❖ fayl turi (oddiy fayllar, kataloglar, maxsus fayllar va boshqalar);
- ❖ fayl egasi (joriy foydalanuvchi);
- ❖ fayl yaratuvchisi (faylni yaratgan foydalanuvchi identifikatori);
- ❖ faylga kirishga uchun parol (faylga kirish uchun parol);
- ❖ faylga kirish operatsiyalariga ruhsat berish to‘g‘risida axborot;
- ❖ yaratilgan vaqti, so‘nggi kirish va so‘nggi o‘zgarish to‘g‘risida axborot;
- ❖ faylning joriy hajmi (yozuvdagi baytlar soni);
- ❖ faylning maksimal hajmi (faylni hajmini oshirish mumkin bo‘lgan baytlar soni);
- ❖ “faqat o‘qish uchun” belgisi;
- ❖ “tizim fayli” belgisi;
- ❖ “arxiv fayli” belgisi;
- ❖ “ikkilik/belgili”belgisi;
- ❖ “vaqtinchalik” (jarayon tugagandan so‘ng yo‘q qilish) belgisi;
- ❖ Bloklash belgisi;
- ❖ Fayl yozuvi uzunligi;
- ❖ Yozuvning muhim qismini ko‘rsatish;
- ❖ Kalit uzunligi.



Fayl atributlar to‘plami fayl tizimining o‘ziga xos xususiyatlarini belgilaydi: fayl xarakteristikasida turli xil turdagi fayl tizimlarida turli xil atribut to‘plamlaridan foydalanish mumkin. Masalan, yuqorida keltirilgan ro‘yxatdagi so‘nggi uchta atribut bo‘lmagan strukturalashmagan faylni qo‘llab-quvvatlovchi fayl tizimida fayl tuzilishi bilan bog‘langan bo‘ladi. Bir foydalanuvchilik OT lar atributlar to‘plami foydalanuvchi va himoya qiluvchi (ya’ni fayl egasi, fayl yaratuvchisi, faylga kirish uchun parol, faylga kirishga ruhsat berish to‘g‘risidagi axborot) layoqatiga ega atributlar bo‘lmaydi.

Foydalanuvchi atributga kirish huquqini olish uchun ushbu maqsadda fayl tizimiga taqdim etilgan vosita yordami bilan olishi mumkin. Odatda istalgan atribut ma’nosini o‘qishga ruhsat beriladi, o‘zgartirishga esa faqat ayrimlariga ruhsat beriladi. Masalan, foydalanuvchi faylga kirish qoidalarini o‘zgartirish mumkin, lekin unga faylni joriy hajmini va yaratilgan sanani o‘zgartirishga ruhsat berilmaydi.

#### **5.4. Fayllarni boshqarish va ular ustida amallar bajarish**

##### ***Faylni mantiqiy tashkil etish***

Fayl tarkibidagi ma’lumot umumiy holatda hech qanday *mantiqiy strukturaga* ega bo‘lmaydi. Bu struktura ushbu ma’lumotlarni qayta ishlash uchun mo‘ljallangan dasturlarni ishlab chiqish asosida bo‘ladi. Masalan, matnni ekranda to‘g‘ri ko‘rsatish uchun, dastur alohida so‘zni, qatorni, xat boshini va boshqalarni ajrata olish imkoniyatiga ega bo‘lishi kerak. Bitta element tuzilishini boshqasidan ajratish belgisiga, belgilangan kodlar ketma-ketligi yoki fayl boshlanishiga taa’luqli bo‘lgan ushbu element tuzilishi qiymatlarini aralashtiruvchi sodda dastur xizmat qilishi mumkin. Birinchi holatda barcha ishlar ma’lumotlar ketma-ketligi strukturalashtirilmagan FT da taqdim etiladigan fayl ajratilgan ilovaga to‘liq tegishli bo‘lgan faylning strukturalash va izohlashdan iborat bo‘ladi. Ilova, tizim vositasi barcha ilovalar uchun umumiy foydalaniladigan fayl tizimiga kiritish/chiqarish so‘rovlarini shakllantiradi, masalan, yozish va o‘qish uchun zarur bo‘ladigan bayt miqdori va fayl boshlanishidan siljitishini ko‘rsatish. Mantiqiy dasturda joylashgan mos tegishli bayt oqimlari ilovalarga kelib tushadi. Masalan, kompilyator generatsiyalaydi. Shu sababli obyekt moduli saqlanadigan fayl formati faqat dasturga ma’lum bo‘ladi. Aytib o‘tish

joizki, ma'lumotlar to'g'ri yoki noto'g'riligini tekshirish ularning fayl tizimida saqlanish yo'llariga hech qanday bog'liq emas.

Bayt (oqim)ni strukturalashmagan ketma-ketlikda taqdim etadigan faylni o'z ichiga olgan fayl modeli Unix OT bilan birga keng tarqalgan, bugungi kunda esa Windows OT larining barcha versiyalarida keng foydalanilmoqda. Faylni strukturalashtirilmagan modeli bir nechta ilovalar o'rtasida faylni taqsimlashni oson tashkil etishga imkon beradi: turli xil ilovalar o'zining tuzilishiga ega bo'lishi va fayl ichidagi ma'lumotlarni sharhlashi mumkin.

Fayl tizimi yetarli miqdorda istalgan yozuvni ajratish maqsadida fayl tuzilishi to'g'risida axborotga ega bo'lishi kerak. FT yozuvga kirish huquqini ilovaga taqdim etadi, so'ng bu yozuvning ichidagi ma'lumotlarni qayta ishlashni ilova bajaradi. Bu yo'lni rivojlantirish natijasida faqat murakkab strukturalarni qo'llab-quvvatlash emas, balki ular bilan bog'liq bo'lgan ma'lumotlar omborini boshqarish tizimi (MOBT) bo'ladi.

Mantiqiy yozuv dasturchi tashqi qurilmalar bilan ma'lumotlar almashish operatsiyasini bajarishi mumkin bo'lgan kam miqdordagi ma'lumotlar elementi bo'ladi. Xatto agar qurilma bilan ma'lumotlarni fizik almashish katta o'lchov birligida amalga oshirilsa, operatsion tizim dasturchini alohida mantiqiy yozuvga kirishini ta'minlashi kerak bo'ladi.

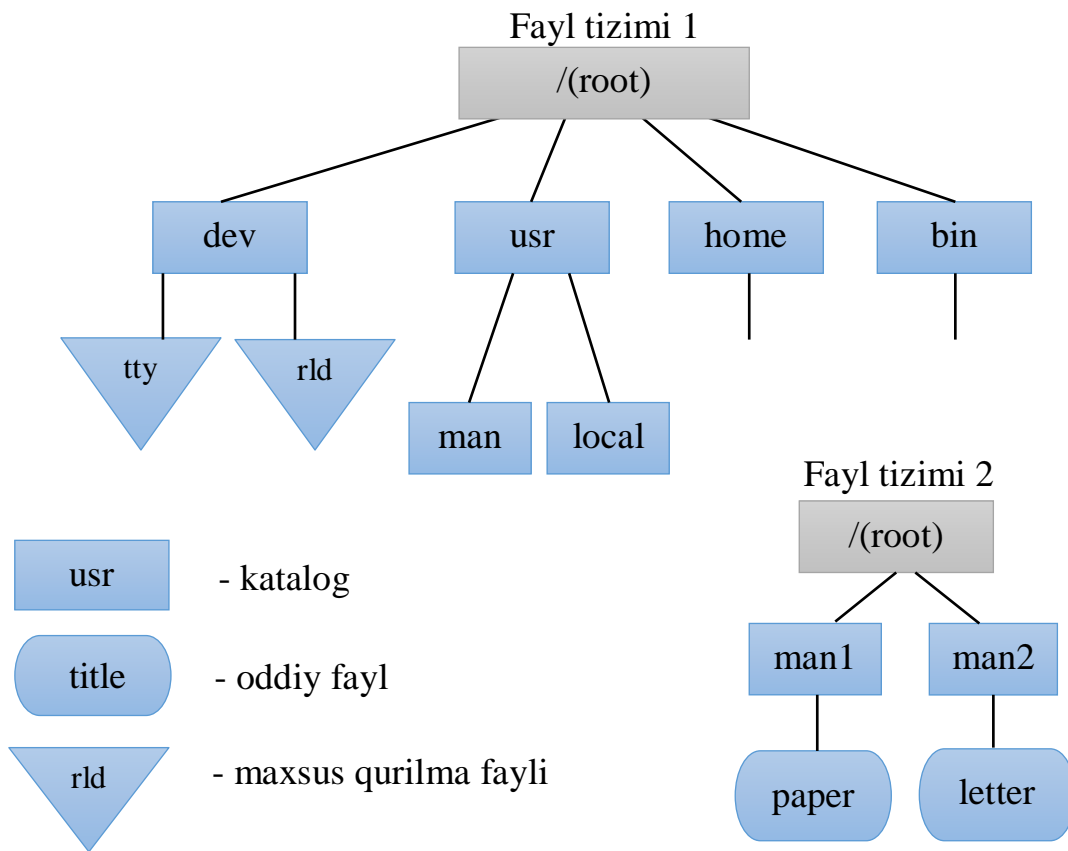
Fayl tizimi mantiqiy yozuvga kirishni ikkita usulidan foydalanishi mumkin: o'qish yoki mantiqiy yozuvlar ketma – ketligini (kirish ketma-ketligi) yozish yoki ko'rsatilgan raqam bilan yozuvga faylni joylashishi (to'g'ridan-to'g'ri kirish).

Shak shubhasiz OT faylda ma'lumotlarni strukturalashning barcha yo'llarini qo'llab-quvvatlashi mumkin emas, shu sababli OT, faylni mantiqiy tashkil etish sxemasida keng tarqalgan uncha katta bo'lmagan sonlar uchun joriy qilishni qo'llab-quvvatlovchi faylni mantiqiy strukturalashni qo'llab-quvvatlaydi.

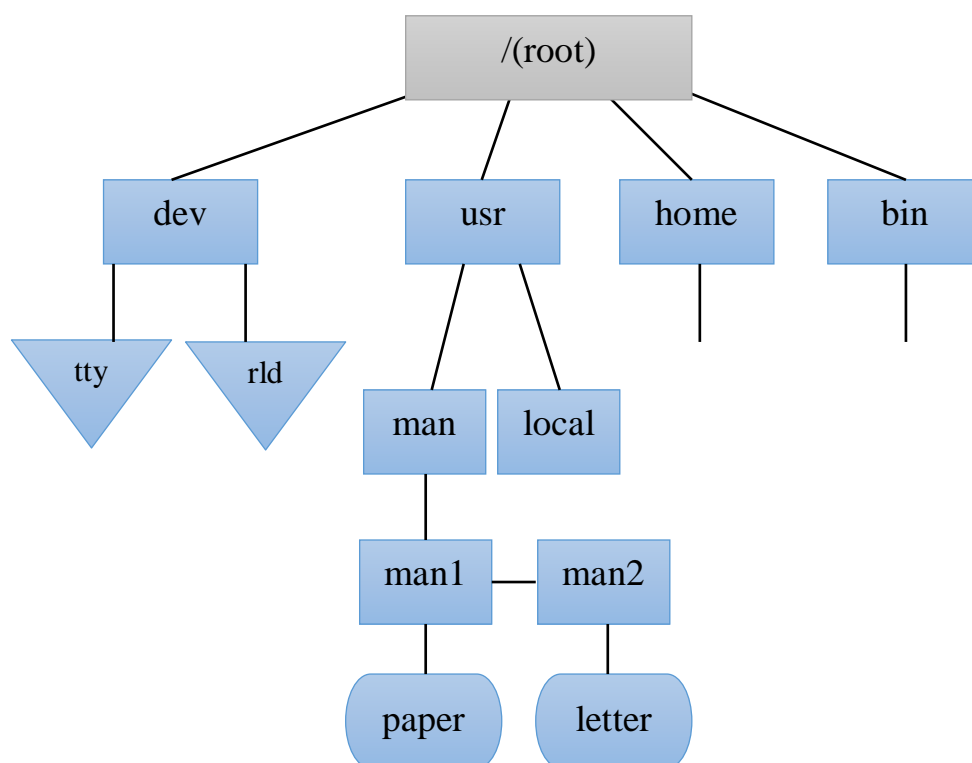
### ***Montaj qilish***

Umuman olganda, hisoblash tizimida bir nechta disk qurilmalari bo'lishi mumkin. Hatto oddiy shaxsiy kompyuterda odatda bitta qattiq disk, bitta disket va CD diskovod mavjud. Kuchli kompyuterlar, qoida tariqasida, ko'plab disk to'plamlari bilan jihozlangan, ularda disk paketlari o'rnatilgan. Bundan tashqari, operatsion tizimdan foydalanadigan bitta fizik qurilma ham bir nechta mantiqiy qurilmalar ko'rinishida taqdim

qilinishi mumkin, xususan, disk maydonini bo'limlarga bo'lish orqali. Savol tug'iladi, bir nechta tashqi xotira qurilmalari bo'lgan tizimda fayllarni saqlashni qanday tashkil qilish kerak?



5.5- rasm. Montaj qilishdan oldingi 2 fayl tizimi



5.6- rasm. Montajdan keyingi umumiy fayl tizimi

Birinchiyechimshundaki, harbirqurilmadaalohidafayltizimi, ya'niushbuqurilmadajoylashganfayllarmavjud; ularboshqaqurilmalardagikatalogdaraxtlaribilanhechqandaybog'liqbo'l magankatalogdaraxtitomonidantasvirlangan. Bunday holda, faylni noyob identifikatsiyalash uchun foydalanuvchi aralash ramziy fayl nomi bilan birga mantiqiy qurilmaning identifikatorini ko'rsatishi kerak. Fayl tizimlarining bunday avtonom mavjudligiga misol operatsion tizimdir. MS-DOS da to'liq fayl nomi mantiqiy diskning harf identifikatorini o'z ichiga olgan. Shunday qilib, A diskida joylashgan faylga kirishda foydalanuvchi ushbu diskning nomini ko'rsatishi kerak: A:\privat\letter\uni\letl.doc1. Boshqa variant - bu faylni saqlashni tashkil qilish, bunda foydalanuvchiga turli xil qurilmalarda joylashgan fayl tizimlarini yagona katalog daraxti tomonidan tasvirlangan yagona fayl tizimiga birlashtirish imkoniyati beriladi. Bunday operatsiya montaj qilish deb ataladi.

Ushbu operatsiya Unix operatsion tizimi misolida qanday amalga oshirilishini ko'rib chiqamiz. OT tizimdagi barcha mantiqiy disk qurilmalari orasidan tizim qurilmasi deb nomlangan bitta qurilmani ajratadi. Turli xil mantiqiy disklarda ikkita fayl tizimlari bo'lsin, disklardan biri tizim diskidir (5.5- rasm). Tizim diskida

joylashgan fayl tizimiga ildiz (root) deb belgilanadi. Ildiz ko‘rinishga ega fayl tizimi bilan fayl ierarxiyasini bog‘lash uchun ba‘zi mavjud kataloglar tanlanadi, bizning misolimizda bu “man” katalogi. Tanlanilgan “man” katalogini montaj qilingandan so‘ng, ikkinchi fayl tizimi katalog ildizi bo‘lib qoladi. Ushbu katalog orqali umumiy daraxtga daraxt osti sifatida bog‘lanib fayl tizimiga montaj qilinadi (5.6- rasm). Umumiy fayl tizimini montaj qilingandan so‘ng, foydalanuvchi uchun ildiz va montaj qilingan fayl tizimlari o‘rtasida mantiqiy farq bo‘lmaydi, xususan, fayl nomlari xuddi tizim boshidan birlashtirilgandek amalga oshiriladi.

### *Fayl operatsiyalari*

Fayllardagi operatsiyalar to‘plami va ayniqsa ularning belgilari tizimdan tizimga qarab farq qilishi mumkin, ammo fayllar bo‘yicha quyidagi asosiy operatsiyalarni ajratib ko‘rsatish mumkin:

- ❖ **Faylni yaratish** (create). Fayl sarlavhasi yaratiladi;
- ❖ **Faylga yozish** (write). Qoida tariqasida, bu yozuvlar yoki bloklar orqali amalga oshiriladi - ma‘lumotlarning katta mantiqiy birliklari, kiritish/chiqarish operatsiyalarini optimallashtirish uchun bir nechta yozuvlarni birlashtiradi;
- ❖ **Fayldan o‘qish** (read). Odatda yozuvlar yoki bloklar tomonidan ham amalga oshiriladi;
- ❖ **Fayl ichidagi manzilni (joylashgan joyini) qidirish** (seek); Joylashuv yozuv yoki blok raqami bilan yoki faylning boshlanishini (birinchi yozuvdan oldingi holat) yoki faylning oxirini ko‘rsatuvchi maxsus nomlar bilan belgilanadi (oxirgi yozuvdan keyingi holat);
- ❖ **Faylni o‘chirish** (delete). Fayl tizimining bajarilishiga qarab, xatolarni o‘chirib tashlash halokatli (UNIX) yoki tuzatilishi mumkin (MS DOS);
- ❖ **Faylni qisqartirish** (Truncate);
- ❖ **Faylni ochish** (open) – katalog tuzilmasidan faylni belgili nomi (yo‘li) bo‘yicha qidirish va uni o‘qish;
- ❖ **Faylni yopish** (close) – bufer tarkibini fayl bloklariga yozish, tashqi xotiradagi faylni hozirgi holatiga muvofiq yangilash, fayl bilan bog‘liq bo‘lgan asosiy xotiradagi barcha tuzilmalarni bo‘shatish.
- ❖ **Faylni nusxalash** (copy) – fayldan nusxa olish;

❖ **Faylni ko‘chirish** (move) – faylni boshqa joyga ko‘chirib o‘tkazish.

Fayllarni almashish operatsiyalarini bajarish uchun (o‘qish, yozish), qoida tariqasida, fayl ochilishi kerak. Faylni yopish foydalanuvchi jarayonlari majburiyati hisoblanadi.

## **5.5. Fayllarni himoyalash va ularga murojaatlarni boshqarish**

Axborot kompyuter tizimida saqlanganda, biz uni fizik shikastlanishdan (ishonchlilik muammosi) va noto‘g‘ri kirishdan (himoya muammosi) himoya qilishimiz kerak. Ishonchlilik odatda fayllar nusxalari tomonidan ta‘minlanadi. Ko‘pgina kompyuterlarda fayl tizimi ishdan chiqqan yoki buzilgan taqdirda ularni qayta tiklash uchun, disk fayllarini ma‘lum bir vaqtda (kuniga bir marta, haftada yoki oyda bir marta) avtomatik ravishda (yoki kompyuter operatorining aralashuvi bilan) ularni nusxalarini lentalarga saqlash uchun ko‘chiradigan tizim dasturlari mavjud. Qurilma bilan bog‘liq muammolar tufayli (o‘qish yoki yozish xatoliklari kabi) fayl tizimlari shikastlanishi mumkin, yuklanishni ortishi yoki uzilishlar, o‘qish/yoziq boshining ishdan chiqishi, haroratning haddan tashqari o‘zgarishi orqali bo‘lishi mumkin. Fayllar tasodifan o‘chirilishi mumkin. Fayl tizimi dasturidagi xatolar, shuningdek, fayl tarkibining yo‘qolishiga olib kelishi mumkin. Himoya turli xil usullar bilan ta‘minlanishi mumkin. Bir foydalanuvchilik tizimda noutbuk yoki ish stoli kompyuterlari xavfsizligini xonani qulflash yoki boshqa usullar orqali himoya qila olamiz. Biroq, ko‘p foydalanuvchilik tizimlarda boshqa xavfsizlikni ta‘minlash mexanizmlariga ehtiyoj seziladi.

Birinchi operatsion tizimlar foydalanuvchilar o‘rtasida fayl almashishni qo‘llab-quvvatlay olmadi. Masalan, oldingi tizimlarda 10 ta foydalanuvchiga xizmat ko‘rsatish uchun kompilyatorning 10 ta nusxasi talab qilingan. Bugungi tizimlar tizimdagi faol dasturlarning sonidan qat‘iy nazar barchaga xizmat qilish uchun bitta nusxani talab qiladi. Aslida, har qanday faylni almashish mumkin - ma‘lumotlar fayllari va foydalanuvchi dasturlari fayllaridan tortib tizim fayllarigacha. Fayllarni almashishning foydalari juda ko‘p.

### ***Kirish usullari***

Fayllarni himoya qilish zarurati fayllarga kirish imkoniyatining bevosita natijasidir. Boshqa foydalanuvchilarning fayllariga kirishga ruxsat bermaydigan tizimlar himoyaga muhtoj emas. Shunday qilib, kirishni taqiqlash orqali biz to'liq himoyani ta'minlashimiz mumkin. Shu bilan birga, siz hech qanday himoyasiz erkin kirishni ta'minlashingiz mumkin. Umumiy foydalanish uchun ikkala yondashuv ham juda ekstremaldir. Nima kerak - bu boshqariladigan kirish. Xavfsizlik mexanizmlari amalga oshirilishi mumkin bo'lgan fayllarga kirish turlarini cheklash orqali kirishni boshqarishni ta'minlaydi. Bir nechta omillarga bog'liq holda kirishga ruxsat beriladi yoki rad etiladi. Bir necha xil operatsiyalar turlarini boshqarish mumkin:

- ❖ Read. Fayldan o'qish.
- ❖ Write. Faylni yozish yoki qayta yozish.
- ❖ Execute. Faylni xotiraga yuklash va uni bajarish.
- ❖ Append. Fayl ohiriga yangi ma'lumot qo'shish.
- ❖ Delete. Faylni o'chirish va qayta foydalanish uchun joyini bo'shatish.

- ❖ List. Fayl nomi va atributlarining ro'yhati.

Fayl nomini o'zgartirish, nusxalash va tahrirlash kabi boshqa operatsiyalarni ham boshqarish mumkin. Biroq, ko'plab tizimlar uchun ushbu yuqori darajadagi funksiyalar pastki darajadagi tizim chaqiriqlarini amalga oshiradigan tizim dasturi tomonidan amalga oshirilishi mumkin. Himoya faqat pastki darajada ta'minlanadi. Masalan, faylni nusxalash oddiygina o'qish talablari ketma-ketligi bilan amalga oshirilishi mumkin. Bunday holda, o'qish huquqiga ega foydalanuvchi, shuningdek faylni nusxalash, chop etish va boshqalarga olib kelishi mumkin. Ko'plab himoya mexanizmlari taklif qilingan. Ularning har biri o'zining afzalliklari va kamchiliklariga ega va mo'ljallangan maqsadga mos bo'lishi kerak. Tadqiqot guruhining faqat bir nechta a'zolari foydalanadigan kichik kompyuter tizimi, masalan, tadqiqot, moliya va kadrlar operatsiyalari uchun foydalaniladigan yirik korporativ kompyuter kabi himoya turlariga muhtoj bo'lmasligi mumkin.

### ***Kirishni boshqarish***

Himoyalash muammosiga eng keng tarqalgan yondashuv - foydalanuvchining identifikatoriga bog'liq holda kirishni ta'minlash.

Turli foydalanuvchilarga fayl yoki katalogga kirishning har xil turlari kerak bo'lishi mumkin. Shaxsga asoslangan kirishni amalga oshirishning eng keng tarqalgan sxemasi bu - har bir foydalanuvchi uchun ruxsat berilgan foydalanuvchi nomlari va kirish turlarini belgilaydigan kirishni boshqarish ro'yxatini (ACL - access-control list) har bir fayl va katalog bilan bog'lashdan iboratdir. Foydalanuvchi ma'lum bir faylga kirishni talab qilganda, operatsion tizim ushbu fayl bilan bog'liq kirishlar ro'yxatini tekshiradi. Agar ushbu foydalanuvchi so'raladigan kirish uchun ro'yhatda ko'rsatilgan bo'lsa, kirish huquqiga ega bo'ladi. Ushbu yondashuv murakkab kirish metodologiyalaridan foydalanishning afzalliklariga ega. Kirish ro'yxatlaridagi asosiy muammo bu ularning uzunligidir. Agar biz hammaga faylni o'qishga ruxsat berishni istasak, unda biz o'qish huquqiga ega bo'lgan barcha foydalanuvchilarni ro'yxatga olishimiz kerak. Ushbu usul ikkita nomaqbul oqibatlariga ega:

- ❖ Bunday ro'yxatni yaratish juda zerikarli va foydasiz vazifa bo'lishi mumkin (ayniqsa tizimdagi foydalanuvchilar ro'yxatini oldindan bilmasak).

- ❖ Ilgari belgilangan o'lchamdagi katalog yozuvi endi o'zgaruvchan o'lchamga ega bo'lishi kerak, bu bo'sh maydonni boshqarish yanada murakkablashishiga olib keladi.

Ushbu muammolarni kirish ro'yxatining qisqartirilgan versiyasidan foydalangan holda hal qilish mumkin. Kirishni boshqarish ro'yxatining uzunligini qisqartirish uchun, ko'pgina tizimlar har bir fayl bilan bog'liq ravishda foydalanuvchilarning uchta sinfini tan oladi:

- ❖ Egasi (Owner). Faylni yaratgan foydalanuvchi uning egasidir;
- ❖ Guruh (Group). Fayldan birgalikda foydalanadigan va shunga o'hshash kirishni talab qiladigan foydalanuvchilar guruhi – bu guruh yoki ishchi guruh.

- ❖ Boshqalar (Others). Tizimdan foydalanishi mumkin bo'lgan barcha boshqa foydalanuvchilar.

So'nggi paytlarda eng keng tarqalgan yondashuv bu - kirishni boshqarish ro'yxatlarini yuqorida tavsiflangan umumiy egasi, guruhi va boshqa foydalanuvchilari uchun kirishni boshqarish sxemasini birlashtirishdir (va amalga oshirish osonroq). Masalan, Solaris uchta standart kirish huquqi toifalarini ishlatadi, ammo batafsil kirishni boshqarish kerak bo'lganda, kirishni boshqarish ro'yxatlarini



muayyan fayllar va kataloglarga qo‘shishga imkon beradi. Tasavvur qilish uchun, yangi kitob yozayotgan Sora ismli odamni olaylik. U loyihaga yordam berish uchun uchta aspirantni (Jim, Don va Jill) yollagan. Kitob matni book.tex deb nomlangan faylda saqlanadi. Ushbu fayl bilan bog‘liq himoya quyidagicha:

❖ Sara (ism) fayllar bilan barcha operatsiyalarni bajarish imkoniyatiga ega bo‘lishi kerak;

❖ Jim, Don va Jill faqat faylni o‘qish va yozish imkoniyatiga ega bo‘lishi kerak; ularga faylni o‘chirishga ruxsat bermaslik kerak.

❖ Boshqa barcha foydalanuvchilar faylni o‘qiy oladilar, lekin yoza olmaydilar. (Sara ushbu matnni iloji boricha ko‘proq odamlar o‘qishlarini va ularni fikrlarini olishga qiziqadi).

Ushbu himoyaga erishish uchun biz Jim, Dawn va Jill a‘zolari bilan yangi guruh tuzishimiz kerak, aytaylik, matnli. Keyin guruh matni nomini book.tex fayli bilan bog‘lash kerak va kirish huquqlari yuqorida keltirilgan qoidalarga muvofiq o‘rnatilishi kerak. Endi Sara 1-bobga vaqtinchalik kirishni xohlagan mehmonni ko‘rib chiqimiz. Matn guruhiga tashrif buyuruvchini qo‘shish mumkin emas, chunki bu unga barcha boblarga kirish huquqini beradi. Fayl faqat bitta guruhda bo‘lishi mumkinligi sababli, Sara 1-bobga boshqa guruhni qo‘sha olmaydi. Shu bilan birga, kirishni boshqarish ro‘yxatining funksional imkoniyatlari qo‘shilishi bilan tashrif buyuruvchi 1-bobning kirish huquqlarini boshqarish ro‘yxatiga qo‘shilishi mumkin.

### ***Unix tizimida ruhsatlar***

UNIX tizimlarida katalog va fayllarni himoya qilish shunga o‘xshash tarzda amalga oshiriladi. Har bir pastki katalog bilan bog‘liq uchta maydon mavjud - egasi, guruhi va boshqalar - ularning har biri uchta rwx bitidan iborat. Shunday qilib, agar foydalanuvchi tegishli maydonchada r bit o‘rnatilgan bo‘lsa, foydalanuvchi pastki katalog tarkibini ro‘yxatlashi mumkin. Huddi shu tarzda, foydalanuvchi o‘z katalogini boshqa joriy katalogga o‘zgartirishi mumkin (aytaylik, foo), agar tegishli maydonda foo pastki katalogi bilan bog‘liq x bit o‘rnatilgan bo‘lsa.

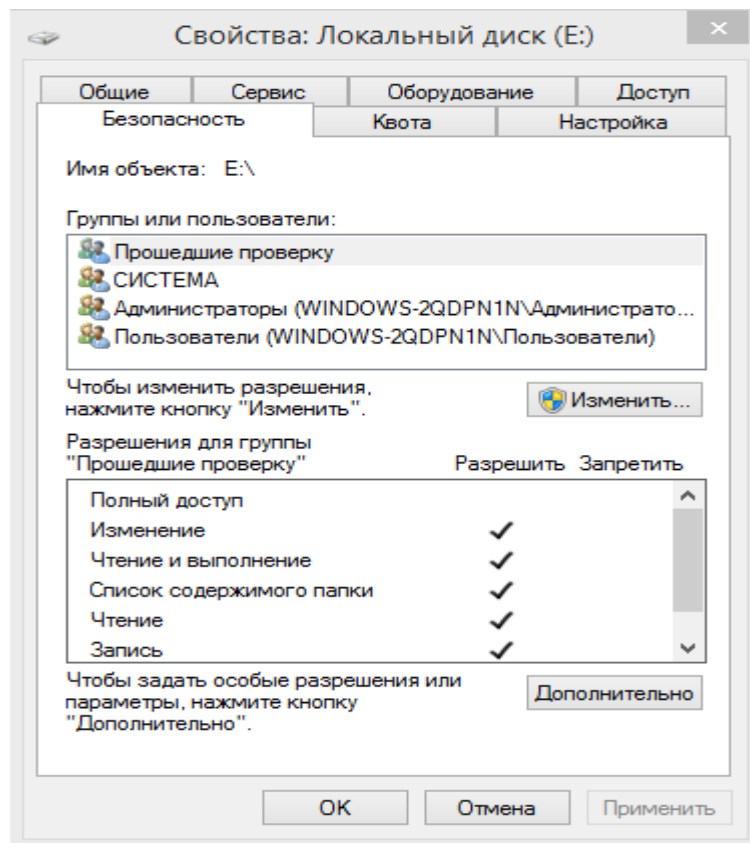
UNIX muhitidan kataloglar namunalari ro‘yxatiga misol quyida keltirilgan:

```
-rw-rw-r-- 1 pbg staff 31200 Sep 3 08:30 intro.ps
```

drwx-----	5 pbg	staff	512	Jul 8 09:33	private/
drwxrwxr-x	2 pbg	staff	512	Jul 8 09:35	doc/
drwxrwx---	2 jwg	student	512	Aug 3 14:13	Student-proj/
-rw-r--r--	1 pbg	staff	9423	Feb 24 2012	program.c
-rwxr-xr-x	1 pbg	staff	20471	Feb 24 2012	program
drwx--x--x	4 tag	faculty	512	Jul 31 06:52	lib/
drwx-----	3 pbg	staff	1024	Aug 29 06:52	mail/
drwxrwxrwx	3 pbg	staff	512	Jul 8 09:35	test/

Birinchi maydon fayl yoki katalogning himoyasini tavsiflaydi. d birinchi belgi sifatida pastki katalogni bildiradi. Shuningdek, faylga havolalar soni, egasining nomi, guruh nomi, baytdagi fayl hajmi, oxirgi o'zgartirilgan sanasi va nihoyat fayl nomi (ixtiyoriy kengaytma bilan) ko'rsatilgan.

Ushbu sxema to'g'ri ishlashi uchun siz ruxsatlarni va kirish ro'yxatlarini diqqat bilan nazorat qilishingiz kerak. Ushbu nazorat bir necha usul bilan amalga oshirilishi mumkin. Masalan, UNIX tizimida guruhlarni faqat obyekt boshqaruvchisi (yoki har qanday superuser) yaratishi va o'zgartirishi mumkin. Cheklangan himoya tasnifi bilan himoyani aniqlash uchun faqat uchta maydon kerak. Ko'pincha har bir maydon bitlarning to'plamini tashkil qiladi va har bir bit u bilan bog'liq kirishga ruxsat beradi yoki rad etadi. Masalan, UNIX tizimi har birida 3 bitdan iborat uchta maydonni aniqlaydi, rwx, bu erda r o'qishni boshqarish, w yozishni boshqarish va x bajarishni boshqarish. Fayl egasi, fayllar guruhi va boshqa barcha foydalanuvchilar uchun alohida maydon saqlanadi. Ushbu sxemada xavfsizlik ma'lumotlarini yozish uchun har bir fayl uchun 9 bit talab qilinadi. Shunday qilib, bizning misolimiz uchun Book.tex fayli uchun xavfsizlik maydonchalari quyidagicha: Sara egasi uchun barcha bitlar o'rnatildi; guruhli matn uchun r va w bitlari o'rnatiladi; va boshqa foydalanuvchilar uchun faqat r bit o'rnatiladi. Yondashuvlarni birlashtirishda qiyinchiliklardan biri bu foydalanuvchi interfeysi. Foydalanuvchilar fayl uchun qo'shimcha ACL ruxsatlari qachon o'rnatilishini aniqlashlari kerak. Solaris misolida standart ruxsatlarga "+" qo'shiladi, chunki:



5.7- rasm. Windowsda fayllarga ruhsat berish oynasi

19 -rw-r--r--+ 1 jim staff 130 May 25 22:13 file1

ACL-ni boshqarish uchun alohida buyruqlar to'plami, setfacl va getfacl ishlatiladi. Windows foydalanuvchilari odatda kirishni boshqarish ro'yxatlarini grafik interfeys orqali boshqaradilar.

5.7- rasmda Windows da NTFS fayl tizimidagi fayllarga ruxsat oynasi ko'rsatilgan. Bu orqali foydalanuvchilarga fayllardan foydalanish huquqlarini belgilash mumkin. Unix, Linux operatsion tizimlarida bo'lgani kabi o'qish, yozish, bajarish, o'zgartirish kabi huquqlarni belgilash mumkin.

### ***Boshqa himoya usullari***

Xavfsizlik muammosiga yana bir yondashuv - parolni har bir fayl bilan bog'lash. Ko'pincha kompyuter tizimiga kirish parol bilan boshqariladi, har bir faylga kirishni ham shu tarzda boshqarish mumkin. Agar parollar tasodifiy tanlansa va tez-tez o'zgartirilsa, ushbu sxema faylga kirishni cheklashda samarali bo'lishi mumkin. Biroq, parollardan foydalanish bir nechta kamchiliklarga ega. Birinchidan, foydalanuvchi eslab qolishi kerak bo'lgan parollar soni ko'payib ketishi mumkin. Ikkinchidan, agar barcha fayllar uchun bitta

parol ishlatilsa, u topilgandan so'ng barcha fayllar mavjud bo'ladi; hech narsa yoki hech narsa himoyasi. Ikkinchidan, agar barcha fayllar uchun bitta parol ishlatilsa, u topib olingandan so'ng barcha fayllarga kirish mumkin bo'ladi; Ba'zi tizimlar foydalanuvchiga ushbu muammoni hal qilish uchun parolni alohida fayl bilan emas, balki qo'shimcha katalog bilan bog'lash imkonini beradi.

Ko'p darajali katalog tuzilishida nafaqat alohida fayllarni, balki pastki kataloglarda fayllar to'plamlarini ham himoya qilishimiz kerak; ya'ni katalogni himoya qilish mexanizmini taqdim etishimiz kerak. Himoyalani kerak bo'lgan katalog operatsiyalari fayl operatsiyalaridan biroz farq qiladi. Katalogdagi fayllarni yaratish va o'chirishni nazorat qilishni xohlaymiz. Bunga qo'shimcha ravishda, biz foydalanuvchi katalogda fayl mavjudligini aniqlay oladimi yoki yo'qligini nazorat qilishni xohlaymiz. Ba'zan, faylning mavjudligi va nomini bilishni o'zi muhimdir. Shuning uchun katalog tarkibini ro'yxatga olish himoyalangan operatsiya bo'lishi kerak. Shunga o'xshab, agar yo'l nomi katalogdagi faylga tegishli bo'lsa, foydalanuvchi katalogga ham, faylga ham kirish huquqiga ega bo'lishi kerak. Fayllar bir nechta yo'l nomlariga ega bo'lishi mumkin bo'lgan tizimlarda (masalan, siklik va umumiy grafikalar), foydalanuvchining ishlatiladigan yo'l nomiga qarab ma'lum bir faylga kirish huquqiga ega bo'lamiz.

### **Nazorat savollari**

1. Fayl tizimi nima?
2. Fayl nima?
3. Fayl menejerining vazifasi nimalardan iborat?
4. Fayldan foydalanishning asosiy maqsadlari nima?
5. Fayl operatsiyalarini keltiring.
6. Fayl turlarini keltiring.
7. Oddiy fayllarga misol keltiring.
8. Kataloglar nima?
9. Maxsus fayllarga misol keltiring.
10. Nomlangan kanallar qanday fayllar?
11. Faylga kirish mexanizmlarini tushuntiring.
12. Fayl tizimlarining ierarxik tuzilishini tushuntiring.
13. Montaj qilish nima va u nima uchun kerak?
14. Fayl atributlari nima?

15. Disklar, qism, sektor va klaster nima?
16. Fayllarga kirish usullari?
17. Fayllarga kirishni boshqarishni tushuntiring.
18. Unix operatsion tizimida fayllarga kirishni boshqarish usullari.
19. Windows operatsion tizimida fayllarga kirishni boshqarish usullari.
20. Unix operatsion tizimida ruhsatlarni boshqarishni tushuntiring.
21. Windows va Unix operatsion tizimlarida fayllarga kirishni boshqarishning farqlarini tushuntiring.
22. Fayl atributlari o'z ichiga qanday ma'lumotlarni oladi?
23. Fayllarni himoyalash usullari.
24. Maxsus fayl va nomlangan fayllarni bir-biridan farqini keltiring.
25. Maxsus fayllarga misol keltiring.

## **VIBOB. KOMPYUTER TARMOQLARINI QURISH VA BOSHQARISH**

### **6.1. Kompyuter tarmoqlari, ularning turlari va asosiy tushunchalar**

Tarmoq texnologiyalarini va tarmoqda ishlashni qo'llab-quvvatlash zamonaviy operatsion tizimlarning ajralmas qismidir. Zamonaviy sharoitda kompyuterlardan foydalanish alohida holda emas, balki taqsimlangan tizim va tarmoqlarda uy yoki ofisning mahalliy tarmog'i, Internet, korporativ tarmoqlarda va boshqalarda amalga oshiriladi.

**Kompyuter tarmog'i** - bu aloqa tizimiga ulangan va tegishli dasturiy ta'minot bilan jihozlangan kompyuterlar majmui bo'lib, tarmoq foydalanuvchilariga ushbu kompyuterlarning resurslariga kirishga imkon beradi. Turli xil kompyuterlar tarmoqni tashkil qilishi mumkin, masalan, kichik mikroprotessorlar, ishchi stansiyalar, mini kompyuterlar, shaxsiy kompyuterlar va superkompyuterlar. Aloqa tizimi har qanday juft kompyuter o'rtasida xabar almashishni ta'minlaydigan kabellar, kommutatorlar, marshrutizator va boshqa qurilmalarni o'z ichiga olishi mumkin.

**Tarmoq operatsion tizimi**- foydalanuvchiga o'z kompyuterlari bilan mustaqil ishlashga imkon beradi va shu bilan tarmoqdagi boshqa kompyuterlarning axborot va qurilma resurslariga kirish imkoniyatini oshiradi.

Tarmoq operatsion tizimining soddalashtirilgan ishlashi quyidagicha amalga oshiriladi. A kompyuteridagi foydalanuvchi o'z faylini tarmoqdagi boshqa B kompyuteriga qo'yishga qaror qildi deylik. Buning uchun u klaviaturada tegishli buyruqni kiritadi. Foydalanuvchi interfeysi uchun javobgar bo'lgan operatsion tizimning dasturiy moduli ushbu buyruqni oladi va uni A kompyuterning mijoz qismiga o'tkazadi. OT mijoz qismi boshqa kompyuter resurslariga to'g'ridan-to'g'ri kira olmaydi (masalan, B kompyuter diskleri yoki kompyuterlariga). U faqat ushbu resurslar joylashgan kompyuterda ishlaydigan OT server qismini "so'rashi" mumkin. Ushbu "so'rovlar" tarmoq orqali uzatiladigan xabarlar shaklida ifodalanadi. Xabarlarda

nafaqat muayyan harakatlarni bajarish buyruqlari, balki ma'lumotlarning o'zi ham bo'lishi mumkin, masalan, ma'lum bir faylning tarkibi.

**Tarmoq (network)** deb kompyuterlar, terminallar, va boshqa qurilmalarning ma'lumot almashishini ta'minlaydigan aloqa kanallari bilan o'zaro bog'langan majmuiga aytiladi. Barcha tarmoq tizimlarining umumiy maqsadi foydalanuvchilarga tarmoq resurslaridan foydalanishda qulaylikni ta'minlash hisoblanadi. Kompyuterlarni tarmoqga birlashtirish quyidagi masalalarni yechishga imkon beradi:

- ❖ ma'lumotlardan birgalikda foydalanish (misol uchun, fayllar);

- ❖ qurilma vositalaridan birgalikda foydalanish (misol uchun, printerlar, modemlar va hokazolar);

- ❖ dasturiy resurslardan birgalikda foydalanish (misol uchun, mijoz-server ko'rinishidagi dasturlar);

- ❖ tarmoq tugunlarida yagona xavfsizlik siyosatini ta'minlash (misol uchun, lokal tarmoqni Internetga ulash davrida serverda ishchi stansiya xavfsizligini sozlash);

- ❖ tarmoq tugunlarini vakolatlarini chegaralash (misol uchun, korxonaning har xil bo'linmalari orasida vakolatlarni taqsimlash);

- ❖ birgalikda foydalaniladigan ma'lumotlarni xavfsizligini ta'minlash (misol uchun, server tomonidan zaxira nusxalarni olish);

- ❖ tarmoq tugunlari orasida ma'lumot almashinuvini ta'minlash (misol uchun, elektron pochta ishlatishda);

- ❖ masofadagi saytlarda fayllarni almashish va chop etish. Tarmoqlar foydalanuvchilarga, boshqa tarmoq foydalanuvchilari foydalanishlari uchun, masofadagi saytlarda ma'lumotlarni chop etishga imkon beradi;

- ❖ taqsimlangan ma'lumotlar omborida axborotlarni qayta ishlash;

- ❖ hisoblashlarni tezlashtirish.

Funksiyalar tarmoqdagi kompyuterlar o'rtasida qanday taqsimlanganiga qarab, ular uchta turli rollarni bajarishi mumkin:

- ❖ faqat boshqa kompyuterlarning so'rovlariga xizmat ko'rsatishga mo'ljallangan kompyuter ajratilgan tarmoq serveri rolini o'ynaydi;

❖ boshqa mashina resurslariga so'rovlar yuboradigan kompyuter mijoz tuguni sifatida ishlaydi;

❖ mijoz va server funksiyalarini birlashtirgan kompyuter – bir rangli tugun hisoblanadi.

Tarmoqdagi kompyuterlarning vakolatlarini taqsimlanishi bo'yicha bir rangli, serverli va gibridliga bo'lish mumkin.

❖ Bir rangli tarmoqda barcha kompyuterlar bir xil huquqga va majburiyatga ega bo'ladi. Har bir kompyuter o'z resursini tarmoqning boshqa a'zolariga beradi va bir vaqtning o'zida ularning resurslaridan foydalanadi.

❖ Serverli tarmoqlarda bir yoki bir nechta kompyuterlar (serverlar) o'z resurslarini tarmoqdagi boshqa kompyuterlarga (mijozlarga) beradi. Bunda server mijozlarning resurslaridan foydalanmaydi.

❖ Gibrid tarmoqlarda bir rangli va server tarmoqlarning belgilari birlashgan tarmoq hisoblanadi.

Misol uchun, bir tarmoq, bir qism kompyuterlar uchun server bo'lib turib, boshqa serverning mijoz bo'lishi mumkin. Ko'pchilik tarmoqlar gibrid bo'lishadi.

Tarmoq texnologiyalari – bu shunday texnologiyaki ular shaxsiy kompyuterda ma'lumotlarni to'plash, saqlash, uzatish va qayta ishlash texnologiyasini aloqa texnikasi bilan birlashtiradi.

Tarmoqlar ko'pincha shartli ravishda quyidagi turlarga bo'linadi:

- ❖ LAN (Local Area Network);
- ❖ MAN (Metropolitan Area Network);
- ❖ GAN (Global Area Network);
- ❖ WLAN (Wireless Local Area Network).

### Mahalliy tarmoq

Mahalliy tarmoq (LAN – Local area network) — bitta ofis binosida, omborxonada, kampusda yoki shunga o'xshash hisoblash muhitida joylashgan konfiguratsiyani belgilaydi. Bunday tarmoqqa odatda bitta tashkilot tomonidan egalik qilinadi, foydalaniladi va boshqariladi va kompyuterlarga umumiy aloqa liniyasi orqali to'g'ridan-to'g'ri aloqa o'rnatishga imkon beradi. Qoida tariqasida, bu bitta umumiy zonada joylashgan shaxsiy kompyuterlar yoki ishchi stansiyalar to'plami hisoblanadi.

Garchi mahalliy tarmoq aniq belgilangan mahalliy hudud bilan fizik jihatdan chegaralangan bo'lsa-da, uning aloqasi bu sohada



cheklanmagan, chunki mahalliy tarmoq katta aloqa tarmog'ining komponenti bo'lishi mumkin va boshqa tarmoqlarga ko'prik (bridge) yoki shlyuz (gateway) orqali kirishni ta'minlaydi.

Ko'prik - bu bir xil protokollar yordamida ikki yoki undan ko'p geografik jihatdan uzoq masofadagi mahalliy tarmoqlarni bog'laydigan qurilma va uni boshqarish uchun dasturiy ta'minotdir. Masalan, Ethernet tarmoq texnologiyasidan foydalangan holda ikkita mahalliy tarmoqni ulash uchun oddiy ko'prikdan foydalanish mumkin. Mahalliy tarmoq shina, xalqa yoki yulduz topologiyalari asosida tashkil qilinadi (tarmoq topologiyasi ushbu mavzuda keyinroq muhokama qilinadi).

O'z navbatida, shlyuz, bu ikki yoki undan ko'p mahalliy tarmoqlarni yoki turli protokollardan foydalanadigan tizimlarni ulash uchun ishlatiladigan yanada murakkab qurilma va dasturiy ta'minotdir. Shlyuz qurilma va dasturiy ta'minotdagi nomuvofiqliklarni bartaraf qilib, bitta tarmoqning protokolini boshqasiga tarjima qiladi. Yuqori tezlikda ishlaydigan mahalliy tarmoqlarda ma'lumotlarni uzatish tezligi sekundiga 100 megabitdan 40 gigabitgacha oshirmoqda.

#### Hududiy tarmoq

Hududiy tarmoq (MAN – Metropolitan area network) — mahalliy tarmoqqa qaraganda katta, lekin global tarmoqdan (WAN) kichik bo'lgan (masalan, shahar, viloyat, tuman) fizik hududni qamrab oladigan tarmoq infratuzilmasi hisoblanadi. Hududiy tarmoqni boshqarish bitta tashkilot, yirik korxonalar orqali amalga oshiriladi.

#### Global tarmoq

Global tarmoq (GAN – Global area network) — juda katta hududda boshqa tarmoqlarga ulanishni ta'minlaydigan tarmoq infratuzilmasi hisoblanadi. Global tarmoqlarni boshqarish odatda aloqa operatorlari yoki Internet-provayderlar orqali boshqariladi. Global tarmoqning asosiy komponentlari quyidagilarni tashkil qiladi:

- ❖ Global tarmoq shaharlar, hududlar, davlatlar, yoki qit'alar kabi juda katta hududlardagi lokal tarmoqlarni birlashtiradi;
- ❖ Global tarmoqlarni boshqarishni odatda turli aloqa operatorlari amalga oshiradi.

Global tarmoqlarda keng doiradagi aloqa vositalaridan, masalan, sun'iy yo'ldosh va mikro to'lqinlardan foydalaniladi. Global

tarmoqlar mahalliy tarmoqlarga qaraganda nisbatan sekinroq hisoblanadi.

#### Simsiz mahalliy tarmoq

Simsiz mahalliy tarmoq (WLAN – Wireless local area network) — bu tarmoq doirasida joylashgan kompyuterlarni yoki ish stansiyalarini ulash uchun simsiz texnologiyalardan foydalanadigan mahalliy tarmoq.

Tarmoqlarga nisbatan, operatsion tizimlarni tarmoq operatsion tizimlari va taqsimlangan operatsion tizimlarga bo‘lish mumkin. Tarmoq operatsion tizimi muhitida ishlayotgan foydalanuvchi, kompyuter tarmog‘idagi har qanday mashinada biron bir vazifani bajarishi mumkin bo‘lsa ham, har doim o‘z vazifasini qaysi mashina bajarishini biladi. Odatda, foydalanuvchi ishi foydalanuvchi kirgan mashinada bajariladi. Agar u boshqa kompyuterda vazifani bajarishni istasa, u masofadan kirish buyrug‘idan foydalanib ushbu mashinaga masofadan mantiqiy kirishni amalga oshirishi yoki masofadan turib bajariladigan maxsus buyruqni kiritishi kerak. Ideal holda, tarmoq OT foydalanuvchiga tarmoq shaklida emas, balki, yagona markazlashtirilgan virtual mashinaning resurslari shaklida tarmoq resurslarini taqdim etishi kerak. Bunday operatsion tizim uchun maxsus nom ishlatiladi - taqsimlangan operatsion tizimlar. Taqsimlangan operatsion tizimlar kompyuter tizimi miqyosida yagona operatsion tizim sifatida mavjud. Taqsimlangan operatsion tizimlar boshqaruvida ishlaydigan barcha tarmoq kompyuterlari ushbu global operatsion tizim funksiyalarining bir qismini bajaradi. Taqsimlangan operatsion tizim foydalanuvchisi, umuman aytganda, uning ishi qaysi mashinada bajarilayotgani to‘g‘risida ma’lumotga ega bo‘lmaydi. Hozirgi vaqtda deyarli barcha operatsion tizimlar tarmoq sinfiga tegishlidir. Zamonaviy kompyuter tarmog‘i har bir kompyuterga o‘rnatilgan bir qator tarmoq operatsion tizimlari nazorati ostida ishlaydi. Ushbu operatsion tizimlar bir xil yoki boshqacha bo‘lishi mumkin. Masalan, bir xil Unix operatsion tizimlar tarmoqdagi barcha kompyuterlarda ishlashi mumkin. Yana ham aniqroq variant - bu turli xil operatsion tizimlarga ega tarmoqdir, masalan, ba’zi kompyuterlar Unix FreeBSD, boshqalari CentOS, qolganlari Windows Server-da ishlaydi. Ushbu operatsion tizimlarning barchasi bir-biridan mustaqil ravishda ishlaydi, ularning har biri o‘z jarayonlarini yaratish, yakunlash va mahalliy resurslarni

boshqarish bo'yicha mustaqil qarorlar qabul qiladi. Ammo har qanday holatda ham, tarmoq orqali ishlaydigan kompyuterlarning operatsion tizimlari tarmoqdagi turli xil kompyuterlarda ishlaydigan jarayonlarning o'zaro ta'sirini tashkil qilish va ushbu kompyuterlarning resurslarini tarmoq foydalanuvchilari o'rtasida almashish uchun o'zaro kelishilgan aloqa protokollar to'plamini o'z ichiga olishi kerak.

## **6.2. Kompyuter tarmoqlarining tuzilishi va topologiyalari**

Har qanday tarmoq tizimidagi saytlar ma'lum bir topologiyada, tugunlarni bog'laydigan ulanishlarning (kabellar, simsiz tarmoqlar yoki ikkalasi) geometrik joylashuvida bir-biriga fizik yoki mantiqan ulanishi mumkin. Eng keng tarqalgan topologiyalar shina, yulduz, xalqa, daraxtdir. Har bir topologiyada barcha saytlar o'rtasida tezkor aloqa zarurligi, sayt yoki aloqa liniyasi ishlamay qolish (nosozliklar tufayli) holatlari, uzun aloqa liniyalarining narxi va bitta saytni boshqa ko'plab saytlarga ulashning qiyinligi o'rtasida o'zaro bog'liqliklar mavjud. Tarmoqni loyihalashtiruvchisi uchun juda ko'p yechimlar mavjud va ularning barchasi mijozning talablariga javob berishi mumkin. Loyihalashtiruvchilar qaysi konfiguratsiyani ishlatishni tanlashda to'rtta mezonni hisobga olishlari kerak:

- ❖ Asosiy xarajat - tizimdagi turli saytlarni bog'lash uchun zarur bo'lgan xarajatlari;

- ❖ Aloqa narxi - xabarni bir saytdan boshqasiga yuborish uchun ketadigan vaqt;

- ❖ Ishonchlilik - tizimda havola yoki sayt ishlamay qolgan taqdirda ham ko'plab saytlar bir-biri bilan aloqada bo'lishining kafolati;

- ❖ Foydalanuvchi muhiti - biznesni muvaffaqiyatli investitsiyalash uchun tarmoq javob berishi kerak bo'lgan muhim parametrlar.

Har bir mijoz uchun bir nechta yechimlar bo'lishi mumkin. Eng yaxshi tanlov barcha to'rtta mezonni hisobga olishi kerak.

### ***Shina topologiyasi***

Shina topologiyasida barcha tugunlar 6.1- rasmda ko'rsatilganidek tarmoqning butun uzunligi bo'ylab ishlaydigan bitta aloqa liniyasiga

ulanadi. Qurilmalar kabellar bilan fizik bir-biriga bog‘langan, lekin kabellar markazlashtirilgan boshqaruv mexanizmi orqali o‘tmaydi. Barcha topologiya oxirida signallarni ifodolovchi terminatorlar joylashadi. Xabarlar bitta mashina orqali hammaga jo‘natiladi va faqat u kimga mo‘ljallangan bo‘lsa o‘sha uni qabul qiladi. Bunday tarmoq juda arzon va sozlash oson hisoblanadi. Kamchiligi umumiy kabel va terminator tizimidan chiqishda ishni rad etishi hisoblanadi. Shuningdek tarmoqni shikastlangan joyini topish qiyin.



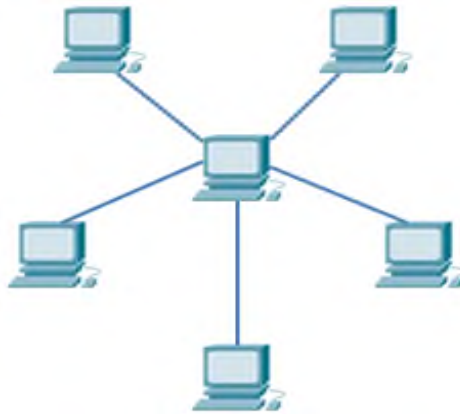
6.1- rasm. Shina topologiyasi asosida qurilgan tarmoq

Bunday ulanishda kompyuterlar axborotni faqat navbat bo‘yicha uzatishi mumkin, chunki aloqa liniyasi bir dona bo‘ladi. Aks holda ustma ust (konflikt) tushishi natijasida uzatiladigan axborot buziladi. Ko‘p holatlarda, shinadan foydalanayotganda boshqa topologiyalarga nisbatan ulanadigan kabelni eng kam miqdori talab qilinadi. Shuni hisobga olish kerakki, har bir kompyuterga (ikkita chetdagilardan tashqari) ikkita kabel keladi, bu esa har doim qulay bo‘lmaydi.

### ***Yulduz topologiyasi***

Ba‘zan konsentrator yoki markazlashtirilgan topologiya deb nomlanadigan yulduz topologiyasida 6.2- rasmda ko‘rsatilganidek, jo‘natuvchidan qabul qiluv-chiga o‘tish paytida barcha uzatiladigan ma‘lumotlar markaziy boshqaruvchidan o‘tishi kerak bo‘lgan a‘nanaviy yondashuvdir, chunki unda bitta markaziy kompyuterga qolgan barcha kompyuterlar ulanadi, shu bilan birga har biri o‘zining alohida aloqa liniyalaridan foydalanadi.

Yulduz turidagi topologiyalarning umumiy kamchiligi, boshqa topologiyalarga qaraganda juda ko‘p kabel sarflanishi hisoblanadi. Bu esa butun tarmoqning narxiga sezilarli ta‘sir ko‘rsatadi. Yana bir kamchiligi sifatida markaziy tugunning ishdan chiqishi butun tarmoqning ishini to‘xtatib qo‘yadi. Topologiyaning bu ko‘rinishi yuqori samaradorligi, tarmoqning to‘g‘ri loyihalashtirilganligi va shikastlangan joylarning oson aniqlashligi bilan ajralib turadi.



6.2- rasm. Yulduz topologiyasi asosida qurilgan tarmoq

### *Xalqa topologiyasi*

Xalqa topologiyasi har bir punktga faqat ikkita liniyalar birlashtirilgan tapmoqni tavsiflaydi. Xalqa topologiyasi optik kabel yordamida lokal kompyuter tapmoqlari, transport tarmoqlari va abonent kirish tarmoqlarida keng qoʻllaniladi.

Xalqa topologiyasida qurilma oʻz qoʻshnisiga ikkita vosita orqali ulanadi. Maʼlumotlar paketi toʻqnash kelmasligi uchun qurilmadan-qurilmaga bir yoʻnalishda uzatiladi. Har bir qurilma maʼlumotlar uzatish uchun oʻz navbatini kutadi. Tizimdagi barcha qurilmalardan istalgan birining rad etilishi chiqishda muammolar keltirib chiqaradi va shikastlangan joyni topish qiyin.



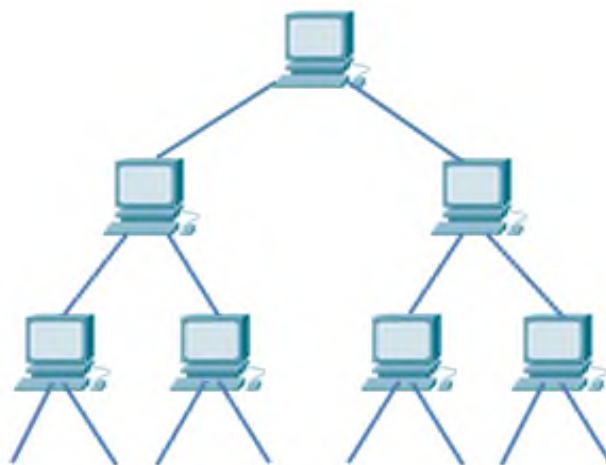
6.3- rasm. Xalqa topologiyasi asosida qurilgan tarmoq

Xalqa topologiyada aloqa liniyalari orqali har bir kompyuter faqat ikkita boshqalari bilan bogʻlanadi: Faqat u biridan axborot oladi,

ikkinchisiga esa – faqat uzatadi. Har bir aloqa liniyasida, yulduz holatiga o‘xshab, faqat bitta uzatuvchi va bitta qabul qiluvchi ishlaydi. Bu narsa tashqi terminallarni qo‘llashdan ozod qiladi. Halqani kamchiliklaridan (yulduzga taqqoslanganida) shuni hisoblash kerakki, tarmoqdagi har bir kompyuterga ikkita kabel olib kelinishi kerak. Ikki qavat xalqa oraliq punktlar o‘rtasida fizik ulanishni juftliklari bilan tashkil qilinadi, bunda axborot oqimi ikki yo‘nalishda yo‘naltiriladi, ulardan biri asosiy, boshqasi esa –zahira bo‘ladi.

### ***Daraxt topologiyasi***

Daraxtsimon topologiya - bir nechta yulduz topologiyalarining kombinatsiyasi hisoblanadi. Ushbu topologiya passiv yoki faol kommutatorlarni to‘g‘ridan-to‘g‘ri ulashni ta‘minlaydi. Daraxt har bir yuqori darajadagi tugunni yulduzlarning birlashmasidan tashkil etuvchi, past darajadagi yulduz shaklidagi tugunlar bilan bog‘langan tarmoqlarning topologiyasi. Bundan tashqari, daraxt ierarxik yulduz deb ham ataladi. Daraxt nomi grafik nazariyasidan kelib chiqqan. Daraxtning birinchi tuguni ildiz deb ataladi. Daraxtlar faol va passiv bo‘lishi mumkin. Faol daraxtlarda kompyuterlar tugun sifatida ishlatiladi, passiv daraxtlar esa kommutatorlarda ishlatiladi. Topologiya ko‘pincha mahalliy tarmoqlarni kichik hajmdagi qurilmalar bilan o‘rnatish uchun asosan korporativ kommutatorlarni yaratishda ishlatiladi. Daraxtsimon topologiya katta tarmoq bo‘lgani uchun, barcha kompyuterlar tarmoqqa yaxshi kirish imkoniyatiga ega bo‘ladi. Bu, aslida, bir nechta kompyuterni bitta daraxtga ulashning eng samarali usuli hisoblanadi.



6.4- rasm. Daraxt topologiyasi asosida ulangan tarmoq

### 6.3. Ma'lumotlar kommutatsiyasi va marshrutlash mexanizmi

#### *Telekommunikatsiya tarmog'i klassifikatsiyasi*

Telekommunikatsiya tarmog'i bir nechta parametrlar bo'yicha toifalanishi mumkin.

Tarmoq hajmi bo'yicha:

❖ Lokal tarmoqlar (*Local Area Network, LAN*) – tarmoq binosi yoki tashkilot;

❖ Hududiy tarmoq (*Metropolitan Area Network, MAN*) – shahar va milliy darajadagi tarmoq;

❖ Global tarmoq (*Wide Area Network, WAN*) – o'n minglab va yuz minglab kompyuterlarni o'z ichiga olgan va katta hududlarni qamrab olgan tarmoq;

Kommutatsiya turi bo'yicha:

❖ Paketlikommutatsiyaasosidagitaroq (masalan, TCP/IP, IPX/SPX, ATM, 3Gmobilaloqataroq'i);

❖ Kanallar kommutatsiyasi asosidagi tarmoq (1G va 2G mobil aloqa tarmog'i, UfTT);

❖ Aralash turdagi tarmoq (masalan, 2,5G mobil aloqa tarmog'i).

Virtual kanallarni o'rnatish bo'yicha:

❖ Virtual kanallarni o'rnatishli (masalan, X.25, Frame Relay, ATM tarmog'i UfTT);

❖ Virtual kanal o'rnatishsiz bog'lanish (masalan, TCP/IP, IPX/SPX);

❖ Protokol stekidan foydalanish bo'yicha.

Tarmoq orqali aloqani tashkil etish muammosi

Aloqa tarmog'ini loyihalashda quyidagi asosiy muammolarni hal qilish kerak:

❖ Nomlash va ruhsat. Ikki jarayon aloqa uchun bir-birini qanday topadi?

❖ Marshrutlash strategiyalari. Xabarlar tarmoq orqali qanday yuboriladi?

❖ Ulanish strategiyalari. Qanday qilib ikkita jarayon xabar almashadi?

❖ Nizolarni hal qilish. Tarmoq bu umumiy resurs. Uni ishlatish bo'yicha ziddiyatli so'rovlar qanday hal qilinadi?

**Tarmoqda nomlarni belgilash va ruxsat.** Tarmoqdagi tizimlar (mashinalar) nomlarga ega. Xabarlar jarayon raqamlari bilan aniqlanadi. Masofadagi tizimda jarayon <host-nomi, jarayon-identifikatori> juftligi bilan aniqlanadi. Tarmoqdagi nomlarga global ruxsat uchun domen nomi tizimi (DNS - Domain Name System) ishlatiladi, bu mashinalarning nomlanish tuzilishini, shuningdek, nomni IP manzilga o'zgartirishni ta'minlaydi. Internetdagi mashinaning domen nomiga misol: tuit.uz bu Toshkent axborot texnologiyalari universitetining domen nomidir. Protokol va DNS xizmatlaridan foydalangan holda, domen nomlari tarmoqdagi maxsus mashinalarning raqamli IP manzillariga aylantiriladi, masalan, 100.200.150.2.

### *Marshrutlash algoritmi*

Har bir tarmoq ma'muri uchun marshrutlash jarayoni eng muhim ahamiyatga ega. Marshrutlash - bu aloqa tarmoqlarida ma'lumotlarning yo'nalishini aniqlash jarayoni. Yo'nalishlar ma'muriy ravishda belgilanishi yoki marshrutlash protokollari yordamida olingan topologiya va tarmoq holati to'g'risidagi ma'lumotlarga asoslangan marshrutlash algoritmlari yordamida hisoblanishi mumkin. Tarmoqlarda marshrutlash uchun maxsus tarmoq qurilmalari marshrutizatorlar (routers) ishlatiladi.

Marshrutizator - bu asosan dasturiy ta'minot asosida boshqariladigan tarmoqlararo qurilma bo'lib, ikkita turli xil mahalliy tarmoqlar yoki turli xil protokol manzillariga ega bo'lgan ikkita tarmoq segmentlari o'rtasida trafikni yo'naltiradi.

Bizga ma'lumki, marshrutlashning statik va dinamik turlari mavjud. Kichik tarmoqlarda statik marshrutlashdan foydalanish ayni muddao. Ammo tarmoq hajmining ortishi (ya'ni foydalanuvchilar soni keskin oshishi) va katta tarmoqlarda statik marshrutlash tarmoq ma'murlariga yetarlicha qiyinchilik tug'diradi. Bunday hollarda marshrutlashning dinamik usuli ancha samarali hisoblanadi.

**Statik marshrutlash.** A dan B gacha bo'lgan yo'l oldindan belgilanadi, qurilmaning ishdan chiqishi sababli undan foydalanishning imkoni bo'lmasa, u o'zgartiriladi. Bunday holda:

❖ Qisqa yo'lni tanlanganligi sababli, aloqa xarajatlari



minimallashtiriladi;

❖ Belgilangan marshrutlashni yuklashni o'zgartirish uchun moslashtirib bo'lmaydi;

❖ Xabarlar qanday yuborilgan bo'lsa, huddi shu tartibda qabul qilinishi ta'minlanadi.

**Dinamik marshrutlash.** A-dan B-ga xabar yuborish uchun yo'l, ushbu xabarni yuborish vaqtida aniqlanadi. Odatda tizim boshqa tizimga ulanish orqali xabar yuboradi. Usul yuklanishlarni o'zgartirishga moslashtirilgan, chunki u juda ko'p ishlatiladigan ulanishlar orqali xabarlarni yuborishdan qochadi. Dinamik marshrutlash algoritmlari tarmoqning o'zgaruvchan sharoitlariga real vaqtda moslashadi. Buni, kelgan marshrutni yangilash xabarlarini tahlil qilish orqali amalga oshiradilar. Xabarda tarmoq almashinuvi sodir bo'lganligini bildirsa, marshrutlash dasturi marshrutlarni qayta hisoblab chiqadi va yangi marshrutlash (yangilanish) xabarlarini yuboradi. Dinamik marshrutlash algoritmlari, kerak bo'lganda, statik marshrutlarni to'ldirishi mumkin.

Ma'lumotlarni uzatish har doim ham muammosiz amalga oshirilmaydi. Masalan, noto'g'ri tugun yoki tarmoq tufayli manzilga yetib bormaslik kabi xatolarga olib keladigan sharoitlar yuzaga kelishi mumkin. Bunday hollarda, marshrutizator va marshrutlash protokollari garchi ular xatoni tuzatishga harakat qilmasa ham, xatolik haqida xabar beradi. Xatolarni tuzatish tarmoq arxitekturasining boshqa darajalaridagi protokollarga qoldiriladi. Quyida biz eng ko'p ishlatiladigan dinamik marshrutizatsiya protokollarini ko'rib chiqamiz.

**RIP** (Routing Information Protocol) – eng oddiy marshrutlash protokoli hisoblanadi. U kichik kompyuter tarmoqlarida qo'llaniladi. U marshrutizatorlarga qo'shni marshrutizator tomonidan olingan yo'l (marshrut) ma'lumotlarini dinamik ravishda yangilashga imkon beradi. RIP protokoli 15 tagacha marshrutizatorlarni qo'llab-quvvatlaydi. RIP protokolida eng kam sonli o'tish har doim tanlanadi. Ushbu vektor masofasini aniqlash algoritmini bajarish oson, ammo zamonaviy tarmoq sharoitida u eng yaxshi bo'lmisligi mumkin, chunki u boshqa muhim omillarni, masalan, tarmoqli kengligi, ma'lumot ustuvorligi yoki tarmoq turini hisobga olmaydi.

**OSPF** (Open Shortest Path First) – bu bog'lanish holati (link-state) texnologiyasiga asoslangan dinamik marshrutlash protokoli va

qisqa yoʻlni topish uchun Dijkstra algoritmidan foydalanadi. OSPF protokoli marshrutizatsiya maʼlumotlarini uzatish uchun xabarlarini almashishda 5 ta turdagi paketlarni ishlatadi. Salomlashish (hello) paketi, maʼlumotlar bazasini tavsiflovchi paket, kanal holati paketi, kanal holatini yangilash paketi, kanal holatini tasdiqlash paketi. Bu paketlar qoʻshni marshrutizatorlarni aniqlash uchun va tarmoq toʻgʻrisida aniq maʼlumotga ega boʻlish maqsadida marshrutizatsiya maʼlumotlarini almashish uchun ishlatiladi.

**BGP** (Border Gateway Protocol) – Internet tarmogʻida ishlatiladigan asosiy dinamik marshrutlash protokoli hisoblanadi. BGP marshrutizatorlari tarmoq mavjudligi haqida maʼlumot almashadilar. Tarmoq maʼlumoti bilan birgalikda, ushbu tarmoqlarning turli xususiyati va holatlari uzatiladi, BGP eng yaxshi yoʻnalishni tanlaydi va marshrut siyosatini yuritadi. Yoʻnalish maʼlumoti bilan uzatiladigan asosiy xususiyatlardan biri bu axborot uzatilgan avtonom tizimlar roʻyxatidir. Ushbu maʼlumot BGP ga tarmoqning qayerda avtonom tizimlari joylashganligini aniqlashga, marshrutlashni ilova qilishni istisno qilishga imkon beradi va shuningdek, undan qoidalarni konfiguratsiya qilishda ham foydalanish mumkin. Marshrutlash bir avtonom tizimdan boshqasiga bosqichma-bosqich amalga oshiriladi. Barcha BGP siyosati asosan tashqi/qoʻshni mustaqil tizimlar bilan bogʻliq. Yaʼni ular bilan boʻladigan oʻzaro munosabatlar qoidalari tasvirlangan. BGP katta hajmdagi maʼlumotlar bilan ishlayotganligi sababli (IPv4 uchun mavjud jadval hajmi 450 mingdan ortiq marshrutni tashkil qiladi), uning konfiguratsiyasi va ishlash tamoyillari ichki dinamik marshrutlash protokollaridan (IGP - interior gateway protocol – ichki shlyuz protokoli) farq qiladi.

Aloqa tarmogʻi uzatilayotgan maʼlumotlarning mazmuni bilan emas, balki maʼlumotni bir nuqtadan boshqasiga oʻtkazish bilan bogʻliq. Tarmoq ulanishlarining (kommutatsiya) quyidagi turlari mavjud:

**Kanallar kommutatsiyasi.** Ketma-ket aloqa kanallarining alohida ulangan qismlaridan kanal hosil qilib, joʻnatish va qabul qilish punktlari oʻrtasida fizik aloqa oʻrnatiladi. Kanal aloqa seansining boshida tashkil etiladi, butun seans davomida saqlanadi va uzatish tugaganidan soʻng uziladi.

#### Afzalliklari

❖ interaktiv rejimda va real vaqtda ishlash qobiliyati;

❖ kanalning to‘liq shaffofligini ta‘minlash.

#### Kamchiliklari

❖ uzluksiz aloqa kanalini yaratish ko‘p vaqtni talab qiladi;  
❖ aloqa kanali resurs va qabul qiluvchi tomonidan monopollashtiriladi, bu tarmoqning samaradorligini pasaytiradi.

Kanallar kommutatsiyasiga misol sifatida telefon aloqasini keltirishimiz mumkin.

**Xabarlar kommutatsiyasi.** Vaqtinchalik ulanish xabarni yuborish davrida o‘rnatiladi. Ma‘lumotlar turli uzunlikdagi xabarlar shaklida uzatiladi. Yuboruvchi faqat qabul qiluvchining manzilini ko‘rsatadi. Kommutatsiya tugunlari kanallarning manzili va mavjudligini tahlil qiladi va xabarni qabul qiluvchining yo‘nalishi bo‘yicha eng yaqin joylashgan tarmoq tuguniga uzatadi.

#### Afzalliklari

❖ tarmoq samaradorligining o‘sishi, chunki xabarni tugundan tugunga uzatgandan so‘ng kanal bo‘shatiladi;  
❖ xabarlarni yetkazib berish yo‘nalishini tanlash imkoniyati.

#### Kamchiliklari

❖ kanallar kommutatsiyasiga qaraganda yetkazib berish vaqtining ko‘payishi;  
❖ interaktiv va real vaqtda ishlash qiyinligi.

Xabarlar kommutatsiyasiga misol sifatida elektron pochta va telekonferensiyani keltirish mumkin.

**Paketlar kommutatsiya.** O‘zgaruvchan uzunlikdagi xabarlar belgilangan uzunlikdagi paketlarga bo‘linadi, va manzilga yuboriladi. Paketlar tarmoq orqali turli yo‘llar bilan uzatilishi mumkin. Paketlar kelganidan keyin xabarlarga qayta yig‘ilishi kerak. Bunga misol TCP/IP asosiy Internet protokoli. Paketlarni uzatishda virtual kanal yaratilishi mumkin. Tugunning vaqtinchalik manbai bir nechta foydalanuvchilar o‘rtasida taqsimlanadi, shunda har bir foydalanuvchiga doimiy ravishda minimal vaqt beriladi va doimiy kirish taassurotlari yaratiladi.

Afzalligi: tarmoqning katta o‘tkazish qobiliyati va ma‘lumotlarni uzatishning eng kam kechikishi.

Kamchiligi: interfaol rejimda va real vaqt rejimida ishlaydigan tizimlar uchun qiyinchilikdir va ba’zida undan foydalana olmaslik hisoblanadi.

Ushbu uchta yondashuvning har biri o'zining afzalliklari va kamchiliklariga ega. Kommutatsiya texnikasining har biri uchun an'anaviy dasturlar mavjud, masalan, telefon tarmoqlari kanallar kommutatsiyasi texnikasidan foydalangan holda qurilgan va qurilmoqda, kompyuter tarmoqlari esa asosan paketlar kommutatsiya texnikasiga asoslangan. Paketlar kommutatsiya texnikasi raqobatchisidan ancha yoshroq va uni muayyan sohalar, masalan, telefoniya (Internet yoki IP-telefoniya shaklida) dan siqib chiqarishga harakat qilmoqda, ammo bu tortishuv hali hal qilinmagan va, ehtimol, ikkita kommutatsiya texnikasi uzoq vaqt birga bo'lib, bir-birini to'ldiradi. Shunga qaramay, ko'plab mutaxassislarning uzoq muddatli bashoratlari ko'ra, kelajak paketlar kommutatsiya texnikasiga tegishli, chunki u yanada moslashuvchan va universaldir.

Agar bir vaqtning o'zida bir nechta tizim ma'lumot uzatish uchun bitta tarmoq bo'limiga kirsa, tarmoqda to'qnashuvlar yuzaga keladi. Tarmoqlarda to'qnashuvlarni hal qilish uchun CSMA/CD (Carrier sense with multiple access; collision detection) usuli. Usulning mohiyati quyidagicha. Tizim xabarning bir vaqtning o'zida tarmoqning boshqa qismida boshqa tizim tomonidan yuborilishini aniqlaydi. Agar ikkita yoki undan ko'p tizim bir vaqtning o'zida xabarlarini yuborishni boshlasa, to'qnashuv qayd etiladi va uzatish to'xtaydi. Ushbu usulning kamchiligi: agar tizimda yuklanish ko'p bo'lsa, unda ko'plab to'qnashuvlar yuz berishi mumkin, bu esa ishlash samaradorligining pasayishiga olib keladi. CSMA/CD usuli tarmoqning eng keng tarqalgan turi Ethernet tarmoqlarida muvaffaqiyatli ishlatilgan.

## **6.4. Kompyuter tarmoqlari arxitekturasini va protokollari**

### ***OSI/ISO modeli***

1970-yillarning oxiriga kelib dunyoda ko'p sonli xususiy aloqa protokollari steklari mavjud bo'lib, ular orasida, masalan, DECnet, TCP/IP va SNA kabi mashhur steklarni eslatib o'tish mumkin. Internet tarmog'iga ulanishning bunday xilma-xilligi turli protokollardan foydalangan holda qurilmalarning nomuvofiqligi (mos kelmasligi) muammosiga olib keldi. O'sha paytda ushbu muammoni hal qilishning bir usuli mavjud tizimlarning kamchiliklarini hisobga olgan holda, yaratilgan barcha tizimlar uchun yagona protokol stekiga

universal o'tish edi. Aynan shu maqsadda OSI modeli ishlab chiqilgan. 1980-yillarning boshlarida bir qator xalqaro standartlashtirish tashkilotlari, xususan Xalqaro standartlashtirish tashkiloti (ISO), ochiq tizimlarning o'zaro ta'sirining standart modelini ishlab chiqdilar (Open System Interconnection, OSI). Ushbu model kompyuter tarmoqlarini rivojlantirishda muhim rol o'ynadi. OSI modeli, birinchidan, paketlar kommutatsiyasi tarmoqlarida tizimlarning o'zaro ta'siri darajasini, ikkinchidan, standart daraja nomlarini va uchinchidan, har bir darajani bajarishi kerak bo'lgan funksiyalarni belgilaydi. OSI modeli ma'lum bir protokol to'plamining bajarilishini tavsiflamaydi.

**Tarmoq protokoli** – hisoblash tizimlarining tarmoq komponentlari orqali almashinayotgan xabarlarining shakli va ketma-ketligini belgilovchi rasmiy qoidalar majmui. Barcha protokollar to'plami protokollar oilasi yoki protokollar steki deb nomlanadi.

Tarmoqning murakkabligi tufayli ishlatiladigan tarmoq protokoli modeli juda ko'p miqdordagi mavhumlikni o'z ichiga oladi. Zamonaviy tarmoqlar murakkab tizimdir: ular ko'p sonli tarmoqlardan iborat bo'lib, ularning har biri odatda har xil tarmoq texnologiyalaridan foydalangan holda turli xil aloqa qurilmalarida qurilgan va turli xil topologiyalarga muvofiq aloqa kanallari bilan bog'langan. ISO aloqa modeli standartiga muvofiq aloqa tarmog'i quyidagi asosiy yetti pog'onalariga (qatlam, sath) bo'linadi:

**Fizik pog'ona** - signallarni uzatish uchun mexanik va elektr qurilmalar. Tarmoq aloqasining eng past darajasi hisoblanadi. Tarmoq qurilmalarini o'z ichiga oladi - tarmoq kabellari, ulagichlar, konsentratorlar va boshqalar. Fizik aloqa kanallari orqali bit oqimini uzatishni amalga oshiradi.

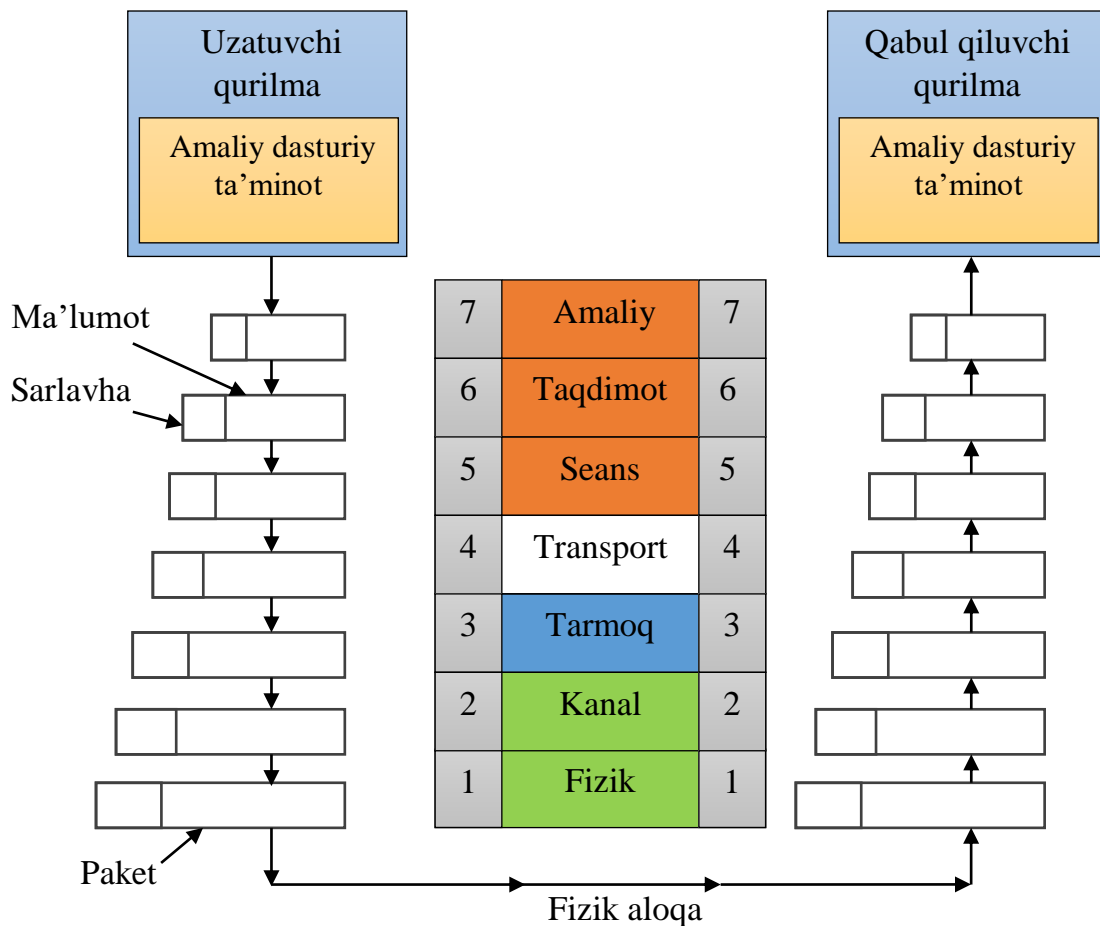
**Kanal pog'onasi** - xatoliklarni aniqlash va fizik darajada xatoliklarni tiklashni o'z ichiga oladi, belgilangan uzunlikdagi kadrlar (frame) yoki qismlarni qayta ishlaydi. Ushbu pog'onada PDU (Protocol Data Unit – ma'lumotlar bloki protokoli) odatda kadr deb ataladi. Paketlar kommutatsiyasi rejimida ishlaydigan birinchi pog'ona hisoblanadi.

**Tarmoq pog'onasi** – aloqa tarmog'ida paketlarni bog'lash va marshrutlashni ta'minlaydi, xususan, chiquvchi paketlarning manzillarini qayta ishlash, kiruvchi paketlarning manzillarini dekoderlash va yuklanish darajasining o'zgarishiga mos keladigan

javob uchun marshrutlash ma'lumotlarini qo'llab-quvvatlashni o'z ichiga oladi.

**Transport pog'onasi** - quyi darajadagi tarmoqqa kirish va mijozlar o'rtasida xabarlarni uzatish, shu jumladan xabarlarni paketlarga ajratish, paketlar tartibini kuzatish, boshqaruv oqimlari va fizik manzillarning generatsiyasi uchun javobgardir.

**Seans pog'onasi** - jarayonlar o'rtasida sessiyalar (aloqani o'rnatadi, tasdiqlaydi va tamomlaydi) yoki aloqa protokollarini amalga oshiradi. Amalda, kam sonli dasturlar seans pog'onasidan foydalanadi va kamdan-kam hollarda alohida protokollar sifatida amalga oshiriladi. Ushbu qatlamning funksiyalari ko'pincha dastur pog'onasining funksiyalari bilan birlashtirilib, bitta protokolda amalga oshiriladi.



6.5- rasm. OSI modeli

**Taqdimot pog‘onasi** - uning nomidan ko‘rinib turibdiki, tarmoq orqali uzatiladigan ma‘lumotlarning tarkibini o‘zgartirmasdan taqdim etilishini ta‘minlaydi. Taqdimot pog‘onasi tufayli, bitta tizimning amaliy pog‘onasi tomonidan uzatiladigan ma‘lumotlar, har doim boshqa tizimning amaliy pog‘onasi tomonidan tushuniladi (ya‘ni tarjimon vazifasini bajaradi). Shu yerda axborot shifrlanadi va deshifrlanadi, va kerak bo‘lgan taqdirda axborotni zichlashtiradi (siqadi). Buning natijasida barcha amaliy xizmatlar uchun ma‘lumotlar almashish mahfiyligi darhol ta‘minlanadi. Bunga misol qilib, TCP/IP stekidagi amaliy pog‘ona protokollari uchun maxfiy xabar almashishni ta‘minlaydigan SSL (Secure Socket Layer) protokolini keltirishimiz mumkin.

**Amaliy pog‘ona** - bu pog‘ona tarmoq foydalanuvchilariga fayllar, printerlar yoki gipermatnli veb-sahifalar kabi umumiy resurslarga kirish, shuningdek, qo‘shma ishlarni tashkil qilish imkonini beradigan turli xil protokollar to‘plami, masalan, elektron pochta protokoli. Amaliy pog‘onada ishlaydigan ma‘lumotlar birligi odatda xabar deb nomlanadi. Bu pog‘onada juda ko‘p turli xil protokollar va tegishli amaliy dastur xizmatlari mavjud. Eng keng tarqalgan tarmoq fayllarini xizmatlarining ba‘zi bir turlari: TCP/IP stekidagi NFS va FTP, Microsoft Windowsdagi SMB, Nowell Netware operatsion tizimidagi NCP.

### ***Ethernet tarmoqlari***

Ethernet (IEEE 802.3) eng keng tarqalgan tarmoqni tashkil etish usuli. Ethernet tarmoqlarining asoschisi Robert Metcalfe (R. Metcalfe 1973). Keyinchalik u tarmoq texnologiyalari sohasidagi eng mashhur kompaniyalardan biri bo‘lgan 3COM kompaniyasiga asos soldi. Ethernet-ning asosiy g‘oyalari har bir tarmoq kompyuteriga birlashtirilgan va tarmoqlarda paketlarning resurslari va qabul qiluvchilarni aniqlash uchun ishlatiladigan koaksial kabel (BNC) va 48 bitli manzil (MAC manzil) dan foydalanishdir. Dastlab Ethernet tezligi 3 Mbit/s gacha bo‘lgan. Hozirgi vaqtda u 100 Gbit/s gacha oshirildi (Gigabit Ethernet).

### ***TCP/IP modeli***

TCP/IP protokoli har bir xostning (kompyuterning) IP-manzillaridan foydalanishga asoslangan: a,b,c,d (barcha to‘rt raqamlar

0 dan 255 gacha oraliqda) va tarmoq orqali xabarda qabul qiluvchi manzili va paket raqamini o'z ichiga olgan belgilangan o'lchamdagi paketlarni yo'naltirishni ta'minlaydi. TCP/IP protokoli tarmoq paketlarini tashishni ta'minlaydi, xabar jo'natuvchi tomonidan paketlarga bo'linadi va xabar qabul qiluvchi tomonidan paketlardan yig'iladi. IP manzil kompyuter foydalanuvchisiga Internet provayder tomonidan doimiy ravishda belgilanishi (statik) mumkin, yoki, Internetga kirishda dinamik (har safar har xil) ravishda tayinlanishi mumkin. TCP/IP uchun eng keng tarqalgan zamonaviy nom Internet protokol to'plami (Internet Protocol Suite) hisoblanadi. TCP/IP stekida 4 ta pog'ona belgilangan.

TCP/IP stekining **amaliy pog'onasi** OSI modelining eng yuqori uchta pog'onasiga mos keladi: amaliy, taqdimot va seans. Tizim tomonidan taqdim etilgan xizmatlarni foydalanuvchi dasturlari bilan birlashtiradi. Turli mamlakatlar va tashkilotlarning tarmoqlarida ko'p yillar davomida ishlatilgan TCP/IP stek ko'plab protokollar va amaliy pog'onadagi xizmatlarni o'z ichiga olgan. Bularga FTP, telnet, SMTP, HTTP va boshqalar kiradi.

TCP/IP stekining **transport pog'onasi** yuqori pog'onaga ikki turdagi xizmatni ko'rsatishi mumkin:

- ❖ kafolatlangan yetkazib berish, TCP (Transmission Control Protocol) protokoli tomonidan ta'minlanadi;

- ❖ imkon darajasida yetkazib berish (ya'ni, yetkazib berish kafolatlanmagan), yoki maksimal kuch bilan, UDP (User Datagram Protocol) protokoli bilan ta'minlanadi.

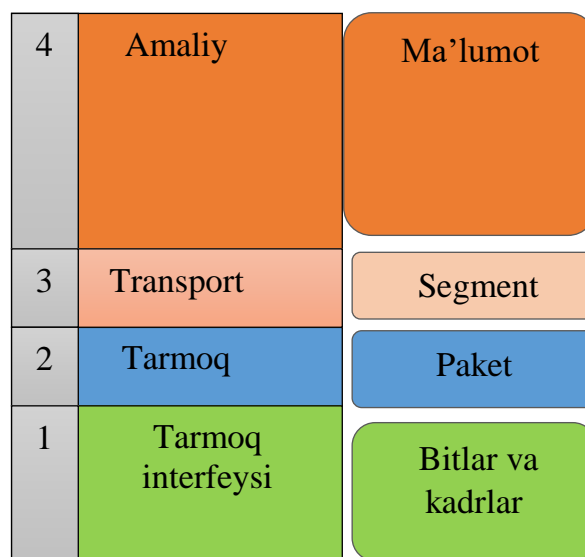
TCP protokoli ma'lumotlarning ishonchli uzatilishini ta'minlash uchun mantiqiy aloqani o'rnatishni ta'minlaydi, bu unga paketlarni raqamlash, qabul qilib olinganligini tasdiqlash, yo'qotilgan taqdirda qayta uzatishni tashkil etish, dublikatlarni tan olish va yo'q qilish, paketlarni ular joylashgan tartibda amaliy pog'onaga yetkazish imkonini beradi. Ushbu protokol, yuboruvchi va qabul qiluvchi kompyuterlarining obyektlariga, ma'lumotlarni ikki tomonlama (dupleks) almashish imkonini beradi. TCP kompyuterlardan birida hosil bo'lgan bayt oqimini tarmoq tarkibiga kiradigan boshqa kompyuterga xatosiz yetkazish imkonini beradi. TCP baytlar oqimini bo'laklarga ajratadi va ularni ishlov beriladigan pog'onaga o'tkazadi.



Ushbu bo‘laklar belgilangan joyga yetkazilgandan so‘ng, TCP yana ularni doimiy ravishda baytlar oqimiga to‘playdi.

Ushbu darajadagi ikkinchi protokol - UDP - bu eng oddiy datagrammali protokol bo‘lib, ma‘lumotlar ishonchli almashinuvi vazifasi umuman qo‘yilmagan yoki yuqori pog‘onada - amaliy pog‘ona yoki foydalanuvchi ilovalari yordamida hal qilinadigan hollarda undan foydalaniladi.

TCP va UDP protokollarini amalga oshiradigan dastur modullari, amaliy pog‘ona protokoli modullari kabi, xostlarga o‘rnatiladi.



6.6- rasm. TCP/IP modeli

Tarmoq pog‘onasi, shuningdek Internet pog‘onasi deb ham ataladi, bu butun TCP/IP arxitekturasining asosi hisoblanadi. Aynan shu pog‘ona funksiyalari OSI modelining tarmoq pog‘onasiga to‘g‘ri keladi, ko‘pgina birlashgan tarmoqlarda paketlarning harakatlanishini ta‘minlaydi. Tarmoq pog‘onasining asosiy protokoli bu IP (Internet Protocol). Uning vazifasi paketni tarmoqlar o‘rtasida tarqatishdir. TCP/IP tarmoq pog‘onasi ko‘pincha IP ga nisbatan yordamchi funksiyalarni bajaradigan protokollarni o‘z ichiga oladi. Avvalo, bu tarmoq topologiyasini o‘rganadigan, marshrutlarni aniqlaydigan va marshrutlash jadvallarini tuzadigan RIP, OSPF, IS-IS va BGP marshrutlash protokollari, ular asosida IP protokoli paketlarni to‘g‘ri yo‘nalishda harakatlantiradi. Xuddi shu sababga ko‘ra tarmoq

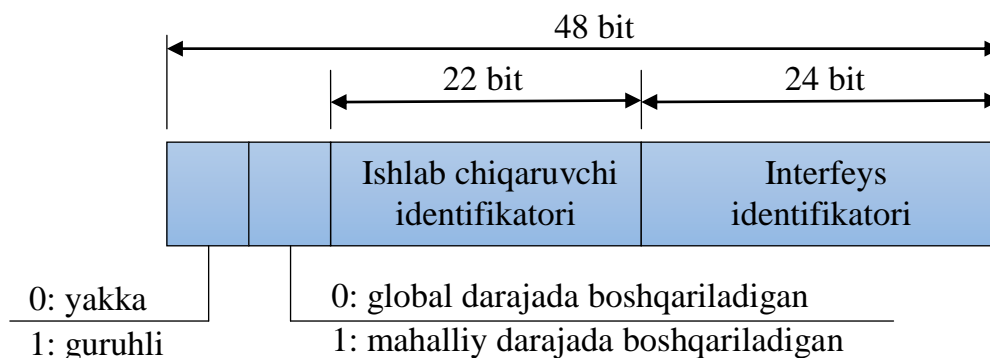
pogʻonasiga yana ikkita protokollar tayinlanishi mumkin: ICMP (Internet Control Message Protocol) bu marshrutizator paketni uzatish paytida yuz bergan xatoliklar toʻgʻrisida resursga maʼlumot uzatish uchun moʻljallangan va IGMP (Internet Group Management Protocol) esa bir vaqtning oʻzida bir nechta manzilga paketlarni yoʻnaltirish uchun ishlatiladi.

OSI modelining pastki pogʻonalari (kanal va fizik) uzatuvchi vositaga kirish, kadrlarni shakllantirish va elektr signallari darajalarini muvofiqlashtirish, kodlash, sinxronizatsiya va baʼzi boshqa funksiyalarni amalga oshiradi. TCP/IP stekining pastki pogʻonasida vazifa ancha sodda - u faqat tarmoq tarkibiga kiritilgan tarmoq texnologiyalari bilan oʻzaro aloqani tashkil qilish uchun javobgardir.

### **MAC manzil**

Tarmoq boʻylab maʼlumotlarni uzatishda tarmoqqa ulangan har bir qurilmaning MAC manzilidan (fizik manzil deb ham nomlanadi) foydalaniladi. MAC manzil («Media Access Control» – qurilmaga kirishni boshqarish) maʼlumotlar uzatish tarmogʻi qurilmasining har bir tarmoq interfeysiga beriladigan noyob identifikator hisoblanadi. Har bir tarmoq qurilmasi oʻzining 6 bayt oʻlchamli MAC manziliga ega.

IEEE standartlari 4 qismga boʻlingan 48 bitli MAC manzilni aniqlaydi (6.7- rasm).



6.7-rasm. MAC manzil tuzilishi

Ular quyidagi vazifalar uchun moʻljallangan:

- ❖ birinchi bit qaysi kadr – yakka (0) yoki guruhli (1) manzillash uchun moʻljallanganligini bildiradi;
- ❖ ikkinchi bit MAC manzil global (0) yoki lokal (1) maʼmuriy boshqariladigan ekanligini koʻrsatadi.

- ❖ 22 bit - ishlab chiqaruvchining identifikatori hisoblanadi;
- ❖ 24 bit - tarmoq qurilmasining har bir interfeysi uchun mustaqil ravishda ishlab chiqaruvchi tomonidan tanlanadi.

Odatda, MAC manzil «-» yoki «:» belgisi bilan ikkitadan guruhga ajratilgan 12 ta o‘n olti xonali raqamlar ko‘rinishida yoziladi.

Misol:

000100001111000100001110001000111010001011010000

10-F1-0E-23-A2-D0 yoki 10:F1:0E:23:A2:D0

### ***IP-manzil***

Bugungi kunda IP (Internet Protocol) manzillarning ikki turi mavjud, ular IPv4 (Internet Protocol version 4) va IPv6 (Internet Protocol version 6). Bu protokol orqali nafaqat Internet tarmog‘i elementlarini manzillash, balki mahalliy tarmoqda ham foydalanuvchilarga noyob manzillarni berish mumkin. Manzillash orqali tarmoq foydalanuvchilari bir-biridan farqlanadi va paketlar aniq belgilangan foydalanuvchiga yetib borishi kafolatlanadi.

**IPv4 protokoli.** IP protokolining 4-versiyasida (IPv4) tarmoq manzili 32 bit uzunlikka ega. Mavjud bo‘lgan manzillar soni  $2^{32}$  darajasini tashkil etadi. O‘qish qulay bo‘lishi uchun IPv4 tarmoq manzillari nuqtalar yordamida 4 baytga (oktetga) ajratiladi, shuningdek, IP-manzildagi har bir bayt o‘nlik sanoq tizimidagi sonlar bilan yoziladi.

Misol:

11000000 10101000 00001010 00000001

11000000.10101000.00001010.00000001

192.168.10.1

**IPv4 manzillari sinflari.** Tarmoq manzili (IP-manzil) ikki qismdan iborat bo‘lishi kerak: tarmoq raqami va ushbu tarmoqdagi tugun raqami. IP-manzillarning butun manzil makoni 5 ta sinfga ajratilgan (A, B, C, D va E sinflar), ularning har biri uchun tarmoqdagi raqam chegaralari va tugun raqami belgilangan.

A sinfdagi IP manzillar juda katta tarmoqlarda, B sinfidagi IP manzillar o‘rta kattalikdagi tarmoqlarda, C sinfidagi IP manzillar eng ko‘p qo‘llaniladigan manzillardan bo‘lib, kichik tarmoqlar (mahalliy

yoki lokal tarmoq deb nomlanadi) uchun ajratilgan. Tarmoqda asosan uchta sinf IP manzillaridan foydalaniladi.

6.1- jadval

Sinf	Bo‘linish	Diapazon	Tarmoq manzili	Kompyuter manzili
A	0111 1111	1.0.0.0 126.255.255.255	dastlabki 8bit	ohirgi 24 bit
B	1011 1111	128.0.0.0 191.255.255.255	dastlabki 16bit	ohirgi 16bit
C	1101 1111	192.0.0.0 223.255.255.255	dastlabki 24bit	ohirgi 8bit
D	1110 1111	224.0.0.0 239.255.255.255	Ma’lumotlarni ko‘p manzilli jo‘natish manzillari	
E	1111 1111	240.0.0.0 255.255.255.255	Zaxiralangan manzillar	

Hozirda IPv4 manzillari tugaganligi sababli IPv6 protokoli tarmoqda qo‘llash ustidagi global miqyosda ish boshlangan. Internet tarmog‘i shu darajada rivojlanmoqda, u taqdim etayotgan xizmatlari ham ko‘payib bormoqda. Internet buyumlari, ya’ni masofadan boshqaruv tizimlari, «aqliy» kabizamonaviy imkoniyatlarni ta’minlash uchun IPv6 ni qo‘llashdan boshqilolmaydi.

**IPv6 protokoli.** IPv6 protokoli yangi avlod protokoli hisoblanib, IPv4 protokolidan farqliravishda keng imkoniyatlarga ega. IPv6 protokoli 6-versiyasida (IPv6) tarmoq manzili 128 bit uzunlikka ega. Mavjud bo‘lgan manzillar soni  $2^{128}$  darajasini tashkil etadi. IPv6 da har bir tarmoq manzillari nuqtalar yordamida har biri 16 baytga teng bo‘lgan 8 taguruhga (oktetga) ajratiladi, shuningdek, IPv6 manzillari har bir bayt IPv4 dan farqliravishda (mavjud manzillar soni ko‘pligini hisobga olingan yozish formati qulay bo‘

Imaganligisababli) o‘noltiliksanoqtizimidagisonlarbilanyoziladi.  
Masalan,

8000:0000:0000:0000:0127:AB68:CD45:EF15

IPv6 protokolini yaratishdan asosiy maqsad quyidagilardan iborat:

- ❖ Tarmoqdagi milliardlab qurilmalarni manzillash;
- ❖ Marshrutizatorlarni ishlashini tezlashtirish uchun protokolni soddalashtirish;

- ❖ Xavfsizlikni ta‘minlash;

- ❖ Sifatli xizmat ko‘rsatish.

IPv6 protokolida manzillarning uchta turi mavjud:

- ❖ Unicast – tarmoqdagi xost manzili (ma‘lumotni faqat bitta xost qabul qiladi).

- ❖ Anycast – guruhli manzil (ma‘lumotni guruhdagi faqat bitta xost qabul qiladi).

- ❖ Multicast – guruhli manzil (ma‘lumotni guruhdagi barcha xost qabul qiladi).

IPv6 protokolida nol bitlik manzillar yozuvini qulayroq qilish uchun, ortiqcha nollarni olib tashlash uchun maxsus sintaksis nazarda tutilgan. « :: » yozuvidan foydalanish 16 ta nollik bitlardan iborat guruhlar borligiga ishora qiladi. « :: » kombinatsiyasi faqatgina manzil yozilishida paydo bo‘lishi mumkin. « :: » ketma-ketligi, shuningdek, yozuvdan manzildagi boshlang‘ich va yakunlovchi nollarni olib tashlash uchun foydalanilishi mumkin.

Masalan: 8000:0000:0000:0000:0127:AB68:CD45:EF15

8000::0127:AB68:CD45:EF15

## 6.5. Operatsion tizim tarmoq funksiyalari va turlari

### *Tarmoq operatsion tizimining funksional komponentlari*

Tarmoq operatsion tizimi asosiy komponentlari:

- ❖ *Lokal resurslarni boshqaruvchi vositalar* kompyuterda avtonom kompyuter OT ning barcha funksiyalarini amalga oshiradi (jarayonlarni boshqarish, operativ va tashqi xotirani boshqarish va boshqalar);

- ❖ *Tarmoq vositalari*, ular uchta komponentga bo‘linadi:

➤ *OT ning Server qismi* – umumiy foydalanish uchun mahalliy resurslar va xizmatlarni taqdim etish vositalari;

➤ *OT ning Mijoz qismi* – masofadagi resurslar va xizmatlardan foydalanishga ruhsat so‘rash vositalari;

➤ *OT ning Transport vositalari* - bu kompyuterlar o‘rtasida xabarlarni uzatishni ta‘minlovchi kommunikatsiya tizimlari bilan birgalikda ishlovchi vositalar.

Tarmoq orqali xabarlarni uzatishda kompyuterning o‘zaro ishlash qoidalari Ethernet, Token Ring, IP, IPX va boshqalar kabi aloqa protokollaridan foydalaniladi. Ikkita kompyuter tarmoq orqali xabar almashish imkoniyatiga ega bo‘lishi uchun ularning operatsion tizimlari ba‘zi bir keng tarqalgan aloqa protokollar to‘plamini qo‘llab-quvvatlashlari kerak.

### ***Tarmoq xizmatlari va servislari***

Tarmoq orqali ma‘lum bir kompyuter resursiga kirishni ta‘minlaydigan operatsion tizimning server va mijoz qismlarining majmuiga tarmoq xizmatlari deb nomlanadi. Tarmoq xizmatlari tarmoq foydalanuvchilariga bir qator xizmatlarni taqdim etadi. Masalan, fayl xizmatlarining aksariyati foydalanuvchi yoki dastur so‘roviga binoan quyidagi amallarni bajarishi (xizmatlarni taqdim qilishi) mumkin: fayllarni yaratish, o‘chirish, nusxalash, qayta nomlash (nomini o‘zgartirish) va hokazolar. Tarmoq xizmatlari ba‘zan tarmoq servislari deb ataladi (inglizcha, service).

*Servis* – bu xizmatlardan foydalanuvchilar va xizmatlarni taqdim etuvchilar o‘rtasidagi interfeys.

Har bir xizmat tarmoq resurslarining ma‘lum bir turi va/yoki ushbu resurslarga kirishning o‘ziga xos usuli bilan bog‘liq. Masalan, chop etish xizmati tarmoq foydalanuvchilariga umumiy tarmoq printerlariga kirishni ta‘minlaydi va chop etish xizmatini taqdim etadi, pochta xizmati esa tarmoq ma‘lumot resurslariga elektron xatlarga kirishni ta‘minlaydi. Resurslarga kirish usuli har xil, masalan, masofadan kirish xizmati - bu kompyuter tarmog‘ining foydalanuvchilariga kommutatsiyalangan telefon kanallari orqali barcha resurslarga kirishni ta‘minlaydi. Printer kabi ma‘lum bir resursga masofadan kirish huquqiga ega bo‘lish uchun, masofadan kirish xizmati chop etish xizmati bilan aloqa qiladi. Tarmoq xizmatlari orasida biz oddiy foydalanuvchilarga emas, balki, tarmoq ma‘murlari

uchun yoʻnaltirilgan xizmatlarni ajratib koʻrsatishimiz mumkin. Bunday xizmatlar tarmoqning ishlashini tashkil qilish uchun moʻljallangan. Masalan, markazlashtirilgan maʼlumotnoma xizmati yoki katalog xizmati tarmoq foydalanuvchilari maʼlumotlar bazasini, uning barcha dasturiy va qurilma komponentlarini (komponentlarini) saqlash uchun moʻljallangan. Maʼmurga xizmat koʻrsatadigan tarmoq xizmatlarining boshqa misollariga tarmoqni boshqarish va tahlil qilish imkonini beruvchi tarmoqni boshqarish xizmati kiradi, uning vazifalari, xususan, parolni tekshirish bilan mantiqiy kirish protsedurasini bajarish, zaxira va arxivlash xizmatini oʻz ichiga olishi mumkin. Tarmoq xizmatlari mijoz-server tizimlari tomonidan amalga oshiriladi. Har qanday tarmoq xizmatini amalga oshirishda soʻrov manbai (mijoz) va soʻrovni bajaruvchi (server) tabiiy ravishda paydo boʻladi, shuning uchun har qanday tarmoq xizmatida ikkita nosimmetrik qism mavjud - mijoz va server. Odatda, server oʻz resurslarini mijozga beradi va mijoz ulardan foydalanadi. Shuni taʼkidlash kerakki, tarmoq xizmati maʼlum bir xizmatni taqdim etganda, resurslardan nafaqat server, balki mijoz ham foydalanadi. Mijoz oʻz resurslarining katta qismini (disk maydoni, protsessor vaqti va boshqalar) tarmoq xizmatini taʼminlash uchun sarflashi mumkin. Masalan, pochta xizmatini amalga oshirayotganda, maʼlumotlar bazasining lokal nusxasi, uning keng qamrovli yozishmalarini mijozning diskida saqlash mumkin. Bunday holda, mijoz turli xil formatlarda, shu jumladan, murakkab multimediya, xabarlar yozish paytida juda koʻp ishlarni amalga oshiradi, manzillar kitobini saqlashni qoʻllab-quvvatlaydi va qoʻshimcha yordamchi vazifalarni bajaradi. Mijoz va server oʻrtasidagi farq shundaki, tarmoq xizmatining tashabbuskori har doim mijoz boʻlib, server har doim soʻrovlarni kutish rejimida boʻladi. Masalan, pochta serveri pochta foydalanuvchi kompyuteriga faqat pochta mijozidan soʻrov kelib tushganda yetkazib beradi. Odatda, mijoz va server qismlari oʻrtasidagi oʻzaro aloqa standartlashtirilgan, shuning uchun bitta turdagi server har xil turdagi mijozlar bilan ishlash uchun moʻljallangan boʻlishi mumkin. Buning yagona sharti shundaki, mijozlar va server umumiy standart aloqa protokollarini qoʻllab-quvvatlashlari kerak.

### ***Oʻrnatilgan tarmoq xizmatlari va tarmoq qobiqlari***

Amaliyotda tarmoq operatsion tizimlarini qurishda bir nechta yondashuvlar ishlab chiqilgan, ular operatsion tizimda tarmoq xizmatlarini amalga oshirishda quyidagilar bilan farqlanadi:

- ❖ tarmoq xizmatlari to‘plam sifatida birlashtirilgan - qobiq;
- ❖ tarmoq xizmatlari alohida mahsulot sifatida ishlab chiqariladi va yetkazib beriladi;
- ❖ tarmoq xizmatlari OTga chuqur joylashtirilgan.

Birinchi yondashuv mavjud mahalliy operatsion tizim va uning ustiga o‘rnatilgan tarmoq qobig‘ining kombinatsiyasi sifatida yaratilgan birinchi tarmoq operatsion tizimlariga xos edi.

**Tarmoq qobig‘i** - bu mustaqil dasturiy mahsulot sifatida ishlab chiqilgan, qoida tariqasida, o‘zaro muvofiqlashtirilgan tarmoq xizmatlari to‘plamidir. Tarmoq qobiqlarini mijoz va serverga bo‘lish mumkin. Qobiq tarmoq xizmatlarining mijoz qismlarini o‘z ichiga olsa, mijoz qobig‘i deyiladi. Server qobig‘i kamida ikkita asosiy tarmoq xizmatlarining server komponentlarini o‘z ichiga oladi – fayl va chop etish xizmatlari. Hozirda faqat tarmoqda ishlash uchun yaratilgan maxsus operatsion tizimlar mavjud. Ushbu operatsion tizimlarning tarmoq funksiyalari tizimning asosiy modullariga joylashtirilgan, bu operatsion tizimning mantiqiy uyg‘unligini, shuningdek, yaxshiroq ishlashini ta‘minlaydi. Agar barcha tarmoq xizmatlari yaxshi integratsiyalangan bo‘lsa va operatsion tizimning ajralmas qismi deb hisoblansa, unda bunday operatsion tizimning barcha ichki mexanizmlari tarmoq funksiyalarini bajarish uchun optimallashtirilishi mumkin.

### ***Tarmoq OT lariga qo‘yiladigan talablar***

1. Masshtabiylik;
2. Moslashuvchanlik;
3. Oxirgi qurilmalardagi turli OT larni qo‘llab-quvvatlash;
4. Protokollar stekini qo‘llab-quvvatlash (TCP/IP, IPX/SPX, NetBIOS, DECnet, AppleTalk, OSI);
5. Ko‘p serverli tarmoqlarni qo‘llab-quvvatlash va boshqa OT lar bilan samarali integrallashish;
6. Markazlashgan ma‘lumotnomalar xizmatini bo‘lishi;
7. Servislar tizimini rivojlanishi: fayl-servis, print-servis, ma‘lumotlar xavfsizligi, arxivlash, ma‘lumotlar bazasi va boshqalar;



8. Turli standartlardagi tarmoq qurilmalarini (Ethernet, Token Ring, ARCnet, FDDI) va tarmoqni boshqarish standartlarini qo‘llab-quvvatlash.

### ***Tarmoq OT larining turlari***

Kompyuterlar o‘rtasida funksiyalarni taqsimlanishiga qarab, ular ajratilgan server yoki klient tuguni rolda namoyon bo‘lishadi. Tarmoq quyidagi sxemalar asosida qurilgan bo‘lishi mumkin:

- ❖ Server va klient funksiyalari birlashtirilgan kompyuterlardan tashkil topgan tarmoq – birrangli tarmoq;
- ❖ Klient va server asosida qurilgan tarmoqlar – ajratilgan serverli tarmoq;
- ❖ Barcha turdagi tugunlardan tashkil topgan tarmoq – gibril tarmoq deb nomlanadi.

### ***Birrangli tarmoqda operatsion tizim***

Birrangli tarmoqlarda barcha kompyuterlar bir-birlarining resurslaridan foydalanishda tengdir. Har bir foydalanuvchi, uning hohishiga binoan, kompyuteridagi har qanday resursni umumiy deb e‘lon qilishi, shundan so‘ng boshqa foydalanuvchilar undan foydalanishlari mumkin. Birrangli tarmoqlarda barcha kompyuterlarga operatsion tizim o‘rnatilgan bo‘lib, u tarmoqdagi barcha kompyuterlarga teng imkoniyatlarni taqdim etadi. Ushbu turdagi tarmoq operatsion tizimlari birrangli operatsion tizimlar deb nomlanadi. Shubhasiz, birrangli operatsion tizimlar tarmoq xizmatlarining ham server, ham mijoz komponentlarini o‘z ichiga olishi kerak. Birrangli operatsion tizimlar sifatida Windows 2000 Professional, Windows XP/Vista lar xizmat qilishi mumkin. O‘z navbatida tarmoqda o‘z resurslarini bo‘lishishni istamaydigan foydalanuvchilar ham uchraydi. Bunday holda, ularning operatsion tizimlarining server imkoniyatlari faollashtirilmaydi va kompyuterlar faqat mijoz rolini o‘ynaydi.

### ***Ajratilgan serverli tarmoqda operatsion tizim***

Ajratilgan serverlari bo‘lgan tarmoqlarda maxsus server roli uchun optimallashtirilgan tarmoq operatsion tizimlari qo‘llaniladi. Ushbu tarmoqlarda foydalanuvchi kompyuterlari mijoz operatsion tizimlari boshqaruvida ishlaydi. Server sifatida ishlash uchun ixtisoslashgan operatsion tizimlar, server operatsiyalari

samaradorligini oshirishning tabiiy usulidir. Agar tarmoqda yuzlab va hatto minglab foydalanuvchilar bo'lsa, umumiy resurslarga yuboriladigan so'rovlarning intensivligi juda katta bo'lishi mumkin va server ushbu so'rovlar oqimini hech qanday kechiktirmasdan bajara oladi. Ushbu muammoning aniq yechimi, server sifatida kuchli qurilma platformasi va server funksiyalari uchun optimallashtirilgan operatsion tizimga ega kompyuterdan foydalanishdir. Ajratilgan serverli operatsion tizimlar sifatida Windows server NT/2008/2012/2016/2019, Solaris, Linux, FreeBSD va boshqalarni aytishimiz mumkin.

### ***Taqsimlangan ilovalar va tarmoq xizmatlari modellari***

Tarmoqlardagi ilovalar ishlashini tashkil etishga ko'ra uchga bo'linadi:

- ❖ Ilova qismlari turli tarmoq kompyuterlariga taqsimlanadigan tarmoqlar;
- ❖ Maxsus serverlarda barcha ilovalar uchun bir qancha umumiy funksiyalar bajaradigan tarmoqlar;
- ❖ Turli kompyuterlarda taqsimlanib bajariladigan funksiyalarni o'zaro birgalikdagi harakatli tarmoqlar.

### **Nazorat savollari**

1. Kompyuter tarmog'i nima?
2. Tarmoq operatsion tizimi turlari?
3. Kompyuterni tarmoqqa birlashtirish qanday masalalarni yechishga imkon beradi?
4. Tarmoq turlari va bir-biridan farqlarini tushuntiring.
5. Tarmoq operatsion tizimining asosiy komponentlari.
6. Tarmoq operatsion tizimlariga qo'yiladigan talablar nima?
7. Tarmoq operatsion tizimlarining qanday turlari mavjud?
8. Tarmoqni qurish topologiyalari turlari?
9. Shina topologiyasini tushuntiring.
10. Yulduz topologiyasini tushuntiring.
11. Xalqa topologiyasini tushuntiring.
12. Daraxt topologiyasini tushuntiring.
13. Marshrutlash algoritmini tushuntiring.
14. Statik marshrutlash nima?

## **VIIBOB. OPERATSION TIZIMLARNING TARMOQ XIZMATLARI**

Tarmoq yoki ichki tarmoqning maqsadi foydalanuvchilarga xizmat ko'rsatishdir: tarmoqning mahalliy qismdagi foydalanuvchilar masofadagi server xizmatlaridan foydalanishni istaydilar. Ushbu maqsadni amalga oshirish usullaridan biri sifatida, ikkita dasturiy ta'minotni ishga tushirish hisoblanadi. Mahalliy kompyuter masofadagi kompyuterga xizmat so'rovlarini jo'natish uchun dasturiy ta'minotni ishga tushiradi, masofadagi kompyuter esa so'rovlarga mos xizmatlarni taqdim etish uchun dasturiy ta'minotni ishlatadi. Bundan shuni anglash mumkinki, internet orqali bog'langan ikkita kompyuterlarning biri xizmatlar so'rovini jo'natishi va ikkinchisi esa bu so'rovlarga mos xizmatlarni taqdim etish uchun dasturiy ta'minotni ishlatadilar.

### **7.1. Mijoz-server arxitekturasi, tuzilishi va asosiy tushunchalari**

Bir qarashda ikkita amaliy dasturlar o'rtasida aloqani tashkil etish sodda ishdek tuyiladi, ulardan biri mahalliy qismda va ikkinchisi masofadan ishlatiladi. Ammo bu yondashuvni amalga oshirishda ko'plab savollar tug'iladi. Ushbu savollardan ba'zilar quyidagilar bo'lishi mumkin:

1. Har ikkala amaliy dasturlar ham xizmatlar uchun so'rovlar jo'natish va xizmatlarni taqdim qilish imkoniyatiga ega bo'lishi kerakmi, yoki, ulardan biri faqat so'rov jo'natish va ikkinchisi faqat xizmatni ta'minlash imkoniyatiga ega bo'lishi kerakmi? Buning yechimlaridan biri mahalliy mashinada ishlovchi va xizmatlar uchun so'rov jo'natuvchi *mijoz* deb nomlanuvchi amaliy dasturning ishlatilishi, va masofadagi mashinada so'rovlarga mos xizmatni ta'minlovchi *server* deb ataladigan amaliy dasturning mavjud bo'lishidir. Boshqacha qilib aytganda, so'rovlar jo'natish va xizmat bilan ta'minlash bir-biridan ajratilgan. Amaliy dastur yoki so'rovlar jo'natuvchi mijoz, yoki xizmatni ta'minlovchi server bo'lishi kerak. Ushbu amaliy dasturlar birgalikda qo'llanilib, bir hil nom bilan – mijoz va server deb yuritiladi.

2. Server faqat bitta maxsus mijoz uchun xizmat taqdim etishi kerakmi, yoki istalgan mijozning server ta'minlay oladigan xizmatlariga bo'lgan so'rovlariga javob berishi kerakmi? Eng keng tarqalgan yechim bu –ixtiyoriy mijozning server ta'minlay oladigan xizmatlarga so'rovini qo'llab-quvvatlashdir, bunda maxsus xizmatlar nazarda tutilmaydi. Boshqacha aytganda, server – mijoz munosabati birni – ko'pga tamoyiliga asoslanadi.

3. Kompyuter faqat yagona amaliy dasturni ishlatishi kerakmi (mijoz yoki server)? Agar kerakli amaliy dasturiy ta'minot mavjud bo'lsa, Internetga ulangan ixtiyoriy kompyuter mijoz dasturini ishlata olishi kerak. Server dasturlari doimiy ishlovchi kompyuterlarda ishlatilishi kerak bo'ladi. Bu haqida keyingi matnlarda batafsil yoritilgan.

4. Qachon amaliy dastur ishlatilishi kerak? Har doimmi yoki faqat xizmatga ehtiyoj sezilgandami? Umuman olganda, xizmatga so'rov yuboruvchi mijoz dasturi faqat xizmatga ehtiyoj tug'ilganda ishlatiladi. Lekin xizmatlar bilan ta'minlovchi server dasturi doimiy ishlashi talab etiladi. Chunki, uning xizmatlariga mijozlar tomonidan ehtiyoj qachon paydo bo'lishi va so'rovlar yuborilishi haqida ma'lumot mavjud bo'lmaydi.

5. Foydalanuvchi istagan ixtiyoriy xizmatlarni taqdim etuvchi yagona, universal amaliy dasturiy ta'minot bo'lishi kerakmi? Yoki har bir xizmat uchun alohida dasturiy ta'minot bo'lishi talab etiladimi? TCP/IPda ko'plab mijozlar tomonidan tez-tez so'rov jo'natiladigan xizmatlar uchun mijoz-server amaliy dasturlar mavjud. Masalan, fayllarga kirish, elektron pochta jo'natmalari va boshqa xizmatlar uchun alohida mijoz-server amaliy dasturlar mavjud. Moslashtirilgan xizmatlardan foydalanish uchun masofadagi kompyuterda mavjud bo'lgan ushbu xizmatni foydalanuvchilarga taqdim etuvchi yagona umumiy amaliy dastur bo'lishi kerak. Masalan, shunday mijoz-server amaliy dasturi bo'lishi kerakki, uning yordamida foydalanuvchi masofadagi kompyuterga kirishi (log onto) va u kompyuter taqdim etadigan xizmatlardan foydalanishi mumkin bo'lsin.

### Server

Server bu – mijozlarga xizmatlarni taqdim qiluvchi masofadagi kompyuterda ishlatiluvchi dastur hisoblanadi. Bu dastur ishga tushishi bilan mijozlardan kelayotgan so'rovlarga “eshiklar ochiladi”, lekin hech qachon so'rov kelmaguncha xizmat ko'rsatishni boshlamaydi.

Server dasturi vaqt bo'yicha cheksiz hisoblanadi. U ishga tushganidan boshlab muammolar yuzaga kelgunicha cheksiz davrgacha ishlashi mumkin. So'rovlar kelganida server ularga iteratsion yoki bir vaqtning o'zida javob beradi.

### Mijoz

Mijoz bu – mahalliy mashinada o'rnatilgan dastur bo'lib, u serverdagi xizmatlardan foydalanish uchun so'rovlar yuboradi. Mijoz dasturi vaqt bo'yicha cheklangan xususiyatga ega. U foydalanuvchi (yoki boshqa amaliy dastur) tomonidan ishga tushirilib, xizmat yakunlanganida to'xtatiladi. Odatda, mijoz masofadagi host kompyuterning IP manzilidan va u kompyuterda ishlayotgan xizmat dasturining port manzilidan foydalangan holda aloqa kanalini ochadi. Aloqa kanali ochilganidan so'ng, mijoz o'zining so'rovini jo'natadi va javobni (xizmat) qabul qiladi. Albatta, so'rov – javob qismi bir necha bor takrorlanishi mumkin, biroq umumiy jarayon vaqt bo'yicha chegaralangan va u ohir-oqibat yakunlanadi.

### Bir vaqtda bajarish (Concurrency)

Ham mijoz, ham server bir vaqtda ishlatish tartibida bo'lishi mumkin. Quyida ularning har biri uchun dasturlarni bir vaqtda ishlatish imkoniyatlari haqida so'z yuritiladi.

### Mijozdagi bir vaqtda bajarish

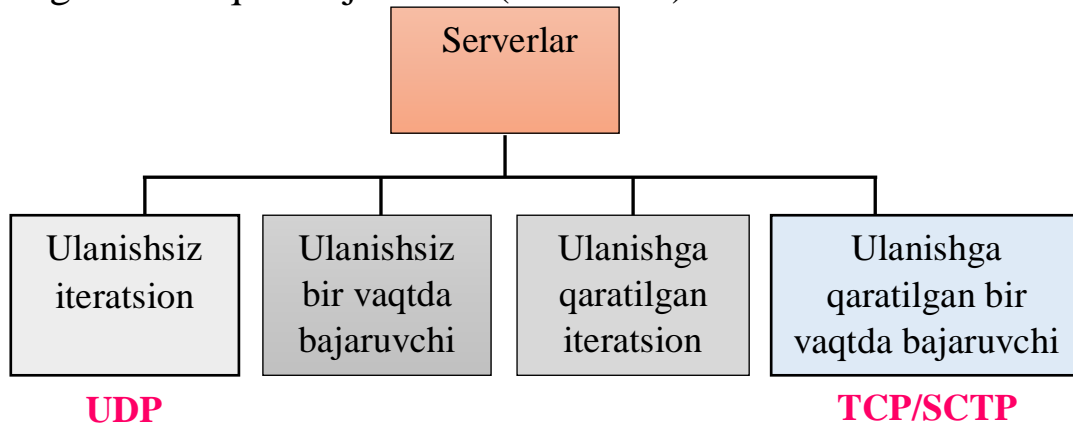
Mijoz dasturlar mashinada iteratsion yoki bir vaqtda bajarilishi mumkin. Mijoz dasturlarni iteratsion bajarish – bu ularni birin-ketin ishlatilishini ifodalaydi, ya'ni, bir mijoz dastur boshlanadi, bajariladi va yakunlanadi, shundan so'ng mashina keyingi mijozni bajarilishini boshlaydi. Bugungi kunda kompyuterlarning asosiy qismi bir vaqtda bajarilish imkonini beradi, bu ikki va undan ortiq mijoz dasturlari bir vaqtda ishlashini nazarda tutadi.

### Serverdagi bir vaqtda bajarish

Iteratsion server bir vaqtda faqat bir so'rovni qayta ishlashi mumkin: u birinchi so'rovni qabul qiladi, uni qayta ishlaydi va mijozga javobni yo'llaydi, shundan so'ng ikkinchi so'rovni boshlaydi. Bir vaqtda bajarish imkonini beruvchi serverlar bir nechta so'rovlarni bir vaqtning o'zida qayta ishlash orqali so'rovlar o'rtasida vaqtni taqsimlash tamoyilini qo'llaydi.

Server yoki ulanishsiz transport sathi protokoli UDP, yoki ulanishga qaratilgan transport sathi protokoli TCP/SCTP protokolini ishlatadi. Shuning uchun server amaliyoti ikkita omilga: transport

sathi protokoli va xizmat uslubiga bog‘liq bo‘ladi. Nazariy jihatdan 4 xil serverlar bo‘lishi mumkin: ulanishsiz iteratsion, ulanishsiz bir vaqtda bajaruvchi, ulanishga qaratilgan iteratsion va ulanishga qaratilgan bir vaqtda bajaruvchi (7.1- rasm).

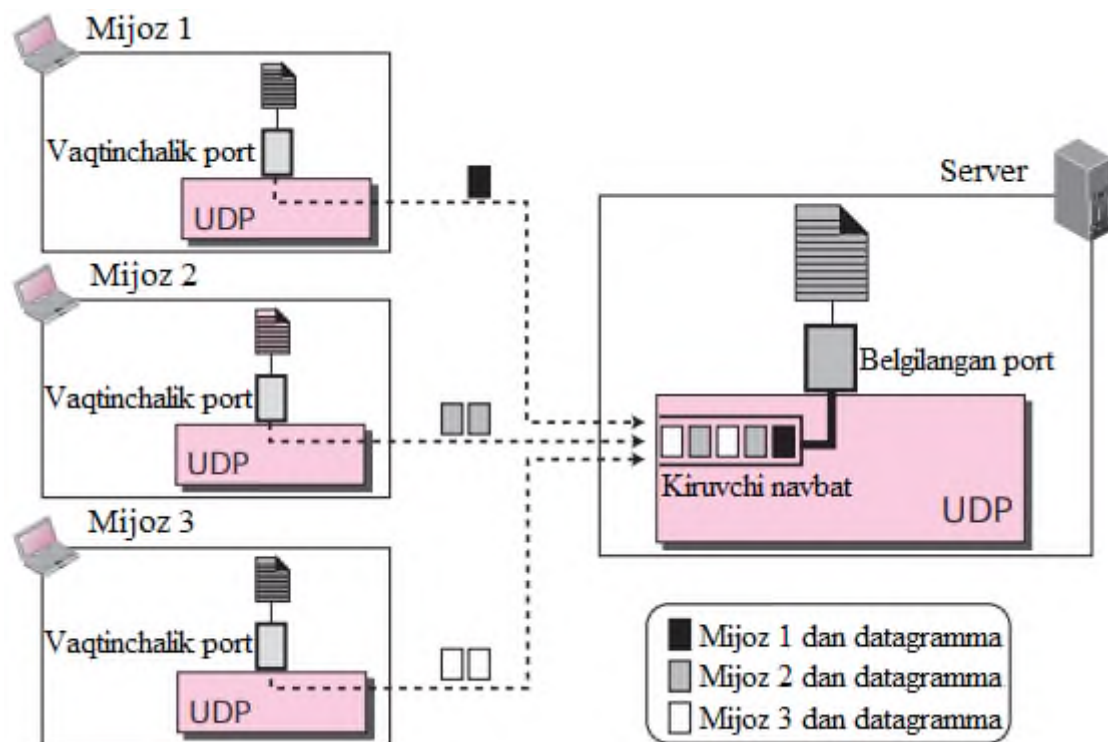


7.1- rasm. Serverturlari

### Ulanishsiz iteratsion

UDPni ishlatadigan serverlar odatda iterativdir, yuqori data'kidlanganidek, server bir vaqtning o'zida bitta so'rovni qayta ishlaydi. Server UDP dan datagramda olingan so'rovni oladi, so'rovni qayta ishlaydi va mijozga yuborish uchun UDP ga javob beradi.

7.2- rasmda ulanishsiz iteratsion server ko'rsatilgan.



7.2- rasm. Ulanishsiz iteratsion server

Serverboshqama'lumotlarjadvallarigaahamiatbermaydi.

Ushbudatagramlarxizmatnikutib, navbatdasaqlanadi. Ularning barchasi bitta mijozdan yoki ko'plab mijozlardan bo'lishi mumkin. Ikkala holatda ham ular kelish tartibida birma-bir qayta ishlanadi.

Bu maqsadda server bitta tanlangan portdan foydalanadi, bu belgilangan port. Ushbu portga kelgan barcha datagramlar 7.2- rasmda tasvirlanganidek xizmat ko'rsatilishini navbatda kutadi.

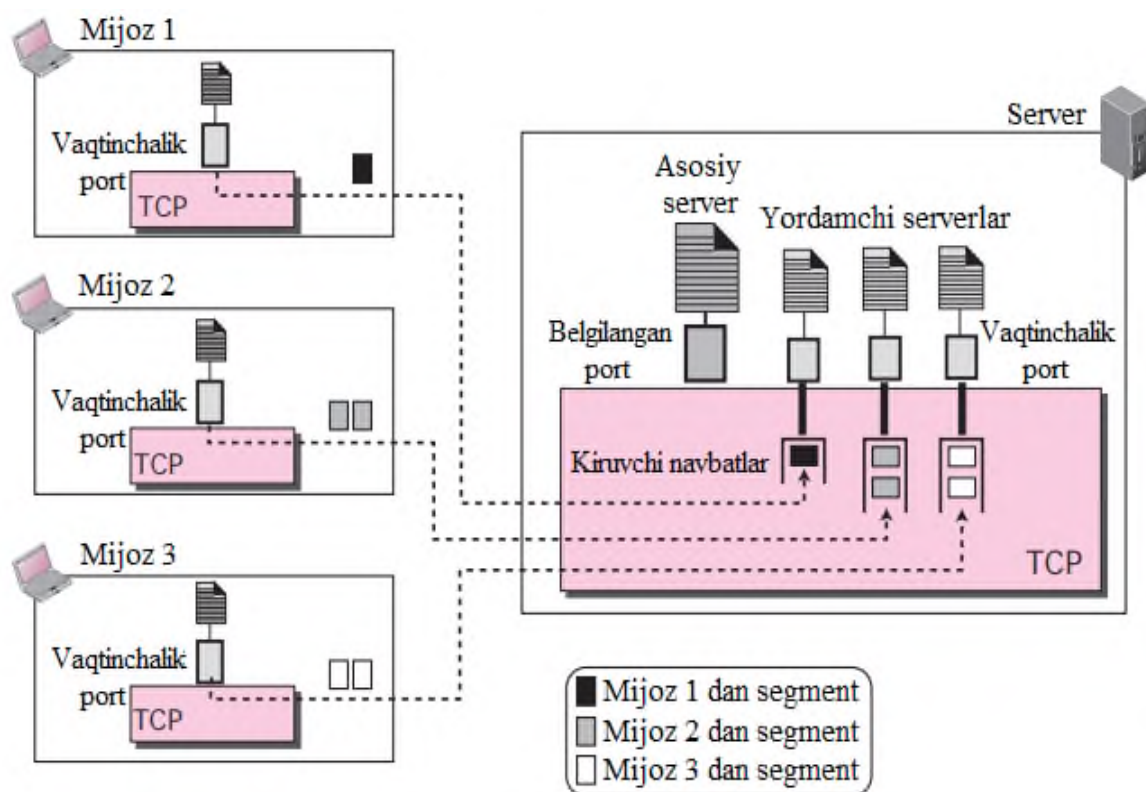
#### Ulanishga – qaratilgan bir vaqtda bajaruvchi server

TCP (yoki SCTP) dan foydalanadigan serverlar odatda bir vaqtning o'zida bir nechta so'rovlarni qayta ishlaydi. Bu shuni anglatadiki, server bir vaqtning o'zida ko'plab mijozlarga xizmat ko'rsatishi mumkin. Aloqa - bu ulanishga yo'naltirilgan, ya'ni so'rov bir necha segmentlarga kelishi mumkin bo'lgan va so'rovlarga javoblar bir necha segmentlarni egallashi mumkin bo'lgan baytlar oqimidir. Server va har bir mijoz o'rtasida aloqa o'rnatiladi va ulanish butun oqim qayta ishlanib tugallanmaguncha ochiq qoladi.

Ushbu turdagi server faqat bitta portdan foydalana olmaydi, chunki har bir ulanish portga muhtoj va ko'plab ulanishlar bir vaqtning o'zida ochilishi mumkin. Ko'pgina portlarga ehtiyoj bor, ammo server faqat bitta belgilangan portdan foydalanishi mumkin.

Yechim bitta belgilangan port va ko'pgina qisqa muddatli portlarga ega bo'lishdir. Server ulanish so'rovlarini belgilangan portda qabul qiladi. Mijoz ulanishni amalga oshirish uchun ushbu portga o'zining dastlabki yondashuvini taklif qilishi mumkin. Ulanish amalga oshirilgandan so'ng, server belgilangan portni bo'shatish uchun ushbu ulanishga vaqtinchalik portni tayinlaydi. Endi ma'lumotlarni uzatish ushbu ikkita vaqtinchalik portlar o'rtasida amalga oshirilishi mumkin, biri mijoz qismida, ikkinchisi esa server qismida. Endi belgilangan port boshqa mijozga ulanishni amalga oshirish uchun bo'shdir. Bir vaqtning o'zida bir nechta mijozlarga xizmat ko'rsatish uchun server dastlabki jarayonning (parent process) nusxalarini (child process) yaratadi.

Serverda har bir ulanish uchun bitta navbat bo'lishi kerak. Segmentlar mijozdan kelib chiqadi, tegishli navbatda saqlanadi va ularga bir vaqtda server tomonidan xizmat ko'rsatiladi. Ushbu konfiguratsiya uchun 7.3-rasmga qarang.



7.3- rasm. Ulanishga yo'naltirilgan parallel server

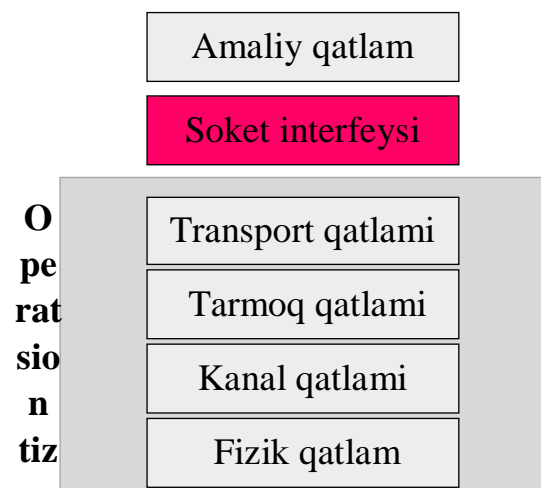
### *Soket interfeyslari*



Qanday qilib mijoz jarayoni server jarayoni bilan aloqa o‘rnatishi mumkin? Kompyuter dasturi - bu kompyuterda nima qilish kerakligini aytadigan oldindan belgilangan ko‘rsatmalar to‘plamidir. Kompyuter dasturida matematik operatsiyalar bo‘yicha ko‘rsatmalar to‘plami, satrlarni boshqarish bo‘yicha boshqa ko‘rsatmalar to‘plami, kirish va chiqish uchun kirish bo‘yicha ko‘rsatmalar to‘plami mavjud. Agar boshqa mashinada ishlaydigan dastur bilan aloqa qilish uchun bizga dastur kerak bo‘lsa, transport sathiga ulanishni ochish, boshqa tomondan ma’lumotlarni yuborish va ma’lumotlarni olish hamda ulanishni yopish uchun buyruqlar to‘plamiga ehtiyoj sezamiz. Ushbu turdagi ko‘rsatmalar to‘plami odatda interfeys deb ataladi.

Bir nechta interfeyslar aloqa uchun mo‘ljallangan. Ulardan uchta keng tarqalgan: soket interfeysi, transport sathi interfeysi (TLI) va STREAM.

Soket interfeysi 1980-yillarning boshlarida Berkli Universitetida UNIX muhitining bir qismi sifatida boshlangan. Soket interfeysini yaxshiroq tushunish uchun UNIX yoki Windows kabi asosiy operatsion tizim va TCP/IP protokoli to‘plami o‘rtasidagi munosabatni ko‘rib chiqish kerak. 7.4- rasmda operatsion tizim va TCP/IP protokoli to‘plami o‘rtasidagi konseptual bog‘liqlik ko‘rsatilgan.



7.4- rasm. Operatsion tizim va TCP/IP to‘plami o‘rtasidagi bog‘liqlik

Soket interfeysi, ko‘rsatmalar to‘plami sifatida, operatsion tizim va amaliy dasturlar o‘rtasida joylashgan. TCP/IP protokoli to‘plamida

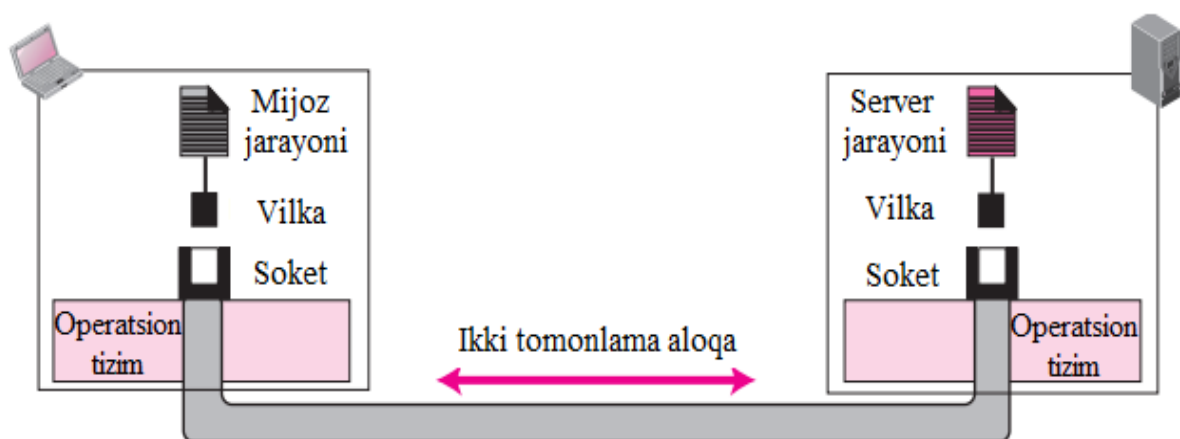
taqdim etiladigan xizmatlarga kirish uchun dastur socket interfeysida belgilangan ko'rsatmalardan foydalanishi kerak.

### Misol

Dasturlash tillarining ko'pida dastur interfeysi, dasturchiga faylni ochish, fayldan o'qish, faylga yozish, faylda boshqa operatsiyalarni bajarish va nihoyat faylni yopish imkonini beradigan ko'rsatmalar to'plami mavjud. Agar dastur faylni ochishi kerak bo'lsa, u operatsion tizimga ma'lum bo'lganidek fayl nomidan foydalanadi. Fayl ochilganda, operatsion tizim o'qish va yozish kabi boshqa ko'rsatmalar uchun ishlatilishi mumkin bo'lgan faylga (butun son yoki ko'rsatgich) ma'lumotni qaytaradi.

### **Soket**

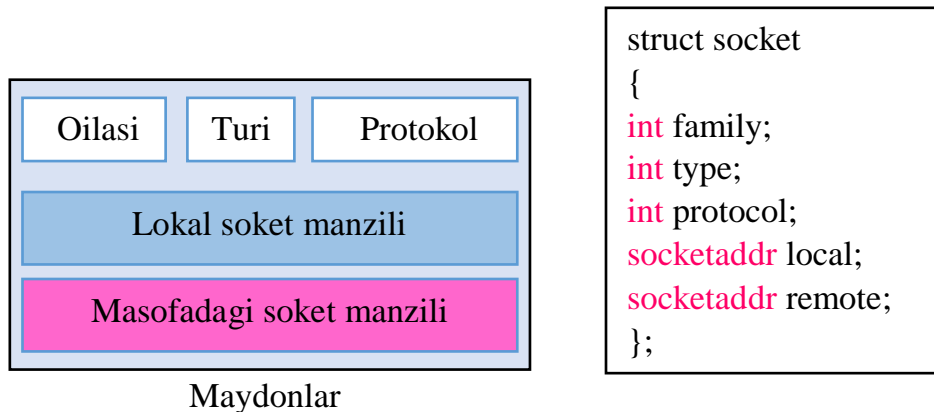
Soket - bu bizning kundalik hayotimizda ko'radigan apparat soketni simulyatsiya qiladigan abstrakt dasturdir. Aloqa kanalidan foydalanish uchun amaliy dasturdan (mijoz yoki server) soket yaratish uchun operatsion tizimni so'rash kerak. Keyin dastur ma'lumot uzatish va qabul qilish uchun soketga ulanishi mumkin. Ma'lumotlar uzatilishi amalga oshishi uchun har bir aloqaning bir uchida ikkita soket kerak bo'ladi. 7.5- rasmda biz kundalik hayotimizda ishlatadigan **rozetka** va **vilkasidan** (masalan, telefon uchun) foydalangan holda, ushbu abstraktsiyani simulyatsiya qilamiz. Internetda soket – bu muhokama qiladiganimiz kabi dasturiy ta'minot ma'lumotlarining tuzilishi.



7.5- rasm. Soket tushunchasi

### ***Soket ma'lumotlar tuzilishi***

Soketni aniqlash uchun ma'lumotlar tuzilishining formati jarayonlar ishlatadigan tilga bog'liq. Masalan, C tilida socket 7.6-rasmda ko'rsatilgandek, beshta maydonli tuzilma (tuzilish yoki yozuv) sifatida aniqlanadi.



7.6- rasm. Soket ma'lumotlarining tuzilishi

E'tibor bering, dasturchi ushbu tuzilmani qayta aniqlamasligi kerak; u allaqachon aniqlangan. Dasturchi faqat ushbu ta'rifni o'z ichiga olgan sarlavha faylidan foydalanishi kerak.

Keling, ushbu tuzilishda foydalaniladigan maydonlarni qisqacha ko‘rib o‘tamiz:

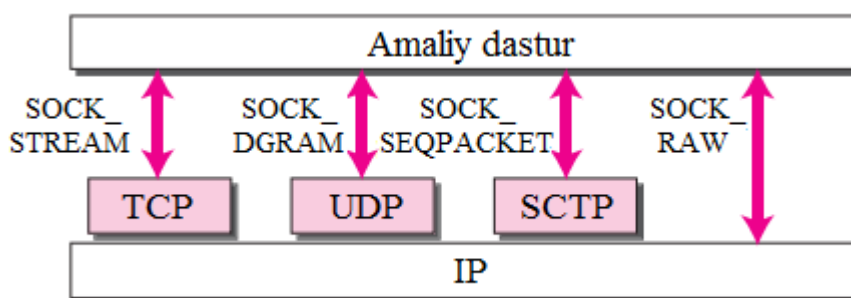
**Oilasi (Family).** Ushbu maydon protokol guruhini belgilaydi: IPv4, IPv6, UNIX domen protokollari va boshqalar. Biz TCP/IP da ishlatadigan oilasi (family) turi IPv4 protokoli uchun doimiy IF\_INET va IPv6 protokoli uchun IF\_INET6 tomonidan belgilanadi.

**Turi (Type).** Ushbu maydon soketning to‘rt turini belgilaydi: SOCK\_STREAM (TCP uchun), SOCK\_DGRAM (UDP uchun), SOCK\_SEQPACKET (SCTP uchun) va SOCK\_RAW (IP xizmatlaridan bevosita foydalanadigan ilovalar uchun) 7.7-rasmda ko‘rsatilgan.

**Protokol (Protocol).** Ushbu maydon interfeysni ishlatadigan protokolni belgilaydi. TCP/IP protokoli to‘plami uchun 0 ga o‘rnatiladi.

**Lokal soket manzili (Local socket address).** Ushbu maydon mahalliy soket manzilini belgilaydi. Soket manzili IP manzil va port raqamining kombinatsiyasi hisoblanadi.

**Masofadagi soket manzili (Remote socket address).** Bu maydon masofadagi soket manzilni aniqlaydi.



7.7- rasm. Soket turlari

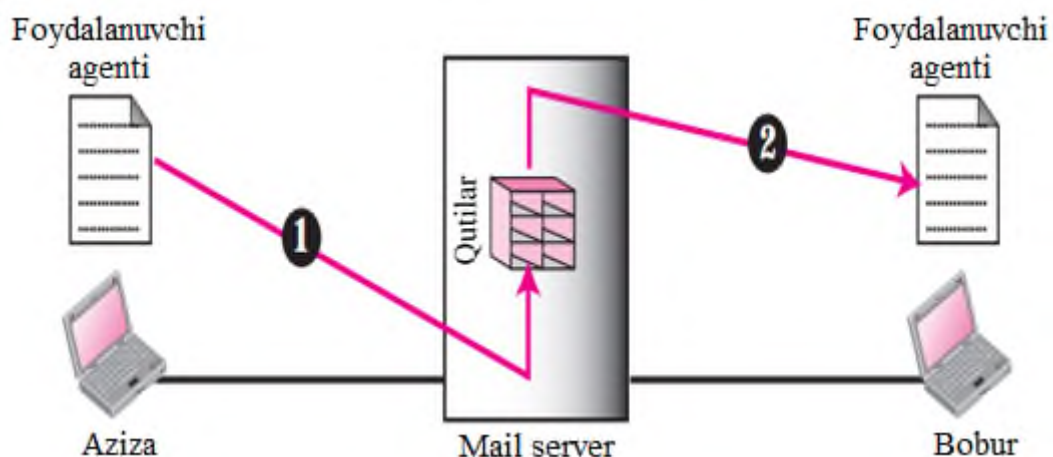
## 7.2. Tarmoq operatsion tizimlari va ularning asosiy tarmoq xizmatlari

### *E-mail arxitekturasi*

Elektron pochta arxitekturasini tushunish uchun quyidagi to‘rt senariy ko‘rib o‘tiladi.

**Birinchi senariy.** Birinchi holatda elektron pochta jo‘natuvchi va qabul qiluvchi bitta pochta serveridagi foydalanuvchilar (yoki

dasturlar); ular to‘g‘ridan-to‘g‘ri umumiy pochta serveriga ulangan. Ma‘mur qabul qilingan xabarlar saqlanadigan har bir foydalanuvchi uchun bitta pochta qutisini yaratdi. Pochta qutisi mahalliy qattiq diskning bir qismi, ruxsat cheklovlari bo‘lgan maxsus fayl. Faqat pochta qutisi egasi unga kirish huquqiga ega. A foydalanuvchi B foydalanuvchiga xabar yuborishi kerak bo‘lganida, u xabarni tayyorlash va B ning pochta qutisiga saqlash uchun foydalanuvchi agenti (user agent - UA) dasturini ishga tushiradi. Xabarda jo‘natuvchi va qabul qiluvchining pochta qutisi manzillari (fayllarning nomlari) mavjud. B foydalanuvchi agentidan foydalangan holda o‘z pochta qutisi tarkibini o‘zi xohlagan paytda olishi va o‘qishi mumkin. 7.8-rasmda mazkur misol ko‘rsatilgan. Bu ofisdagi xodimlar o‘rtasidagi an’anaviy qaydlarni almashinishga o‘xshaydi. Pochta xonasi mavjud, u yerda har bir xodimda uning ismi yozilgan pochta qutisi mavjud. A, B ga eslatma yuborishi kerak bo‘lganida, u xatni yozadi va B ning pochta qutisiga kiritadi.

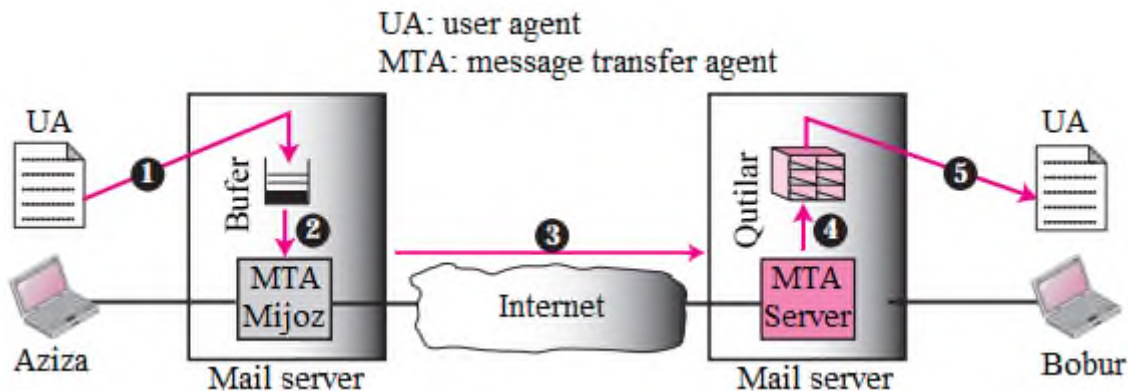


7.8- rasm. Birinchi senariy

B pochta qutisini tekshirib ko‘rganda, A ning qaydlarini topadi va o‘qiydi. Demak, elektron pochta jo‘natuvchisi va qabul qiluvchisi bitta pochta serverida bo‘lsa, faqat ikkita foydalanuvchi agenti kerak bo‘ladi.

**Ikkinchi senariy.** Ikkinchi holatda elektron pochta yuboruvchi va qabul qiluvchi ikki xil pochta serverlaridagi foydalanuvchilar (yoki dastur dasturlari). Xabar Internet orqali yuborilishi kerak. Bu yerda bizga 7.9- rasmda ko‘rsatilgandek foydalanuvchi agentlari (user agent

- UA) va xabarlarni uzatish agentlari (message transfer agents - MTA) kerak.



7.9- rasm. Ikkinchi senariy

Afoydalanuvchi o'z xabarini o'z pochta serveriga yuborish uchun foydalanuvchi agent dasturidan foydalanish kerak. Pochta serveri yuborilishini kutayotgan xabarlarni saqlash uchun navbatdan foydalanadi. Shuningdek, B ham pochta qutisida saqlangan xabarlarni olish uchun foydalanuvchi agenti dasturiga muhtoj. Xabarni Internet orqali A dan B ga yuborish kerak. Bu yerda ikkita xabarni uzatish agenti kerak: bitta mijoz va bitta server. Ko'pgina Internetdagi mijoz-server dasturlari singari, server doimo ishlashi kerak, chunki mijoz qachon ulanishni so'rashini bilmaydi. Boshqa tomondan, mijoz yuborishi kerak bo'lgan xabar mavjud bo'lganda tizim tomonidan ishga tushishi mumkin.

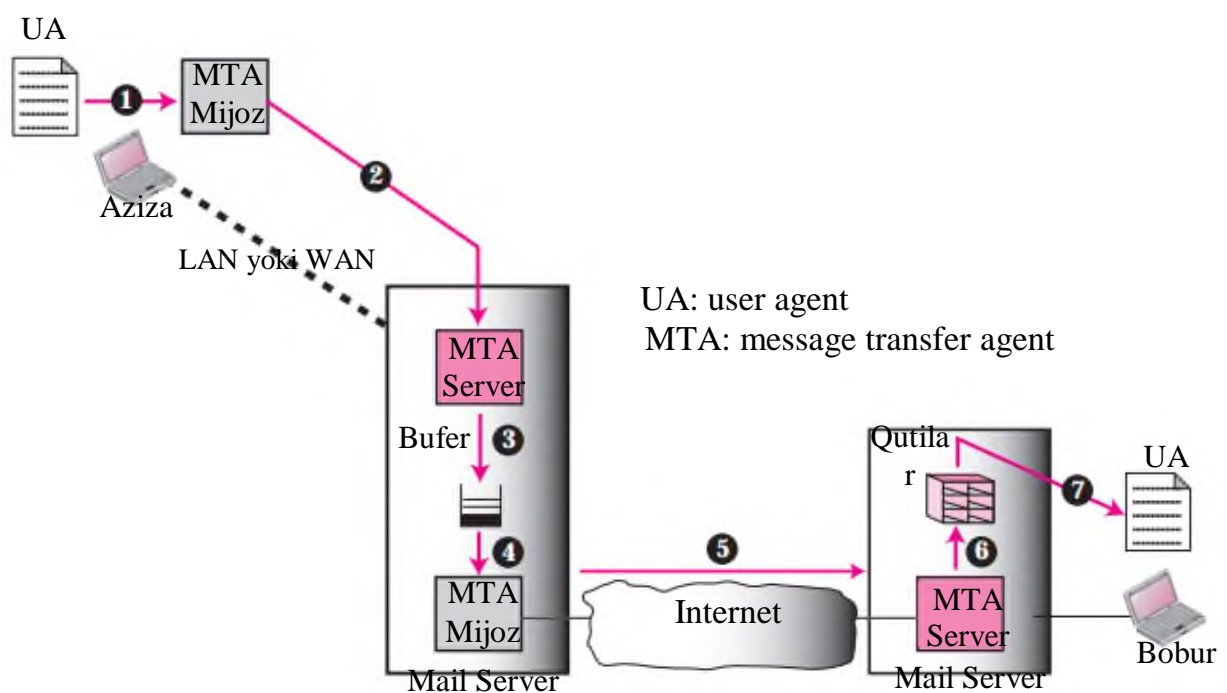
Elektron pochta jo'natuvchisi va qabul qiluvchisi turli xil pochta serverlarida bo'lganda, ikkita foydalanuvchi agenti (UA) va bir juft xabarlarni uzatish agentlari (MTA, mijoz va server) kerak bo'ladi.

**Uchinchi senariy.** 7.10- rasmda uchinchi senariy ko'rsatilgan. B, ikkinchi senariyda bo'lgani kabi, to'g'ridan-to'g'ri pochta serveriga ulangan. Ammo, A o'zining pochta serveridan ajratilgan. A pochta serveriga yoki point-to-point WAN orqali ulangan - masalan, dial-up modem, DSL yoki kabel-modem yoki bitta pochta serveridan foydalanish uchun tashkilotda LAN ga ulangan elektron pochta xizmatiga ulangan, bu orqali barcha foydalanuvchilar o'z xabarlarini ushbu pochta serveriga yuborishlari kerak.

Xabarni tayyorlash uchun A ga hali ham foydalanuvchi agenti kerak. Keyin u LAN yoki WAN orqali xabarni yuborishi kerak. B uni xabarlarni uzatish agentlari (mijoz va server) orqali amalga oshirish

mumkin. A har safar xabar yuborishi bilan u foydalanuvchi agentini chaqiradi, u esa o'z navbatida MTA mijozini chaqiradi. MTA mijoz doimiy ravishda ishlaydigan tizimdagi MTA serveri bilan aloqani o'rnatadi. Tizim qabul qilingan barcha xabarlarni navbatga qo'yadi. Keyin MTA mijozidan xabarlarni B ga yuborish uchun foydalanadi; tizim xabarni o'ladi va uni B ning pochta qutisiga saqlaydi.

O'ziga qulay bo'lganida, B o'zining foydalanuvchi agentidan xabarni olish uchun foydalanadi va uni o'qiydi. E'tibor bering, bizga ikkita juft MTA mijoz-server dasturlari kerak. Yuboruvchi pochta serveriga LAN yoki WAN orqali ulanganda bizga ikkita UA va ikki juft MTA (mijoz va server) kerak bo'ladi.

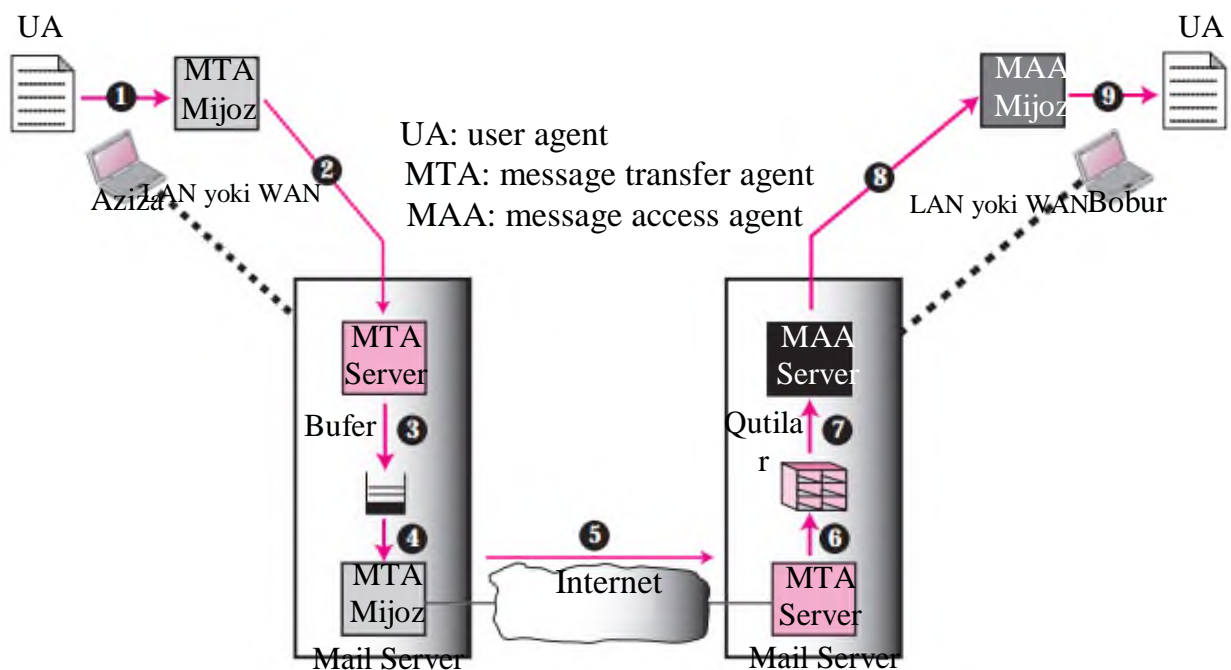


7.10- rasm. Uchinchi senariy

**To'rtinchi senariy.** To'rtinchi va eng keng tarqalgan holatda, B o'z pochta serveriga WAN yoki LAN orqali ulanadi. Xabar B ning pochta serveriga yetib borgandan so'ng, B uni qaytarib olishi kerak. Bu yerda bizga xabarlarga kirish agentlari (message access agents - MAA) deb nomlanadigan mijoz-server agentlarining yana bir to'plami kerak. B o'zining xabarlarini olish uchun MAA mijozidan foydalanadi. Mijoz har doim ishlaydigan MAA serveriga so'rov yuboradi va xabarlarni uzatishni talab qiladi. Vaziyat 7.11- rasmda keltirilgan.

Bu yerda ta'kidlashimiz kerak bo'lgan ikkita muhim jihat mavjud. Birinchidan, B pochta serverini chetlab o'ta olmaydi va to'g'ridan-to'g'ri MTA serveridan foydalana olmaydi. To'g'ridan-to'g'ri MTA serveridan foydalanish uchun B MTA serverini doimo ishga tushirishi kerak edi, chunki u qachon xabar kelishini bilmaydi. Bu shuni anglatadiki, agar u o'z tizimiga LAN orqali ulangan bo'lsa, B o'z kompyuterini doimo tarmoqa ulangan holda ushlab turishi kerak. Agar u WAN orqali ulangan bo'lsa, u doimo aloqani saqlab turishi kerak. Bugungi kunda ushbu holatlarning ikkalasi ham amalga oshirilmaydi.

Ikkinchidan, B ning mijoz-server dasturlariga yana bir juft dastur kerak: xabarlarga kirish dasturlari. Buning sababi, MTA mijoz-server dasturi **jo'natish (push)** dasturidir: mijoz xabarni serverga yuboradi. B ga **tortish (pull)** dasturi kerak. Mijoz xabarni serverdan tortib olishi kerak. 7.12- rasmda farq ko'rsatilgan.



7.11- rasm. To'rtinchi senariy



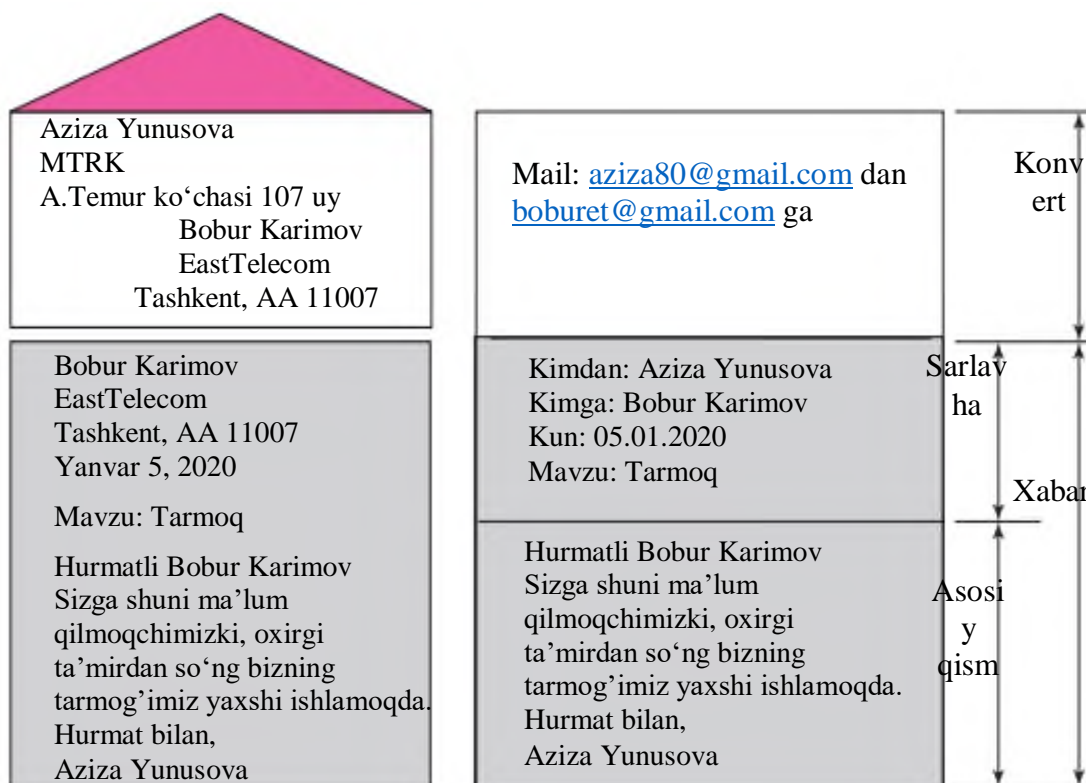


## 7.12- rasm. a) Mijoz xabarlarini jo‘natish, b) Mijoz xabarlarini tortish

Pochta serveriga LAN yoki WAN orqali ulanganda qabul qiluvchiga ikkita UA, ikkita juft MTA (mijoz va server) va bir juft MAA (mijoz va server) kerak bo‘ladi. Bu bugungi kunda eng keng tarqalgan holat.

### ***Foydalanuvchi agenti (User agent – UA)***

Elektron pochta tizimining birinchi komponenti foydalanuvchi agentidir (UA). Bu foydalanuvchi uchun xabar yuborish va qabul qilish jarayonini osonlashtiradigan xizmatni taqdim etadi. Foydalanuvchi agenti - bu xabarlarni tuzadigan, o‘qiydigan, javob beradigan va qayta yuboradigan dasturiy ta‘minot to‘plami. Shuningdek, u foydalanuvchi kompyuterlarida mahalliy pochta qutilariga ishlov beradi.



### 7.13- rasm. Elektron pochtaning formati

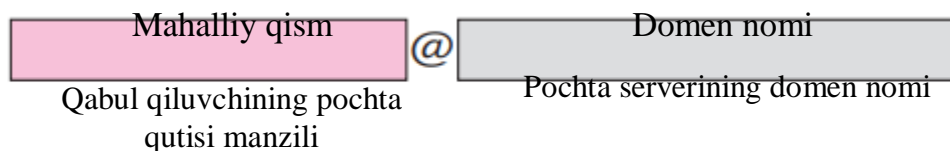
Pochta xabarini yuborish uchun foydalanuvchi UA orqali pochta jo‘natmalariga juda o‘xshash pochta xabarlarini yaratadi. U konvert va xabarga ega (7.13- rasmga qarang).

Konvertida odatda jo'natuvchining manzili, qabul qiluvchining manzili va boshqa ma'lumotlar mavjud bo'ladi. Xabarda sarlavha va asosiy qism mavjud. Xabarning sarlavhasi yuboruvchini, qabul qiluvchini, xabar mavzusini va boshqa ba'zi ma'lumotlarni aniqlaydi. Xabarning asosiy qismida qabul qiluvchi tomonidan o'qilishi kerak bo'lgan haqiqiy ma'lumotlar mavjud.

Foydalanuvchi agenti foydalanuvchi (yoki taymer) tomonidan ishga tushiriladi. Agar foydalanuvchida pochta bo'lsa, UA foydalanuvchini ogohlantirish bilan xabardor qiladi. Agar foydalanuvchi pochta qutisini o'qishga tayyor bo'lsa, ro'yxat ko'rsatiladi, unda har bir satr pochta qutisidagi ma'lum bir xabar to'g'risidagi ma'lumotlarning qisqacha mazmuni mavjud. Qisqacha bayon sifatida odatda yuboruvchining pochta manzili, xabar mavzusi va xabar yuborilgan yoki olingan vaqtni keltiriladi. Foydalanuvchi istalgan xabarlarni tanlab, undagi ma'lumotlarni ekranda ko'rsatishi mumkin.

### ***Manzil***

Pochtani yetkazib berish uchun pochta bilan ishlash tizimi noyob manzillarga ega manzillar tizimidan foydalanishi kerak. Elektron pochta manzili ikki qismdan iborat: mahalliy qism va @ belgisi bilan ajratilgan domen nomi (7.14- rasmga qarang).



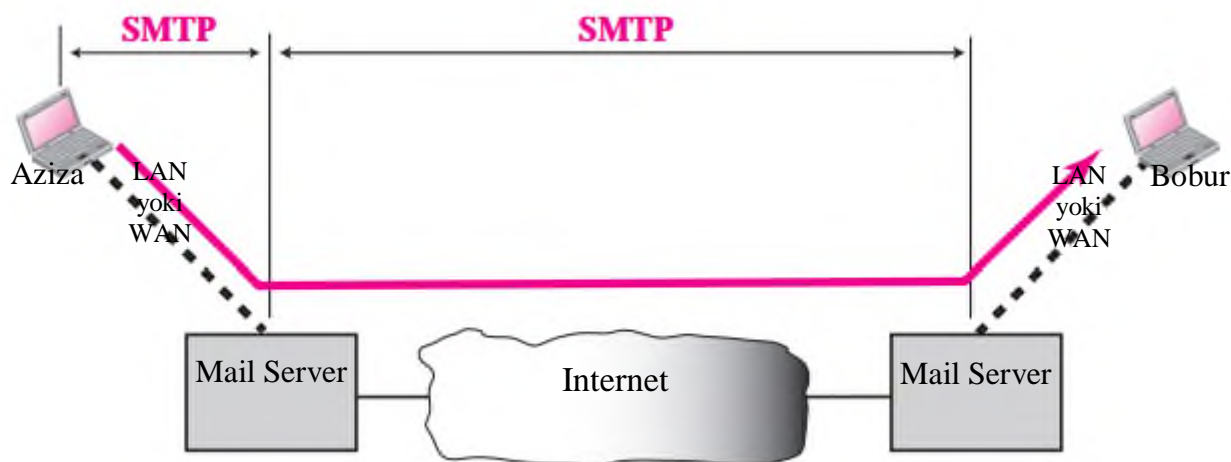
7.14- rasm. Pochta manzili

Mahalliy qism foydalanuvchi pochta qutisi deb nomlangan maxsus faylning (domenda takrorlanmas) nomini belgilaydi, unda foydalanuvchi uchun qabul qilingan barcha xabarlar xabarlarga kirish agenti tomonidan olish uchun saqlanadi.

Manzilning ikkinchi qismi - bu domen nomi. Tashkilot odatda elektron pochta xabarlarini olish va yuborish uchun bitta yoki bir nechta xostlarni tanlaydi; ular ba'zan pochta serverlari yoki almashtiruvchilar deb ataladi. Har bir pochta almashinuvchisiga tayinlangan domen nomi DNS ma'lumotlar bazasidan keladi yoki mantiqiy nomdir (masalan, tashkilot nomi).

## ***SMTP - Simple Mail Transfer Protocol***

Haqiqiy pochta o'tkazmasi xabarlarini uzatish agentlari (MTA) orqali amalga oshiriladi. Pochta xabarini yuborish uchun tizimda MTA mijoz bo'lishi kerak va pochta olish uchun tizimda MTA serveri bo'lishi kerak. MTA mijoz va Internetdagi serverni belgilaydigan rasmiy protokol SMTP (Simple Mail Transfer Protocol) - oddiy pochta uzatish protokoli deb nomlanadi. Yuqorida aytib o'tganimizdek, eng ko'p tarqalgan vaziyatda (to'rtinchi senariy) MTA mijozlariga xizmat ko'rsatuvchi ikkita juft dastur ishlatiladi. 7.15-rasmda ushbu senariyda SMTP protokoli oraliq'i ko'rsatilgan.



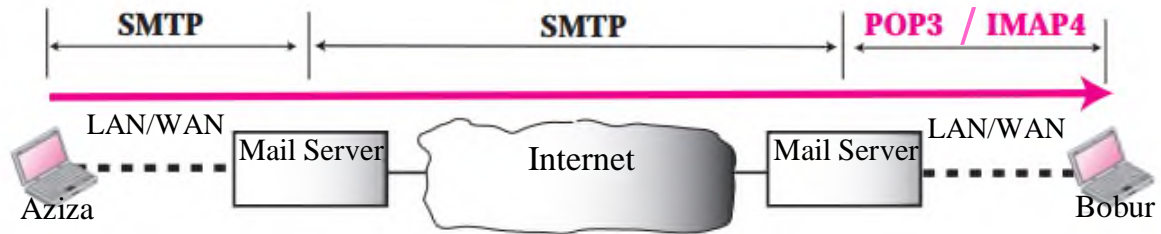
7.15- rasm. SMTP oraliq'i

SMTP ikki marta, yuboruvchi va jo'natuvchining pochta serverlari o'rtasida va ikkala pochta serverlari o'rtasida ishlatiladi. Pochta serveri va qabul qiluvchi o'rtasida boshqa protokol talab qilinadi. SMTP oddiygina buyruqlar va javoblar qanday qilib u yoki bu foydalanuvchiga yuborilishini belgilaydi. Har bir tarmoq joriy etish uchun dasturiy paketni erkin tanlaydi.

## ***POP va IMAP***

Hozirda xabarlariga kirishning ikkita protokoli mavjud: Post Office Protocol version 3 (POP3) va Internet Mail Access Protocol, version 4 (IMAP4 - Internet-pochtaga kirish protokoli, 4-versiya).

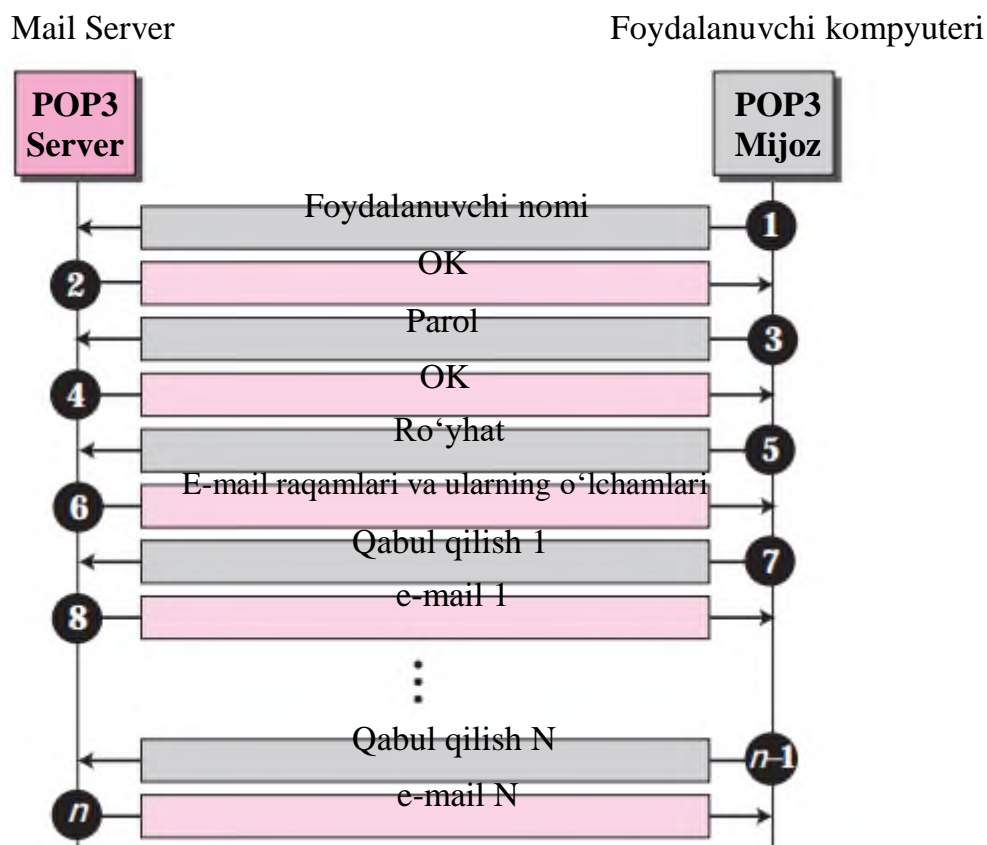
7.16- rasmda ushbu ikkita protokolning eng keng tarqalgan vaziyatdagi holati ko‘rsatilgan (to‘rtinchi senariy).



7.16- rasm. POP3 va IMAP4

### ***Post Office Protocol, version 3 – POP3***

Post Office Protocol, version 3 (POP3) sodda va cheklangan. Mijoz POP3 dasturi qabul qiluvchining kompyuteriga o‘rnatiladi; server POP3 dasturi pochta serveriga o‘rnatilgan. Foydalanuvchi o‘z elektron pochta qutisidagi pochta xabarlarini pochta serveridagi pochta qutisidan yuklab olishi kerak bo‘lganda mijoz dasturining pochtaga kirishi orqali jarayon boshlanadi. Mijoz TCP 110 portidagi serverga ulanishni ochadi. Keyin pochta qutisiga kirish uchun foydalanuvchi nomi va parolini yuboradi. Keyin foydalanuvchi pochta xabarlarini ketma-ket ro‘yxatlashi va foydalanishi mumkin. 7.17- rasmda POP3 dan foydalanib pochta xabarlarini yuklab olish misoli keltirilgan.



7.17- rasm. POP3 protokolimisol

POP3 ikkita rejimga ega: o'chirish va saqlash rejimlari. Ochirish rejimida, har bir qabul qilishdan so'ng, pochta qutisidan xabar o'chiriladi. Saqlash rejimida pochta (xabar) qabul qilinganidan keyin pochta qutisida qoladi. O'chirish rejimi odatda foydalanuvchi doimiy kompyuterida ishlayotganda foydalaniladi va o'qigan yoki javob bergandan so'ng qabul qilingan xabarlarni saqlash va tartiblashi mumkin.

Odatda, foydalanuvchipochtasigaasosiykompyuteridan (masalan, noutbuk) uzoqdakirganda, saqlashrejimidanfoydalaniladi. Pochta o'qiladi, lekin keyinchalik uni olish va tashkillashtirish uchun tizimda saqlanadi.

### ***Internet Mail Access Protocol, version 4 (IMAP4)***

Yana bir pochtaga kirish protokoli bu - Internet Mail Access Protocol, version 4 (IMAP4 - Internet pochtaga kirish protokoli, 4-versiya). IMAP4 POP3 ga o'xshash, ammo u ko'proq xususiyatlarga ega; IMAP4 yanada kuchli va murakkabroq. POP3ning bir necha jihatda kamchiligi bor. Foydalanuvchi o'z pochtagini serverda tartibga solishiga yo'l qo'ymaydi; foydalanuvchi serverda turli xil papkalarga

ega bo'la olmaydi (albatta, foydalanuvchi o'z kompyuterida papkalarini yaratishi mumkin.) Bundan tashqari, POP3 foydalanuvchiga yuklab olishdan oldin pochta tarkibini qisman tekshirishga ruxsat bermaydi.

IMAP4 quyidagi qo'shimcha funksiyalarni ta'minlaydi:

- ❖ Foydalanuvchi yuklab olishdan oldin elektron pochta xabarining sarlavhasini tekshirishi mumkin;

- ❖ Foydalanuvchi elektron pochta tarkibini yuklab olishdan oldin ma'lum bir belgilar qatoridan qidirishi mumkin;

- ❖ Foydalanuvchi elektron pochta qisman yuklab olishi mumkin. Bu, ayniqsa, o'tkazish qobiliyati cheklangan va elektron pochta yuqori o'tkazish qobiliyatini talab etuvchi multimediyalar kontent mavjud bo'lganida foydalidir;

- ❖ Foydalanuvchi pochta serverida pochta qutilarini yaratishi, o'chirishi yoki nomini o'zgartirishi mumkin;

- ❖ Foydalanuvchi elektron pochta saqlash papkasida pochta qutilarining ierarxiyasini yaratishi mumkin.

### **7.3. Tarmoq kompyuterlarini tashkil etish va boshqarish mexanizmlari**

#### ***Peer-to-peer arxitekturasi***

Garchi bugungi kunda Internetda mavjud bo'lgan ko'pgina dasturlar mijoz-server arxitekturasidan foydalanishsada, "peer-to-peer" (P2P) deb nomlangan arxitekturani ishlatish g'oyasi e'tiborni jalb qildi. Ushbu arxitekturada ikki tengdosh kompyuter (noutbuk, ish stoli yoki mainframe) o'zaro almashish xizmatlari uchun bir-biri bilan aloqa o'rnatishi mumkin. Ushbu arxitektura ba'zi sohalarda, masalan, mijoz audio-video fayli kabi katta faylni uzatmoqchi bo'lsa, mijoz-server arxitekturasi server mashinasiga juda ko'p yuklama berilganida. Ikkita tengdosh serverga bormasdan bir-birlariga ixtiyoriy fayl yoki ma'lumot almashishi kerak bo'lsa, bu g'oya qiziq hisoblanadi. Shu bilan birga, P2P arxitekturasi mijoz-server arxitekturasini e'tiborsiz qoldirmasligini ta'kidlashimiz kerak.

Aslida, bu jarayonda ishtirok etishni istagan ba'zi foydalanuvchilar tomonidan server vazifasini bo'lishishdir. Masalan,

bir nechta mijozlarga ulanishga ruxsat berishning o'rniga va har biri katta faylni yuklab olishning o'rniga, server har bir mijozga faylning bir qismini yuklab olishga va keyin uni bir-birlari bilan bo'lishishga ruxsat berishi mumkin. Faylning bir qismini yuklab olish yoki yuklab olingan faylni almashish jarayonida kompyuter mijoz va boshqa serverning rolini o'ynashi kerak. Boshqacha aytganda, kompyuter bir lahzada ma'lum bir dastur uchun mijoz, boshqa vaqtda esa server bo'lishi mumkin. Ushbu dasturlar endi tijorat tomonidan boshqariladi va Internetning rasmiy qismi emas.

### ***DHCP***

TCP/IP protokoli to'plamidan foydalanadigan har bir kompyuter o'z IP manzilini bilishi kerak. Agar kompyuter sinfsiz manzilni ishlatsa yoki tarmoqostining a'zosi bo'lsa, shuningdek, uning quyi tarmoq maskasini bilishi kerak. Bugungi kunda aksariyat kompyuterlar ikkita ma'lumotga muhtoj: boshqa tarmoqlar bilan bog'lanish uchun standart protokol manzili va manzillar o'rniga nomlardan foydalanish uchun serverining manzili. Boshqacha qilib aytganda, odatda to'rtta ma'lumot kerak:

- Kompyuterning IP manzili;
- Kompyuterning tarmoqosti maskasi;
- Marshrutizatorning IP manzili;
- Nomlar serverining IP manzili.

Ushbu to'rtta ma'lumot konfiguratsiya faylida saqlanishi mumkin va yuklash jarayonida kompyuter tomonidan unga kirish mumkin. Ammo disksiz ish stansiyasi yoki birinchi marta ishga tushirilgan diskka ega kompyuter haqida nima deyish mumkin?

Disksiz kompyuter bo'lsa, operatsion tizim va tarmoq dasturlari faqat o'qish uchun xotirada saqlanishi mumkin (ROM). Biroq, yuqoridagi ma'lumotlar ishlab chiqaruvchiga ma'lum emas va shuning uchun uni ROM-da saqlash mumkin emas. Ma'lumotlar mashinaning shaxsiy konfiguratsiyasiga bog'liq va mashina ulangan tarmoqni belgilaydi. DHCP xost konfiguratsiyasining rasmiy protokoli bo'lishidan oldin, ushbu taklif uchun ba'zi boshqa protokollar ishlatilgan. Biz ularni bu yerda qisqacha tasvirlaymiz.

### ***RARP***

Internet davrining boshida, yuklangan kompyuter uchun IP manzilini ta'minlash uchun Reverse Address Resolution Protocol (RARP) protokoli ishlab chiqilgan. RARP aslida ARP ning versiyasi bo'lib, ARP IP manzilni fizik manzilga xaritada joylashtiradi: RARP fizik manzilni IP manzilga xaritada joylashtiradi. Biroq, bugungi kunda RARP ikkita sababga ko'ra eskirgan. Birinchidan, RARP ma'lumotlar bog'lanish sathining uzatish xizmatidan foydalangan, ya'ni RARP serveri har bir tarmoqda bo'lishi kerak. Ikkinchidan, RARP faqat kompyuterning IP manzilini taqdim qilishi mumkin, ammo bugungi kunda kompyuter yuqorida aytib o'tilgan barcha to'rta ma'lumotga muhtoj.

### ***BOOTP***

Bootstrap Protocol (BOOTP) - bu DHCP ning dastlabki avlodidir. Bu RARP protokolining ikkita kamchiliklarini bartaraf etish uchun mo'ljallangan mijoz-server protokoli. Birinchidan, u mijoz-server dasturi bo'lganligi sababli BOOTP serveri Internetning istalgan joyida bo'lishi mumkin. Ikkinchidan, yuqorida biz aytib o'tgan barcha ma'lumotlarni, shu jumladan IP manzilni ham berishi mumkin. Yuqorida tavsiflangan to'rta ma'lumotni berish uchun, u RARP protokoli bo'yicha barcha cheklovlarni olib tashlaydi. BOOTP - bu statik konfiguratsiya protokoli. Mijoz o'z IP manzilini so'raganda, BOOTP serveri mijozning fizik manzili va uning IP manzili bilan mos keladigan jadvalga murojaat qiladi. Bu fizik manzil va mijozning IP manzili o'rtasidagi bog'lanish allaqachon mavjudligini anglatadi. Ulanish oldindan belgilanadi.

Dinamik konfiguratsiya protokoli kerak bo'lgan ba'zi holatlar mavjud. Masalan, xost bir fizik tarmoqdan boshqasiga o'tganda, uning fizik manzili o'zgaradi. Boshqa bir misol sifatida, host vaqtincha IP manzilidan foydalanilishini xohlaydigan holatlar mavjud. BOOTP bu vaziyatlarni bajara olmaydi, chunki fizik va IP manzillar orasidagi bog'lanish statik va ma'mur tomonidan o'zgartirilmaguncha jadvalda o'rnatiladi. DHCP ushbu kamchiliklarni bartaraf etish uchun ishlab chiqilgan.

### ***DHCP***

Dynamic Host Configuration Protocol (DHCP) - bu disksiz kompyuter yoki birinchi marta ishga tushirilgan kompyuter uchun to'rta ma'lumotni berish uchun yaratilgan mijoz-server protokoli.



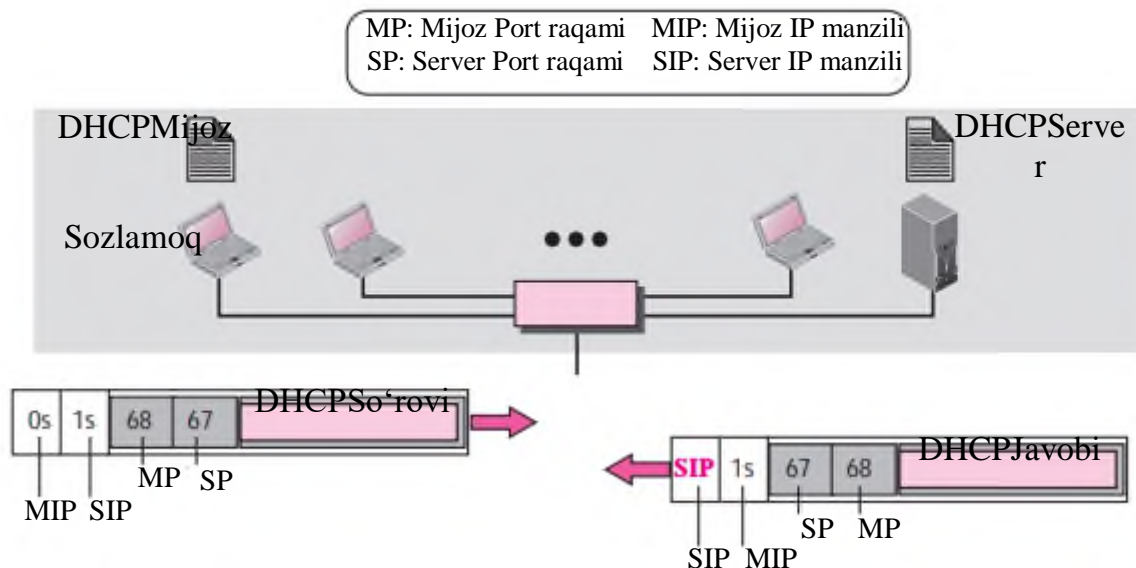
DHCP BOOTP ning vorisi hisoblanadi. BOOTP eskirgan deb hisoblansa ham, konfiguratsiya uchun BOOTP dan foydalanadigan ba'zi tizimlar mavjud bo'lishi mumkin. DHCP mijoz va serveri bitta tarmoqda yoki turli tarmoqlarda bo'lishi mumkin. Keling, har bir holatni alohida muhokama qilaylik.

**Bitta tarmoq**

Amaliyotda juda keng tarqalgan bo'lmasa ham, ma'mur mijoz va serverni 7.18- rasmda ko'rsatilgandek bir xil tarmoqqa joylashtirishi mumkin.

Ushbu holatda operatsiyalar quyidagicha tavsiflanishi mumkin:

1. DHCP server 67- raqamli UDP portida passiv ochiq buyruq chiqaradi va mijozni kutadi.
2. Yuklangan mijoz 68-portda faol ochiq buyruq chiqaradi. Xabar 67-sonli port va 68-raqamli manba portidan foydalanib, UDP foydalanuvchi datagramiga kiritilgan. UDP foydalanuvchi datagrami o'z navbatida, IP datagrammasida kapsullangan. O'quvchi, o'z IP manzilini (manbaa manzili) va serverning IP manzilini (belgilangan manzilni) bilmagan holda, qanday qilib mijoz IP datagramini yuborishi mumkinligini so'rashi mumkin. Mijoz barcha 0 larni manba manzili sifatida va hamma 1 larni belgilangan manzil sifatida ishlatadi.



7.18- rasm. Mijoz va server bir tarmoqda

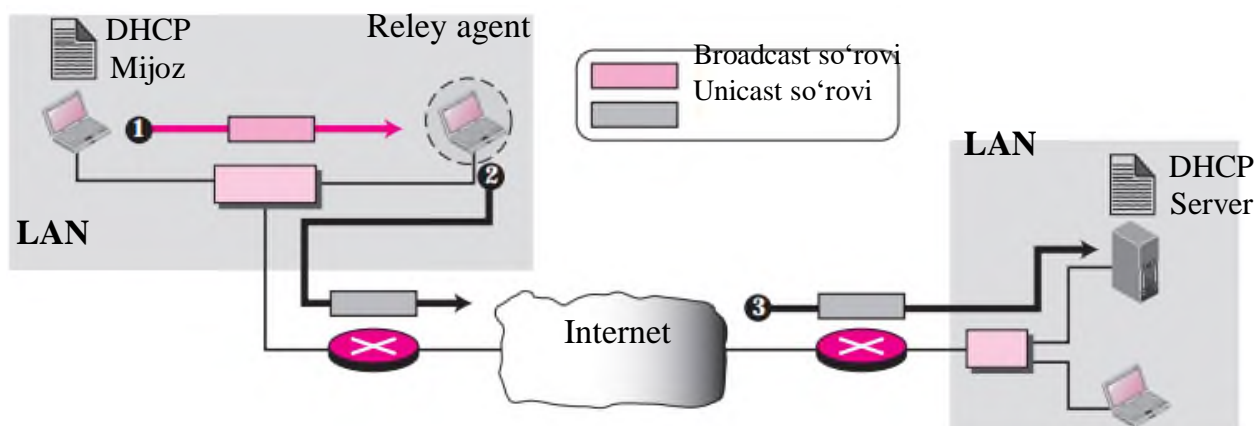
3. Server UDP manbasi 67- va 68- raqamli portlar yordamida translyatsiya yoki unicast xabar orqali javob beradi. Javob unicast bo'lishi mumkin, chunki server mijozning IP manzilini biladi. Shuningdek, u mijozning fizik manzilini ham biladi, ya'ni manzili xaritalash uchun mantiqiy ARP xizmatlariga muhtoj emasligini anglatadi. Biroq, ba'zi tizimlar ARPni chetlab o'tishga imkon bermaydi, natijada translyatsiya manzilidan foydalaniladi.

### ***Turli tarmoqlar***

Amaliy sathning boshqa jarayonlarida bo'lgani kabi, mijoz bitta tarmoqda, server esa bir necha boshqa tarmoqlar bilan ajratilgan holda bo'lishi mumkin. 7.19-rasmda vaziyat ko'rsatilgan.

Biroq, hal qilinishi kerak bo'lgan bitta muammo mavjud. DHCP so'rovi translyatsiya qilinadi, chunki mijoz serverning IP manzilini bilmaydi. Efirning (Broadcast) IP datagrami biron bir marshrutizator orqali o'taolmaydi. Bunday paketni olgan marshrutizator uni rad etadi. Eslatib o'tamiz, barcha 1 larning IP manzili cheklangan tarqatish manzili.

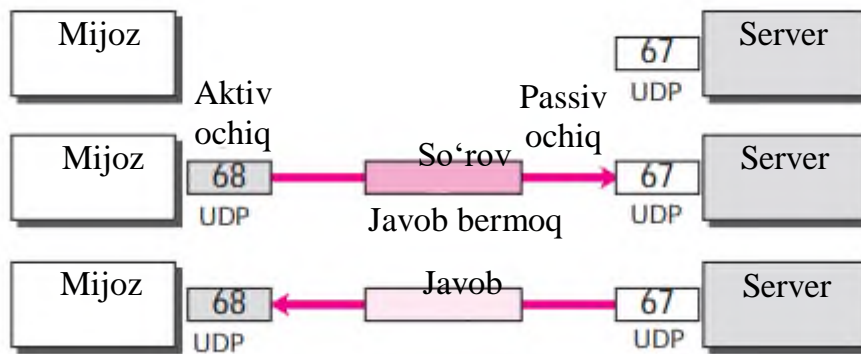
Muammoni hal qilish uchun vositachiga ehtiyoj bor. Reley (retranslyator) sifatida hostlardan biri (yoki amaliy sathda ishlashga sozlanishi mumkin bo'lgan marshrutizator) ishlatilishi mumkin. Bunday holatda host reley agenti deb ataladi. Reley agenti DHCP serverining unicast manzilini biladi va 67- portida eshitiladigan xabarlarni tinglaydi. Ushbu paketni qabul qilganda, xabarni unicast ma'lumotlar datagramiga singdiradi va so'rovni DHCP serveriga yuboradi. Paketda bitta belgilangan manzilga ega paket har qanday marshrutizator tomonidan yo'naltiriladi va DHCP serveriga yetib boradi. DHCP server xabarning reley agentidan (retranslyator) kelganligini biladi, chunki so'rov xabaridagi maydonlardan biri retranslyator agentining IP manzilini belgilaydi. Releyagenti javobni olgandan keyin uni DHCP mijoziga yuboradi.



7.19- rasm. Mijoz va server ikki turli xil tarmoqda

### ***UDP portlar***

7.20- rasmda mijoz va DHCP-serverning o'zaro ta'siri ko'rsatilgan. Serverda odatda 67- port mashhur. Mijoz juda mashhur bo'lgan 68-portdan foydalanadi. Vaqtinchalikportning o'rniga belgilangan 68-portni tanlashning sababi, serverdan mijozga javob yuborilganda muammoning oldini olishdir. Muammoni tushunish uchun keling, vaqtinchalik portidan foydalanilgan vaziyatni ko'rib chiqaylik. Aytaylik, tarmoqdagi A xodimi DHCP mijozini 2017 vaqtinchalik portida (tasodifiy tanlangan) ishlatmoqda. Xuddi shu tarmoqdagi xost B, vaqtinchalik portida DAYTIME mijozidan foydalanmoqda (tasodifan bir xil). Endi DHCP serveri manzil porti raqami 2017 va FFFFFFFF<sub>16</sub> IP manzilini yuborgan holda javob qaytaradi. Har bir host ushbu IP-manzilni o'z ichiga olgan paketni ochishi kerak. Host A dasturini vaqtinchalik portida 2017 xabarini topadi. To'g'ri xabar DHCP mijoziga yetkaziladi. Noto'g'ri xabar DAYTIME mijoziga yetkazilgan. Chalkashlik IP manzili va port raqamining kombinatsiyasi bo'lgan socket manzili asosida paketlarning demultipleksifikatsiyasi bilan bog'liq. Bunday holda, ikkalasi ham bir xil.



7.20- rasm. UDP portning qo'llanilishi

Belgilangan portdan foydalanish (1024 dan kichik) bir xil ikkita belgilangan (destination) port raqamlaridan foydalanishni oldini oladi. Host B 68 vaqtinchalik porti sifatida tanlay olmaydi, chunki vaqtinchalik port raqamlari 1023 dan katta.

Qiziquvchan o'quvchi B hosti DHCP mijozini ishlaganda nima bo'lishini so'rashi mumkin. Bunday holda, socketning manzili bir xil va ikkala mijoz ham xabarni olishadi. Bunday vaziyatda uchinchi identifikatsiya raqami mijozlarni farqlaydi. DHCP o'tkazma identifikatori deb nomlangan boshqa raqamdan foydalanadi, DHCP bilan bog'liq har bir ulanish uchun tasodifiy tanlanadi. Ikkala hostning bir vaqtning o'zida bir xil identifikatorni tanlashi mumkin emas.

### ***Xatoliklarni nazorat qilish***

Agar so'rov yo'qolsa yoki buzilsa nima bo'ladi? Agar javob buzilgan bo'lsa nima bo'ladi? DHCP ni ishlatishda xatolarni boshqarish kerak. DHCP xatolarni boshqarishni ta'minlamaydigan UDP dan foydalanadi. Shuning uchun, DHCP xato nazoratini ta'minlashi kerak. Xatolarni boshqarish ikkita strategiya orqali amalga oshiriladi:

1. DHCP UDP tekshiruv jadvalidan foydalanishni talab qiladi. Yodda tutingki, UDP da nazorat summasidan foydalanish majburiy emas.

2. Agar DHCP mijoz so'rovga DHCP javobini olmasa, taymerlar va retranslyatsiya siyosatidan foydalanadi. Biroq, bir nechta hostlar so'rovni qayta yuborishlari kerak bo'lgan hollarda, masalan, tirbandlikni oldini olish uchun (masalan, elektr uzilishidan keyin),

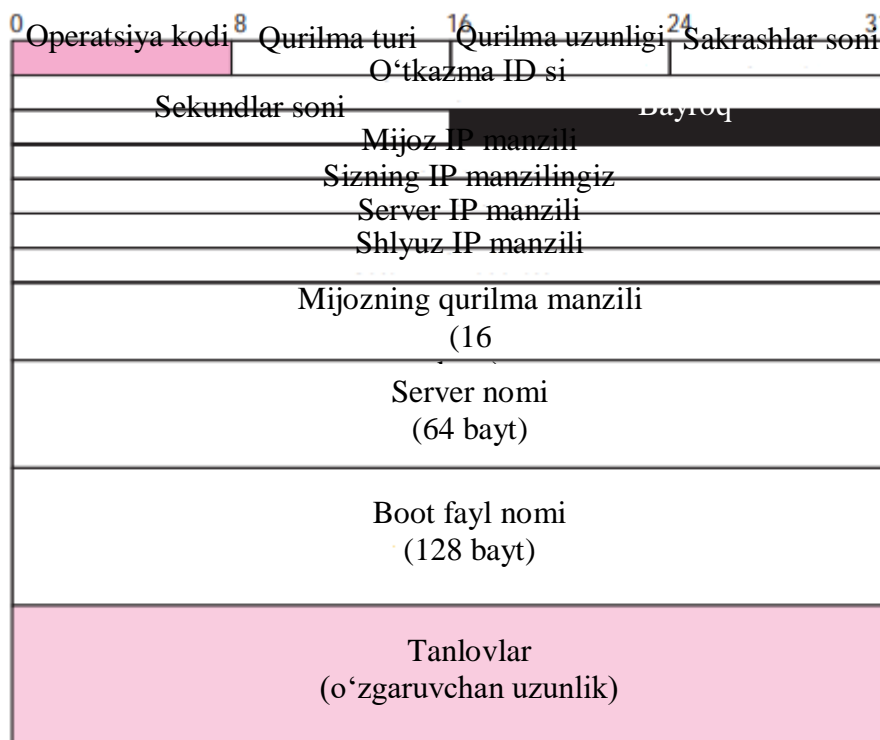
DHCP mijozni taymerlarini o‘rnatish uchun tasodifiy raqamdan foydalanishga majbur qiladi.

### ***DHCP paket formati***

Har bir maydon quyidagicha tavsiflanadi:

❖ **Operatsiya kodi.** Ushbu 8 bitli maydon DHCP paketining turini aniqlaydi: so‘rov (1) yoki javob (2).

❖ **Qurilma turi.** Bu fizik tarmoq turini aniqlaydigan 8 bitli maydon. Har bir tarmoq turiga butun son berilgan. Masalan, Ethernet uchun qiymat 1 ga teng.



7.21- rasm. DHCP paketning formati

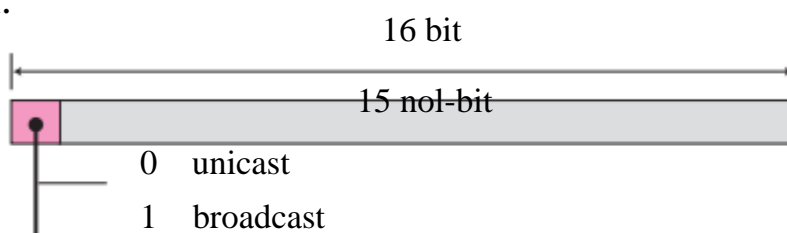
❖ **Qurilma uzunligi.** Bu baytlarda fizik manzil uzunligini belgilaydigan 8 bitli maydon. Masalan, Ethernet uchun qiymat 6 ga teng.

❖ **Sakrashlar soni.** Bu 8 bitli maydon bo‘lib, paketning sayohatga tushadigan maksimal sakrashlar (hops) sonini belgilaydi.

❖ **O‘tkazma ID si.** Bu 4 baytli butun sonni o‘z ichiga olgan maydon. Tranzaktsiyani identifikatsiya qilish mijoz tomonidan o‘rnatiladi va javobni so‘rov bilan moslashtirish uchun ishlatiladi. Server javobida bir xil qiymatni qaytaradi.

❖ **Sekundlar soni.** Bu 16 bitli maydon bo‘lib, mijoz yuklashni boshlagan vaqtdan boshlab o‘tgan soniyalar ko‘rsatadi.

❖ **Bayroq.** Bu 16 bitli maydon bo‘lib, unda faqat chap tomondagi bit ishlatiladi va qolgan bitlar 0 larga o‘rnatilishi kerak. Chap tomonida bit serverdan yuborilgan majburiy javobni (unicast o‘rniga) belgilaydi. Agar javob mijozga bir xilda bo‘lishi kerak bo‘lsa, belgilangan IP paketning IP manzili mijozga tayinlangan manzildir. Mijoz o‘z IP manzilini bilmaganligi sababli paketni tashlab yuborishi mumkin. Agar IP datagram translyatsiya qilinsa, har bir host uzatilgan xabarini qabul qiladi va qayta ishlaydi. 7.22-rasmda bayroq formati ko‘rsatilgan.



7.22-rasm. Bayroq formati

❖ **Mijozning IP manzili.** Bu mijozning IP manzilini o‘z ichiga olgan 4 baytlik maydon. Agar mijozda bunday ma’lumotlar bo‘lmasa, bu maydon 0 qiymatiga ega.

❖ **Sizning IP manzilingiz.** Bu mijozning IP manzilini o‘z ichiga olgan 4 baytlik maydon. U mijozning talabiga binoan server tomonidan (javob xabarida) to‘ldiriladi.

❖ **Server IP manzili.** Bu serverning IP manzilini o‘z ichiga olgan 4 baytlik maydon. Bu server tomonidan javob xabarida to‘ldiriladi.

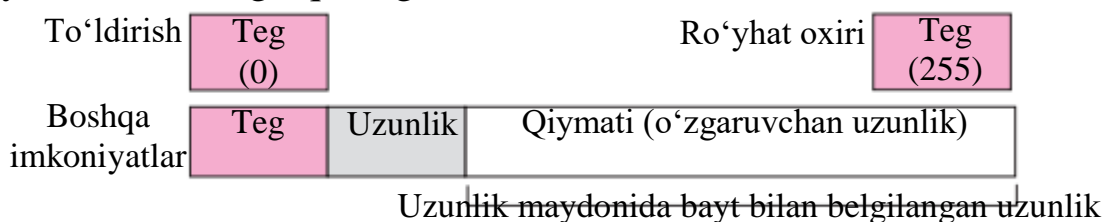
❖ **Shlyuzning IP-manzili.** Bu marshrutizator IP manzilini o‘z ichiga olgan 4 baytlik maydon. Bu server tomonidan javob xabarida to‘ldiriladi.

❖ **Mijozning qurilma manzili.** Bu mijozning fizik manzili. Server ushbu manzilni mijoz tomonidan yuborilgan freymdan aniqlashi mumkin bo‘lsa ham, agar so‘rov xabari manzili mijoz tomonidan aniq yetkazib berilsa, samaraliroq bo‘ladi.

❖ **Server nomi.** Bu 64 baytlik maydon bo‘lib, ixtiyoriy server tomonidan javoblar paketida to‘ldiriladi. Unda serverning domen nomidan iborat nol bilan tugagan satr mavjud. Agar server ushbu maydonni ma’lumotlar bilan to‘ldirishni istamasa, server uni barcha 0 lar bilan to‘ldirishi kerak.

❖ **Boot fayl nomi.** Bu 128 baytli maydon bo‘lib, uni ixtiyoriy ravishda javoblar paketida to‘ldirish mumkin. Unda yuklash faylining to‘liq nomi ko‘rsatilgan nol bilan tugagan satr mavjud. Mijoz ushbu yo‘ldan boshqa yuklash ma’lumotlarini olish uchun foydalanishi mumkin. Agar server ushbu maydonni ma’lumotlar bilan to‘ldirishni istamasa, server uni barcha 0 lar bilan to‘ldirishi kerak.

❖ **Tanlovlar.** Bu 64 baytli maydon ikki tomonlama maqsadga ega. U qo‘shimcha ma’lumotni (masalan, tarmoq maskasi yoki standart marshrutizator manzili) yoki sotuvchiga oid ba’zi ma’lumotlarni olishi mumkin. Maydon faqat javob xabarida ishlatiladi. Server 99.130.83.99 Request line qiymatiga ega bo‘lgan IP manzil formatida “**magic cookie**” deb nomlangan raqamdan foydalanadi. Mijoz xabarni o‘qib bo‘lgach, bu **magic cookie** faylni qidiradi. Agar mavjud bo‘lsa, keyingi 60 bayt variantlardir. Variant uchta maydondan iborat: 1 baytlik teg maydoni, 1 bayt uzunlik maydoni va o‘zgaruvchan uzunlikdagi qiymat maydoni. Uzunlik maydoni butun tanlovni emas, balki qiymat maydonining uzunligini belgilaydi. 7.23-rasmga qarang.



7.23- rasm. Variant formati

IP manzillarini o‘z ichiga olgan maydonlarning uzunligi 4 baytga ko‘payadi. Faqat uzunligi 1 bayt bo‘lgan to‘ldirish tanlovi faqat moslash uchun ishlatiladi. Faqatgina 1 bayt uzunlikdagi ro‘yhat oxirida parametrlar maydonining tugashini bildiradi. Javob xabarida qo‘shimcha ma’lumot berish uchun sotuvchilar 128 dan 254 gacha teglardan foydalanishlari mumkin.

Tanlovlar ro‘yxati 7.1- jadvalda keltirilgan

7.1- jadval

Teg	Uzunlik	Qiymat	Tavsif
0			To‘ldirish
1	4	Tarmoqosti maskasi	Tarmoqosti maskasi
2	4	Kun vaqti	Vaqtini qoplash
3	O‘zgaruvchi	IP manzil	Standart

			marshrutizator
4	O'zgaruvchi	IP manzil	Vaqt serveri
5	O'zgaruvchi	IP manzil	IEN 16 serveri
6	O'zgaruvchi	IP manzil	DNS server
7	O'zgaruvchi	IP manzil	Jurnallar (log) serveri
8	O'zgaruvchi	IP manzil	Quote server
9	O'zgaruvchi	IP manzil	Print server
10	O'zgaruvchi	IP manzil	Impress
11	O'zgaruvchi	IP manzil	RLP server
12	O'zgaruvchi	DNS nom	Host nomi
13	2	Butun son	Boot fayl o'lchami
53	1		Dinamik sozlash uchun ishlatiladi
128-254	O'zgaruvchi	Maxsus axborot	Sotuvchiga hos
255			Ro'yxat yakuni

#### **7.4. Tarmoq fayllarini tashkil etish va boshqarish mexanizmlari**

##### ***FTP – File Transfer Protocol***

Fayl uzatish protokoli (FTP) bu faylni bir hostdan boshqasiga nusxalash uchun TCP/IP tomonidan ta'minlanadigan standart mexanizm. Fayllarni bir tizimdan boshqasiga o'tkazish oddiy va sodda ko'rinadigan bo'lsada, birinchi navbatda ba'zi muammolarni hal qilish kerak. Masalan, ikkita tizim turli xil fayl nomlari tizimidan foydalanishi mumkin. Ikki tizimda matn va ma'lumotlarni taqdim etishning turli xil usullari bo'lishi mumkin. Ikki tizim turli xil katalog tuzilmalariga ega bo'lishi mumkin. Ushbu muammolarning barchasi FTP tomonidan juda sodda va ishonchli yechim bilan hal qilindi.

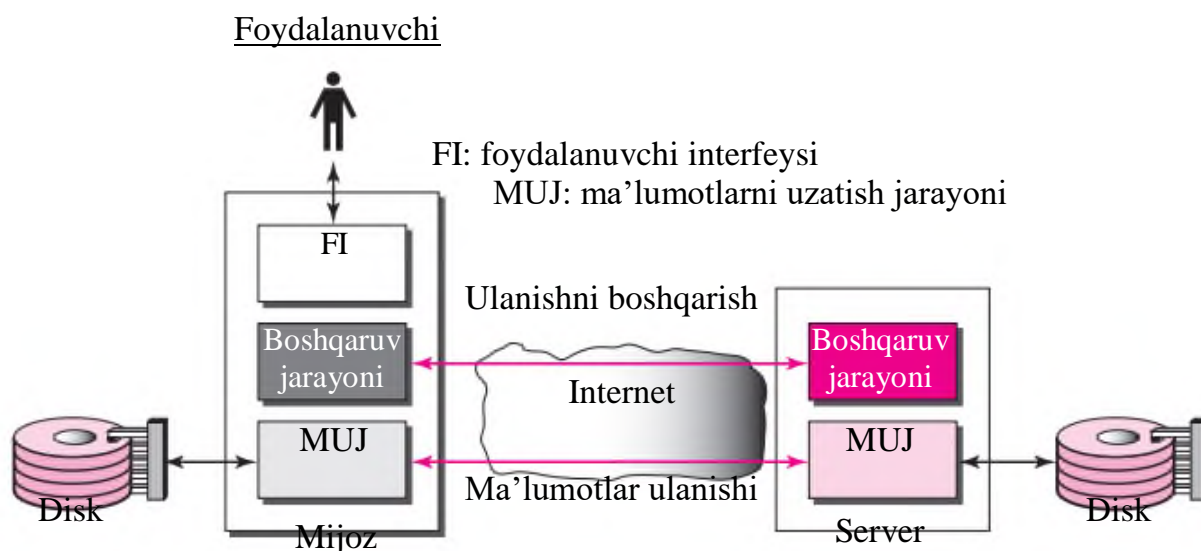
FTP boshqa mijoz-server dasturlaridan farq qiladi, chunki u hostlar o'rtasida ikkita ulanishni o'rnatadi. Bir ulanish ma'lumot uzatish uchun ishlatiladi, ikkinchisi boshqarish ma'lumotlari uchun (buyruqlar va javoblar). Buyruqlarni ajratish va ma'lumotlarni uzatish FTP ni yanada samaraliroq qiladi. Tekshirish aloqasi juda oddiy aloqa qoidalaridan foydalanadi. Bir vaqtning o'zida faqat buyruqlar qatorini yoki javoblar qatorini o'tkazish kerak. Boshqa tomondan, ma'lumotlar ulanishi, uzatiladigan ma'lumotlar turlarining xilma-xilligi sababli yanada murakkab qoidalarga muhtoj.



FTP ikkita belgilangan TCP portlaridan foydalanadi: 21-port bu boshqarish ulanishi uchun va 20-port ma'lumotlar ulanishi uchun ishlatiladi.

FTP TCP xizmatlaridan foydalanadi. Unga ikkita TCP ulanishi kerak. Belgilangan 21-port nazorat ulanishi uchun va ma'lumot ulanish uchun belgilangan 20-port ishlatiladi.

7.24- rasmda FTP ning asosiy modeli ko'rsatilgan. Mijoz uchta kompo-nentdan iborat: foydalanuvchi interfeysi, mijozni boshqarish jarayoni va mijoz ma'lumotlarini uzatish jarayoni. Server ikkita komponentdan iborat: serverni boshqarish jarayoni va server ma'lumotlarini uzatish jarayoni. Boshqarish aloqalari boshqarish jarayonlari o'rtasida amalga oshiriladi. Ma'lumot aloqalari ma'lumotlar uzatish jarayonlari o'rtasida amalga oshiriladi.



7.24- rasm. FTP protokoli

Boshqarish aloqasi butun interfaol FTP seansi davomida ulanishda qoladi. Ma'lumotlar ulanishi ochiladi va keyin har bir uzatilgan fayl uchun yopiladi. Har safar fayllarni uzatishni o'z ichiga olgan buyruqlar ishlatilganda ochiladi va fayl uzatilganda yopiladi. Boshqacha qilib aytganda, foydalanuvchi FTP seansini boshlaganda, boshqaruv ulanishi ochiladi. Boshqarish aloqasi ochiq bo'lsa ham, agar bir necha fayllar uzatilsa ma'lumotlar ulanishi bir necha marta ochilishi va yopilishi mumkin.

### **Aloqalar**

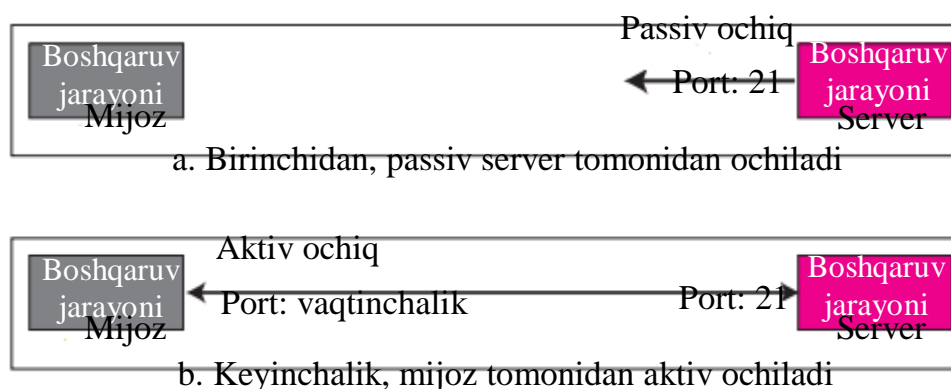
## Ikkita FTP Ulanishlari

boshqarish va ma'lumotlarning turlari xil strategiyalar va turlari xil port raqamlarida foydalanadi.

**Boshqarish aloqasi.** Boshqarish aloqasi shu paytgacha tasvirlangan boshqa amaliy dasturlar singari yaratilgan. Ikki bosqich mavjud:

1. Server belgilangan 21-portda passiv ochiladi va mijozni kutadi.
2. Mijoz vaqtinchalik portidan foydalanadi va faol ochiladi.

Ulanish butun jarayon davomida ochiq qoladi. IP protokoli tomonidan foydalaniladigan xizmat turi kechikishni minimallashtiradi, chunki bu foydalanuvchi va server o'rtasidagi interfaol aloqadir. Foydalanuvchi buyruqlarni yozadi va javoblarni sezilarli kechiktirmasdan olishni kutadi. 7.25- rasmda server va mijoz o'rtasidagi dastlabki ulanish ko'rsatilgan.



7.25- rasm. Boshqaruv aloqasini ochish

**Ma'lumot ulanishi.** Ma'lumot ulanishi server tarafdagi belgilangan 20- portdan foydalanadi. Dastlabki ma'lumot ulanishini yaratish bosqichlari 7.26- rasmda keltirilgan.

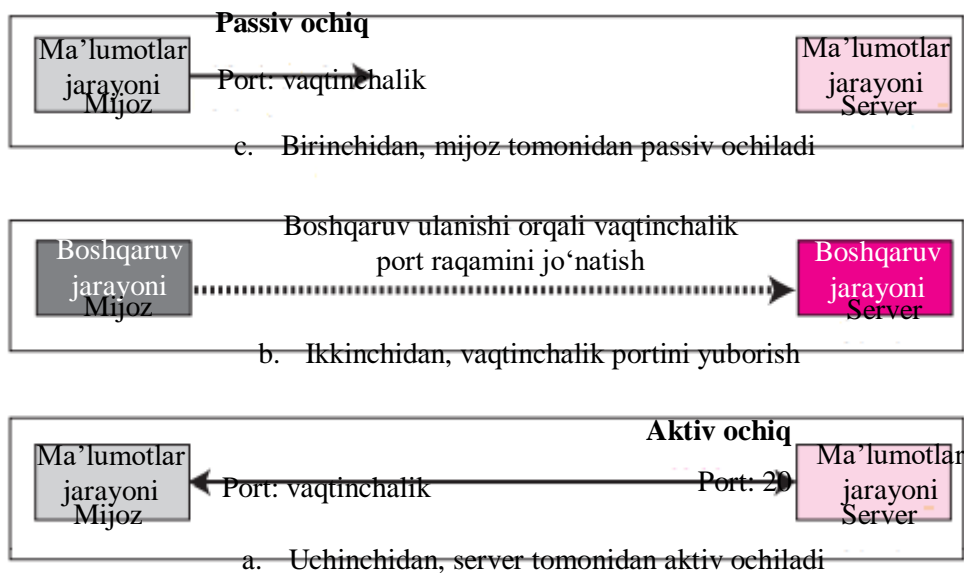
Quyida FTP ma'lumotlar ulanishini qanday yaratishi ko'rsatilgan:

1. Server emas, balki mijoz vaqtinchalik portidan foydalangan holda passiv ochishni amalga oshiradi.

Bu mijoz tomonidan bajarilishi kerak, chunki fayllarni uzatish buyruqlarini mijoz amalga oshiradi.

2. Mijoz ushbu port raqamini PORT buyrug'i yordamida serverga yuboradi.

3. Server port raqamini oladi va belgilangan 20- port va olingan vaqtinchalik port raqamidan foydalanib, faol ochilishni amalga oshiradi.

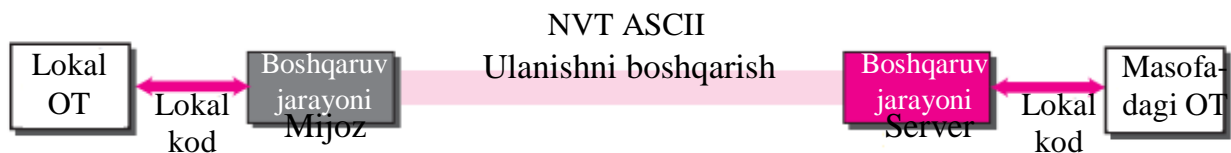


7.26- rasm. Ma'lumotlar ulanishini yaratish

### **Kommunikatsiya (communication)**

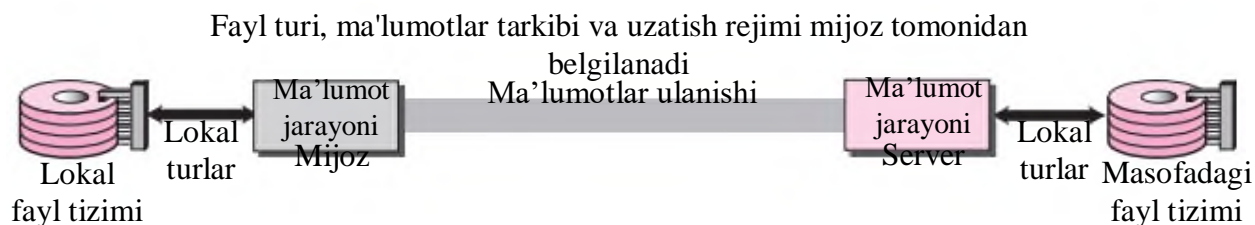
Turli xil kompyuterlarda ishlaydigan FTP mijoz va serveri biri-biri bilan aloqa o'rnatishi kerak. Ushbu ikkita kompyuter turli xil operatsion tizimlardan, turli xil belgilar to'plamlaridan, turli xil fayl tuzilmalaridan va har xil fayl formatlaridan foydalanishi mumkin. FTP ushbu har xillikni moslashtirishi kerak. FTPda ikki xil yondashuv mavjud, biri boshqaruv ulanishi uchun va ikkinchisi ma'lumotlar ulanishi uchun. Biz har bir yondashuvni alohida o'rganamiz.

**Tekshirish ulanishi orqali aloqa.** FTP ham SMTP singari boshqarish ulanishi orqali bog'lanish yondoshuvidan foydalanadi. U NVT ASCII belgilar to'plamidan foydalanadi (7.27- rasmga qarang). Muloqot buyruqlar va javoblar orqali amalga oshiriladi. Ushbu oddiy usul nazorat ulanishi uchun yetarli, chunki biz bir vaqtning o'zida bitta buyruq (yoki javob) yuboramiz. Har bir buyruq yoki javob faqat bitta qisqa qatordir, shuning uchun fayl formati yoki fayl tuzilishi haqida tashvishlanmaslik kerak.



7.27- rasm. Boshqaruv ulanishidan foydalanish

**Ma'lumot-ulanish orqali aloqa.** Ma'lumotlar ulanishining maqsadi va bajarilishi boshqarish ulanishining maqsadidan farq qiladi. Ma'lumotlar ulanishi orqali fayllarni uzatishni maqsad qilinadi. Mijoz uzatiladigan fayl turini, ma'lumotlarning tuzilishini va uzatish rejimini belgilashi kerak. Faylni ma'lumotlar ulanishi orqali yuborishdan oldin, boshqarish ulanishi orqali uzatishga tayyorlanadi. Turli xillik muammosi aloqaning uchta atributini aniqlash orqali hal qilinadi: fayl turi, ma'lumotlar tuzilishi va uzatish rejimi (7.28- rasmga qarang).



7.28- rasm. Ma'lumot ulanishidan foydalanish

**Fayl turi.** FTP ma'lumotlar ulanishi orqali quyidagi fayl turlaridan birini uzatishi mumkin:

- ❖ **ASCII (American Standard Code for Information Interchange) fayli.** Bu matnli fayllarni uzatish uchun standart format. Har bir belgi NVT ASCII (NVT - Network Virtual Terminal) yordamida kodlangan. Yuboruvchi faylni o'z ko'rinishidan NVT ASCII belgilariga, qabul qiluvchi esa NVT ASCII belgilarini asl ko'rinishiga o'zgartiradi.

- ❖ **EBCDIC (Extended Binary Coded Decimal Interchange Code) fayli.** Agar ulanishning bir yoki ikkala uchi EBCDIC kodlashdan foydalansa, fayl EBCDIC kodlash yordamida uzatilishi mumkin.

- ❖ **Tasvir fayli.** Bu ikkilik fayllarni uzatish uchun standart format. Fayl hech qanday izoh va kodlamasdan bitlarning uzluksiz oqimi sifatida yuboriladi. Bu asosan kompilyatsiya qilingan dasturlar kabi ikkilik fayllarni uzatish uchun ishlatiladi.

❖ **Ma'lumotlar tuzilmasi.** FTP ma'lumotlar tuzilmasi haqida quyidagi izohlardan birini ishlatib, ma'lumot uzatishda ulanishni amalga oshirishi mumkin:

❖ **Fayl tuzilishi (standart).** Fayl tuzilishga ega emas. Bu baytlarning doimiy oqimi.

❖ **Yozuv tuzilishi.** Fayl yozuvlarga bo'linadi. Buni faqat matnli fayllarda ishlatish mumkin.

❖ **Sahifa tuzilishi.** Fayl sahifalarga bo'linadi, har bir sahifada sahifa raqami va sahifa sarlavhasi bo'ladi. Sahifalar tasodifiy yoki ketma-ket saqlanishi va shu tartibda ularga kirishi amalga oshirilishi mumkin.

❖ **Uzatish rejimi.** FTP faylni ma'lumotlar uzatish tarmog'i orqali quyidagi uchta uzatish usullaridan biri yordamida uzatishi mumkin.

❖ **Oqim rejimi.** Bu standart rejim. Ma'lumotlar FTP dan TCP ga uzluksiz ravishda baytlar oqimi sifatida yetkaziladi. TCP ma'lumotlarni segmentlar kattaligiga mos qismlarga ajratish uchun javobgardir. Agar ma'lumotlar shunchaki baytlar oqimi bo'lsa (fayl tuzilmasi), hech qanday Fayl-ohiri (EOF-end-of-file) talab etilmaydi. Bu holda fayl-oxiri jo'natuvchi tomonidan ma'lumotlar ulanishining to'xtatilishi hisoblanadi. Agar ma'lumotlar yozuvlarga bo'linadigan bo'lsa (yozuvlar tuzilishi), har bir yozuvda 1 baytlik yozuv-oxiri (EOR- end-of-record) belgisi bo'ladi va faylning oxirida 1 baytli fayl-oxiri belgisi bo'ladi.

❖ **Bloklash rejimi.** Ma'lumotlar FTP dan TCP ga bloklarda etkazilishi mumkin. Bunday holda, har bir blok oldida 3 baytli sarlavha bo'ladi. Birinchi bayt "blok tavsiflovchisi" deb ataladi; keyingi ikki bayt blok hajmini baytlarda belgilaydi.

❖ **Siqilgan rejim.** Agar fayl katta bo'lsa, ma'lumotlarni siqish mumkin. Odatda ishlatiladigan siqishni usuli - bu uzunlikdagi kodlash. Ushbu usulda ma'lumotlar blokining ketma-ket paydo bo'lishi bitta holat va bir nechta takrorlanishlar bilan almashtiriladi. Matn faylida bu odatda bo'sh joylar sifatida qaraladi. Ikkilik faylda odatda nol belgilar siqiladi.

### ***HTTP - Hypertext Transfer Protocol***

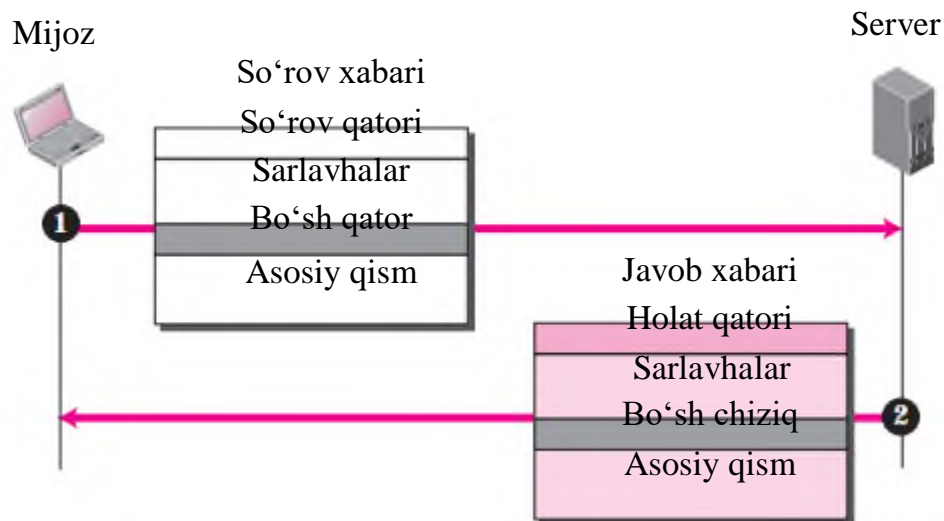
Gipermatnni uzatish protokoli (HTTP) bu asosan Internetga kirish uchun ishlatiladigan protokoldir. HTTP bir qarashda FTP va

SMTP ning funksiyalari jamlanmasi kabi imkoniyatlarga ega. FTP ga o'xshash, chunki u fayllarni uzatadi va TCP xizmatlaridan foydalanadi. Biroq, bu FTP ga qaraganda ancha sodda, chunki u faqat bitta TCP ulanishidan foydalanadi. Alohida nazorat aloqasi yo'q; faqat ma'lumotlar mijoz va server o'rtasida uzatiladi.

HTTP SMTP ga o'xshaydi, chunki mijoz va server o'rtasida uzatiladigan ma'lumotlar SMTP xabarlariga o'xshaydi. SMTP dan farqli o'laroq, HTTP xabarlari odamlar tomonidan o'qilishi uchun mo'ljallanmagan; ularni HTTP serveri va HTTP mijoz (brauzer) o'qiydi va inson uchun tushinarli ko'rinishda taqdim qiladi. SMTP xabarlari saqlanadi va yo'naltiriladi, ammo HTTP xabarlari darhol yetkaziladi. Mijozdan serverga buyruqlar so'rov xabariga kiritilgan. Talab qilingan fayl yoki boshqa ma'lumotlar tarkibi javob xabariga joylashtirilgan. HTTP belgilangan 80-portda TCP xizmatlaridan foydalanadi.

### **HTTP o'tkazmasi**

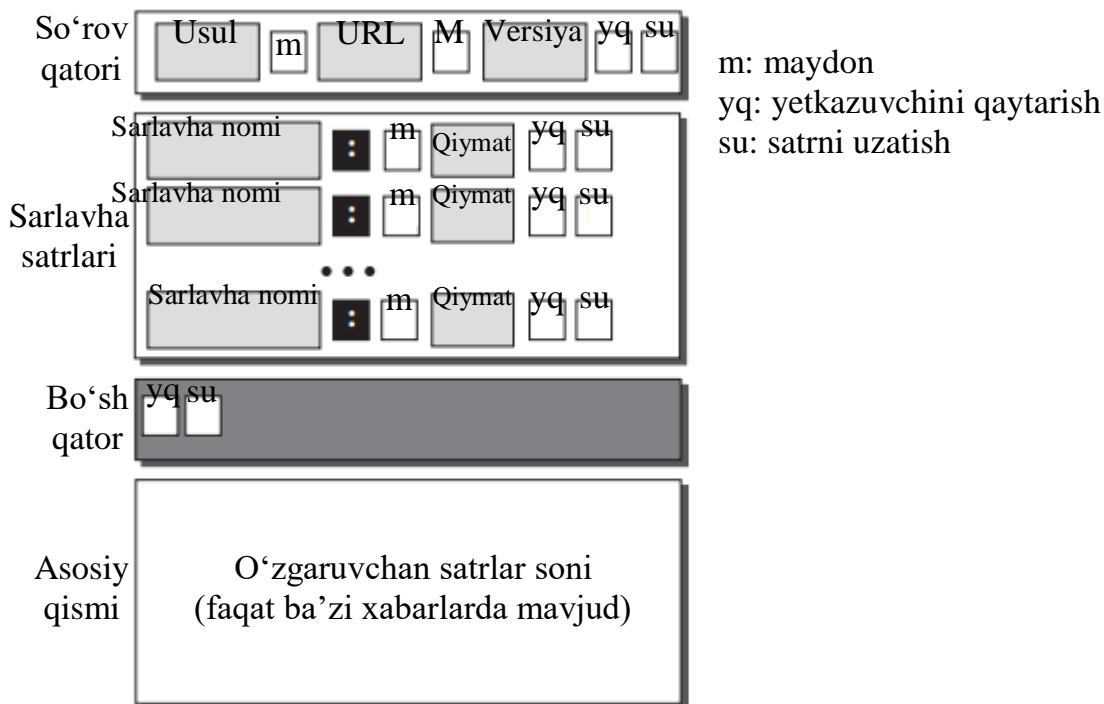
7.29- rasmda mijoz va server o'rtasidagi HTTP o'tkazmasi ko'rsatilgan. HTTP TCP xizmatlaridan foydalansa ham, HTTP o'zi holatsiz protokol bo'lib, server mijoz haqida ma'lumot saqlamaydi. Mijoz operatsiyani so'rov yuborish orqali amalga oshiradi va server javob yuboradi.



7.29- rasm. HTTP o'tkazmasi

**So'rov xabari.** So'rovning shakli 7.30- rasmda keltirilgan. So'rov xabari so'rovlar qatoridan, sarlavhadan va ba'zan asosiy qismdan (body) iborat.

**So‘rov qatori.** So‘rov xabaridagi birinchi qator so‘rovlar qatori deb nomlanadi. 7.30- rasmda ko‘rsatilgandek, ushbu qatorda ajratilgan uchta maydon mavjud. Maydonlar usullar, URL va versiya deb nomlanadi. Ushbu uchlikni bo‘sh joy belgisi bilan ajratish kerak. Ikkita belgidan so‘ng, yetkazuvchini qaytarish va satr tarjimasidan keyin qator tamomlanadi. Usul maydoni so‘rov turini belgilaydi.



7.30- rasm. So‘rov xabarining formati

HTTP 1.1 versiyasida, 7.3- jadvalda ko‘rsatilgandek, bir nechta usullar aniqlangan.

7.3- jadval

Usul	Harakat
GET	Serverdan hujjatni so‘raydi
HEAD	Hujjat haqida axborotni so‘raydi, lekin hujjatning o‘zini so‘ramaydi
POST	Mijozdan serverga ba‘zi axborotlarni jo‘natadi
PUT	Serverdan mijozga hujjatni jo‘natadi
TRACE	Kiruvchi so‘rovni akslantiradi
CONNECT	Zahirilangan
DELETE	Veb sahifani o‘chirish
OPTIONS	Mavjud variantlar haqida ma’lumot so‘rash

Ikkinchi maydon, URL - tegishli veb-sahifaning manzili va nomini aniqlaydi. Uchinchi maydon, versiya, protokol versiyasini taqdim etadi, masalan, HTTP ning 1.1 versiyasi.

**Talab qilingan xabarning sarlavhasi.** So‘rov satridan so‘ng biz nol yoki undan ko‘p so‘rov sarlavhalari qatoriga ega bo‘lishimiz mumkin. Har bir sarlavha qatori mijozdan serverga qo‘shimcha ma’lumotlarni yuboradi. Masalan, mijoz hujjatning maxsus formatda yuborilishini talab qilishi mumkin. Har bir sarlavhada sarlavha nomi, ustun, bo‘sh joy va sarlavha qiymati mavjud (7.30- rasmga qarang). 7.4- jadvalda so‘rovda tez-tez ishlatiladigan ba’zi sarlavhalar ko‘rsatilgan. Qiymat maydoni har bir sarlavha nomi bilan bog‘liq bo‘lgan qiymatlarni belgilaydi. Qiymatlar ro‘yxatini tegishli RFC larda (request for comment) topish mumkin.

7.4- jadval

Sarlavha	Tavsif
User-agent	Mijoz dasturini aniqlaydi
Accept	Mijoz qabul qilishi mumkin bo‘lgan media formatni aniqlaydi
Accept-charset	Mijoz ishlay olishi mumkin bo‘lgan belgilar to‘plamini ko‘rsatadi
Accept-encoding	Mijoz ishlay olishi mumkin bo‘lgan kodlash sxemasini ko‘rsatadi
Accept-language	Mijoz qabul qilishi mumkin bo‘lgan tilni ko‘rsatadi
Authorization	Mijoz gataqdimetilgan ruxsatlarni ko‘rsatadi
Host	Mijozning host va port raqamlarini ko‘rsatadi
Date	Joriy sanani ko‘rsatadi
Upgrade	Afzal aloqa protokolini belgilaydi
Cookie	Cookie-ni serverga qaytaradi
If-Modified-Since	Cookie-ni serverga qaytaradi

**So‘rov xabardagi tana (Body In Request Message).** Tana so‘rov xabarida mavjud bo‘lishi mumkin. Odatda, u yuborilishi kerak bo‘lgan sharhni o‘z ichiga oladi.

**Holat satri.** Javob xabaridagi birinchi qatorga holat satri deyiladi. Ushbu satrda uchta maydon mavjud bo‘lib, ular bo‘shliqlar bilan ajratilgan va jo‘natuvchini qaytarish va satr tarjimasini bilan tugatilgan. Birinchi maydon HTTP protokoli versiyasini aniqlaydi, masalan, 1.1. Holat kodi maydoni so‘rovning holatini belgilaydi. U uchta raqamdan iborat. 100 diapazonidagi kodlar faqat ma’lumot bo‘lsa, 200 diapazonidagi kodlar muvaffaqiyatli so‘rovni bildiradi. 300 diapazonidagi kodlar mijozni boshqa URLga yo‘naltiradi va 400 diapazonidagi kodlar mijozning saytida xato borligini ko‘rsatadi. Va nihoyat, 500 diapazonidagi kodlar server tarafidagi

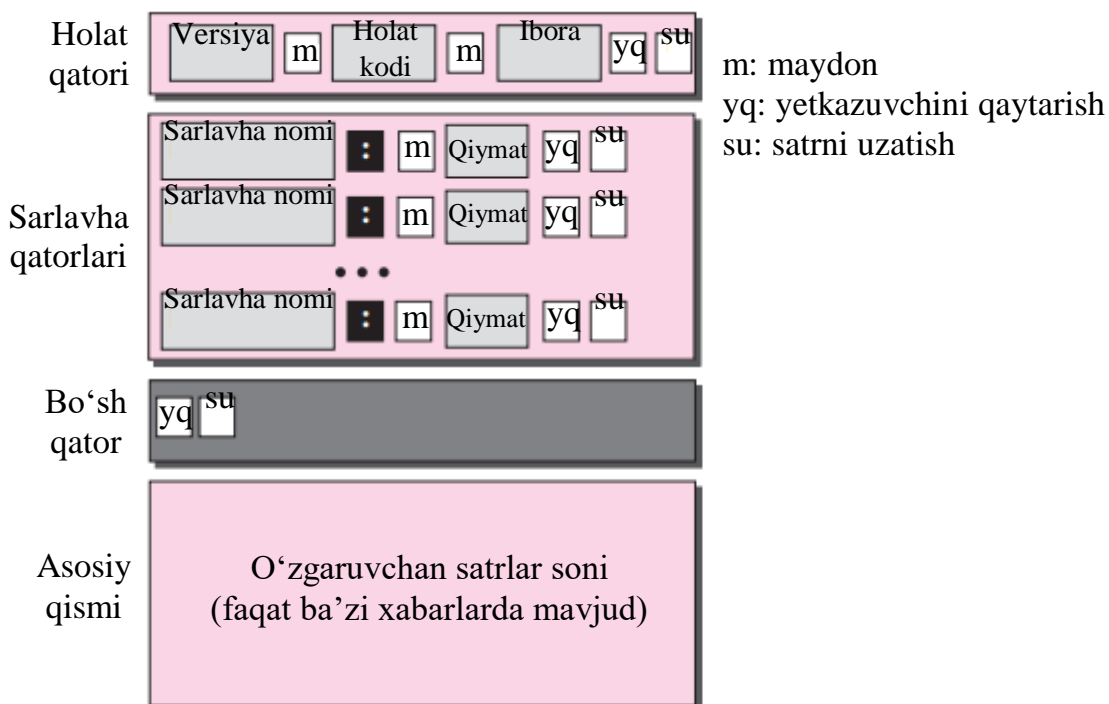


xatoni ko'rsatadi. Holat iborasi holat kodini matn shaklida ifodalaydi. Holat kodi va holat iborasi uchun mumkin bo'lgan qiymatlar 7.5- jadvalda keltirilgan.

7.5- jadval

Holat kodi	Holat matni	Tavsif
<b>Axborot</b>		
100	Continue	So'rovning dastlabki qismi qabul qilindi, davom ettirish
101	Switching	Server kommutatsiya protokollariga mos kelmoqda
<b>Muvaffaqiyat</b>		
200	OK	So'rov muvaffaqiyatli amalga oshirildi
201	Created	Yangi URL yaratildi
202	Accepted	So'rov qabul qilindi, ammo u darhol amalga oshirilmadi
204	No content	Tanada kontent mavjud emas
<b>Qayta yo'naltirish</b>		
301	Moved permanently	So'ralgan URL server tomonidan endi foydalanilmaydi
302	Moved temporarily	So'ralgan URL vaqtincha ko'chirilgan
304	Not modified	Hujjat o'zgartirilmagan
<b>Mijoz xatosi</b>		
400	Bad request	So'rovda sintaksis xato mavjud
401	Unauthorized	So'rovda tegishli avtorizatsiya mavjud emas
403	Forbidden	Xizmat ras etildi
404	Not found	Hujjat topilmadi
405	Method not allowed	Usul bu URLda qo'llab-quvvatlanmadi
406	Not acceptable	So'ralgan formatni qabul qilish imkoni mavjud emas
<b>Server xatosi</b>		
500	Internal server error	Server tarafda xatolik mavjud, masalan ishdan chiqish
501	Not implemented	Talab qilingan amalni bajarib bo'lmaydi
503	Service unavailable	Xizmat vaqtincha ish faoliyatida emas

**Javob xabari.** Javob xabari formati 7.31- rasmda keltirilgan. Javob xabari holat satridan, sarlavha satrlaridan, bo'sh satr va ba'zan asosiy qismdan iborat.



7.31- rasm. Javob xabari formati

**Javob xabaridagi sarlavha satrlari.** Holat satridan so'ng nol yoki undan ko'p javob sarlavhalariga ega bo'lish mumkin. Har bir sarlavha satri serverdan mijozga qo'shimcha ma'lumotlarni yuboradi. Masalan, jo'natuvchi hujjat haqida qo'shimcha ma'lumot yuborishi mumkin. Har bir sarlavhao'z ichiga sarlavha nomi, ustun, bo'sh joy va sarlavha qiymatini oladi. 7.6- jadvalda javob xabarida tez-tez ishlatiladigan ba'zi nomlar ko'rsatilgan.

7.6- jadval

Sarlavha	Tavsif
Date	Joriy sanani ko'rsatadi
Upgrade	Afzal aloqa protokolini belgilaydi
Server	Server haqida ma'lumot beradi
Set-Cookie	Server mijozdan cookieni saqlashni so'raydi
Content- Encoding	Kodlash sxemasini aniqlaydi
Content-Language	Tilni aniqlaydi
Content-Length	Hujjatning uzunligini ko'rsatadi
Content-Type	Media turini aniqlaydi
Location	Mijozdanboshqasaytgaso'rovjo'natishniso'raydi
Accept-Ranges	Server so'ralgan bayt diapazonlarini qabul qiladi
Last-modified	Ohirgi o'zgartirishning sanasi va vaqtini beradi

**Asosiy qism.** U o'z tarkibida serverdan mijozga yuboriladigan hujjatni saqlaydi. Agar javob xato xabar bo'lmasa, u holda asosiy qism mavjud.

## **7.5. Tarmoq dasturlarini tashkil etish va boshqarish mexanizmlari**

Ushbu bo‘limda tarmoq dizaynerlari tomonidan hal etilishi kerak bo‘lgan dasturiy mahsulot bilan bog‘liq bo‘lgan to‘rt masalani ko‘rib o‘tamiz:

- ❖ Qanday qilib saytlar boshqa saytlarni topish uchun manzillardan foydalanadilar?
- ❖ Xabarlar qanday yo‘naltiriladi va ular qanday yuboriladi?
- ❖ Jarayonlar bir-biri bilan qanday aloqa qiladi?
- ❖ Resurslarga bo‘lgan ziddiyatli talablar qanday hal qilinadi?

### ***Manzillash bo‘yicha kelishuvlar***

Tarmoq saytlari bir-birlari bilan aloqa qilishlari va bir-birlarining manbalariga kirishlari uchun o‘z foydalanuvchilarini qanday qilib aniqlab olishni belgilashlari kerak. Point-to-Point bog‘lanishdan tashqari saytlar bir-birlari bilan to‘g‘ridan-to‘g‘ri bog‘lanmaganligi uchun ismlar, manzillar va marshrutlar talab qilinadi. Shuning uchun manzillash protokollari tarmoq topologiyasi va har bir saytning geografik joylashuvi bilan chambarchas bog‘liq. Ba‘zi hollarda, o‘z tizimida birlik ma‘lum bo‘lgan nomni anglatadigan "mahalliy nom" va birlikni o‘z tizimidan tashqarida bildiruvchi nom - "global nom" o‘rtasida farq bor. Bu farq foydalidir, chunki u har bir saytga mahalliy miqyosda amalga oshirish qiyin bo‘lgan yagona nomlash qoidalarini o‘rnatmasdan, o‘zlarining standartlari bo‘yicha o‘z bo‘linmalarini aniqlash imkoniyatini beradi. Boshqa tomondan, global nom standart uzunliklar, formatlar va boshqa global kelishuvlarga muvofiq bo‘lishi kerak.

Oddiy misol sifatida Internet manzilidan foydalanganda, biz chapdan o‘ngga quyidagi tartibda ierarxik tuzilishga moslikni ko‘rishimiz mumkin: mantiqiy foydalanuvchidan host-mashinasiga, host-mashinadan tarmoq mashinasiga, tarmoq mashinadan klastergacha va klasterdan tarmoqqa. Masalan, har bir Internet manzilida: info@tuit.uz yoki igss12@aber.ac.uk, har bir komponentni ajratish uchun nuqtalar ishlatiladi. Esda tutish juda oson bo‘lgan ushbu elektron pochta manzillari mos keladigan qurilma manzillarga

o'zgartirilishi kerak. Ushbu o'zgartirish kompyuter operatsion tizimining tarmoq bo'limi tomonidan amalga oshiriladi.

Yuqorida keltirilgan misollar Domen nomlari xizmati (DNS) protokoliga asoslangan bo'lib, uning asosiy vazifasi Internet manzillarini hal qilishdir. Agar biz info@tuit.uz ni tarkibiy qismlarga ajratsak, bizda quyidagilar bo'ladi:

- ❖ info – mahalliy foydalanuvchi;
- ❖ tuit – info deb nomlangan mahalliy foydalanuvchi uchun host;
- ❖ uz – Toshkent axborot texnologiyalari universiteti uchun tarmoq.

### ***Marshrutlash strategiyalari***

Marshrutizator - bu, asosan, dasturiy ta'minot asosida ishlaydigan, har xil protokoli manzili bo'lgan ikki xil mahalliy tarmoq yoki ikkita tarmoq segmentlari o'rtasida trafikni marshrutlovchi ichki tarmoq qurilmasi. Yo'naltirish ma'lumotni tarmoqdagi bir nuqtadan boshqasiga olish imkonini beradi. Buning uchun har bir yo'nalishni aniq belgilash kerak. Ma'lumotlar tarmoqqa chiqarilganidan so'ng, marshrutizator tarmoqdagi to'g'ri (mos) tugun uni qabul qilishiga ishonch hosil qiladi. Tarmoq dizayni o'zgarganda marshrutizatorlarning roli o'zgaradi. Marshrutizatorlar saytlarni bir-biriga va Internetga ulash uchun keng qo'llaniladi. Ular turli funktsiyalarda, shu jumladan oldindan belgilangan joylarda hosil bo'ladigan ma'lumotni himoyalash, bir nuqtadan boshqasiga eng tezkor marshrutni tanlash va qo'shimcha tarmoq ulanishlarini ta'minlash uchun ishlatilishi mumkin, shunda bir hududdagi tarmoq muammosi boshqa tarmoqdagi ishni yomonlashtirmaydi.

Marshrutlash protokollarida manzillash, manzilni aniqlash, xabar formati va xatolar to'g'risida xabarlar bo'lishi kerak. Marshrutlash protokollarning aksariyati har bir tugunni aniqlash uchun tarmoq va tugun raqamidan foydalanadigan manzil formatiga asoslangan. Tarmoq yoqilganda, har bir protokol jadvalga to'g'ridan-to'g'ri ulangan tarmoqlarning manzillarini yozib oladi. Marshrutlash protokollari marshrutizatorlarning o'zaro ta'sirini ta'minlashga imkon berganligi sababli, har bir protokol o'z xizmatlarini amalga oshirishi natijasida olingan tarmoq manzillarini almashish osonlashadi. Belgilangan vaqt oralig'ida tarmoq ichidagi har bir protokol o'zining

barcha marshrutlash jadvalining nusxasini uzatadi. Oxir-oqibat, barcha marshrutizatorlar har xil maqsadli tarmoqlarga qanday yetib borishni bilishadi.

Garchi manzillar marshrutizatorlarga ma'lumotni bitta tarmoqdan boshqasiga yuborish imkonini beradigan bo'lsa-da, lekin ularni bitta tarmoqdagi bir nuqtadan ikkinchi nuqtaga olish uchun foydalanib bo'lmaydi. Buni marshrutlash orqali amalga oshirish kerak, bu marshrutizatorga apparat manzili bo'yicha asl manzilni xaritaga tushirish va xaritani kelajakda uzatish uchun foydalaniladigan jadvalda saqlashga imkon beradi.

Turli xil xabar formatlari marshrutlash protokollari bilan belgilanadi. Ushbu xabarlar protokolga tarmoqdagi yangi tugunlarni topish, ularning ishlayotganligini aniqlash, xato holatlari to'g'risida xabar berish, marshrutlash ma'lumotlarini almashish, ulanishlarni o'rnatish va ma'lumotlarni uzatish kabi funksiyalarini bajarish uchun ishlatiladi.

Ma'lumot uzatish har doim ham oson amalga oshirilmaydi. Masalan, tugun yoki tarmoq ishlamay qolganligi sababli manzilga yetib borolmaslik kabi xatolarga olib keladigan sharoitlar yuzaga kelishi mumkin. Bu kabi holatlarda, marshrutizator va marshrutlash protokollari xatoning holati to'g'risida xabar berishadi, garchi ular xatoni tuzatishga urinmasa ham xatolarni tuzatish tarmoq arxitekturasi boshqa darajalaridagi protokollarga qoldiriladi. Internetda eng ko'p ishlatiladigan marshrutlash protokollaridan ikkitasi – a) marshrutlash ma'lumotlari protokoli va b) dastlab qisqa yo'lni ochishdir.

Marshrutlash ma'lumotlari protokoli. Marshrutlash ma'lumotlari protokolida (Routing information protocol - RIP) ma'lumotni bir tarmoqdan boshqasiga uzatish yo'lini tanlash, manba va qabul qiluvchi manzil o'rtasidagi oraliq tugunlar soniga asoslanadi. Eng kam tugunlardan o'tuvchi yo'l har doim tanlanadi. Ushbu masofaviy vektor algoritmini bajarish juda oson, ammo bugungi tarmoq sharoitida u eng yaxshi bo'lmasligi mumkin, chunki u boshqa muhim omillarni, masalan, tarmoqning kengligi, ma'lumot ustuvorligi yoki tarmoq turini hisobga olmaydi. Ya'ni, u tezroq yoki ishonchli yo'llarni tanlashga to'sqinlik qilishi mumkin, chunki bu yo'llarda ko'proq tugunlar mavjud.

Marshrutlash ma'lumotlari protokolining yana bir cheklanishi marshrutlash jadvallariga bog'liq. O'zgarishlar bo'ladimi yoki yo'qmi, jadval to'liq yangilanadi va har 30 sekundda qayta chiqariladi; bu tarmoq ichidagi trafikni oshiradi va xabarlarini etkazib berishga salbiy ta'sir qiladi. Bundan tashqari, jadvallar bitta marshrutizatoridan boshqasiga tarqaladi. Shunday qilib, 15 ta tugun bilan ishlaydigan ichki tarmoq holatida, o'zgarish boshqa tomonga ma'lum bo'lishi uchun 7 daqiqadan ko'proq vaqt ketadi. Barcha marshrutizatorlar ichki tarmoq haqida bir xil ma'lumotga ega bo'lmagani uchun, biron bir xatolik barcha xabarlar trafigi uchun beqaror muhitni yaratishi mumkin.

Dastlab qisqa yo'lni ochish (Open Shortest Path First - OSPF), uzatish yo'lini tanlash tarmoqning holati aniqlanganidan keyingina amalga oshiriladi, shunda oraliq tugun ishlamay qolsa, uning xizmati qayta tiklanmaguncha darhol foydalanishdan chiqariladi. Yo'nalishni yangilash to'g'risidagi xabarlar faqat marshrutlash muhitida o'zgarishlar yuz berganda yuboriladi, shu bilan ichki tarmoq ichidagi xabarlar soni kamayadi va marshrutlash jadvalini to'liq yubormasdan xabarlar hajmi kamayadi. Shu bilan birga, Dastlab qisqa yo'lni ochish Marshrutlash ma'lumotlari protokoliga qaraganda ko'proq ma'lumotni kuzatib borishi sababli xotira sarfi oshadi. Bundan tashqari, o'tkazish qobiliyatini tejash eng qisqa yo'lni hisoblash uchun talab qilinadigan yuqori protsessor ishlatilishi bilan qoplanadi. Bunda bosqichma-bosqich davom ettirish va yo'l uzunligini orttirish orqali berilgan manbadan boshqa yo'nalishlarga eng qisqa yo'llarni topish mumkin.

Marshrutizator topologik ma'lumotlar bazasini yaratib, ichki tarmoqdagi har bir manzilga yetib borish uchun barcha turli yo'llarni hisoblab chiqadi. Ushbu ma'lumotlar tuzilishi Dastlab qisqa yo'lni ochish usuli tomonidan qo'llab-quvvatlanadi va ixtiyoriy xatolik tufayli to'xtalishlar yuzaga kelganida yangilanadi. Shuning uchun, marshrutizator shunchaki yo'lning mavjudligini aniqlash uchun o'z topologik ma'lumotlar bazasini tekshiradi va muvaffaqiyatsiz ulanishni atrofida eng qisqa yo'l daraxtini yaratadi.

### ***Ulanish modellari***

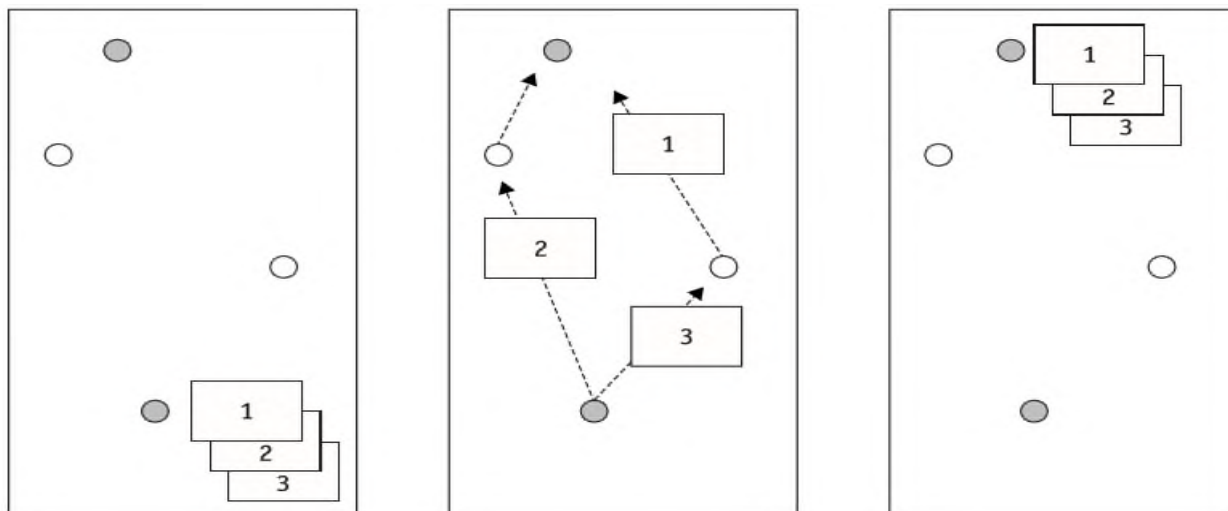
Aloqa tarmog‘i uzatilayotgan ma‘lumotlarning mazmuni bilan emas, balki ma‘lumotni bir nuqtadan boshqasiga o‘tkazish bilan bog‘liq. Tarmoqdagi har bir tugunni boshqa barcha tugunlarga ulash taqiqlanganligi sababli, tugunlar aloqa xarajatlarini minimallashtirish va barcha ulangan qurilmalar o‘rtasida to‘liq ulanishni ta‘minlash uchun mo‘ljallangan aloqa tarmog‘iga ulangan. Bir vaqtning o‘zida tarmoqqa kirish ma‘lumotlari bir nuqtadan tugunga o‘tish orqali, kanallar kommutatsiyasi yoki paketlar kommutatsiyasi orqali yo‘naltiriladi.

Kanallar kommutatsiyasi - bu ikkita kompyuter o‘rtasida ajratilgan aloqa yo‘li o‘rnatiladigan aloqa modelidir. Yo‘l - bog‘langan ulanishlar ketma-ketligi va ikkala nuqta orasidagi aloqa ulardan biri uzilib qolguncha mavjud. Ma‘lumot uzatish boshlanishidan oldin ulanish yo‘lini sozlash kerak. Shuning uchun, agar butun yo‘l mavjud bo‘lmasa, xabarlar uzatilishi mumkin emas, chunki kanal to‘liq bo‘lmasligi mumkin. Telefon tizimi kanallar kommutatsiya tarmog‘iga eng yaxshi misoldir. Ishlash jihatidan, ulanishni o‘rnatish paytida signal uzatilishidan oldin kechikish mavjud. Shu bilan birga, kanal to‘liq bo‘lishiga olib kelgandan so‘ng, tarmoq foydalanuvchilar uchun shaffof bo‘ladi va ma‘lumotlar belgilangan tezlikda, oraliq tugunlarda ahamiyatsiz kechikishlar bilan uzatiladi.

Paketlar kommutatsiyasi - bu asosan saqlash va o‘naltirishga asoslangan usul bo‘lib, unda xabar paketlar deb ataladigan bir xil o‘lchovli birliklarga bo‘linadi va ular tarmoq orqali o‘z manzillariga yuboriladi va asl formatga qayta yig‘iladi. 7.32-rasm. Paketlar kommutatsiyasi uzoq masofalarga ma‘lumotlarni uzatish uchun samarali texnologiya bo‘lib, kanallar kommutatsiyasiga qaraganda ko‘proq moslashuvchanlikni ta‘minlaydi, chunki u har xil tezliklarda ma‘lumotlarni qabul qiladigan yoki uzatadigan qurilmalar o‘rtasida ma‘lumot uzatishga imkon beradi. Biroq, xabarlar paketlarga bo‘linganidan so‘ng, paketlarning barchasi belgilangan manzilga bir xil yo‘lda yurishlari yoki ular fizik ketma-ketlikda kelishlariga kafolat yo‘q.

Bunga qo‘shimcha ravishda, bitta xabardagi paketlar boshqa xabarlarining paketlari o‘z manzillariga qarab ketayotganlarida ular bilan o‘zaro aloqada bo‘lishi mumkin. Shuning uchun, paketni uzatilishidan oldin har bir paketga paket haqida tegishli ma‘lumotni

o‘z ichiga olgan sarlavha qo‘yiladi. Paketsarlavhasidajoylashganma’lumotlartarmoqotomonidanishlatiladiganmarshrutlashusuligako‘rafarqqiladi.



7.32- rasm. Paketlarkommutatsiyasigamisol

G‘oyapaketlarniyetkazibberishtizimiorqali 30 tama’lumotnomalarseriyasiniyuborishbilano‘xshashdir. Oltita quti har birida beshta jilddan iborat bo‘lib, har bir qutiga uning tartib raqami (masalan, 6-sonli 2-quti), shuningdek, belgilangan manzil ko‘rsatilgan. Yetkazib berishda tashuvchilarda bo‘sh joy paydo bo‘lishi bilan, har bir quti markaziy kommutatsiya markaziga yuboriladi, u bo‘sh joy paydo bo‘lgunga qadar, uni o‘z manziliga yaqinroq joylashgan keyingi kommutatsiya markaziga yuborish uchun saqlanadi. Oxir oqibat, barcha oltita qutilar kelganida, ular asl tartibida joylashtiriladi, 30 jild ochiladi va asl tartib tiklanadi.

7.7- jadvalda ko‘rsatilgandek, paketlar kommutatsiyasi kanallar kommutat-siyadan tubdan farq qiladi. Shuningdek, markaziy kommutatsiya nuqtasi tomonidan butun xabar qabul qilinadigan va quyidagi ikkita hodisadan bittasi sodir bo‘lganda, manzilga yo‘naltirilgan saqlash va yo‘naltirish texnikasi qo‘llaniladi: barcha kanallar birdaniga butun xabarni yuborish imkoniyatiga ega bo‘lganida yoki qabul qiluvchi nuqta o‘zining saqlangan xabarlarini talab qilib so‘rov yuborganida.

-7.7- jadval

Kanallar kommutatsiyasi	Paketlar kommutatsiyasi
-------------------------	-------------------------



Real vaqtda uzatadi	Partiyalarda uzatadi
Kichik hajmli tarmoqlarda maqulroq	Katta hajmli tarmoqlarda ma'qulroq
Liniya samaradorligi past	Liniya samaradorligi yuqori
Bitta uzatishga mo'ljallangan	Ko'plab uzatishlar o'rtasida taqsimlanadi
Tovushli aloqada ma'qulroq	Tovushli aloqa uchun yaxshi emas
Oson yuklanganlik ortadi	Paketlar orasida turli ustuvorliklarni qo'llaydi

Bu, shuningdek, kanallar kommutatsiyasi tarmoqlaridan farqli o'laroq, ular haddan tashqari yuklanganda, u kamayguncha yangi ulanishlarni qabul qilishni rad etadi. Katta bayramlar paytida shaharlararo telefon qo'ng'iroqlarini amalga oshirayotganda, siz biron bir band signalni oldingizmi? Ushbu muammo kanallar kommutatsiya tarmog'ining haddan tashqari yuklanishi bilan bog'liqdir.

Paketlar kommutatsiyasi foydalanuvchilarga o'z xabarlariga ustuvorliklarni ajratishga imkon beradi, shunda marshrutizator uzatish uchun navbatga qo'yilgan bir nechta paketlar orasidan birinchi navbatda ustuvor paketlarni yuborishi mumkin. Bundan tashqari, paketlar kommutatsiyasiga asoslangan tarmoqlar boshqa turlarga qaraganda ancha ishonchli, chunki ko'pchilik tugunlar bir nechta aloqa orqali ulanadi, shuning uchun bitta kanalda to'xtalish yuz bersa, tugunlar o'rtasida mutlaqo boshqacha yo'l o'rnatilishi mumkin.

### ***Ziddiyatlarni hal qilish***

Tarmoq umumiy uzatish qobiliyatini birgalikda foydalanuvchi qurilmalardan iborat bo'lganligi sababli, ushbu umumiy resursga teng va adolatli kirishni ta'minlash uchun muhitdan foydalanishni boshqarishning ba'zi usullari zarur. Birinchidan, biz foydalanishni boshqarishning ba'zi bir o'rta texnik usullarini tavsiflaymiz: round robin, rezervlash va ziddiyatlar.

Kirishni boshqarish usullari. Round Robin tarmoqdagi har bir tugunga aloqa vositasidan foydalanishga imkon beradi. Agar tugunni yuborish uchun ma'lumotlar bo'lsa, uzatishni tugatish uchun ma'lum vaqt beriladi, so'ngra imkoniyat keyingi tugunga o'tadi. Agar tugunni yuborish uchun ma'lumot bo'lmasa yoki u vaqt o'tmasdan uzatishni tugatsa, keyingi tugun o'z navbatini boshlaydi. Round robin - bu uzoq vaqt davomida ko'plab tugunlar mavjud bo'lgan samarali uzatish

usulidir. Ammo, uzoq vaqt davomida bir nechta tugunlar uzatilganda, tugundan tugunga o'tishda ortiqcha sarflar bo'lishi mumkin. Bu esa boshqa usullarni translyatsiyaning qisqa va uzluksiz bo'lishiga yoki fayllarni uzatishning katta sessiyalaridagi kabi uzoq va uzluksiz bo'lishiga bog'liq ravishda qo'llash afzalliklariga asos bo'ladi.

Zahiralash texnikasi uzoq va doimiy trafik uchun juda mos keladi. Muhitning kirish vaqti slotlarga bo'linadi hamda tugun kelajakda foydalanish uchun kelgusi vaqt slotlarini rezervlashi mumkin. Ushbu usul host kompyuterga bitta kirish/chiqish porti orqali ulangan bir nechta terminallar bilan konfiguratsiya uchun yaxshi bo'lishi mumkin.

Ziddiyatlar usuli qisqa va uzilishli trafik uchun yaxshiroqdir. Kimning navbati uzatilishini aniqlashga harakat qilinmaydi, shuning uchun tugunlar muhitga kirish uchun raqobatlashadi. Shuning uchun, u yengil trafikdan mo'tadil trafikka qadar yaxshi ishlaydi, ammo yuklanganlik oshganida xatolik va to'xtalishlarga moyil. Ushbu usulning asosiy afzalligi shundaki, uni amalga oshirish oson. Hozirda foydalanilayotgan kirish protokollari avval aytib o'tilgan usulga asoslangan bo'lib, bu yerda lokal tarmoq muhitidagi roli xususida muhokama qilindi.

## **7.6. Domen tizimi va internet xizmatlarini tashkil etish**

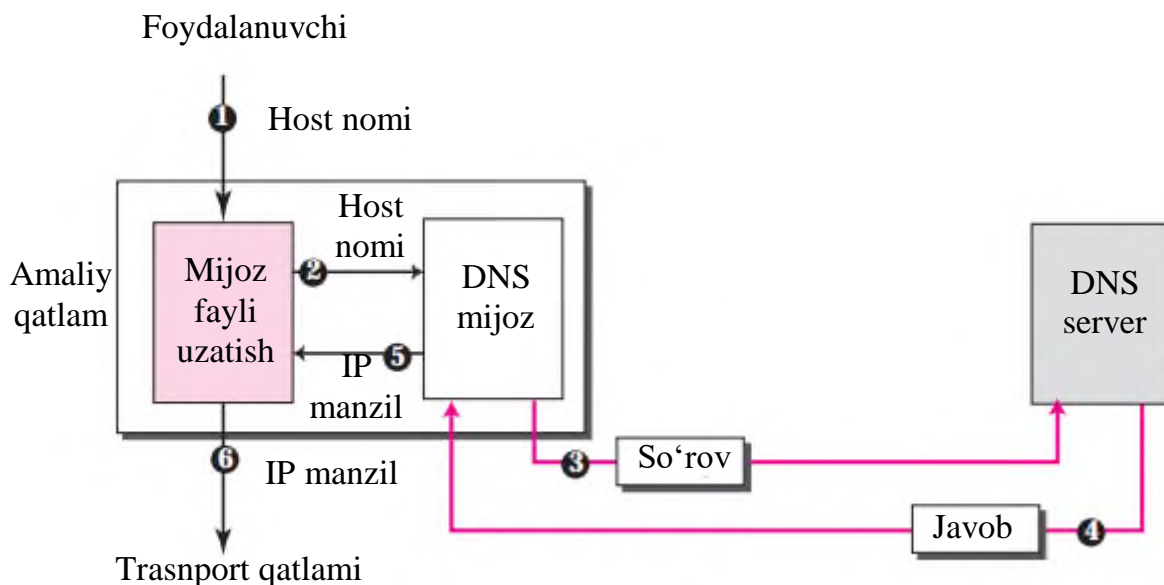
### ***Domain Name System (DNS)***

Korxonani identifikatsiya qilish uchun TCP/IP protokollari hostning Internetga ulanishini aniqlaydigan IP manzilidan foydalanadi. Biroq, odamlar raqamli manzillar o'rniga ismlardan foydalanishni afzal ko'rishadi. Shuning uchun, bizga manzilni manzilga yoki nomga manzilni xaritada ko'rsatadigan tizim kerak.

Internet kichkina bo'lganda, host-fayl yordamida xarita tuzildi. Host faylida faqat ikkita ustun mavjud edi: ism va manzil. Har bir host o'z diskida host faylini saqlashi va vaqti-vaqti bilan uni asosiy host fayldan yangilab turishi mumkin. Dastur yoki foydalanuvchi nomni manzilga xaritaga tushirishni istaganida, host host faylni ko'rib chiqdi va xaritani topdi. Ammo, bugungi kunda har bir manzilni nom bilan va aksincha bog'lash uchun bitta host-faylga ega bo'lish mumkin emas. Host fayli har bir hostda saqlash uchun juda katta bo'lishi kerak

edi. Bunga qo‘shimcha ravishda, har safar biron bir o‘zgarish bo‘lganda, barcha host fayllarini yangilab bo‘lmaydi. Bitta yechim host faylni bitta kompyuterda saqlash va xaritalashga muhtoj bo‘lgan har bir kompyuterga ushbu markazlashtirilgan ma’lumotdan foydalanishga imkon berishdir. Ammo biz bilamizki, bu Internetda katta hajmdagi trafikni keltirib chiqaradi.

Bugungi kunda ishlatiladigan yana bir yechim bu katta hajmdagi ma’lumotni kichik qismlarga bo‘lish va har bir qismini boshqa kompyuterda saqlashdir. Ushbu usulda xaritalashni talab qiladigan host kerakli ma’lumotlarni ushlab turgan eng yaqin kompyuter bilan bog‘lanishi mumkin. Ushbu usul Domen Nomlari Tizimi (DNS) tomonidan qo‘llaniladi. 7.33- rasmda TCP/IP qanday qilib DNS mijozdan va DNS serverni nomni manzilga xaritada ko‘rsatish uchun qanday foydalanishi keltirilgan, teskari xaritalash ham huddi shu ketma-ketlikda amalga oshiriladi.



7.33- rasm. DNS maqsadi

7.33- rasmda foydalanuvchi masofaviy hostda ishlaydigan fayl uzatish ser-veriga kirish uchun fayl uzatish mijozidan foydalanishni xohlaydi. Foydalanuvchi faqat fayl uzatish serverining nomini biladi, masalan forouzan.com. Biroq, TCP/IP to‘plamiga ulanish uchun fayl uzatish serverining IP manzili kerak. Quyidagi oltita bosqichda host nomini IP manzilga xaritalash ko‘rsatilgan.

1. Foydalanuvchi host nomini fayl uzatish mijoziga jo‘natadi.
2. Fayl uzatish mijozu DNS mijozga host nomini yuboradi.

3. Har bir kompyuter yuklangandan so‘ng bitta DNS serverning manzilini biladi. DNS mijoz DNS serverga DNS serverning ma‘lum IP manzilidan foydalanib, fayl jo‘natish serverining nomini beradigan so‘rov bilan xabar yuboradi.

4. DNS server kerakli fayl jo‘natish serverining IP manzili bilan javob beradi.

5. DNS mijoz IP manzilni fayl jo‘natish serveriga yuboradi.

6. Endi fayllarni jo‘natish mijoz fayllarni jo‘natish serveriga kirish uchun qabul qilingan IP manzilidan foydalanadi.

E‘tibor bering, Internetga kirishning maqsadi fayllarni jo‘natish mijoz va server o‘rtasida aloqa o‘rnatishdir, ammo bu sodir bo‘lishidan oldin DNS mijoz va DNS server o‘rtasida boshqa ulanish kerak. Boshqacha aytganda, biz ikkita ulanishga muhtojmiz; masalan, birinchisi, nomni IP manzilga solishtirish uchun; ikkinchisi fayllarni jo‘natish uchun.

### ***Nom maydoni***

Mashinalarga tayinlangan nomlar aniq bo‘lishi uchun, nomlar va IP manzillari orasidagi bog‘lanishni to‘liq nazorat qilib, diqqat bilan tanlanishi kerak. Boshqacha qilib aytganda, nomlar noyob bo‘lishi kerak, chunki manzillar noyobdir. Har bir manzilni noyob nom bilan xaritada aks ettiradigan nom maydoni ikki shaklda tashkil etilishi mumkin: tekis (qat‘iy) yoki ierarxik.

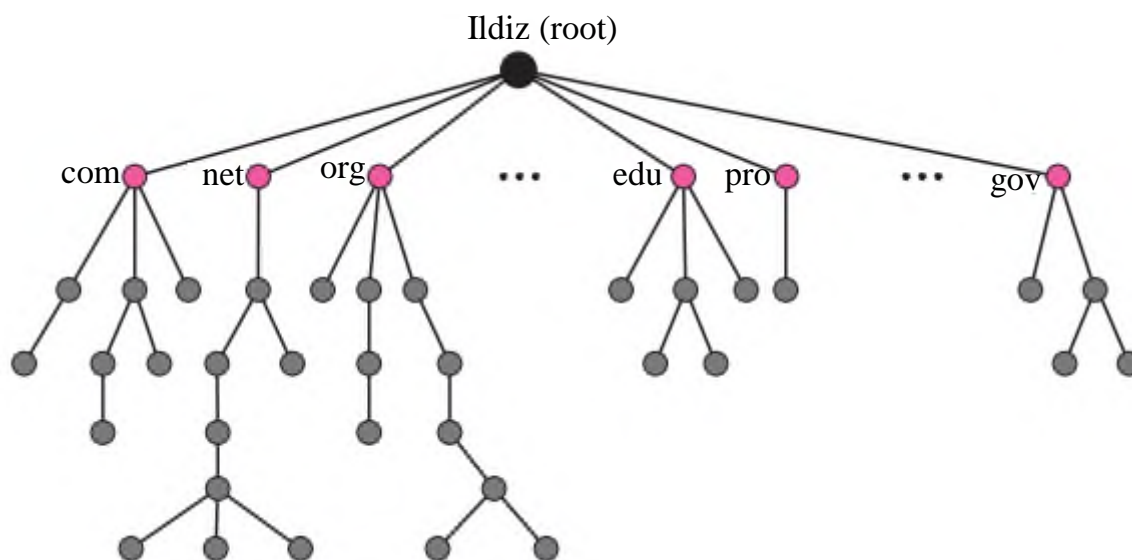
*Tekis (qat‘iy) nom maydoni.* Tekis maydon nomi bu manzilga beriladigan nomdir. Bu maydondagi nom strukturalanmagan belgilar ketma-ketligi sifatida qaraladi. Nomlar umumiy qismga ega bo‘lishi yoki bo‘lmasligi mumkin, lekin bu hech qanday ma‘noga ega emas. Tekis (qat‘iy) nom maydonining asosiy kamchiligi shundaki, uni Internet kabi katta tizimda ishlatish mumkin emas, chunki noaniqlik va takrorlanishning oldini olish uchun markazlashtirilgan boshqaruv kerak.

*Ierarxik nom maydoni.* Ierarxik nomlar maydonida har bir nom bir necha qismdan iborat. Birinchi qism tashkilotning xususiyatini, ikkinchi qismi tashkilot nomini, uchinchi qismi tashkilotdagi bo‘limlarni va boshqalarni belgilashi mumkin. Bunday holda, nom maydonlarini tayinlash va boshqarish vakolati markazlashtirilishi mumkin. Markaziy tashkilot tashkilot nomini va tashkilot xususiyatini belgilaydigan qismni tayinlashi mumkin. Qolgan nomning javobgarligi tashkilotning o‘ziga berilishi mumkin.

Tashkilot o‘z hostini yoki manbalarini aniqlash uchun nomga suffikslarni (yoki prefikslarni) qo‘shishi mumkin. Tashkilot rahbariyati host uchun tanlangan prefiks boshqa tashkilot tomonidan qabul qilinadi, deb xavotirlanmaslik kerak, chunki manzilning bir qismi bir xil bo‘lsa ham, butun manzil boshqacha. Masalan, ikkita kollej va kompaniya o‘z kompyuterlaridan birini challenger deb ataydi deylik. Birinchi kollejga fhda.edu nomi va ikkinchi kollejga berkeley.edu nomi berilgan hamda kompaniyaga smart.com nomi berilgan. Ushbu tashkilotlarning har biri allaqachon berilgan nomga challenger nomini qo‘shganda, natija uchta turli xildagi nomga ega bo‘ladi: challenger.fhda.edu, challenger.berkeley.edu, va challenger.smart.com. Nomlar markaziy tashkilot tomonidan tayinlanishini talab etmasdan noyobdir. Markaziy tashkilot nomning to‘liq qismini emas, balki faqat bir qismini boshqaradi.

### ***Domen nomi maydoni***

Nomlarning ierarxik maydoniga ega bo‘lish uchun, domen nomining maydoni yaratilgan. Ushbu dizayndagi nomlar tepada joylashgan ildiz (root) bilan teskari ko‘rinishdagi daraxt shaklida tuzilgan. Daraxt faqat 128 darajaga ega bo‘lishi mumkin: 0 darajasidan (ildiz, yoki root) 127 darajagacha (7.34- rasmga qarang).



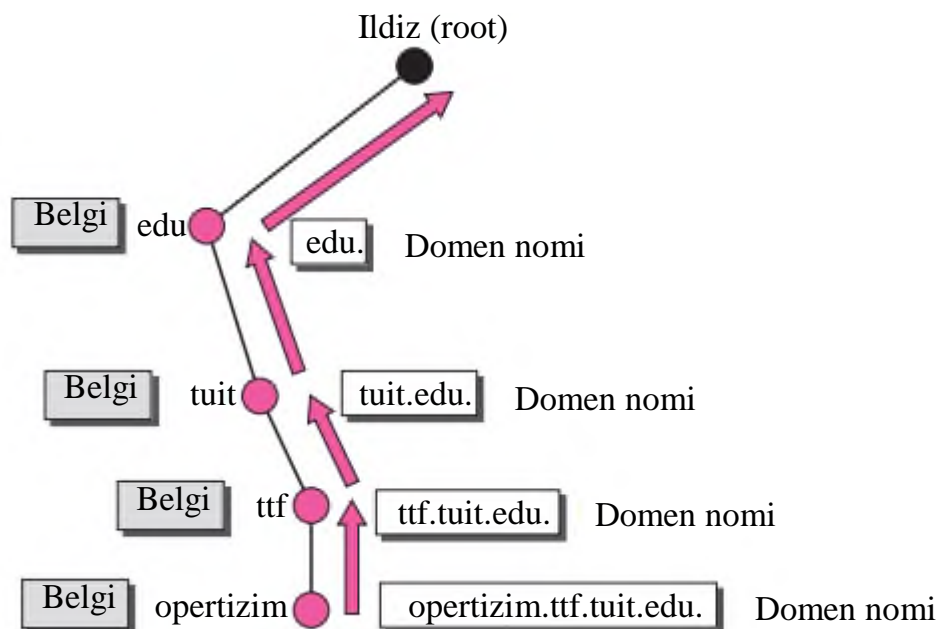
7.34- rasm. Domen nomi maydoni

### ***Belgi***

Daraxtning har bir tugunida belgi mavjud, u 63 belgidan iborat satr. Ildiz (root) belgisi - bu null string, ya'ni bo'sh satr. DNS, tugunning ostki qismlari (bitta tugunning ostki tugunlar) turli xil belgilarga ega bo'lishini talab qiladi, bu domen nomlarining noyobligini kafolatlaydi.

### ***Domen nomi***

Daraxtdagi har bir tugun domen nomiga ega. To'liq domen nomi - bu nuqta (.) bilan ajratilgan belgilar ketma-ketligi. Domen nomlari har doim tugundan ildizgacha (root) o'qiladi. Oxirgi belgi - bu ildizning belgisi (null). Bu shuni anglatadiki, to'liq domen nomi har doim nol belgisida tugaydi, ya'ni oxirgi belgi nuqta degan ma'noni anglatadi, chunki bo'sh qator hech narsa emas. 7.35- rasmda ba'zi domen nomlari keltirilgan.

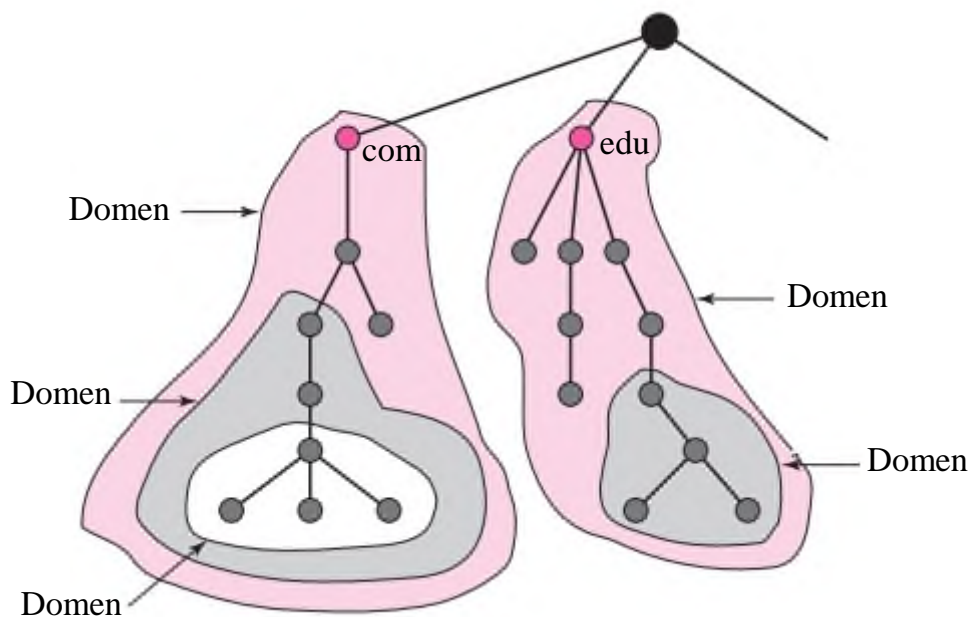


7.35- rasm. Domen nomi va belgilar

### ***Domen***

Domen bu domen nomi maydonining ost-daraxti. Domenning nomi ost-daraxtning yuqori qismida joylashgan tugunning nomi. 7.36- rasmda ba'zi domenlar ko'rsatilgan. Domenlar boshqa domenlar uchun foydalanilishi mumkin, odatda bu ost-domenlar deb ataladi.

Ildiz (root)



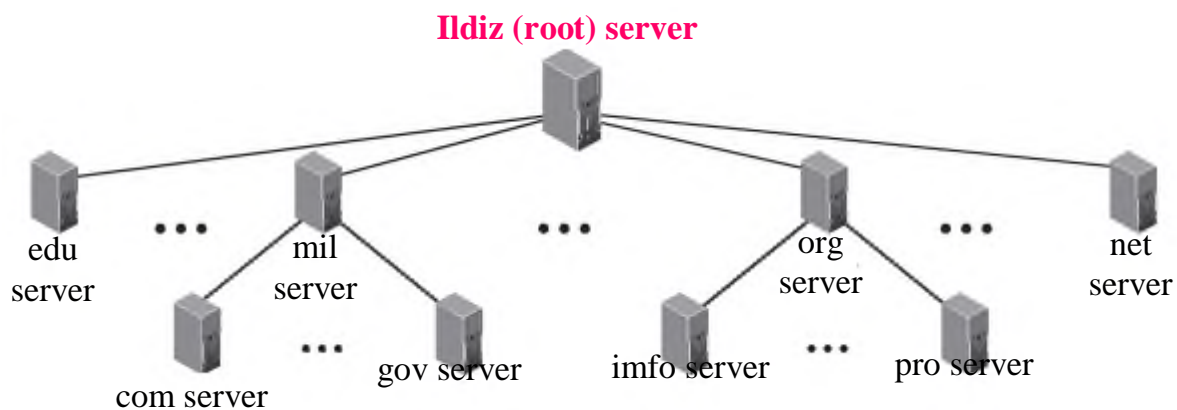
7.36- rasm. Domenlar

### ***Nom maydonlarini taqsimlash***

Domen nomlari maydonida joylashgan ma'lumotlar saqlanishi kerak. Biroq, juda ko'p ma'lumotga ega bo'lgan bitta kompyuter xotirasiga ega bo'lish juda samarasiz va ayni paytda ishonchli emas. Dunyoning barcha mamlakatlaridan kelgan so'rovlarga javoban tizimga katta yuklama tushishi hisobiga bu samarasiz hisoblanadi. Har qanday nosozlik ma'lumotlarga kirishga imkon bermasligi sababli bu ishonchsizdir.

### ***Nom serverlari ierarxiyasi***

Ushbu muammolarning yechimi ma'lumotlarni DNS serverlari deb nomlangan ko'plab kompyuterlar orasida tarqatishdir. Buning bir usuli - bu butun maydonni birinchi darajaga asoslangan ko'plab domenlarga bo'lish. Boshqacha qilib aytganda, biz ildizni (root) yolg'iz qoldiramiz va ko'p darajali domenlar yaratamiz (pastki darajalar), chunki birinchi darajali tugunlar mavjud. Shu tarzda yaratilgan domen juda katta bo'lishi mumkinligi sababli, DNS domenlarni kichikroq domenlarga (pastki domenlarga) bo'lish imkonini beradi. Har bir server katta yoki kichik domen uchun javobgar (vakolatli) bo'lishi mumkin. Boshqacha qilib aytganda, bizda nomlar ierarxiyasi bo'lgani kabi serverlar ierarxiyasi ham mavjud (7.37- rasmga qarang).



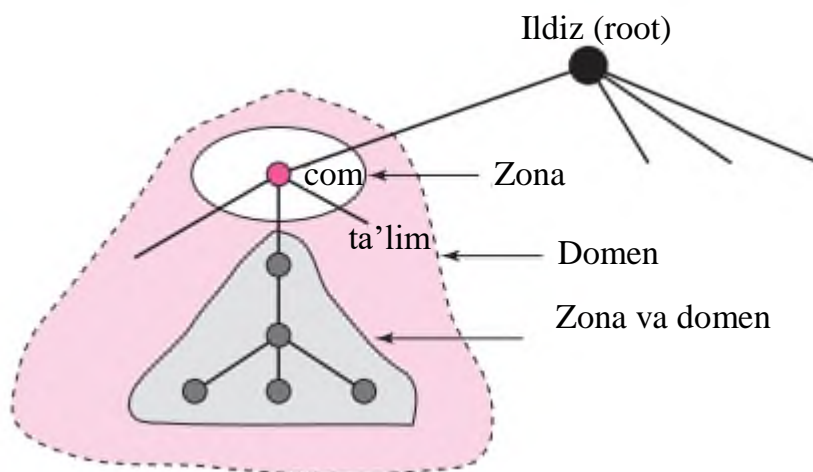
7.37- rasm. Server nomlari ierarxiyasi

### **Zona (Zone)**

To'liq domen nomlari ierarxiyasini bitta serverda saqlash imkoniyati bo'lmaganligi sababli, u ko'p serverlar orasida taqsimlanadi. Server javobgar bo'lgan yoki uning vakolatiga ega bo'lgan tushuncha zona deb nomlanadi. Biz zonani butun daraxtning qo'shni qismi sifatida belgilashimiz mumkin. Agar server domen uchun javobgarlikni o'z zimmasiga olsa va domenni kichik domenlarga ajratmasa, «domen» va «zona» aynan bir tushunchani anglatadi. Server zonali fayl deb nomlangan ma'lumotlar bazasini yaratadi va ushbu domen ostidagi har bir tugun uchun barcha ma'lumotlarni saqlaydi. Agar server o'z domenini quyi domenlarga ajratsa va o'z vakolatlarining bir qismini boshqa serverlarga topshirsa, "domen" va "zona" turli xil tushunchalarni anglatadi. Ost-domomenlardagi tugunlar to'g'risidagi ma'lumotlar pastki darajadagi serverlarda saqlanadi va yuqori darajadagi server ushbu pastki darajadagi serverlarga havolani saqlaydi. Albatta, yuqori darajadagi server o'zini javobgarlikdan butunlay ozod qilmaydi: u hanuzgacha zonaga ega, ammo batafsil ma'lumot quyi darajadagi serverlarda saqlanadi (7.38- rasmga qarang).

Server, shuningdek, o'z domenining bir qismini ajratishi va javobgarlikni o'z zimmasiga olishi mumkin, ammo baribir domenning bir qismini o'zi uchun saqlab qoladi. Bunday holda, uning zonasi vakolat berilmagan domenning qismi uchun batafsil ma'lumot va vakolat berilgan qismlarga havolalar mavjud.





7.38- rasm. Zonalar va domenlar

### ***Ildiz (root) server***

Ildiz server - bu butun daraxtdan tashkil topgan server. Ildiz server odatda domenlar to'g'risida hech qanday ma'lumot saqlamaydi, lekin o'z vakolatlarini boshqa serverlarga topshiradi va o'sha serverlarga murojaatlarni saqlaydi. Bir nechta ildiz serverlari mavjud bo'lib, ularning har biri butun domen nomi maydonini o'z ichiga oladi. Ildiz serverlari butun dunyoga tarqalgan.

### ***Asosiy va ikkilamchi serverlar***

DNS ikki xil serverni aniqlaydi: asosiy va ikkilamchi. Asosiy server bu vakolat berilgan zona to'g'risidagi faylni saqlaydigan server. Bu zona faylini yaratish, saqlash va yangilash uchun javobgardir. Zona faylini mahalliy diskda saqlaydi.

Ikkilamchi server - bu zona haqida to'liq ma'lumotni boshqa serverdan (asosiy yoki ikkilamchi) jo'natuvchi va faylni o'zining mahalliy diskida saqlaydigan server. Ikkilamchi server zona fayllarini yaratmaydi va yangilamaydi. Agar yangilanish kerak bo'lsa, uni yangilangan versiyasini ikkinchi darajaga yuboradigan asosiy server amalga oshirishi kerak.

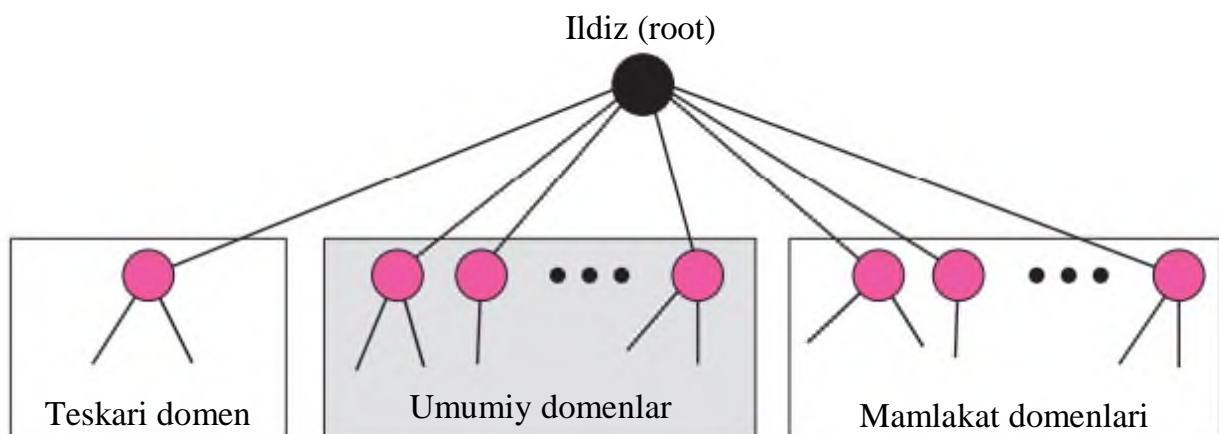
Asosiy va ikkilamchi serverlar o'zlari xizmat ko'rsatadigan zonalar uchun vakolatli hisoblanadi. G'oya ikkinchi darajali serverni past darajasiga qo'yish emas, balki bitta server ishlamay qolsa, boshqasi mijozlarga xizmat ko'rsatishni davom ettirish uchun ma'lumotlarning zaxirasini yaratishdir. Shuni ham unutmangki, server ma'lum bir zona uchun asosiy va boshqa zona uchun ikkinchi darajali server bo'lishi mumkin. Shuning uchun, asosiy yoki ikkilamchi server

deb aytganda, qaysi zonaga murojaat qilishimizdan ehtiyot bo‘lishimiz kerak.

Asosiy server barcha ma’lumotlarni diskdan yuklaydi; ikkinchi server barcha ma’lumotlarni asosiy serverdan yuklaydi. Ikkilamchi server asosiy serverdan ma’lumotlarni yuklab olganida, uni zonani **o‘tkazish (transfer)** deyiladi.

### ***Internetda DNS***

DNS - bu turli platformalarda ishlatilishi mumkin bo‘lgan protokol. Internetda domen nomlari maydoni (daraxt) uch xil qismga bo‘linadi: umumiy domenlar, mamlakat domenlari va teskari domen (7.39- rasmga qarang).

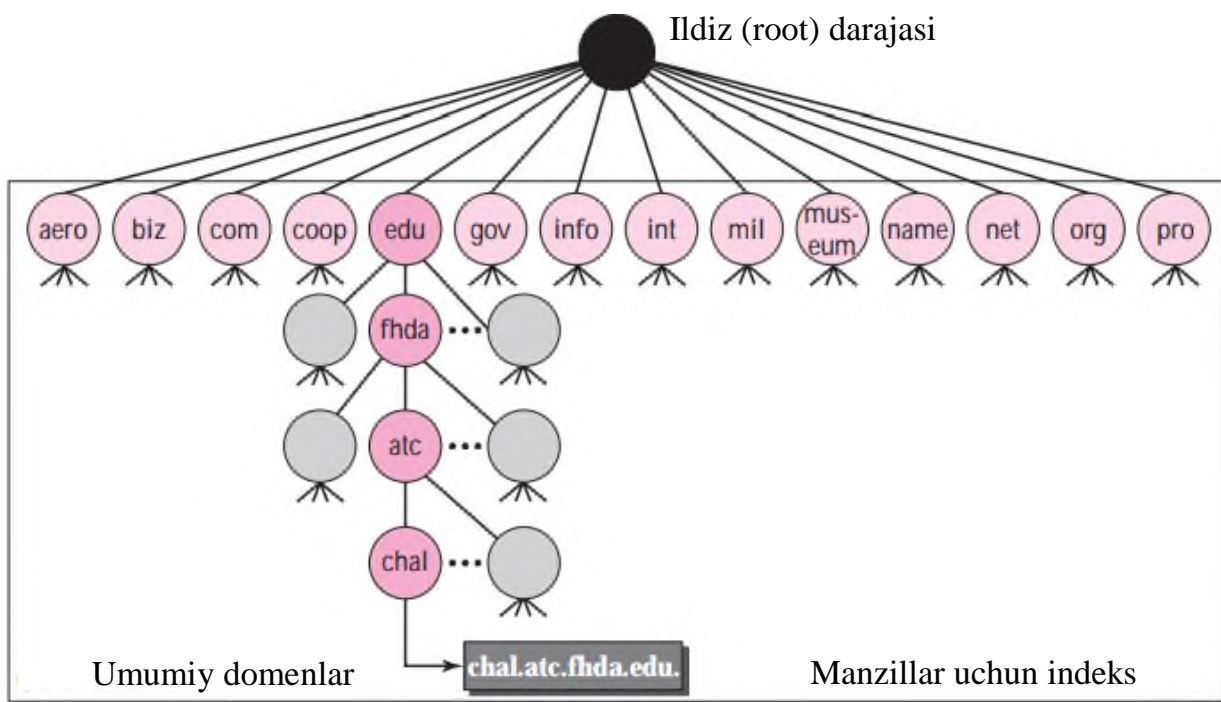


7.39- rasm. DNS ning Internetda qo‘llanilishi

### ***Umumiy domenlar***

Umumiy domenlar umumiy xarakterga ko‘ra ro‘yxatdan o‘tgan hostlarni aniqlaydi. Daraxt ichidagi har bir tugun, domen nomi maydonining ma’lumotlar bazasi uchun indeks bo‘lgan domenni belgilaydi (7.40- rasmga qarang).

Daraxtga nazar tashlasak, umumiy domenlar bo‘limidagi asosiy daraja 14 ta belgilarga imkon beradi. Ushbu belgilar 7.8- jadvalda keltirilgan tashkilot turlarini tavsiflaydi.



7.40- rasm. Umumiy domenlar

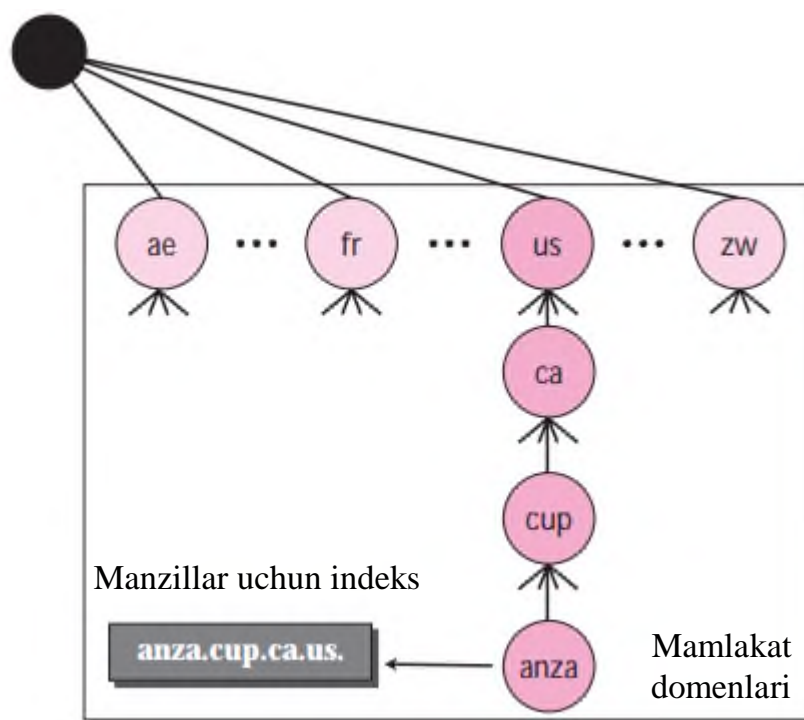
7.8- jadval

Belgilanishi	Tavsifi
aero	aviakompaniyalar va aerokosmik kompaniyalar
biz	korxonalar yoki firmalar ("com" ga o'xshash)
com	tijorat tashkilotlari
coop	kooperativ biznes tashkilotlari
edu	o'quv muassasalari
gov	davlat idoralari
info	axborot xizmatlarini yetkazib beruvchilar
int	xalqaro tashkilotlar
mil	harbiy guruhlar
museum	muzeylar va boshqa notijorat tashkilotlar
name	shaxsiy ismlar
net	tarmoqni qo'llab-quvvatlash markazlari
org	notijorat tashkilotlari
pro	professional individual tashkilotlar

### ***Mamlakat domenlari***

Mamlakat domenlari qismida ikki belgili mamlakat nomlaridan kelib chiqib qisqartirishlar qo'llaniladi (masalan, O'zbekiston uchun uz). 7.41- rasmda mamlakatning domenlari ko'rsatilgan.

Ildiz (root)  
darajasi



7.41- rasm. Mamlakat domenlari

Ikkinchi teglar tashkiliy, yoki ular aniqroq va milliy belgilar bo‘lishi mumkin. Masalan, Amerika Qo‘shma Shtatlari shtatdagi qisqartirishlarni ‘us’ning ost-qismi sifatida ishlatadi (masalan, ca.us.).

Anza.cup.ca.us manzilini AQShning Kaliforniya shtatidagi Cupertino shahridagi De Anza kollejiga tegishli ekanligini ko‘rish mumkin.

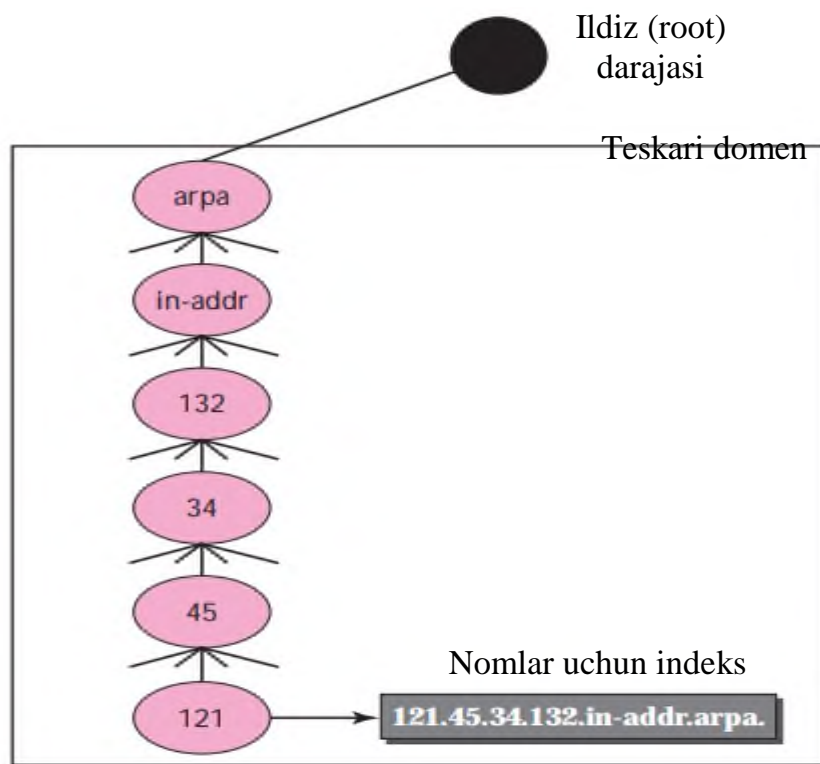
### ***Teskari domenlar***

Teskari domen nomni manzilga xaritalashda ishlatiladi. Masalan, server mijozdan biron bir vazifani bajarish uchun so‘rov olganida ro‘y berishi mumkin. Serverda vakolatli mijozlar ro‘yxati bo‘lgan fayl mavjud bo‘lsa-da, faqat mijozning IP manzili (qabul qilingan IP paketidan olingan) keltirilgan. Server mijozning vakolatli ro‘yxatida ekanligini yoki yo‘qligini aniqlash uchun manzilni xaritaga olish uchun DNS serveriga so‘rov yuborishni so‘raydi.

Ushbu turdagi so‘rov teskari yoki ko‘rsatgichli (pointer - PTR) so‘rov deb ataladi. Ko‘rsatkich so‘rovini bajarish uchun, teskari domen domen nomi maydoniga arpa deb nomlangan birinchi darajali tugun bilan qo‘shadi (tarixiy sabablarga ko‘ra). Ikkinchi daraja,

shuningdek, in-addr deb nomlangan bitta tugun (teskari manzil uchun) qo‘shadi. Qolgan domen IP manzillarini belgilaydi.

Teskari domenni boshqaradigan serverlar ham ierarxik bo‘ladi. Bu shuni anglatadiki, manzilning aniq qismi tarmoqosti id qismiga qaraganda yuqoriroq, va tarmoqosti id qismi host id qismiga nisbatan yuqoriroq bo‘lishi kerak. Shu tarzda, butun saytga xizmat ko‘rsatadigan server har bir tarmoqostiga xizmat ko‘rsatadigan serverlarga qaraganda yuqori darajaga ega bo‘ladi. Ushbu konfiguratsiya umumiy yoki mamlakat domeni bilan taqqoslaganda domenni teskari ko‘rinishga olib keladi. Domen belgilarini pastdan yuqoriga qarab o‘qish qoidalariga rioya qilish uchun 132.34.45.121 kabi IP manzil (132.34 net sinfi bilan B sinf manzili) 121.45.34.132.in-addr.arpa sifatida o‘qiladi. Teskari domen konfiguratsiyasini tasvirlash uchun 7.42- rasmga qarang.



7.42- rasm. Teskari domen

### Nazorat savollari

1. Mijoz-server arxitekturasini tushuntiring.
2. Mijoz-server tuzilishi va asosiy tushunchalari.

3. Soket nima?
4. Soket ma'lumotlar tuzilishini tushuntiring.
5. Tarmoq operatsion tizimlari va ularning asosiy tarmoq xizmatlari.
6. E-mail arxitekturasini tushuntiring.
7. SMTP protokoli vazifasi va imkoniyatlari.
8. POP va IMAP protokollarining vazifalarini tushuntiring.
9. Tarmoq kompyuterlarini tashkil etish va boshqarish mexanizmlari.
10. Peer-to-peer arxitekturasini tushuntiring.
11. DHCP protokolini tushuntiring.
12. Tarmoq fayllarini tashkil etish va boshqarish mexanizmlari.

## VIII BOB. OPERATSION TIZIMLARDA XAVFSIZLIK MASALALARI

### 8.1. Operatsion tizimning asosiy xavfsizlik masalalari

Kompyuter xavfsizligi mavzusi juda keng bo'lib, fizik va ma'muriy nazorat, shuningdek, avtomatik boshqaruv turlari masalalarini qamrab oladi. Kompyuter tizimlarida saqlanadigan ma'lumotlar hajmi oshgani sayin, axborotni himoya qilish zarurati ham oshadi. Ma'lumotni ruxsatsiz kirishdan himoya qilish operatsion tizimlarning asosiy vazifalaridan biridir. Afsuski, ushbu maqsadga erishish tobora qiyinlashib bormoqda. Ma'lumotlarga ruhsatsiz kirish va foydalanishdan himoya qilish barcha operatsion tizimlarning asosiy vazifasiga aylanadi. Ushbu bobda operatsion tizimlar uchun kompyuter xavfsizligi masalalari ko'rib chiqiladi.

Avtonom kompyuter tizimini turli xil yo'llar bilan tashqi hujumlardan samarali himoya qilish mumkin, masalan, parol qo'yish, qattiq diskni yechish va uni xavfsiz yerga (seyfga) joylash. Tarmoqda ishlaydigan kompyuterlarda xavfsizlikni ta'minlash ancha murakkab vazifadir.

Axborot tizimlarining xavfsizligi sohasida odatda ikkita muammo mavjud: kompyuter xavfsizligi va tarmoq xavfsizligi.

❖ **Kompyuterni xavfsizligi:** avtonom tizim sifatida qaraladigan kompyuterda qayta ishlanadigan va saqlanadigan ma'lumotlar himoyasi bilan bog'liq barcha muammolarni o'z ichiga oladi.

Ushbu muammolar operatsion tizimlar va amaliy dasturlar, masalan, ma'lumotlar bazasini boshqarish tizimi yoki korxonalar resurslarini boshqarish tizimi, shuningdek, o'rnatilgan kompyuter qurilmalari yordamida hal qilinadi.

❖ **Tarmoq xavfsizligi:** tarmoqdagi qurilmalarning o'zaro ta'siri bilan bog'liq barcha muammolarni anglatadi:

- Aloqa liniyalari orqali ma'lumotlarning uzatilish davrida himoyalash;
- Tarmoqqa masofadan ruhsat etilmagan murojaatlardan himoyalash.

Ba'zida kompyuter va tarmoq xavfsizligi muammolarini bir-biridan ajratish qiyin bo'lsa ham, ular bir-biri bilan chambarchas bog'liq, ammo tarmoq xavfsizligi o'ziga xos xususiyatlarga ega.

Tarmoq xavfsizligining xususiyatlari:

❖ Tarmoqda ishlayotgan vaqtingizda kompyuteringizdan boshqa foydalanuvchiga mantiqiy kirishkasbiy, lavozimli holat bo'ladi. Bunday holatda axborot xavfsizligini ta'minlash uchun har bir foydalanuvchini nazoratga olish zarur, ya'ni ularni tarmoqda aniq belgilangan funksiyalari mavjud bo'lishi va resurslarga murojaat qilish darajalari belgilab berilgan bo'lishi zarur.

❖ Tarmoq tabiati bo'yicha yana bir turdagi xavf mavjud, ya'ni tarmoq bo'ylab uzatilayotgan axborotni tutib olish va tahlil qilish, shuningdek «yolg'on» trafiklarni yaratishdir. Tarmoq xavfsizligini ta'minlashga qaratilgan vositalarni kattagina qismi aynan shu turdagi buzilishlarni oldini olishga qaratilgan.

### ***Operatsion tizimning xavfsizlikda tutgan o'rni***

Operatsion tizim kompyuter tizimining xavfsizligida muhim rol o'ynaydi, chunki u tizimning barcha qismlariga kirish huquqiga ega. Operatsion tizim darajasidagi har qanday zaifliklar butun tizimni hujum uchun ochadi. Operatsion tizim qanchalik murakkab va kuchli bo'lsa, uning hujumlarga nisbatan zaiflik ehtimoli shunchalik yuqori bo'ladi. Natijada, tizim ma'murlari o'zlarining operatsion tizimlarini hujumlardan va mumkin bo'lgan nosozliklardan himoya qilishning barcha vositalarini ko'rib chiqishlari kerak.

### ***Tizimning yashovchanligi***

Tizimning yashovchanligi va unga erishish strategiyalarining to'rtta asosiy xususiyatlari mavjud:

- ❖ Hujumlarga qarshilik;
- ❖ Hujumlar va zararni aniqlash;
- ❖ Hujumdan keyin muhim va to'liq xizmatlarni tiklash;
- ❖ Kelajakdagi hujumlarning samaradorligini pasaytirish uchun moslashish va evolyutsiya.

8.1- jadvalda tizimning yashovchanligi va unga erishish strategiyasining asosiy xususiyatlari keltirilgan.



## 8.1- jadval

Asosiy xususiyati	Ta'rif	Strategiya misollari
Hujumlarga qarshilik	Hujumlarni qaytarish strategiyalari	Haqiqiylikni tekshirish Kirishni boshqarish Shifrlash Xabarlarini filtrlash Tizimni samaradorligini oshirish Funksional ximoyalanganlik
Hujumlar va zararni aniqlash	Hujumlarni aniqlash va zararni baholash strategiyalari	Buzilishlarni aniqlash Butunlikni tekshirish
Hujumdan keyin muhim va to'liq xizmatlarni tiklash	Zararni cheklash, buzilgan ma'lumotlar yoki funksiyalarni tiklash, xizmat muddati davomida asosiy xizmatlarni saqlash yoki tiklash, to'liq xizmatlarni tiklash strategiyalari.	Ortiqcha komponentlar Ma'lumotlarni qaytarish Tizimni zaxiralash va tiklash Favqulodda vaziyatlarni rejalashtirish
Kelajakdagi hujumlarning samaradorligini pasaytirish uchun moslashish va evolyutsiya	Tuzilishdan olingan bilimlarga asoslangan tizimning yashovchanligini oshirish strategiyalari	Buzilishni aniqlash shablони

**Himoyalash darajasi**

Agar tizim hujumga uchragan bo'lsa, tizimdagi har bir faylning yaxlitligiga va ushbu fayllardagi ma'lumotlarga endi ishonib bo'lmaydi. Har bir kompyuter konfiguratsiyasi uchun tizim ma'muri buzilish xavfini baholashi kerak, bu esa o'z navbatida 8.2- jadvalda ko'rsatilgan tizimga ulanish darajasiga bog'liq.

**Xavfsizlik muammosi**

**Mahfiylikni buzish.** Ushbu turdagi buzishlar ma'lumotlarni ruxsatsiz o'qishni (yoki ma'lumotni o'g'irlashni) o'z ichiga oladi. Odatda, maxfiylikni buzish buzg'unchining maqsadi hisoblanadi. Shaxsiy ma'lumotlarni o'g'irlash uchun tizimdan yoki ma'lumotlar oqimidan kredit karta ma'lumotlari yoki identifikatsiya ma'lumotlari kabi maxfiy ma'lumotlarni olish, buzg'unchi uchun to'g'ridan-to'g'ri pul ishlashiga olib kelishi mumkin.

Konfiguratsiya	Himoya qilishning soddaligi	Nisbiy xavf	Zaiflik
Yagona kompyuter (elektron pochta va Internetsiz)	Yuqori	Past	Parollar kombinatsiyasi, viruslar
LAN ga ulangan (Internetsiz)	O'rta	O'rta	Tarmoq trafigini analiz qilish (Sniffers), axborotni qalbakilashtirish (spoofing), parollar, viruslar)
LAN ga ulangan (Internet bilan)	Past	Yuqori	E-mail, Web serverlar, FTP, Telnet (tarmoq trafigini analiz qilish, axborotni qalbakilashtirish, parollar, viruslar)

**Yaxlitlikni buzish.** Ushbu qoidabuzarlik ma'lumotlarni ruxsatsiz o'zgartirishni o'z ichiga oladi. Bunday hujumlar, masalan, aybsiz tomonga javobgarlikni topshirish yoki muhim tijorat dasturining dastlabki kodini o'zgartirishga olib kelishi mumkin.

**Imkoniylikni buzish.** Ushbu qoidabuzarlik ma'lumotlarni ruxsatsiz yo'q qilishni o'z ichiga oladi. Veb saytni buzilishi bu turdagi xavfsizlik buzilishining keng tarqalgan namunasidir.

**Xizmatni o'g'irlash.** Ushbu qoidabuzarlik resurslardan ruxsatsiz foydalanish bilan bog'liq. Masalan, buzg'unchi (yoki kirish dasturi) fayl serveri vazifasini bajaradigan tizimga demonni o'rnatishi mumkin.

**Xizmatni rad etish.** Ushbu buzilish tizimdan qonuniy foydalanishning oldini olish uchun mo'ljallangan. Xizmatni rad etish (DOS – Denial of service) hujumlari ba'zida tasodifiy bo'ladi. Buzg'unchilar xavfsizlikni buzishda bir nechta standart usullardan foydalanadilar.

Tizimni himoya qilish uchun to'rt darajadagi xavfsizlik choralari ko'rilishi kerak:

❖ **Fizik.** Kompyuter tizimlarini o‘z ichiga olgan sayt yoki saytlar zararli yoki yashirin buzg‘unchilardan fizik himoyalangan bo‘lishi kerak. Mashina xonalari ham, mashinalarga kirish huquqiga ega terminallar yoki ishchi stansiyalar ham xavfsiz bo‘lishi kerak.

❖ **Inson.** Avtorizatsiya ehtiyotkorlik bilan bajarilishi kerak, tizimga faqat tegishli foydalanuvchilar kira olishini ta‘minlash. Shunga qaramay, hatto vakolatli foydalanuvchilar ham boshqalarga ularning tizimga kirish huquqlaridan foydalanishga ruhsat berish uchun (masalan, pora evaziga) "taklif qilinishi" mumkin.

❖ **Operatsion tizim.** Tizim tasodifiy yoki qasddan buzilishlardan o‘zini himoya qilishi kerak. Masalan, stekni to‘lib qolishi ruhsatsiz jarayonni boshlashga imkon berishi mumkin.

❖ **Tarmoq.** Zamonaviy tizimlardagi kompyuter ma‘lumotlari-ning aksariyati xususiy ijaraga berilgan liniyalar, Internet kabi ulanish liniyalari, simsiz ulanishlar yoki kommutatsiya liniyalari orqali uzatiladi. Tarmoqda uzatilayotgan ma‘lumotlarni tutib olish kompyuterga kirish kabi zararli bo‘lishi mumkin.

Xavfsiz axborot tizimi - bu

1. Ruhsat etilmagan kirishlardan ma‘lumotlarni himoyalovchi,
2. Foydalanuvchiga doimo ma‘lumotlarni taqdim etishga tayyor turuvchi,
3. Shuningdek, ma‘lumotlarni ishonchli saqlash va ularni o‘zgarmasligini kafolatlovchi tizimdir.

Xavfsiz tizim maxfiylik, yaxlitlik va imkoniylik bilan aniqlanadi.

### ***Ma‘lumotni maxfiyligi, yaxlitligi va imkoniyligi***

Xavfsizlikka bag‘ishlangan ko‘plab ishlarda axborot tizimlarining xavfsizligi uch qismga bo‘linadi: maxfiylik, imkoniylik va yaxlitlik.

8.3- jadval

<b>Vazifalar</b>	<b>Tahdidlar</b>
Maxfiylik	Ma‘lumotlarni himoyalanganligi
Imkoniylik	Ma‘lumotlarni qalbakilashtirish
Yaxlitlik	Xizmatdan bosh tortish

Uchala komponent birgalikda CIA (Confidentiality, Integrity, Availability) deb nomlanadi. Ularning funksiyalari quyidagilardan iborat:

❖ **maxfiylik** (confidentiality) — maxfiy ma'lumotlarga faqat ruhsat berilgan (ro'yxatdan o'tgan) foydalanuvchilargina murojaat qila olishini kafolatlash;

❖ **imkoniylik** (availability) — ruhsat berilgan (ro'yxatdan o'tgan) foydalanuvchilar har doim ma'lumotlarga murojaat qila olishini kafolatlash;

❖ **yaxlitlik** (integrity) — ma'lumotni to'g'ri qiymatini saqlash, ruhsat berilmagan (ro'yxatdan o'tmagan) foydalanuvchilarni ma'lumotlarga qandaydir o'zgartirish kiritishi, buzishi yoki o'chirishini ta'qiqlashni kafolatlash.

Xavfsizlik talablari tizimning maqsadiga, ishlatilgan ma'lumotlarning xarakteri va mavjud tahdidlarning turiga qarab farq qilishi mumkin. Butunlik va imkoniylik xususiyatlari muhim bo'lmagan tizimni tasavvur qilish qiyin, ammo maxfiylik har doim ham talab qilinmaydi. Masalan, agar siz Internetda veb-serverda ma'lumotni nashr qilsangiz va sizning maqsadingiz uni keng doirada odamlarga taqdim qilish bo'lsa, unda bu holda maxfiylik talab qilinmaydi. Shunga qaramay yaxlitlik va imkoniylik talablari dolzarbligicha qolmoqda. Darhaqiqat, agar siz ma'lumotlar yaxlitligini ta'minlash uchun maxsus choralarni ko'rmasangiz, buzg'unchi sizning serveringizdagi ma'lumotlarni o'zgartirishi va shu bilan korxonangizga zarar yetkazishi mumkin. Masalan, jinoyatchi veb-serverga joylashtirilgan narxlar ro'yxatiga o'zgartirish kiritishi mumkin, bu sizning kompaniyangizning raqobatdoshligiga salbiy ta'sir ko'rsatishi yoki kompaniyangiz tomonidan erkin tarqatiladigan dasturiy mahsulotning kodlarini o'g'irlashi mumkin, bu albatta uning biznes imidjiga ta'sir qiladi. Ushbu misolda ma'lumotlarning imkoniyligi ham muhim hisoblanadi. Internetda serverni yaratish va qo'llab-quvvatlash uchun katta pul sarflab, kompaniya daromadga ishonishga haqlidir: mijozlar sonining ko'payishi, sotuvlar sonining ortishi va boshqalar. Ammo buzg'unchi hujumni boshlashi va natijada serverni boshqa barcha so'rovlarga kirish imkonini yo'qqa chiqarishi mumkin. Maxfiylik, imkoniylik va yaxlitlik tushunchalari nafaqat ma'lumotga, balki kompyuter tarmog'ining boshqa resurslariga, masalan, tashqi qurilmalar yoki ilovalarga nisbatan ham belgilanishi

mumkin. Noqonuniy" foydalanish ehtimoli xavfsizlik tizimini buzilishiga olib keladigan ko'plab tizim resurslari mavjud. Masalan, chop etish qurilmasiga cheklanmagan kirish buzg'unchiga chop etilgan hujjatlarning nusxalarini olish, sozlamalarni o'zgartirish, ish tartibining o'zgarishiga va hatto qurilmaga zarar yetkazilishiga olib kelishi mumkin. Chop etish qurilmasiga nisbatan maxfiylik xususiyati shundaki, qurilmadan foydalanish faqat unga kirish huquqi berilgan foydalanuvchilargina beriladi. Qurilmalar turli xil xizmatlarni taqdim etishlari mumkin (chop etish, fakslarni yuborish, Internetga kirish, elektron pochta va boshqalar), ulardan noqonuniy foydalanish, korxonaga moddiy zarar yetkazish ham tizim xavfsizligini buzish hisoblanadi. Ular quyidagicha:

❖ Axborotni imkoniyligi, maxfiyligi, yoki yaxlitligini buzilishiga qaratilgan har qanday harakatlar, boshqa tarmoq resurslaridan noqonuniy foydalanish **tahdid** deb ataladi.

❖ Axborot tizimining kamchiligidan foydalanib, muvaffaqiyatli amalga oshirilgan tahdid **hujum** deb ataladi.

❖ **Xavf** — bu muvaffaqiyatli amalga oshirilgan hujum natijasida axborot resursining egasiga yetkazilishi mumkin bo'lgan zarar qiymatini ehtimoliy baholash. Xavf qiymati qancha katta bo'lsa, mavjud xavfsizlik tizimi shuncha **zaif** degani va hujumni amalga oshirish ehtimolligi ham shuncha katta bo'ladi.

## **8.2. Operatsion tizimga bo'ladigan tahdidlar va ularning turlari**

**Bilmay qilingan tahdidlar** holis, sodiq ishchilarni noto'g'ri xatti-harakatlari oqibatida yuzaga keladi, bu ularni malakasi pastligi yoki ma'sulyatsizligi tufayli yuzaga keladi. Bundan tashqari, tizimning dasturiy va apparat vositalarining ishonchsiz ishlashi oqibatlari ushbu turdagi tahdid bilan bog'liq. Shunday qilib, masalan, disk, disk kontrolleri yoki fayl serverining ishdan chiqishi sababli korxonaning ishlashi uchun muhim bo'lgan ma'lumotlar mavjud bo'lmasligi mumkin. Shu sababli, xavfsizlik masalalari texnik vositalarning ishonchliligi, nosozliklarga chidamliligi bilan chambarchas bog'liqdir.

Xavfsizlikga bog‘liq muammolar bilan birga ishonchlilik, texnik vositalarning rad etishlarga bardoshlilik muammolari keladi. Bunda oldini olish yo‘llarini takomillashtirish, apparat darajasida zaxiralash yoki ma’lumotlar bazasi darajasida nusxalash choralari ko‘riladi.

**Bilib (qasddan) qilingan tahdidlar** tizimni monitoring yoki ma’lumotlarni o‘qishini cheklash (passiv) yoki axborotni imkoniyligi va yaxlitligini buzilishi, qurilmalar va ilovalarni ishlash darajasini yo‘qotishi (aktiv) bilan bog‘langan. Qasddan qilingan tahdidlar xakerlar xatti-harakatlari natijasida yuzaga keladi va ochiqdan-ochiq tashkilotni zarar ko‘rishiga qaratilgan bo‘ladi. Yetarli zaxiralash va tiklash siyosatiga ega bo‘lish va arxivlashning boshqa usullarini bajarish ko‘plab hisoblash tizimlari uchun standart ish tartibidir. Ko‘plab tizim mamurlari ko‘p darajali zaxira nusxalari jadvalidan foydalanadilar. Ya’ni, ular butun tizimning zaxira nusxasini haftada bir marta va faqatgina o‘sha kuni o‘zgartirilgan fayllarni zahira nusxalarini olishadi. Qo‘shimcha xavfsizlik chorasi sifatida tizimning to‘liq zaxira nusxalari uch-olti oy davomida xavfsiz joyda saqlanadi.

Hisoblash tarmog‘ida qasddan qilingan taqdidlarni quyidagi turlarga ajratish mumkin:

- ❖ Tarmoqdagi kompyuterlardan biriga qonuniy (legal) foydalanuvchi nomi ostida noqonuniy kirish;
- ❖ Dastur-virus yordamida tizimlarni buzish;
- ❖ Qonuniy (legal) foydalanuvchilarni noqonuniy xatti-harakatlari;
- ❖ Tarmoqdagi ichki trafikni «eshitish».

Noqonuniy kirishlar quyidagi holatlar orqali amalga oshirilishi mumkin:

- ❖ Xavfsizlik tizimida imkoniyatlari xujjatlashtirilmagan operatsion tizimni qo‘llagan holatlarda zaif nuqtalarni vujudga kelishi;
- ❖ Kuzatish, parol faylini shifrlash, ehtimoliy terib ko‘rish yoki tarmoq trafigini analiz qilish yo‘li bilan olingan parollar, ya’ni «begona» parollarni ishlatish;
- ❖ Begona kompyuterlarga «troyan oti» ni kirishi. «Troyan oti» dasturi har doim kerakli utilita yoki o‘yin ko‘rinishida bo‘ladi.
- ❖ Qonuniy (legal) foydalanuvchilarni noqonuniy xatti-harakatlari — legal foydalanuvchi o‘zini lavozimidan foydalangan holatda uning funksiyasiga kiritilmagan xatti-harakatlarni bajarishga urinishi bilan baholanadi.

❖ Tarmoqdagi ichki trafikni «eshitish» — bu noqonuniy tarzda tarmoqni monitoring qilish, tarmoq xabarlarini ushlab olish va tahlil qilishga urinishi bilan baholanadi.

### *Dasturiy tahdidlar (program threats)*

Operatsion tizim jarayonlar va yadro ko'rsatmalarga muvofiq berilgan vazifani bajaradi. Agar foydalanuvchi dasturi ushbu jarayonni zararli ishlarni bajarishga majbur qilsa, u "Dasturiy tahdid" deb nomlanadi. Dasturiy tahdidning keng tarqalgan namunasi bu kompyuterda o'rnatilgan dastur bo'lib, foydalanuvchi ma'lumotlarini saqlash va xakerga tarmoq orqali yuborish imkoniyatini beradi. Quyida ba'zi ma'lum dasturiy tahdidlarning ro'yxati keltirilgan.

**Troyan Otlari** (Trojan Horse). Ko'pgina tizimlarda foydalanuvchilar tomonidan yozilgan dasturlarni boshqa foydalanuvchilar tomonidan bajarilishiga imkon beradigan mexanizmlar mavjud. Agar ushbu dasturlar ijrochi foydalanuvchiga kirish huquqini beradigan domenda ishlasa, boshqa foydalanuvchilar ushbu huquqlardan yomon maqsadlarda foydalanishlari mumkin. Masalan, matn muharriri dasturi tahrirlanadigan faylni qidirish uchun ba'zi kalit so'zlar kodini o'z ichiga olishi mumkin. Agar ular topilsa, butun faylni matn muharriri yaratuvchisi uchun mavjud bo'lgan maxsus joyga ko'chirish mumkin. O'z kod segmenti muhitini suiste'mol qiladigan kod segmenti troyan oti deb nomlanadi. Troyan otlari asosan internet tarmog'i orqali tarqaladi. Ko'pincha UNIX tizimlarida uchraydigan uzoq qidiruv yo'llari troyan ot muammosini yanada kuchaytiradi. Qidiruv yo'li noaniq dastur nomi ko'rsatilganda qidirish uchun kataloglar to'plamini ro'yxatlaydi. Qidiruv yo'li shu nomli faylni qidiradi va fayl bajariladi. Ushbu qidirish yo'lidagi barcha kataloglar xavfsiz bo'lishi kerak, aks holda troyan foydalanuvchi yo'lga kiritilishi va tasodifan ishga tushirilishi mumkin.

Troyan otining varianti - bu kirish dasturini emulyatsiya qiladigan dastur. Shubxalanmagan foydalanuvchi terminaldan tizimga kirishni boshlaydi va unga parolni noto'g'ri kiritganligi to'g'risida xabar beradi. U qaytadan tizimga kirishga urinadi va kirish muvaffaqiyatli amalga oshiriladi. Shunday qilib, uning autentifikatsiya kaliti va paroli o'g'ri terminalda qoldirgan kirish emulyatori tomonidan o'g'irlanadi. Emulyator parolni saqlab qo'yadi, kirish

xatosi haqidagi xabarni chop etadi va chiqadi. Keyin foydalanuvchiga haqiqiy kirish so'rovi taqdim etiladi. Agar operatsion tizim interfaol seans oxirida foydalanish to'g'risidagi xabarni chop etsa yoki barcha zamonaviy Windows operatsion tizimlarida ishlatiladigan control-alt-delete kombinatsiyasi kabi nazoratsiz belgilar ketma-ketligidan foydalansa, ushbu hujum turini yo'q qilish mumkin. Trojan otining yana bir varianti - josuslarga qarshi dastur. Ba'zida josuslik dasturi foydalanuvchi o'rnatishni tanlagan dasturga hamroh bo'ladi. Ko'pincha bu bepul dasturlar yoki sinov asosida taqdim etiladigan bepul dasturlar bilan birga keladi, lekin ba'zida bu tijorat dasturlarida ham kiritiladi. Ushbu dasturlar o'rnatilgandan so'ng foydalanuvchi tizimidan kerakli ma'lumotlarni to'playdi va kerakli manzilga yuboradi. Ushbu jarayon foydalanuvchi josuslik dasturini aniqlaguncha davom etadi.

**Tuzoq eshigi**(Trap Door). Dastur yoki tizimni ishlab chiquvchisi dasturda faqat u foydalanishi mumkin bo'lgan teshikni qoldirishi mumkin. Masalan, kod ma'lum bir foydalanuvchi identifikatori yoki parolini tekshirishi va oddiy xavfsizlik protseduralarini chetlab o'tishi mumkin.

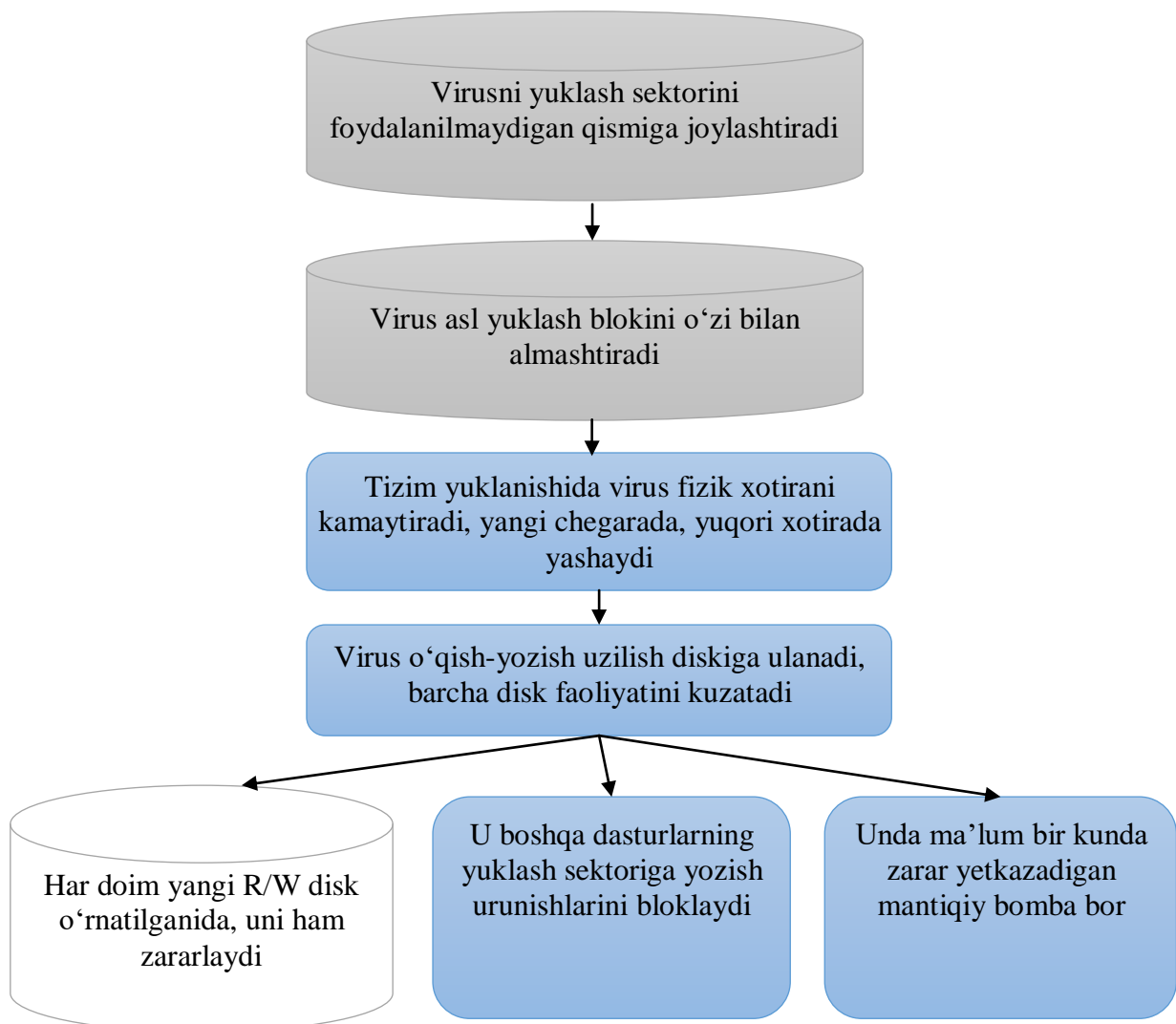
**Logik bomba** (Logic Bomb). Bu ma'lum bir shartlar bajarilgan taqdirdagina dastur noto'g'ri ish tutadigan vaziyat, aks holda u haqiqiy dastur kabi ishlaydi. Buni aniqlash qiyinroq.

### ***Viruslar (Viruses)***

Dasturiy tahdidlarning yana bir shakli bu virusdir. Virus bu qonuniy dasturga kiritilgan kod qismidir. Viruslar o'z-o'zini ko'paytiradigan va boshqa dasturlarga o'zini "yuqtirish" uchun mo'ljallangan. Ular fayllarni o'zgartirish yoki yo'q qilish va tizimning ishdan chiqishi va dasturning ishdan chiqishiga olib kelishi bilan tizimga zarar yetkazishi mumkin. Ko'pgina kirish hujumlarida bo'lgani kabi, viruslar ham arxitektura, operatsion tizim va dasturlarga juda xosdir. Viruslar shaxsiy kompyuterlar foydalanuvchilari uchun ma'lum bir muammodir. UNIX va boshqa ko'p foydalanuvchili operatsion tizimlar odatda viruslarga sezgir emas, chunki bajariladigan dasturlar operatsion tizim tomonidan yozishdan himoyalangan. Agar virus bunday dasturni yuqtirsa ham, uning vakolatlari odatda cheklangan, chunki tizimning boshqa jihatlari himoyalangan. Viruslar odatda elektron pochta orqali yuboriladi, spam eng keng tarqalgan



vektor hisoblanadi. Ular shuningdek, foydalanuvchilar Internet-fayl almashish xizmatlaridan virusli dasturlarni yuklab olganda yoki zararlangan disklarni almashganda ham tarqalishi mumkin. Virusni uzatishning yana bir keng tarqalgan shakli Microsoft Office hujjatlaridan foydalanadi, masalan, Microsoft Word hujjatlari. Ushbu hujjatlarda Office (Word, PowerPoint va Excel) dasturlari avtomatik ravishda ishlaydigan makroslar (yoki Visual Basic dasturlari) bo'lishi mumkin. Ushbu dasturlar foydalanuvchilarning shaxsiy akkauntlari ostida ishlayotgani sababli, makroslar deyarli hech qanday cheklovlarsiz ishlashi mumkin (masalan, foydalanuvchi fayllarini xohlaganicha o'chirish).



8.1- rasm. Kompyuter viruslarining yuklash sektori

Viruslar qanday ishlaydi? Virus maqsadli mashinaga yetib borgach, virusni tomchi virusi deb nomlanadigan dastur tizimga virusni kiritadi. O'rnatishdan so'ng, virus har qanday harakatni amalga oshirishi mumkin. Bugungi kunda minglab viruslar mavjud, ammo ular bir nechta asosiy toifalarga bo'linadi.

**Fayl viruslari** (File virus). Standart fayl virusi tizimga o'zini faylga qo'shish orqali yuqadi. Dastur boshlanishini o'zgartiradi, shunda bajarish uning kodiga o'tadi. Amalga oshirilgandan so'ng, boshqaruvni dasturga qaytaradi, shuning uchun uning bajarilishi sezilmaydi.

**Yuklash virusi** (Boot virus). Yuklash virusi tizimning yuklash sektoriga zarar yetkazadi, bu tizim har safar ishga tushganda va operatsion tizim yuklanmasdan oldin bajariladi. U boshqa yuklanadigan vositalarni ko'radi va ularga zarar yetkazadi. Ushbu viruslar xotira viruslari deb ham nomlanadi, chunki ular fayl tizimida ko'rinmaydi.

**Makro virus** (Macro virus). Ko'pgina viruslar past darajadagi dasturlash tilida yozilgan, masalan, assembler yoki C. Makro viruslar Visual Basic kabi yuqori darajadagi tilda yozilgan. Ushbu viruslar dastur ishga tushirilganda ishga tushadi. Masalan, makro virus elektron jadval faylida bo'lishi mumkin.

**Manba kodi virusi** (Source code virus). Manba kodi virusi dastlabki kodni qidiradi va unga virusni kiritish va virus tarqalishiga yordam berish uchun o'zgartiradi.

**Polimorfik virus** (Polymorphic virus). Polimorfik viruslar bu murakkab fayl infeksiyalari bo'lib, ular aniqlanmasligi uchun o'zi o'zgartirilgan versiyalarini yaratishi mumkin, ammo har bir infeksiyadan keyin bir xil asosiy tartibni saqlab qoladi. Boshqa zararli dasturlar bilan birlashganda, polimorfik viruslar uning qurbonlari uchun yanada katta xavf tug'diradi.

**Tunellash virusi** (Tunneling virus). Ushbu virus o'zini uzilishlarni qayta ishlovchi zanjirga joylashtirib antivirus tekshiruvini yordamida aniqlashni chetlab o'tishga harakat qiladi. Shunga o'xshash viruslar o'zlarini qurilma drayverlariga o'rnatadilar.

**Ko'p qismli virus** (Multipartite virus). Ushbu turdagi virus tizimning bir necha qismlarini, shu jumladan yuklash tarmoqlarini, xotirani va fayllarni yuqtirishi mumkin. Bu uni aniqlashni va saqlashni qiyinlashtiradi.

### ***Tizim tahdidlari (System Threats)***

Tizim tahdidlari - bu tizim xizmatlarini suiste'mol qilish va tarmoq ulanishlari bo'lib, foydalanuvchi uchun muammolar tug'diradi.

**Chualchanglar** (Worms) - bu tizim resurslaridan yuqori darajada foydalanib, tizimning ish faoliyatini pasaytiradigan jarayon. Chualchang jarayoni bir nechta nusxalarni keltirib chiqaradi, ularning har biri tizim resurslaridan foydalanadi va boshqa barcha jarayonlar kerakli resurslarni olishiga xalaqit beradi. Jarayon qurtlari hatto butun tarmoqni o'chirib qo'yishi mumkin.

**Portni skanerlash** (Port scanning) - bu buzg'unchi tizimga hujum qilishi uchun tizimning zaifliklarini aniqlaydigan mexanizm yoki vositadir.

**Xizmat ko'rsatishni rad etish** (Denial of Service) - xizmat ko'rsatishni rad etish odatda foydalanuvchiga tizimdan qonuniy ravishda foydalanishga imkon bermaydi. Masalan, agar brauzer tarkibidagi sozlamalarga hujum qilinsa xizmat ko'rsatish rad etilsa, foydalanuvchi Internetdan foydalana olmaydi.

### **8.3. Operatsion tizimning himoya vositalari va dasturlari**

#### ***Xavfsizlikni ta'minlashga tizimli yondashuv***

##### **Vosita va harakatlar:**

- ❖ etik-ahloqiy (mamlakatda hisoblash vositalarini tarqatish bilan bog'liq xatti-harakatlar me'yorlari);
- ❖ qonuniy (axborotdan foydalanishni tartibga soluvchi qonunlar, hukumat qarorlari va tartiblarni buzganlarga ko'riladigan choralar);
- ❖ ma'muriy (axborot xavfsizligini ta'minlashga qaratilgan tashkilot yoki ishlab chiqarish ma'muriyatini qabul qilgan hatti-harakatlar);
- ❖ psixologik;
- ❖ jismoniy, fizik (tarqalishdan himoyalash uchun binoni ekranlashtirish, qo'yilgan apparatni ishchi holatini va unda «juchok» lar borligini tekshirish, tashqi kuzatish vositalari va qurilma, kompyuter bo'limiga kirishni cheklash va boshqalar);

❖ tarmoqda himoyalash imkonini beruvchi apparat va dasturiy vositalarni oʻrnatish (tarmoq xavfsizligi xizmatlari, tizim himoyasini taʼminlovchilari, misol uchun kirishni nazorat qilish, audit, maʼlumotlarni shifrlash, antivirus himoyasi, autentifikatsiya va avtorizatsiya jarayonlarini, tarmoq trafikini nazorati).

### *Xavfsizlik siyosati*

Xavfsizlik siyosatining asosiy tamoyillari quyidagilar:

❖ Har bir ishchini bajaradigan ish hajmidan kelib chiqib maʼlumotlardan foydalanish darajalarini belgilab berish;

❖ Xavfsizlikni taʼminlashga barchani safarbar qilish;

❖ Koʻp pogʻonali ximoyalash tizimini ishlatish, barcha pogʻonalarda himoya ishonchliligini bir xil balansda boʻlishiga eʼtibor berish;

❖ Rad etishlar yuz berganda maksimal himoyalash holatiga oʻtuvchi vositalardan foydalanish;

❖ Yagona nazorat-oʻtkazish nuqtasi tamoyili, yaʼni barcha kirish va chiqish trafiklari yagona bitta tarmoq tuguni orqali oʻtishidir. Misol uchun, tarmoqlararo ekran (firewall) orqali oʻtishi;

❖ Yuzaga kelishi mumkin boʻlgan xavf va uni bartaraf etish uchun zarar darajasidagi balansni saqlash. Hech qanday xavfsizlik tizimi 100 % maʼlumotlar himoyasini kafolatlamaydi. Shunga koʻra koʻriladigan zarar va xavf oʻrtasida balans boʻlishi lozim.

Internetga chiquvchi (ulangan) tarmoq uchun xavfsizlik siyosatini aniqlashda mutaxassislar masalani ikki qismga ajratishni tavsiya qiladi:

❖ Internet tarmogʻi xizmatlariga kirish imkoniyati siyosatini ishlab chiqish;

❖ Korxonada ichki tarmogʻi resurslariga kirish imkoniyati siyosatini ishlab chiqish.

Internet tarmogʻi xizmatlariga kirish imkoniyati siyosatiga quyidagi bandlarni kiritish zarur:

❖ Internet xizmatlari roʻyxatini tuzib chiqish, yaʼni ichki tarmoq foydalanuvchilar chegaralangan imkoniyatga ega boʻlishi kerak.

❖ Kirish imkoniyati turi orqali chegaralashni aniqlash, masalan SLIP (Serial Line Internet Protocol) va PPP (Point-to-Point Protocol) protokollarini qoʻllash lozim.

❖ Tashqi internet foydalanuvchilar ichki tarmoqqa kirish imkoniyatini mavjud yoki yo'qligini aniqlab olish zarur. Agar mavjud bo'lsa, kimlarga. Ko'pincha faqat korxonada xizmatlarini ishlatishlari uchun ayrimlarga bu imkoniyat beriladi, masalan elektron pochta xizmatlari uchun.

Korxonada ichki tarmog'i resurslariga kirish imkoniyati siyosatini quyidagi ikki tamoyillardan biriga asoslangan bo'lishi lozim:

❖ Aniq shaklda ruhsat berilmagan barchasiga ta'qiq qo'yish yoki;

❖ Aniq shaklda ta'qiq qo'yilmagan barchasiga ruhsat berish.

Tanlangan tamoyilga mos holda tashqi trafikni tarmoqlararo ekran yoki marshrutizator orqali qayta ishlash qoidalari aniqlanadi. Birinchi tamoyilga asoslangan himoyani tashkillashtirish yuqori darajadagi himoyalanganlikni beradi, lekin buning natijasida foydalanuvchilarga ko'pgina noqulayliklar keltirib chiqarishi mumkin, bundan tashqari bu usul tashkilot uchun qimmatga tushadi. Ikkinchi tamoyilga asoslangan tarmoq kam darajadagi himoyalanganlikka ega bo'ladi, lekin foydalanuvchilar ko'pgina qulayliklarga ega bo'lib, tashkilot uchun arzonroq tushadi.

### ***Antivirus dasturlar***

Tizimni zararli dasturlarning hujumidan himoya qilish uchun antivirus dasturini qo'llaniladi. Antivirus dasturlarining himoya darajasi odatda tizimda saqlanayotgan ma'lumotlarining muhimligiga mutanosibdir. Masalan, tibbiy ma'lumotlar yuqori darajada himoyalangan bo'lishi talab etiladi, chunki bu ma'lumotlarda millionlab insonlarning kasallik tarixi saqlanadi. Lekin, talaba yozgan kompyuter dasturlari, ehtimol, bunday yuqori darajadagi xavfsizlikni talab etmaydi.

Viruslarga qarshi qo'llaniluvchi bunday dastur profilaktik yoki diagnostik yoki ikkalasi ham bo'lishi mumkin. Profilaktik dasturlar har bir ishlab chiqilgan dastur uchun qiymatlarni asosiy faylga yozib, nazorat summasini hisoblashi mumkin. Keyinchalik, bu dastur bajarilishidan oldin uning nazorat qiymati olingan qiymat bilan taqqoslanadi. Odatda diagnostika dasturi fayl o'lchamlarini taqqoslaydi (kutilmagan o'zgarishlar bo'lmaganida qo'shilgan kodni tekshirish), nusxalash ko'rsatmalarini qidiradi va odatiy bo'lmagan fayllarni qidiradi. Ba'zi bir dasturlar ma'lum ko'rsatmalarni qidirishi

va dasturlarning bajarilishini kuzatishi mumkin. Ammo shuni esda tutish kerakki, ushbu paketlar sotilgandan ko'p o'tmay, tizim buzg'unchilari ularni oldini olish yo'llarini izlay boshlaydilar. Demak, eng yangi antivirus dasturi eng yangi viruslarni topish mumkin deyish mumkin. Boshqacha aytganda, eski dastur faqat eski viruslarni topadi.

Mavjud viruslar to'g'risida ma'lumot 8.4-jadvalda keltirilgan va ular tizim xavfsizligi bo'yicha ishlovchi kompaniyalar va davlat idoralari tomonidan taqdim etilgan.

8.4- jadval

<b>Manzil</b>	<b>Tashkilot</b>
<a href="http://csrc.nist.gov">http://csrc.nist.gov</a>	AQShning Kompyuter xavfsizligi resurs markazi, Axborot texnologiyasi laboratoriyasi
<a href="http://www.cert.org">www.cert.org</a>	CERT koordinatsiya markazi
<a href="http://www.sans.org">www.sans.org</a>	SANS Instituti
<a href="http://www.mcafee.com">www.mcafee.com</a>	McAfee, Inc
<a href="http://www.symantec.com">www.symantec.com</a>	Symantec Corp.
<a href="http://www.us-cert.gov">www.us-cert.gov</a>	AQShning Kiberxavfsizlik va infrastruktura xavfsizligi agentlig (CISA)

Antivirus dasturi virus bilan zararlangan fayllarni tuzatishga qodir bo'lsada, viruslar va worm dasturlar yoki troyanlar o'rtasidagi tuzilish farqlari tufayli, odatda, worm, troyan yoki aralashgan tahdidlarni tiklay olmaydi. Virus boshqa toza faylga yuqish orqali ishlaydi. Shuning uchun, antivirus dasturi ba'zan infeksiyani olib tashlaydi va qolganini (ya'ni faylni) buzmasdan qoldirishi mumkin. Boshqa tomondan, worm yoki troyan - bu zararli kod. Ya'ni, worm yoki troyan tarkibidagi dasturiy kodning butun tanasi tizimda saqlanayotgan ma'lumotlarga tahdid soladi va u tizimdan umuman olib tashlanishi kerak.

Troyanni yo'q qilishning yagona usuli - zararli dasturning butun tanasini olib tashlashdir. Masalan, agar Internet orqali bepul yuklab olish mumkin bo'lgan kompyuter o'yinlari tizimdagi identifikatorlar (ID) va parollarini o'g'iraydigan worm bo'lsa, o'zida zararli kodni tashuvchi o'yinni tozalash va qolganlarini saqlashning hech qanday usuli yo'q.

## 8.4. Operatsion tizimning autentifikatsiya usullari

### *Autentifikatsiya, avtorizatsiya, audit*

**Autentifikatsiya** (authentication) – bu xohlamagan kishini tarmoqqa kirishini oldini olish va qonuniy foydalanuvchi uchun kirishga ruhsat berishdir. «Autentifikatsiya» atamasi lotin tilidan tarjima qilinganda «aslini oʻrnatish» soʻzini anglatadi. Autentifikatsiyani identifikatsiyadan farqlash zarur.

Identifikatsiya oʻzida tizim foydalanuvchisini identifikatorini aks ettiradi, bu vaqtda autentifikatsiya esa - bu foydalanuvchini oʻzi ekanligini tasdiqlovchi, yaʼni kiritilgan identifikator unga tegishlilikini aniqlash demakdir.

Windows operatsion tizimida parol siyosati

8.5- jadval

Parametr	Odatda	Tavsiya qilinadi
Parol tarixidan fodalanih	Eslab qolish 1 parol	Eslab qolish 24 parol
Parolning maksimal muddati	42 kun	42 kun
Parolning minimal muddati	0 kun	2 kun
Minimal parol uzunligi	0 belgi	8 belgi
Parol murakkablik talablariga javob berishi kerak	Oʻchiq	Yoqish

Hisobga olishni taʼqiqlash siyosati

8.6- jadval

Parametr	Odatda	Tavsiya qilinadi
Akkauntni bloklash muddati	belgilanmagan	30 minut
Akkauntni bloklash darajasi	0	5 urinish
Bloklangan akkauntni qayta tiklash	belgilanmagan	30 minut

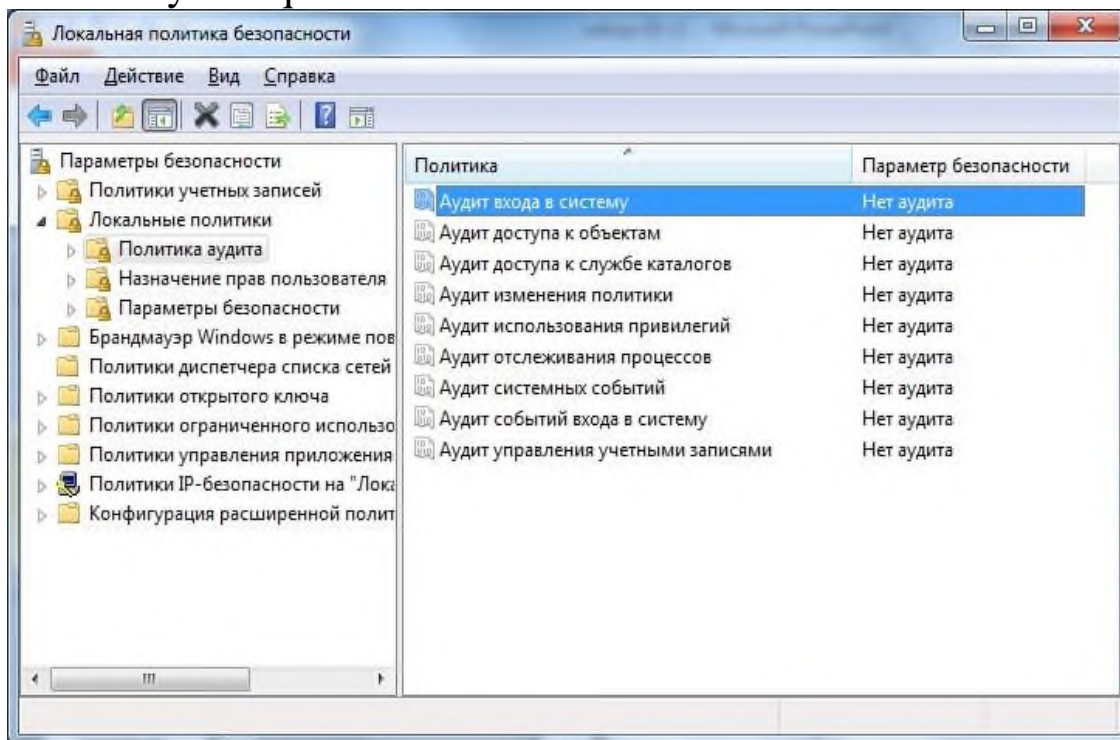
Hisoblashtarmoqlarida autentifikatsiya jarayonidasturiy vositalary ordamidaham tashkilqilinadi, bu jarayon **avtorizatsiya** debataladi.

Autentifikatsiya qonuniy vanoqonuniy foydalanuvchilarni aniqlasa, avtorizatsiya esa faqat qonuniy foydalanuvchilar, ya'ni autentifikatsiya jarayonidan muvaffaqiyatli o'tganlar bilan ish ko'radi.

Avtorizatsiya tizimining asosiy maqsadi har bir qonuniy foydalanuvchi uchun tizim ma'muridan belgilangan kirish darajasiga resurslarga murojaat qilish imkoniyatidara jasidajarayonni tashkilqilib berish hisoblanadi.

**Audit**- himoyalangan tizim resurslariga kirish bilan bog'liq bo'lgan holatlarni tizim holat jurnaliga yozib borish jarayoni. Zamonaviy OTlarda audit tizimi ma'murni qiziqtirgan holatlarni belgilangan tartibda qulay qilib grafik interfeys ko'rinishida hisobot beradi.

Audit hisobga olish va kuzatish vositasi eng muhim xavfsizlik bilan bog'liq holatlarni hosil qilish, kirishga urinish yoki tizim resurslarini o'chirishga urinishlarni aniqlash va yozib qolishda qo'l keladi. Audit asosan, har qanday amalga oshmay qolgan xavflarni ham o'zida yozib qoladi.



8.2- rasm. Windows OTda audit tizimi ko'rinishi



Operatsion tizimning audit tizimi quyidagi talablarni qanoatlantirishi zarur:

❖ Faqat operatsion tizimning o‘zi audit jurnaliga ma’lumotlarni qayd qilishi mumkin;

❖ Nafaqat operatsion tizim balki boshqa subyektlar ham audit jurnalidagi alohida ma’lumotlarni o‘zgartirish va o‘chirish imkoniyatiga ega emas;

❖ Faqat tegishli huquqlarga ega bo‘lgan auditor-foydalanuvchilar audit jurnalini ko‘rishi mumkin;

❖ Faqat auditor-foydalanuvchilar audit jurnalini tozalashi mumkin. Jurnal tozalangandan so‘ng avtomatik ravishda jurnalga uning qachon va kim tozalaganligi to‘g‘risida ma’lumot yoziladi;

❖ Audit jurnalining to‘lishi oqibatida operatsion tizim favqulodda o‘z ishini yakunlaydi.

### ***Parollar***

Foydalanuvchi identifikatorini aniqlashda eng keng tarqalgan yondashuv parollardan foydalanish hisoblanadi. Foydalanuvchi o‘zini foydalanuvchi identifikatori yoki hisob qaydnomasi nomi bilan aniqlasa, undan parol so‘raladi. Agar foydalanuvchi tomonidan taqdim etilgan parol tizimda saqlangan parolga mos kelsa, tizim ushbu hisob egasi tizimga kirish huquqiga ega deb taxmin qiladi. Parol tizimlari ko‘pincha kompyuter tizimidagi obyektlarni himoya qilish uchun foydalaniladi. Ularni kalit yoki imkoniyatlarning alohida holati deb hisoblash mumkin. Masalan, parol har bir resursga (masalan, faylga) bog‘lanishi mumkin. Ularni kalitlar yoki imkoniyatlarning maxsus holati deb hisoblash mumkin. Masalan, parol har bir resursga (masalan, faylga) bog‘liq bo‘lishi mumkin. Resursdan foydalanish so‘ralganda, parol talab qilinadi. Agar parol to‘g‘ri bo‘lsa, kirish huquqi beriladi. Turli xil parollar turli xil kirish huquqlari bilan bog‘liq bo‘lishi mumkin. Masalan, fayllarni o‘qish, fayllarni qo‘shish va fayllarni yangilash uchun turli xil parollardan foydalanish mumkin.

Biometrik parollar (barmoq, ko‘z va boshqalar) eng yaxshi himoya usuli hisoblanadi.

### ***Bir martalik parollar***

Bir martalik parollar oddiy autentifikatsiya bilan birga qo‘shimcha xavfsizlikni ta‘minlaydi. Bir martalik parol tizimida

foydalanuvchi har safar tizimga kirishga harakat qilganida noyob parol talab qilinadi. Bir martalik parol ishlatilgan bo'lsa, uni qayta ishlatib bo'lmaydi. Bir martalik parol turli yo'llar bilan amalga oshiriladi:

❖ Tasodifiy raqamlar - foydalanuvchilarga tegishli alfavit bilan birga raqamlari bosilgan kartalar beriladi. Tizim tasodifiy tanlangan bir nechta alifboga mos keladigan raqamlarni so'raydi;

❖ Maxfiy kalit - foydalanuvchiga foydalanuvchi identifikatori bilan taqqoslangan yashirin identifikatorni yaratishi mumkin bo'lgan apparat ta'minoti taqdim etiladi. Tizim har safar tizimga kirishdan oldin yaratilishi kerak bo'lgan yashirin identifikatorni talab qiladi.

❖ Tarmoq paroli - ba'zi tijorat ilovalari ro'yxatdan o'tish uchun kiritilishi kerak bo'lgan ro'yxatdan o'tgan telefon yoki elektron pochtaga foydalanuvchiga bir martalik parollarni yuboradi.

### ***Parol zaifliklari***

Parol juda keng tarqalgan, chunki ularni tushunish va ishlatish oson. Afsuski, parollarni ko'pincha taxmin qilish, tasodifan ochish, o'g'irlash (eshitish vositasi tomonidan o'qilgan) yoki noqonuniy tarzda vakolatli foydalanuvchidan ruxsatsiz foydalanuvchiga o'tkazish mumkin. Parolni taxmin qilishning ikkita keng tarqalgan usuli mavjud. Buning usullaridan biri – buzg'unchi (inson ham, dastur ham) foydalanuvchini bilish yoki foydalanuvchi to'g'risida ma'lumotga ega bo'lish.

Ko'pgina insonlar tez-tez aniq ma'lumotlardan (masalan, mushuklari yoki turmush o'rtoqlarining ismlari, telefon raqamlari va b.) o'zlarining parollari sifatida foydalanadilar. Boshqa usul - bu parol topilmaguncha taxminiy parollarni ishlatish, raqamlashtirish yoki haqiqiy parol belgilarining barcha mumkin bo'lgan kombinatsiyalari (harflar, raqamlar va boshqa belgilar). Qisqa parollar ushbu usul uchun ayniqsa himoyasiz. Buni oldini olish uchun foydalanuvchilar tizimga parol qo'yishda uzun parollardan foydalanishi kerak. Parollar tizim tomonidan yaratilishi yoki foydalanuvchi tomonidan tanlanishi mumkin. Tizimda yaratilgan parollarni eslab qolish qiyin bo'lishi mumkin va shuning uchun foydalanuvchilar ularni yozib olishlari mumkin. Aytib o'tilganidek, foydalanuvchi tanlagan parollarni topish oson bo'ladi (masalan, foydalanuvchi ismi yoki sevimli mashina nomi).

## ***Himoya***

Kompyuter tizimlari tobora rivojlanib borgan sari ularning yaxlitligini himoya qilish zarurati kuchaymoqda. Zamonaviy himoya tushunchalari umumiy resurslardan foydalanishni ta'minlaydigan har qanday murakkab tizimning ishonchligini oshirish uchun rivojlandi.

Operatsion tizimdagi jarayonlar bir-birining harakatlaridan himoyalangan bo'lishi kerak.

Bunday himoyani ta'minlash uchun biz faqat operatsion tizimdan tegishli ruxsatni olgan jarayonlar fayllar, xotira segmentlari, protsessorlar va tizimning boshqa resurslarida ishlashini ta'minlash uchun turli xil mexanizmlardan foydalanishimiz mumkin. Himoya deganda dasturlar, jarayonlar yoki foydalanuvchilarning kompyuter tizimi tomonidan belgilanadigan resurslarga kirishini boshqarish mexanizmi tushuniladi. Ushbu mexanizm qo'llanilishi kerak bo'lgan boshqarish vositalarini, shuningdek, amalga oshirish vositalari bilan spetsifikatsiyalash uchun vositalarni taqdim etishi kerak. Biz himoya va xavfsizlikni ajratamiz, bu tizim va uning ma'lumotlari yaxlitligini saqlashga bo'lgan ishonchning o'lchovidir.

## ***Viruslardan himoya***

Ko'rib turganimizdek, viruslar tizimga zarar yetkazishi mumkin. Shuning uchun tizimni viruslardan himoya qilish muhim xavfsizlik masalasidir. Ushbu himoyani ta'minlash uchun ko'pincha antivirus dasturlari ishlatiladi. Ushbu dasturlarning ba'zilari faqat ma'lum viruslarga qarshi samarali hisoblanadi. Viruslar ham, antivirus dasturlari ham murakkablashishda davom etmoqda. Ba'zi viruslar boshqa dasturiy vositalarni zararlash orqali o'zlarini o'zgartiradilar va bu antivirus dasturlariga ushbu virusni aniqlashga imkon bermaydi. O'z navbatida, antivirus dasturlari virusni aniqlashning yagona namunasini emas, balki namunaviy oilalarni qidirmoqda. Aslida, ba'zi antivirus dasturlari turli xil aniqlash algoritmlaridan foydalanadilar. Hozirgi kunda juda ko'plab antivirus dasturlari mavjud. Ular quyidagilar Kaspersky total security, Bitdefender antivirus plus, Norton 360 deluxe, Trend micro maximum security, McAfee internet security, ESET smart security premium, Microsoft Windows defender va boshqalar.

### *Tizim va tarmoqni himoya qilish uchun xavfsizlik devori*

Ishonchli kompyuterni qanday qilib ishonchsiz tarmoqqa xavfsiz ulash mumkinligi haqidagi savolga javob beramiz. Bitta echim ishonchli va ishonchsiz tizimlarni ajratish uchun xavfsizlik devoridan (Firewall) foydalanish hisoblanadi. Xavfsizlik devori bu ishonchli va ishonchsizlar orasida joylashgan kompyuter, qurilma yoki marshrutizator. Tarmoq xavfsizlik devori ikkita xavfsizlik domeni, barcha ulanishlarni kuzatib boradi va qayd qiladi. Shuningdek, u resurs yoki belgilangan manzilni, resurs yoki resurs porti yoki bog‘lanish yo‘nalishi bo‘yicha ulanishni cheklashi mumkin. Masalan, veb-serverlar veb-brauzerlar bilan bog‘lanish uchun HTTPdan foydalanadilar. Shuning uchun xavfsizlik devori faqat HTTPga xavfsizlik devori tashqarisidagi barcha xostlardan xavfsizlik devori ichidagi veb-serverga o‘tishga ruxsat berishi mumkin. Firewall – tarmoqlararo himoya qilish tizimi bo‘lib, u umumiy tarmoqni ikki va unda ortiq qismlarga ajratish, ma’lumotlar paketlarini chegara orqali umumiy tarmoqning bir qismidan boshqa qismiga o‘tish shartlarini aniqlaydigan qoidalar to‘plamini amalga oshirishga imkon beradi. Qoidaga ko‘ra, bu chegara korxonaning korporativ (lokal) tarmog‘i va Internet global tarmog‘i o‘rtasida o‘tkaziladi, biroq uni korxonaning korporativ tarmog‘i ichida ham o‘tkazish mumkin. Firewall o‘zi orqali har bir o‘tayotgan paket uchun qarorni, uni o‘tkazish kerakmi yoki tashlab yuborish kerakmi, qabul qilgan holda butun trafikni o‘tkazadi. Firewall buni amalga oshirish uchun, u filtrlashning qoidalar to‘plamini aniqlab olishi kerak. Umumfoydalanuvchi tarmoq tarafidan bo‘ladigan xavflarni to‘shishda Firewall nomini olgan dastur yoki qurilma vositalaridan foydalanadi. Odatda Firewall alohida hisoblash mashinasida amalga oshirilib, bu mashina orqali himoyalangan korporativ tarmoq umumfoydalanuvchi tarmog‘iga ulanadi.

Firewall himoyalangan korporativ tarmoqqa kelayotgan va undan chiqayotgan axborotni nazorat qiladi. Firewall quyidagi vazifalarni bajaradi:

- ❖ Ma’lumotlarni filtrlash;
- ❖ Ekranlovchi agentlardan foydalanish;
- ❖ Manzillarni bog‘lash;
- ❖ Hodisalarni ro‘yhatga olish.



8.3- rasm. Firewall orqali himoyalangan tarmoq

## **8.5. Axborot xafsizligini ta'minlashda kriptografiya masalalari**

Usuldan texnologiyaga qadar butun spektrni qamrab oladigan kompyuter hujumlaridan himoya qilishning ko'plab usullari mavjud. Tizimni ishlab chiquvchilari va foydalanuvchilari uchun mavjud bo'lgan eng keng vosita - bu kriptografiya. Ushbu bo'limda biz kriptografiya va undan kompyuter xavfsizligida foydalanish masalalarini muhokama qilamiz. Yaxshi kriptografiya kutubxonalari keng tarqalgan bo'lib, ular ishlab chiqarish uchun juda yaxshi zamin yaratadi. Tashqi muhitdan himoyalangan kompyuterda operatsion tizim barcha yuboruvchi va qabul qiluvchini jarayon aloqalarini ishonchli aniqlay oladi, chunki u kompyuterdagi barcha aloqa kanallarini boshqaradi. Kompyuter tarmog'ida vaziyat butunlay boshqacha. Tarmoq kompyuterlari bitlarni "sim orqali" qabul qiladi, bu bitlarni qaysi mashina yoki dastur yuborganligini tezkor va ishonchli aniqlashning iloji yo'q. Xuddi shunday, kompyuter bitlarni tarmoqqa yuboradi, keyinchalik ularni kim qabul qilishi mumkinligini bilmaydi. Odatda, tarmoq manzillari potentsial jo'natuvchilar va tarmoq xabarlarini qabul qiluvchilarni aniqlash uchun ishlatiladi. Tarmoq paketlari IP manzili kabi resurs manzili bilan birga keladi.

## *Xavfsizlikni asosiy texnologiyalari. Shifrlash*

Shifrlash — bu axborot xavfsizligi xizmatlarida eng muhim asosiy g'oya, ya'ni tizimni xuddi autentifikatsiyalash yoki avtorizatsiyalash, himoyalangan kanalni hosil qilishdagi vosita yoki ma'lumotlarni xavfsiz saqlash usullariga o'xshash vosita. Shifrlash aloqa xavfsizligi muammolarining keng doirasini hal qilganligi sababli, ko'pincha zamonaviy kompyuterlarning ko'pgina jihatlarida qo'llaniladi. U tarmoq bo'ylab xabarlarini xavfsiz ravishda yuborish, shuningdek ma'lumotlar bazasi ma'lumotlari, fayllar va hatto disklarning tarkibini ruxsatsiz obyektlar tomonidan o'qishdan himoya qilish uchun ishlatiladi. Shifrlash algoritmi yuborilgan xabarni faqatgina ma'lum bir kalitga ega kompyuter o'qishini ta'minlaydi. Albatta, xabarlarini shifrlash qadimgi amaliyot bo'lib, qadimgi zamonlardan beri ko'plab shifrlash algoritmlari mavjud. Ushbu bo'limda biz muhim zamonaviy shifrlash tamoyillari va algoritmlarini tavsiflaymiz. Shifrlash algoritmi quyidagi komponentlardan iborat:

- ❖ K – kalitlar to'plami;
- ❖ M – xabarlar to'plami;
- ❖ C – shifrlangan matnlar to'plami.

Shifrlash funksiyasi  $E: K \rightarrow (M \rightarrow C)$ . Ya'ni, har bir  $k \in K$  uchun  $E_k$  xabarlardan shifrlangan matnlarni yaratish funksiyasidir. Har qanday  $k$  uchun  $E$  va  $E_k$  ikkalasi ham samarali hisoblash funksiyalari bo'lishi kerak. Umuman olganda,  $E_k$  - bu xabarlardan shifrlashlargacha bo'lgan tasodifiy xaritada.

Deshifrlash funksiyasi  $D: K \rightarrow (C \rightarrow M)$ . Ya'ni, har bir  $k \in K$  uchun  $D_k$  shifrlangan matnlardan xabarlar yaratish funksiyasidir. Har qanday  $k$  uchun  $D$  va  $D_k$  samarali hisoblanadigan funksiyalar bo'lishi kerak.

Shifrlash algoritmi ushbu muhim xususiyatni ta'minlashi kerak: agar  $c \in C$  shifrlangan matn bo'lsa, kompyuter  $m$  ni  $E_k(m) = c$  bo'lsa, agar  $u$  kalitga ega bo'lsa hisoblashi mumkin.

Shifrlashni har qanaqa jarayoni oddiy «tushunarli» ko'rinishdan axborotni «o'qilmaydigan» holatga olib kelish, shifrlangan ko'rinishga albatta qo'shimcha jarayon shifrlangan matnni tushunarli ko'rinishga olib keladigan deshifrlash jarayoni mavjud. Shunday qilib, kalitni o'z ichiga olgan kompyuter deshifrlagan matnlarni oddiy

matnga shifrlay oladi, lekin k kompyuteri bo'lmagan kompyuter shifrlangan matnlarni deshifrlay olmaydi.

Shifr matnlari odatda ochiq bo'lganligi sababli (masalan, tarmoq orqali yuborilgan), shifrlangan matnlardan kalitlarni olishning iloji yo'qligi juda muhimdir. Shifrlash algoritmlarining ikkita asosiy turi mavjud: simmetrik va assimetrik.

Shifrlash va deshifrlash jarayoni kriptotizim deb ataladi. Shifrlash va deshifrlash jarayonida ishtirok etadigan axborotni shartli ravishda «matn» deb ataymiz, lekin bu axborot raqamlar to'plami yoki grafik ko'rinishidagi ma'lumot bo'lishi mumkin. Zamonaviy shifrlashda aniq asosiy parametr bu — maxfiy kalit. Agar real vaqtda kalitga yetib borish imkonini beruvchi jarayon topilsa, bunda shifrlash algoritmi ochiq deb hisoblanadi. Algoritm ochilishini qiyinligi kriptotizimni muhim xarakteristikalaridan biri hisoblanadi va kriptoustuvorlikdeb nomlanadi.

### ***Simmetrik va assimetrik shifrlash***

Kriptotizimni ikkita sinfi mavjud — simmetrik va asimetrik.

**Simmetrik** shifrlash sxemasida (klassik kriptografiya) shifrlashda qo'llanilgan maxfiy kalit deshifrlashdagi maxfiy kalitga mos tushadi.

**Asimetrik** shifrlash sxemasida (ochiq kalitli kriptografiya) shifrlashdagi ochiq kalit deshifrlashdagi maxfiy kalitga mos tushmaydi.

### ***Kalitlarni taqsimlash***

Kriptografiyani OSI modelidagi deyarli har qanday pog'onaga kiritish mumkin. Masalan, SSL transport pog'onasida xavfsizlikni ta'minlaydi. Odatda tarmoq sathi xavfsizligi IP Sec-da standartlashtiriladi, bu IP-paketlarning formatlarini belgilaydi, bu autentifikatorlarni va paket tarkibini shifrlashni ta'minlaydi. IPsec simmetrik shifrlashdan foydalanadi va kalitlarni almashish uchun Internet Key Exchange (IKE) protokolidan foydalanadi. IKE ochiq kalitlarni shifrlashga asoslangan. IPsec virtual shaxsiy tarmoqlar (VPN) uchun asos sifatida keng qo'llaniladi, bunda ikkita IPsec so'nggi nuqtalari o'rtasidagi barcha trafik xususiy tarmoqni bitta tarmoqdan ajratish uchun shifrlanadi, aks holda hammaga ochiq bo'lishi mumkin. SSL 3.0 - bu ikkita kompyuterga ishonchli aloqani

ta'minlaydigan kriptografik protokol, ya'ni har biri boshqalarga xabarlarini jo'natuvchi va qabul qiluvchini cheklashi mumkin. Bugungi kunda bu Internetda eng ko'p ishlatiladigan kriptografik protokoldir, chunki bu veb-brauzerlar veb-serverlar bilan ishonchli o'zaro aloqa qiladigan standart protokoldir. To'liqlik uchun SSL Netscape tomonidan ishlab chiqilgan va standart TLS protokoliga aylanganligini ta'kidlash kerak.

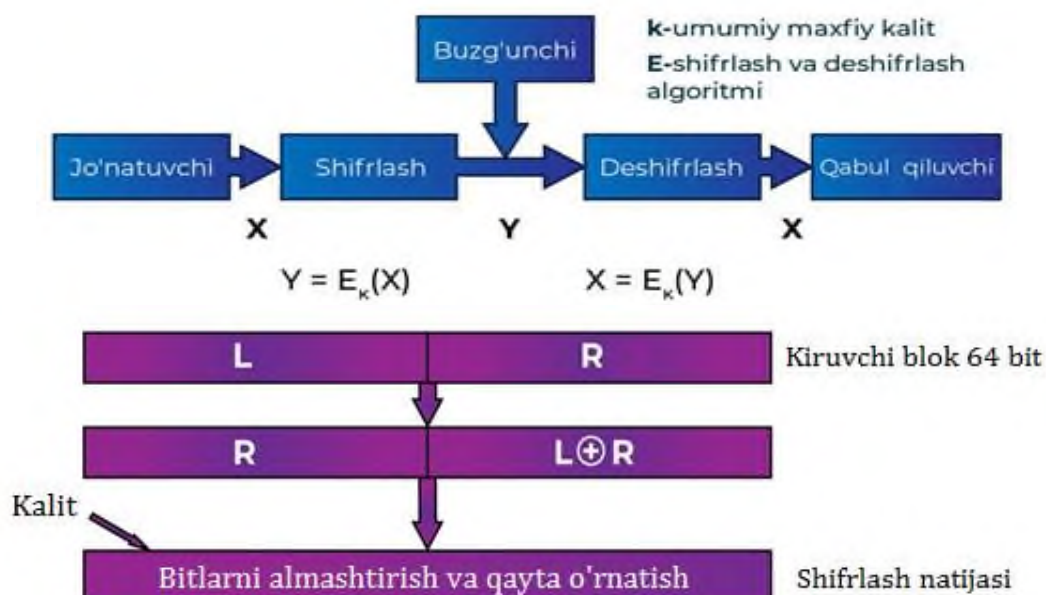
### ***Simmetrik shifrlash algoritmi***

Simmetrik shifrlash algoritmlarida asosiy muammo kalitlar bilan yuzaga keladi.

1. Ko'pgina simmetrik algoritmlarni kriptoustuvorligi kalit sifatiga bog'liq, bu esa kalitlarni generatsiya qiluvchiga talabni oshishiga olib keladi.

2. Asosiy tamoyil kalitni uzatish kanalini ishonchligi ikkinchi qatnashuvchi bilan maxfiy suxbatga bog'liq.

Tizimda ikkita abonent bo'lsa ham kalitlar borasida muammolar kelib chiqadi. Agar tizimda bir nechta abonent «har biri har biri bilan» tamoyili bilan maxfiy ma'lumot uzatishni xohlasa, unda abonentlar sonini kvadratiga teng kalitlar kerak bo'ladi, ko'p abonentlar o'rtasida bunday masalani yechish qiyinchilik tug'diradi. Simmetrik shifrlash algoritmi modeli 8.4- rasmda keltirilgan.



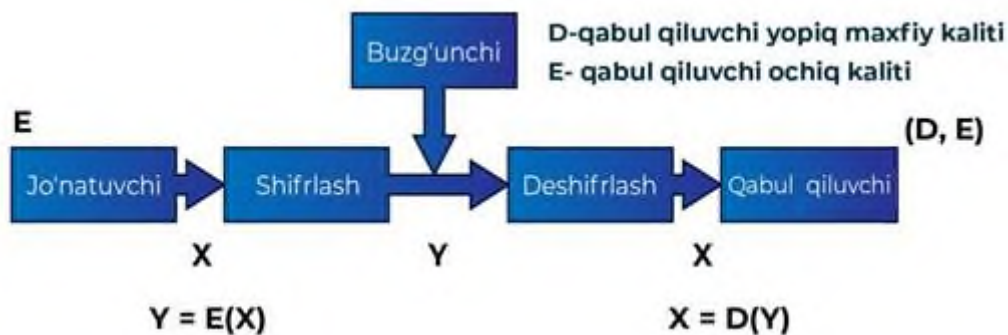
8.4- rasm. DES algoritmi bo'yicha shifrlash sxemasi



## Asimmetrik shifrlash algoritmi

Asimmetrik shifrlash algoritmi ochiq kalitga asoslangan, bu esa simmetrik shifrlashdagi muammoni yechadi.

1970-yillarning o'rtalarida ikkita olim — Vinifild Diffi va Martin Xellman — ochiq kalit yordamida shifrlash tamoyilini tushuntirib berishdi. Ochiq kalit yordamida shifrlashni asosiy jihati shundan iboratki, bunda bir vaqtni o'zida ikkita unikal juft kalitlar generatsiya qilinadi. Birinchi kalit shifrlashda ishlatilsa, ikkinchi kalit deshifrlashda ishlatiladi yoki aksincha (bu shifrlash algoritmida ikkita turli xil kalitlar ishlatiladi). Ushbu algoritmda shifrlash va parolni ochish uchun turli xil kalitlardan foydalaniladigan xususiyatga ega va agar yaxshi tanlangan shifrlash kaliti bo'lsa, tegishli shifrlash kalitini ochish deyarli mumkin emas. Bunday sharoitda shifrlash kaliti oshkor qilinishi mumkin va faqat deshifrlash kaliti sir saqlanadi.

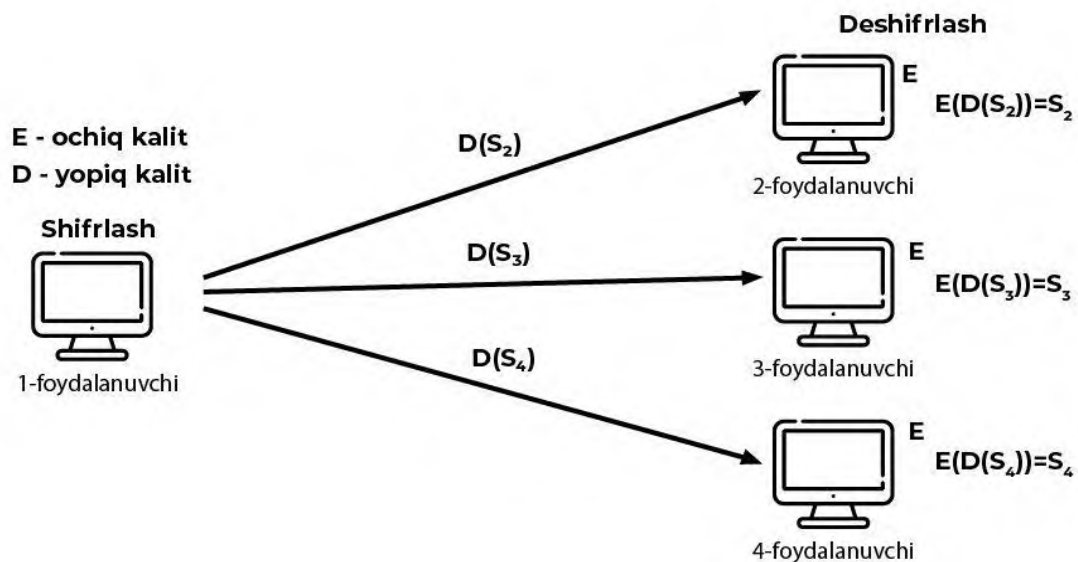
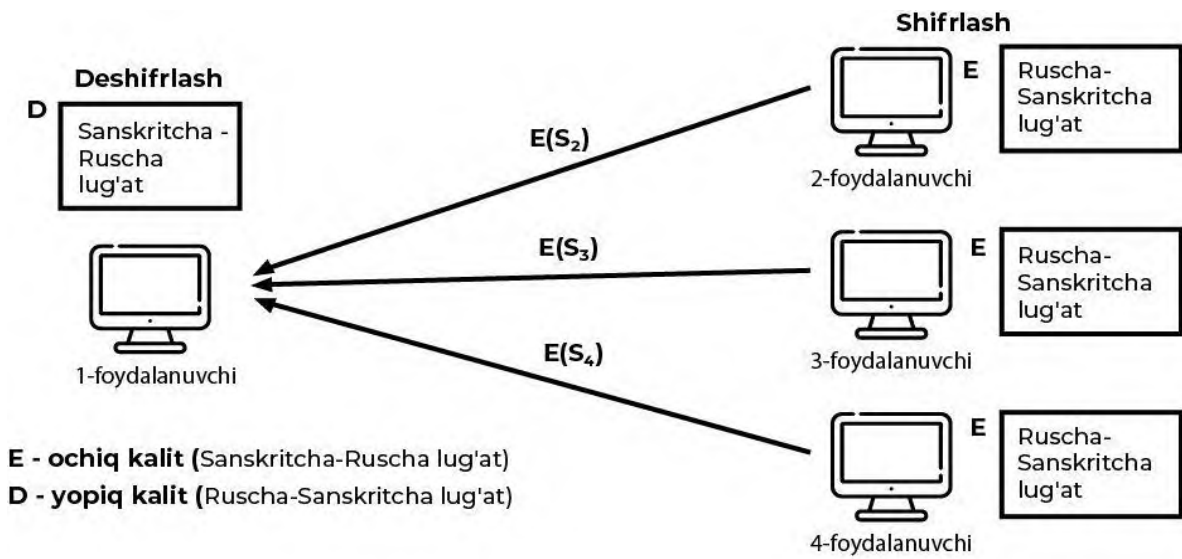


8.5- rasm. Asimmetrik shifrlash algoritmi sxemasi

## RSA kriptotalgoritmi

1978 yilda 3 ta olim (Rivest, Shamir va Adleman) ochiq kalit yordamida shifrlash tizimi Diffi-Xellman tamoyillariga to'liq javob beradigan RSA (Rivest, Shamir, Adleman) kriptotalgoritmini ishlab chiqishdi. Bu usul quyidagi bosqichlardan iborat:

1. Ikkita juda katta oddiy  $r$  va  $q$  tasodifiy tarzda tanlanadi.
2.  $n=p \cdot q$  va  $\varphi(n) = (p - 1) \cdot (q - 1)$  (**Eyler**) tenglamalari hisoblanadi.
3.  $e < \varphi(n)$  va  $\varphi(n)$  bilan umumiy bo'luvchiga ega bo'lmagan  $e$  soni tanlanadi.
4.  $1 < d < \varphi(n)$  va  $\text{mod}(d \cdot e - 1, \varphi(n)) = 0$  shart bo'yicha  $d$  tanlanadi.
5. Ochiq kalit –  $\{e, n\}$ , yopiq kalit –  $\{d, n\}$  juftliklari hosil qilinadi.
6. Shifrlash uchun  $c = \text{mod}(m^e, n)$  hisoblanadi.
7. Deshifrlash uchun  $m = \text{mod}(c^d, n)$  hisoblanadi.



8.6- rasm. Ochiq va yopiq kalitlarni ishlatilishi sxemasi

DES va RSA algoritmlarining solishtirma tavsilotlari 8.7- jadvalda keltirilgan.

8.7- jadval

Tavsilotlari	DES	RSA
Shifrlash tezligi	yuqori	quyi
Shifrlashfunksiyasidan foydalanish	Joyini almashtirish va o'zgartirish	Darajaga ko'tarish
Kalit uzunligi	56 bit	500 bitdan ko'p

Kriptoanalizga sarflash vaqti	Barcha kalitlar makoni bo'yicha ortiqcha	Sonlarni oddiy ko'paytiruvchilarga ajratish
Kalitni generatsiya qilish vaqti	Millisekund	Minut
Kalit turi	Simmetrik	Assimmetrik

### ***Bir tomonlama shifrlash funksiyasi***

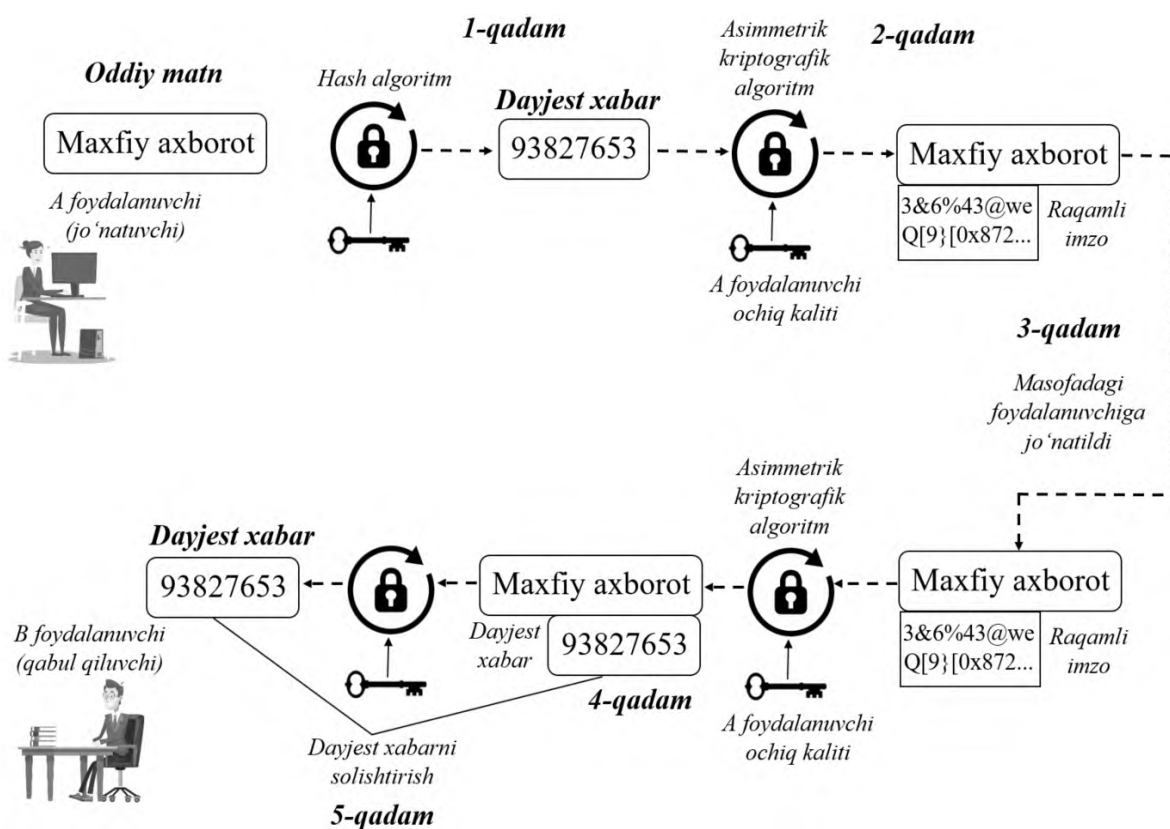
Xavfsizlikni asosiy texnologiyalari yana bir shifrlash usuli bir tomonlama funksiya yordamida shifrlashdir. U yana xesh-funksiya (hash function) yoki daydjest-funksiya (digest function) deb ham ataladi.

Bu funksiya ma'lumotni shifrlagandan keyin (daydjest) qiymat beradi. Daydjest birlamchi xabarlar bilan uzatiladi. Xabarni qabul qilgan shifrlanmagan birlamchi xabarni daydjest qiymatini hisoblaydi va xabar bilan kelgan daydjest qiymati bilan solishtiriladi. Agar qiymat mos tushsa axborot o'zgartirilmagan deb, aks holatda o'zgargan deb hisoblanadi. Daydjest qiymati xech qanaqasiga boshlang'ich axborotni qayta tiklamaydi, lekin uning yaxlitligini tekshirishga yordam beradi.

### **8.6. Xavfsizlik sertifikatleri va elektron raqamli imzo**

Aytaylik, A foydalanuvchi B foydalanuvchidan kelgan deb aytilgan shifrlangan hujjatni oladi. A foydalanuvchi, shifrlangan xabarni boshqa birov tomonidan uzatilayotganligini ko'rmaganligi yoki o'zgartirmaganiga amin bo'lishi mumkin, ammo u B foydalanuvchi haqiqatan ham yuborganligini qanday bilishi mumkin? A foydalanuvchining ochiq kaliti keng tarqalganligi sababli, buzg'unchi C foydalanuvchi soxta hujjat yaratib, uni A foydalanuvchining ochiq kaliti bilan shifrlashi va uni B foydalanuvchi bo'lish uchun da'vo qilib A foydalanuvchiga yuborishi mumkin. A foydalanuvchining kaliti hech kim hujjatni transportatsiya jarayonida o'qimaganligini yoki o'zgartirmaganligini tekshirishi mumkin, ammo u jo'natuvchini tasdiqlay olmaydi.

Jo‘natuvchini tasdiqlash elektron raqamli imzoni yaratish orqali assimetrik kriptografiya bilan ta‘minlanishi mumkin. Xabarni yaratgandan so‘ng, B foydalanuvchi to‘plamni hosil qiladi va A foydalanuvchiga elektron raqamli imzo hamda xabarni yuborishdan oldin shaxsiy kalit bilan to‘plamni shifrlaydi. Ularni qabul qilganda, A foydalanuvchi B foydalanuvchining ochiq kalitidan foydalangan holda raqamli imzoning shifrini ochadi va u to‘plamni ochib beradi. Keyin A foydalanuvchi xabarni B foydalanuvchi ishlatgan hash algoritmi bilan heshlaydi (qismlarga ajratish) va olingan natijani B foydalanuvchidan olgan to‘plam bilan taqqoslaydi. Agar ular mos bo‘lsa, A foydalanuvchi xabarni imzolaganidan beri o‘zgargani yo‘qligiga ishonch hosil qilishi mumkin. Elektron raqamli imzoning ishlash mexanizmi 8.7-rasmda keltirilgan.



8.7- rasm. Elektron raqamli imzo

Biroq, raqamli imzolarning bir kamchiligi bor: ular jo‘natuvchining haqiqiy shaxsini tasdiqlamaydi. Raqamli imzolar faqat jo‘natuvchining shaxsiy kaliti raqamli imzoni shifrlash uchun ishlatilganligini ko‘rsatadi, ammo ular jo‘natuvchining kimligini aniq

isbotlay olmaydi. Agar A foydalanuvchi raqamli imzo bilan B foydalanuvchidan ekanligini da'vo qilsa, u "haqiqiy" B foydalanuvchi ekanligini aniq bila olmaydi, chunki A foydalanuvchi faqat ochiq kalitni oladi.

Tasdiqlash asimmetrik kriptografiya bilan ta'minlanishi mumkin, ammo elektron raqamli imzoni yaratish orqali jo'natuvchini elektron tasdiqlash imkoniyati mavjud bo'ladi. Qog'ozli hujjatga qo'l bilan yozilgan imzo, imzolovchining hujjatni o'qiganligi va rozi bo'lganining isboti bo'lib xizmat qiladi. Elektron raqamli imzo ham huddi shunday, ammo qo'shimcha foydali jihatlari ham mavjud, masalan:

- ❖ Yuboruvchini tasdiqlash. Elektron raqamli imzo elektron xabarni jo'natuvchi shaxsni tasdiqlash uchun xizmat qiladi.

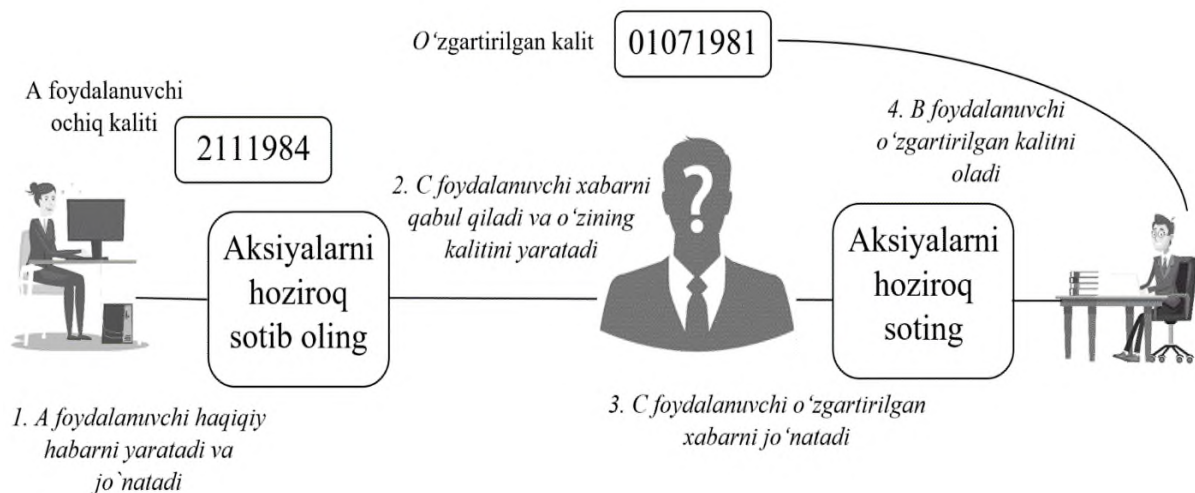
- ❖ Yuboruvchini xabarni rad qilishiga yo'l qo'ymaslik. Imzo qo'yuvchi keyinchalik imzo soxtalashtirilganligini da'vo qilib uni rad etishga harakat qila olmaydi.

- ❖ Xabarning yaxlitligini tasdiqlash. Raqamli imzo, xabar imzolangan vaqtdan beri o'zgartirilmaganligini tasdiqlashi mumkin.

Masalan, B foydalanuvchi raqamli imzo bilan birga xabar yaratdi va uni A foydalanuvchiga yubordi deylik. Biroq, C foydalanuvchi bu xabarni oraliq masofada qabul qildi. Keyin u B foydalanuvchining shaxsini ishlatib, o'zining ochiq va maxfiy kalitlarini yaratdi. Keyin C foydalanuvchi yangi xabar va raqamli imzo (maxfiy kalit bilan) yaratishi va ularni A foydalanuvchiga yuboradi. A foydalanuvchi xabarni va raqamli imzoni olgandan keyin bilmasdan C foydalanuvchinign ochiq kalitni (u B foydalanuvchiga tegishli deb o'ylagan holda) olib tashlaydi va shifrini ochadi. Ushbu jarayon 8.8- rasmda keltirilgan.

Bu raqamli sertifikat ortidagi konsepsiya hisoblanadi. Raqamli sertifikat - bu foydalanuvchi identifikatorini ochiq kalit bilan bog'lash uchun foydalaniladigan va ishonchli uchinchi shaxs tomonidan "raqamli imzo qo'yilgan" texnologiyadir. Ushbu uchinchi tomon (shaxs) imzo egasini va ochiq kalit bu egasiga tegishli ekanligini tasdiqlaydi. B foydalanuvchi A foydalanuvchiga xabar yuborganida, undan markaziy saytdan ochiq kalitini olib qo'yishni so'ramaydi, buning o'rniga, B foydalanuvchi raqamli sertifikatni xabarga qo'shadi. A foydalanuvchi raqamli sertifikat bilan xat olganida, sertifikatdagi ishonchli uchinchi tomon imzosini tekshirishi mumkin. Agar imzo

unga ishongan taraf tomonidan imzolangan bo'lsa, A foydalanuvchi raqamli guvohnomadagi ochiq kalit aslida B foydalanuvchidan ekanligiga ishonishi mumkin. Raqamli sertifikatlar A foydalanuvchiga B foydalanuvchining ushbu kalit unga tegishli ekanligi haqidagi da'vosini tekshirishga va ochiq kalitning egasini ko'rsatadigan o'rtada C foydalanuvchi hujumini oldini olishga imkon beradi.



8.8- rasm. O'zgartirilgan ochiq kaliti

Raqamli sertifikat odatda quyidagi ma'lumotlarni o'z ichiga oladi:

- ❖ Egasining ismi yoki alias nomi;
- ❖ Egasining ochiq kaliti;
- ❖ Sertifikat beruvchining nomi;
- ❖ Sertifikat beruvchining elektron raqamli imzosi;
- ❖ Raqamli sertifikatning seriya raqami;
- ❖ Ochiq kalitning amal qilish muddati.

Raqamli sertifikat asosan ochiq kalit uchun konteynerdir. Shu bilan birga, sertifikatlarda foydalanuvchi tomonidan taqdim etilgan boshqa ma'lumotlar, masalan, elektron pochta manzili, pochta manzili va ro'yxatga olish to'g'risidagi asosiy ma'lumotlar, masalan, mamlakat yoki mintaqa, pochta indeksi, foydalanuvchining yoshi va jinsi bo'lishi mumkin. Raqamli sertifikatlar foydalanuvchilardan boshqa obyektlarni, masalan serverlar va ilovalarni aniqlash uchun ishlatilishi mumkin.

### ***Kriptografik transport protokollari***

Foydalanish va saqlab turish vaqtida ma'lumotlarni himoya qilishdan tashqari, ko'pincha tarmoq ichidagi uzatilayotgan ma'lumotlarini himoya qilish uchun kriptografiya qo'llaniladi. Eng keng tarqalgan kriptografik transport protokollariga Secure Sockets Layer (SSL), Transport Layer Security (TLS), Secure Shell (SSH), Hypertext Transport Protocol Secure (HTTPS) va IP-security (IPsec) kiradi.

**Secure Sockets Layer (SSL).** Eng keng tarqalgan kriptografik transport algoritmlaridan biri bu Secure Sockets Layer (SSL). Ushbu protokol Netscape kompaniyasi tomonidan 1994-yilda Internet xavfsizligi to'g'risida ortib borayotgan xavotirlarga javoban ishlab chiqilgan. SSL-ning maqsadi har qanday platformada yoki operatsion tizimda ishlatilishi mumkin bo'lgan mijoz va server o'rtasida shifrlangan ma'lumotlar yo'lini yaratish edi. SSL zaifroq ma'lumotlarni shifrlash standarti (Data Encryption Standard - DES) o'rniga nisbatan yangi kriptografik algoritmni ilg'or shifrlash standarti (Advanced Encryption Standard - AES) dan foydalandi. Vaqt o'tishi bilan SSL uchun yangilanishlar ishlab chiqarildi. Bugungi kunda SSL 3.0 versiyasi serverlar eng ko'p qo'llab-quvvatlaydigan versiya hisoblanadi.

**Transport Layer Security (TLS)** – bu yana bir kriptografik transport algoritmidir. SSL va TLS ko'pincha bir-birining o'rnida yoki bir-biri bilan (TLS / SSL) ishlatilsa ham, bu to'g'ri yechim bo'lmaydi. SSL 3.0 versiyasi TLS 1.0 versiyasi uchun asos bo'lib xizmat qildi (va ba'zan xatto SSL 3.1 deb nomlanadi). TLS 1.0 versiyasi SSL 3.0 versiyasiga qaraganda qisman xavfsizroq deb hisoblansa-da, TLS ning keyingi versiyalari (1.1 va 1.2) ancha xavfsizroq va SSL 3.0 versiyasi va TLS 1.0 versiyasida mavjud bo'lgan bir nechta zaifliklarni bartaraf qiladi. TLS 1.1 va 1.2 versiyalari SSL 3.0 versiyasiga qaraganda ancha xavfsizroq bo'lishiga qaramay, ko'plab veb-saytlar eski veb-brauzerlar uchun keng ko'lamli moslikni ta'minlash uchun hali ham SSL va TLSning zaif versiyalarini qo'llab-quvvatlamogda.

2014-yil boshida SSL va TLSning ochiq kodli dasturiy ta'minoti bo'lgan OpenSSL da zaiflik aniqlandi. Ushbu zaiflik mijoz-server aloqasida boshqa tomon hali ham faol bo'lishini ta'minlash uchun foydalaniladigan OpenSSLning nisbatan yangi imkoniyatining (Heartbeat Extension) bir qismi edi. "Heartbleed" deb nomlangan

zaiflik, tajovuzkorlarga veb-server xotirasida ma'lumotlarga kirishga va aloqalarni shifrlash va deshifrlash uchun ishlatiladigan kriptografik kalitlarni o'g'irlashga imkon berdi. Veb-serverlar egalari o'zlarining serverlarida zaiflikni tezda tuzatishga majbur bo'lishdi.

Shifr to'plami (cipher suite) - bu SSL va TLS bilan ishlatiladigan shifrlash, autentifikatsiya va xabarlarning autentifikatsiyalash kodi (message authentication code - MAC) algoritmlarining nomlaridan kelib chiqqan kombinatsiyadir. Ular veb-brauzer va veb-server o'rtasida dastlabki ulanishni qo'llab-quvvatlash paytida kelishib olinadi. Tanlangan turli xil algoritmlarga qarab, uzatishning umumiy xavfsizligi kuchli yoki zaif bo'lishi mumkin. Masalan, AES o'rniga RC4 oqimlar shifridan foydalanish shifrlar to'plamini sezilarli darajada zaiflashtiradi. Yana bir omil - bu kalitlarning uzunligi. 2048 bitdan kam kalitlar zaif, 2048 bitli kalitlar yaxshi, 4096 bitli kalitlar kuchli deb hisoblanadi.

Shifr to'plamlari odatda ularning tarkibiy qismlarini ko'rsatish uchun tavsiflovchi nomlardan foydalanadilar. Masalan, CipherSuite SSL\_RSA\_WITH\_RC4\_128\_MD5 tavsiflovchi nomad quyidagilar tushiniladi - RSA shifrlash algoritmi kalitlarni almashish va autentifikatsiya algoritmidan ishlatilishini, 128 bitli kalit yordamida RC4 shifrlash algoritmidan foydalanilishini va MD5 (ma'lumotlarning yaxlitligini tekshirish va kalit so'z xeshlarini saqlash) algoritmi esa shifrlash, autentifikatsiya va xabarlarning autentifikatsiyalash kodi (MAC) algoritmi sifatida qo'llaniladi.

**Secure Shell (SSH)** - bu uzoq masofadagi kompyuterlarga kirish uchun foydalaniladigan Telnet protokoliga shifrlangan alternativa. SSH Linux/UNIXga asoslangan buyruq interfeysi va masofadan turib kompyuterga xavfsiz kirish uchun protokol. SSH aslida uchta yordamchi dasturlarning to'plami bo'lib, slogin, ssh va scp himoyalangan UNIX utilitalarini versiyalarini himoyalash uchun qo'llaniladi. Ular quyidagi maqsadlarda qo'llaniladi:

- ❖ slogin – masofadagi kompyuterga kirish;
- ❖ scp – masofadagi kompyuterlar o'rtasida fayllarni nusxalash;
- ❖ ssh – masofadagi host kompyuterga kirmagan holda buyruqlarni bajarish;

Ulanishning mijoz va server qismlari raqamli sertifikat yordamida autentifikatsiya qilingan va mahfiy so'z (parol)lar shifrlash bilan himoyalangan. SSH-ni hatto tarmoqning xavfsiz zaxira nusxasini



yaratish vositasi sifatida ishlatish mumkin. SSH ning birinchi versiyasi 1995-yilda Xelsinki Texnologiyalar Universitetining tadqiqotchisi tomonidan universitet parolni o'g'irlash xurujining qurboni bo'lganidan keyin chiqilgan.

**Hypertext Transport Protocol Secure (HTTPS).** TLS va SSL-ning keng tarqalgan foydalanish maqsadlaridan biri bu brauzer va veb-server o'rtasida Hypertext Transport Protocol (HTTP) aloqasini ta'minlashdir. Ushbu himoyalangan versiya aslida "oddiy" HTTP bo'lib, u SSL yoki TLS orqali yuborilgan va Hypertext Transport Protocol Secure (HTTPS) deb nomlanadi. HTTPS HTTP 80 portining o'rniga 443 portidan foydalanadi. Foydalanuvchilar URL manzillarini `http://` o'rniga `https://` bilan kiritishlari kerak. HTTP uchun boshqa kriptografik transport protokoli Secure Hypertext Transport Protocol (SHTTP) edi. Biroq, bu HTTPS kabi xavfsiz emas edi va hozir eskirgan deb hisoblanadi.

**IP Security (IPsec) -Internet Protocol Security** - Internet protokoli (IP) aloqasini himoyalash uchun protokol to'plami. IPsec hostlar yoki tarmoqlar o'rtasidagi har bir seansning IP-paketini shifrlaydi va autentifikatsiyalaydi. IPsec SSL yoki TLSga qaraganda ancha kengroq doiradagi dasturlarni himoya qilishi mumkin. IPsec xavfsizlikning shaffof protokoli hisoblanadi. Bu quyidagilar uchun shaffofdir:

- ❖ Ilovalar (applications). IPsec ostida ishlash uchun dasturlarni o'zgartirish kerak emas.

- ❖ Foydalanuvchilar. Ba'zi xavfsizlik vositalaridan farqli o'laroq, foydalanuvchilarga xavfsizlikning muayyan protseduralari bo'yicha o'qitish kerak emas.

- ❖ Dasturiy ta'minot. IPsec firewall yoki marshrutizator kabi qurilmada amalga oshirilganligi sababli, dasturiy ta'minotda hech qanday o'zgarishlar bo'lmasligi kerak.

Foydalanuvchi dasturining bir qismi sifatida amalga oshiriladigan SSLdan farqli o'laroq, IPsec operatsion tizimda yoki aloqa apparatida joylashgan. IPsec boshqa tizim dasturlari va qo'shimcha qurilmalar bilan yaqin hamkorlik qilishi mumkinligi hisobiga tezroq ishlaydi.

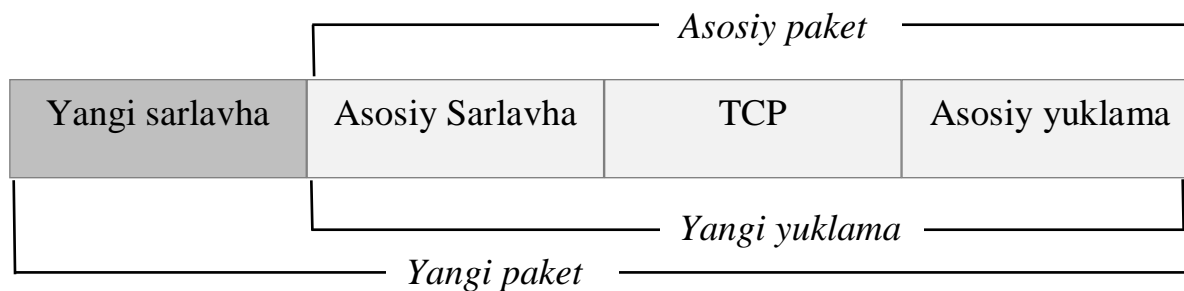
IPsec protokollarga mos keladigan uchta himoya sohasini ta'minlaydi:

❖ *Autentifikatsiya.* Haqiqiylikni tekshirish. IPsec olingan paketlar manbadan yuborilganligini tasdiqlaydi. Bu - paketning sarlavhasida ushbu paket tarkibini o‘zgartiradigan bironta ham tarmoqdagi hujum yoki takroran hujum qilinmaganligiga ishonch hosil qilish uchun belgilangan. Bu Autentifikatsiya sarlavhasi (Authentication Header - AH) protokoli orqali amalga oshiriladi.

❖ *Maxfiylik.* Paketlarni shifrlash bilan IPsec boshqa hech kim tarkibni ko‘rishga imkon bermasligini ta‘minlaydi. Maxfiylikka Encapsulating Security Payload (ESP) protokoli orqali erishiladi. ESP jo‘natuvchini autentifikatsiya qilish va ma‘lumotlarni shifrlashni qo‘llab-quvvatlaydi.

❖ *Kalitlarni boshqarish.* IPsec kalitlarni ularni ruxsatsiz shaxslar tomonidan ushlab qolinmasliklari yoki ulardan foydalanmasliklari uchun boshqaradi. IPsec ishlashi uchun yuborish va qabul qilish moslamalari kalitni bo‘lishishlari kerak. Bu Internet xavfsizlik assotsiatsiyasi va kalitlarni boshqarish protokoli/Oakley (Internet Security Association and Key Management Protocol/Oakley - ISAKMP/Oakley) deb nomlanadigan protokol orqali amalga oshiriladi, u kalitni yaratadi va raqamli sertifikatlar kabi usullardan foydalangan holda foydalanuvchini tasdiqlaydi.

IPsec ikkita shifrlash rejimini qo‘llab-quvvatlaydi: transport va tunnel. Transport rejimi faqat har bir paketning ma‘lumotlar qismini (yuklamani) shifrlaydi, lekin sarlavha shifrlanmagan holda qoladi. Keyinchalik xavfsiz tunnel rejimi sarlavhani va ma‘lumotlar qismini shifrlaydi. IPsec IP paketiga yangi sarlavhalarni qo‘shish orqali transport va tunnel rejimlarini bajaradi. So‘ngra to‘liq to‘plam (sarlavha va yuklama) yangi paketning ma‘lumotlar qismi sifatida ko‘rib chiqiladi. Bu 8.9- rasmda ko‘rsatilgan.



8.9- rasm. Tunel rejimidan foydalanuvchi yangi IPsec paketi

Tunnel rejimi butun paketni himoya qilganligi sababli, odatda tarmoqdan tarmoqqa ulanishda ishlatiladi. Transportlash rejimi qurilma paketni yoʻnaltirishi uchun manba va qabul qiluvchi manzillarini koʻrishi kerak boʻlganda ishlatiladi. Masalan, mijoz kompyuteridan mahalliy IPsec-ga ulangan firewall orqali yuborilgan paket transport rejimida yuboriladi, shunda paket mahalliy tarmoq orqali uzatiladi. U firewallga yetib borgach, Internetga yuborilishidan oldin u tunnel rejimiga oʻzgartirildi. Soʻngra qabul qiluvchi firewall dastlabki paketni oxirgi manzil kompyuteriga yuborilishidan oldin chiqarib tashlaydi, shifrlaydi va haqiqiylikini tekshiradi.

IPv4da IPsec ixtiyoriy protokoldir. IPv6da, IPsec IP protokoli bilan birlashtirilgan va barcha paketlarda mavjud. Barcha IPv6 tugunlari IPsec ga ega boʻlishi kerak, ammo IPsec ning IPv6 dan amalda foydalanish ixtiyoriydir.

### **Nazorat savollari**

1. Kompyuter xavfsizligi nima?
2. Tarmoq xavfsizligi nima?
3. Operatsion tizimning xavfsizlikda tutgan oʻrni?
4. Tizimning yashovchanligi nima?
5. Himoyalashning qanday darajalari mavjud?
6. Xavfsizlik muammolariga nimalar kiradi?
7. Tizimni himoya qilish uchun qanday xavfsizlik choralari koʻrilishi kerak?
8. Maʼlumotni maxfiyligi nima?
9. Maʼlumotni yaxlitliligi nima?
10. Imkoniylik nima?
11. Tahdid turlarini keltiring.
12. Qasddan qilingan tahdidlar nima?
13. Dasturiy tahdidlar nima?
14. Troyan otlari bu nima?

## **IXBOB. AMALIYOTDA OPERATSION TIZIMLARDAN FOYDALANISH**

### **9.1. Linux operatsion tizimi, qurilish tamoyili va ahamiyati**

Linux - bu UNIX operatsion tizimining mashhur variantidir, ammo UNIX ning boshqa ko'plab versiyalari va variantlari mavjud, ular Unix operatsion tizimi mavzusida ko'rib chiqilgan. Ushbu tizim ishchi stansiyalar va serverlarning eski modellari uchun ustun bo'lgan operatsion tizimlardan biri bo'lib, u boshqa tizimlarda - smartfonlardan (Android operatsion tizimi Linux asosida) superkompyuterlarga qadar qo'llaniladi. Bu operatsion tizimlarni qurishning ko'plab muhim tamoyillarini aks ettiradi. Muhokamamizni Linux tarixi va rivojlanish yo'lidan boshlaymiz. Keyinchalik Linux operatsion tizimi haqida umumiy ma'lumotlar va undan qanday foydalanish haqida tushunchalar beriladi. Ushbu sharh faqatgina Windows tizimi bilan tanish bo'lgan o'quvchilar uchun juda muhimdir, chunki tizim foydalanuvchidan tizimning deyarli barcha tafsilotlarini yashiradi. Grafik interfeyslar operatsion tizimni endi boshlagan foydalanuvchilar uchun juda qulay bo'lishi mumkin bo'lsada, ular past moslashuvchanlikka ega va tizim qanday ishlashi haqida tushuncha bermaydi. Keyin ushbu bobning asosiga o'tamiz - Linux tizimidagi xotirani boshqarish, kiritish/chiqarish, fayl tizimi va xavfsizlik masalalariga. Har bir mavzu uchun biz avval asosiy tushunchalarni, so'ngra tizim muammolarini va nihoyat amalga oshirish usullarini muhokama qilamiz. Linus Torvalds tomonidan Linux operatsion tizimini yaratilishiga MINIX operatsion tizimi sabab bo'lgan. Shuning uchun biz mavzuni avval MINIX operatsion tizimidan boshlaymiz.

### ***MINIX***

Barcha zamonaviy UNIX tizimlari umumiy xususiyatga ega: ularning barchasi yirik va murakkab bo'lib, ular qaysidir ma'noda UNIX tizimining asosidagi g'oyaga zid keladi. Tizimlarning barcha boshlang'ich kodlari erkin (bu ko'p hollarda noto'g'ri bo'lgan) bo'lsa ham, baribir, bitta odam ularni tushunishi mumkin emas. MINIX tizimi birinchi Unixga asoslangan mikroyadroli tizimlaridan biri edi. Mikroyadroning g'oyasi, yadroda iloji boricha kamroq funksiyalarni

bajarish va uni ishonchli va samarali qilishdir. Shunga ko'ra, xotira boshqaruvi va fayl tizimi foydalanuvchi jarayonlariga o'tkazildi. Yadro boshqa hech narsa qilmasdan jarayonlar o'rtasida xabarlarni uzatish bilan shug'ullanadi. Yadro C tilidagi 1600 va 800 assembler qatorlaridan iborat edi. Intel 8088 protsessorining arxitekturasi bilan bog'liq texnik sabablarga ko'ra, yadroga kiritish/chiqarish qurilma drayverlari (yana 2900 qo'shimcha qatorlar) joylashtirilgan. Fayl tizimi (C da 5100 satr) va xotira menejeri (C da 2200 satr) ikkita alohida foydalanuvchi jarayoni sifatida ishladi. Mikroyadro tizimining monolit tizimdan ustunligi shundaki, mikroyadro tizimini tushunish va saqlash oson (yuqori modulliligi sababli). Bundan tashqari, kodni yadrodan foydalanuvchi rejimiga o'tkazish tizimni yuqori darajadagi ishonchlilik bilan ta'minlaydi, chunki foydalanuvchi rejimida ishlaydigan jarayonning buzilishi bunday zararni keltirib chiqara olmaydi, chunki yadro rejimida komponent ishdan chiqishi mumkin. Bunday tizimning asosiy kamchiligi bu foydalanuvchi rejimidan yadro rejimiga qo'shimcha o'tish bilan bog'liq bo'lgan biroz past ko'rsatkichdir. Barcha zamonaviy UNIX tizimlarida X Windows oyna tizimi foydalanuvchi rejimida ishlaydi, natijada unumdorlik biroz pasayadi, lekin katta modullikka erishiladi (Windows dan farqli o'laroq, butun foydalanuvchi grafik interfeysi yadroda joylashgan). O'sha davrning boshqa mashhur mikroyadrolariga Mach (Accetta va boshqalar, 1986) va Horus (Rozier va boshqalar, 1988) kiradi. U joriy qilingandan bir necha oy o'tgach, MINIX tizimi o'z yangiliklar guruhining komp.os.minix va 40000 dan ortiq foydalanuvchilari bilan muhim obyektga aylandi. Ko'p foydalanuvchilar buyruqlar va foydalanuvchi dasturlarini o'zlari yoza boshladilar, shuning uchun MINIX tizimi tezda Internetda ko'p sonli foydalanuvchilarning jamoaviy ijod mahsuli bo'lib, keyinchalik paydo bo'lgan boshqa jamoaviy loyihalar uchun prototip bo'lib xizmat qildi. 1997 yilda MINIX tizimining 2.0 versiyasi ishlab chiqildi. Endi bazaviy tizim tarmoq dasturiy ta'minotini o'z ichiga oldi va uning hajmi 62,200 qatorga yetdi. 2004 yilda, MINIX ning rivojlanish yo'nalishi tubdan o'zgardi, tortishish markazi mutlaqo muammosiz va ishonchli tizimni yaratishga yo'naltirildi, u avtomatik ravishda muvaffaqiyatsizliklardan xalos bo'lib, hattoki takrorlangan dastur xatolariga qaramay to'g'ri ishlashni davom ettirishi mumkin edi. Natijada, 1-versiyadagi modullik g'oyasi MINIX 3.0 versiyasida sezilarli darajada kengaydi,

deyarli barcha qurilma drayverlari foydalanuvchi maydoniga ko'chirildi (va barcha drayverlar alohida jarayonlar sifatida ishlaydi). Yadro hajmi keskin kamaydi (bitta dasturchi tushunishi mumkin bo'lgan 4000 kod qatoriga). Xatolarga bardoshlilikini oshirish uchun ichki mexanizmlar ham o'zgartirildi. Bundan tashqari, 650 dan ortiq mashhur UNIX dasturlari Windows X tizimi (ba'zan shunchaki X deb ataladi), turli xil kompilyatorlar (gcc ni o'z ichiga olgan), matn protsessorlari, tarmoq dasturlari, veb-brauzerlar va boshqalarni o'z ichiga olgan MINIX 3.0-ga ko'chirildi. Oldingi versiyalardan farqli o'laroq (asosan ta'limga oid), MINIX 3.0 versiyasidan boshlab, tizim ancha foydaliroq bo'lib qoldi (asosiy urg'u yuqori ishonchlilikka qaratiladi).

### ***Linux***

MINIX tizimini ishlab chiqish va ushbu tizimni Internetda muhokama qilishning dastlabki yillarida, ko'p odamlar yangi va murakkabroq funksiyalarni so'rashgan (va ko'pincha talab qilinadi) va muallif ko'pincha ushbu so'rovlarni rad etgan (ya'ni talabalar bir semestrda to'liq o'zlashtirishlari uchun tizimning hajmini kichik saqlagan). Ushbu doimiy rad etishlar ko'plab foydalanuvchilarni g'azablantirgan. U paytlarda, hali FreeBSD tizimi yo'q edi. Nihoyat, bir necha yil o'tgach, fin talabasi Linus Torvalds UNIX tizimining yana bir klonini o'zi yozishga qaror qildi, va u Linux deb nomlangan. Bu dastlabki MINIX tizimida yetishmayotgan ko'plab xususiyatlarga ega, to'liq ishlab chiqarish tizimi bo'lishi kerak edi. Linux operatsion tizimining birinchi 0.01 versiyasi 1991yilda chiqarilgan. U MINIX kompyuterida ishlab chiqilgan va qurilgan bo'lib, MINIX tizimidan resurs daraxtining tuzilishidan tortib, fayl tizimining tartibiga qadar ko'plab g'oyalarni o'zlashtirgan. Ammo, MINIX mikroyadro tizimidan farqi, Linux monolit tizim edi, ya'ni butun operatsion tizim yadroda joylashgan edi. Dastlabki manba matnning o'lchami C tilida 9300 satr va assemblerda 950 satr bo'lib, ular hajmi va funksiyasi bo'yicha taxminan MINIX versiyasiga to'g'ri keldi. Aslida, bu MINIX tizimining qayta o'zgartirilgani edi, faqat, Torvalds tizimi manba kodiga ega edi. Linux operatsion tizimi hajmi tezda o'sdi va keyinchalik virtual xotiraga, yanada murakkab fayl tizimiga va boshqa qo'shimcha funksiyalarga ega bo'lgan to'liq UNIX kloniga aylandi. Garchi dastlab Linux tizimi faqat Intel 386 protsessorida ishlagan (va

hatto C protseduralarida 386 protsessor uchun oʻrnatilgan assembler kodi boʻlgan), u tezda boshqa platformalarga koʻchirildi va hozirda UNIX kabi juda koʻp mashinalarda ishlaydi. Linux tizimi va UNIX oʻrtasidagi farqni alohida taʼkidlash kerak: u gcc kompilyatorining koʻplab maxsus xususiyatlaridan foydalanadi, shuning uchun, uni standart ANSI C kompilyatori bilan kompilyatsiya qilish uchun koʻp harakat talab etiladi. Linux tizimi va UNIX oʻrtasida bitta farq bor: u gcc kompilyatorining koʻplab maxsus xususiyatlaridan foydalanadi, shuning uchun uni standart ANSI C kompilyatori bilan kompilyatsiya qilish uchun koʻp harakat talab etiladi. Linux tizimining navbatdagi katta versiyasi 1994 yilda paydo boʻlgan 1.0 versiyasi edi. U taxminan 165000 satr kodidan iborat boʻlib, yangi fayl tizimini, xotira manzili maydonida fayllarni koʻrsatish, va BSD bilan mos keladigan tarmoq dasturlari socketlari va TCP/IP larni oʻz ichiga olgan. Shuningdek, u koʻplab yangi qurilma drayverlarini oʻz ichiga olgan. Keyingi ikki yil ichida kichik xatoliklar tuzatildi. Bu vaqtga kelib, Linux operatsion tizimi UNIX bilan toʻliq moslasha boshladi, shuning uchun unga koʻpgina UNIX dasturlari oʻtkazildi, bu esa uning foydaliligini sezilarli darajada oshirdi. Bundan tashqari, Linux operatsion tizimi uning kodi va kengaytmasi ustida ishlashni boshlagan koʻplab odamlarni jalb qildi (Torvaldsning umumiy rahbarligi ostida). Keyingi yirik ishlanma - 2.0 versiyasi 1996 yilda chiqdi. Ushbu versiya C tilidagi taxminan 470000 satrlardan va 8000 qatorli assembler kodlaridan iborat edi. Bu 64-razraydli arxitektura, simmetrik koʻp vazifali, yangi tarmoq protokollari va boshqa koʻplab funksiyalarni qoʻllab-quvvatlashni oʻz ichiga oldi. Ushbu versiyadan keyin, tez-tez qoʻshimcha ishlanmalar chiqarildi. Linux yadrosining versiyasi toʻrtta raqamdan iborat: A.B.C.D (masalan, 2.6.9.11). Birinchi raqam yadro versiyasini koʻrsatadi. Ikkinchi raqam asosiy versiyani koʻrsatadi. 2.6 yadrosidan oldin, hatto versiya raqamlari yadroning barqaror versiyasini koʻrsatgan, toqlari esa nobarqaror. 2.6 yadrosi versiyasidan boshlab, bunday emas. Uchinchi raqam versiya raqamini koʻrsatadi (masalan, yangi drayverlarni qoʻllab-quvvatlash qoʻshilgan). Toʻrtinchi raqam xatolarni tuzatish yoki xavfsizlikni koʻrsatadi. 2011yil iyul oyida Linus Torvalds Linux 3.0 ning chiqarilishini eʼlon qildi, ammo bu biron bir texnik yaxshilanish tufayli emas, balki yadro rivojlanishining 20 yilligi sharafidir. 2013yil holatiga koʻra Linux yadrosida taxminan 16 million kod satrlari mavjud. Standart UNIX

dasturining asosiy dasturiy ta'minot qismi, X Windows va ko'p sonli tarmoq dasturlari Linux ga o'tkazildi. Bundan tashqari, maxsus Linux uchun ikkita raqobatdosh grafik foydalanuvchi interfeysi yozilgan: GNOME va KDE. Umuman olganda, Linux tizimi UNIX ni yoqtiruvchilar talab qilishi mumkin bo'lgan barcha shov-shuvlarga ega bo'lgan to'liq UNIX kloniga aylandi. Linux ning noodatiy xususiyati bu uning biznes modeli: bu bepul dasturiy ta'minot. Uni turli veb-saytlardan yuklab olish mumkin, masalan, [www.kernel.org](http://www.kernel.org). Linux tizimi Free Software Foundation asoschisi Richard Stallman tomonidan ishlab chiqilgan litsenziya bilan ta'minlangan. Linux tizimi bepul bo'lishiga qaramay, GPL (GNU Public License) deb nomlangan ushbu litsenziya Windows operatsion tizimi uchun Microsoft litsenziyasidan uzunroq va kod bilan nima qila olishingizni va qila olmasligingizni ko'rsatadi. Foydalanuvchilar bepul foydalanishlari, nusxa ko'chirishlari, o'zgartirishlari va manba kodlari va ikkilik fayllarni tarqatishlari mumkin. Asosiy cheklash - ikkilik kodni (Linux yadrosi asosida yaratilgan) manba kodisiz alohida sotish yoki tarqatish. Manba kodlari (matnlar) ikkilik fayllar bilan ta'minlanishi yoki talabga binoan taqdim etilishi kerak. Torvalds tizimning yadrosini yetarlicha nazorat qilib turishiga qaramay, foydalanuvchi darajasidagi ko'plab dasturlar Linux ga dastlab MINIX, BSD va GNU tarmog'idan o'tgan boshqa dasturchilar tomonidan yozilgan. Ammo, Linux tizimi rivojlanib borgan sari, Linux hamjamiyatining kichkina bir qismi boshlang'ich kodni talab qilmoqchi (Linux tizimini qanday o'rnatishni va undan qanday foydalanishni tavsiflovchi yuzlab kitoblar, va kodning o'zi yoki qanday ishlashi haqida munozarali bir nechta kitoblar buning isbotidir). Bundan tashqari, ko'plab Linux foydalanuvchilari tizimni ko'plab tijorat kompaniyalari tomonidan tarqatiladigan CD-ROM-larning birini sotib olish orqali Internetdan bepul yuklab olishni afzal ko'rishmoqda. [www.linux.org](http://www.linux.org) veb-saytida Linux distributivlarini sotadigan 100 dan ortiq kompaniyalar ro'yxati keltirilgan. Shunisi qiziqki, Linux modasi jadal rivojlana boshlaganida, kutilmaganda u AT&T tomonidan qo'llab-quvvatlandi. 1992yilda Berkli Universiteti mablag'ni yo'qotib, BSD UNIX-ni 4.4 BSD ning so'nggi versiyasini (keyinchalik FreeBSD uchun asos bo'lgan) ishlab chiqarishni to'xtatishga qaror qildi.



## *Linux vazifasi*

UNIX operatsion tizimi har doim bir vaqtning o'zida bir nechta jarayonlarni va bir nechta foydalanuvchilarni qo'llab-quvvatlashga mo'ljallangan interfaol tizim bo'lib kelgan. U dasturchilar uchun, dasturchilar tomonidan - foydalanuvchilarning aksariyati juda tajribali va dasturiy ta'minotni ishlab chiqish loyihalarida ishtirok etadigan (ko'pincha juda murakkab) sharoitlarda foydalanish uchun ishlab chiqilgan. Ko'p holatlarda, ko'p sonli dasturchilar umumiy tizimni yaratish uchun faol ishlamoqda, shuning uchun UNIX operatsion tizimida insonlar birgalikda ishlashi va ma'lumot almashishni boshqarishga imkon beradigan ko'plab vositalar mavjud. Yaxshi dasturchilar operatsion tizimdan nimani xohlashadi? Birinchidan, ko'pchilik ularning tizimining sodda va mos kelishini xohlaydi. Masalan, eng past darajada fayl faqat baytlar to'plamidan iborat bo'lishi kerak. Tajribali dasturchilar odatda operatsion tizimda ko'rishni istagan boshqa xususiyatlar bu kuchlilik va moslashuvchanlikdir. Bu shuni anglatadiki, tizimda ularni ma'lum bir dasturga moslashtirish uchun birlashtirilishi mumkin bo'lgan oz miqdordagi asosiy elementlar bo'lishi kerak. Linux tizimining asosiy qoidalaridan biri shundaki, har bir dastur bitta vazifani bajarishi kerak - va uni yaxshi bajarish. Ya'ni, kompilyatorlar ro'yxatlarni tuzishda ishtirok etmaydilar, chunki boshqa dasturlar bu vazifani yaxshiroq bajara oladilar. Va nihoyat, ko'pchilik dasturchilarda foydasiz ortiqchalikga nisbatan kuchli dushmanlik mavjud. Istalgan narsani olish uchun cp ni o'zi yetarli bo'lsa, nima uchun copy deb yozish kerak? Bu qimmatbaho xakerlik vaqtining behuda sarflanishidir. F faylidan "ard" satrini o'z ichiga olgan barcha satrlar ro'yxatini olish uchun Linux operatsion tizimidagi dasturchi quyidagi buyruqni kiritadi.

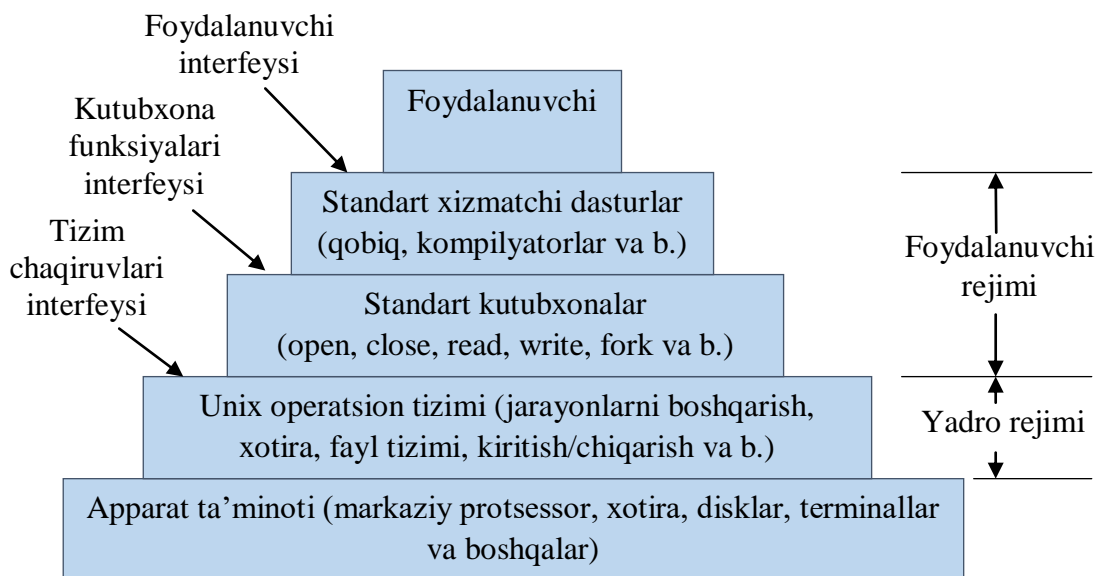
```
grep ard f
```

Qarama-qarshi yondoshuv shundaki, dasturchi avval grep dasturini ishga tushuradi (argumentlarsiz), shundan so'ng grep dasturi dasturchiga: "Salom, men grepman. Men fayllardan namuna izlayapman. Iltimos, namunangizni kiriting". Namunani olgandan so'ng, grep fayl nomini so'raydi. Keyin boshqa fayllar bor-yo'qligini so'raydi. Nihoyat, u nima qilmoqchi ekanligi to'g'risida ma'lumotni ko'rsatib, hamma narsa to'g'ri yoki yo'qligini so'raydi. Bunday

foydalanuvchi interfeysi yangi boshlanuvchilar uchun qulay bo‘lishi mumkin bo‘lsa ham, u tajribali dasturchilarni asabiylashtiradi.

### *Linux tizim interfeysi*

Linux operatsion tizimini piramida deb hisoblash mumkin (9.1-rasm). Piramida bazasida markaziy protsessor, xotira, disklar, monitor va klaviatura, shuningdek, boshqa qurilmalardan iborat. Uning vazifasi qurilmani boshqarish va barcha dasturlarni tizim chaqiriqlari interfeysi bilan ta’minlashdir. Ushbu tizim chaqiriqlari foydalanuvchi dasturlariga jarayonlar, fayllar va boshqa manbalarni yaratish va boshqarish imkonini beradi. Dasturlar argumentlarni registrarga (yoki ba’zida stekka) joylashtirish va foydalanuvchi rejimidan yadro rejimiga o‘tish uchun, emulyatsiya qilingan uzilish buyrug‘ini bajarish orqali tizim chaqiriqlarini amalga oshiradilar. C tilida emulyatsiya qilingan uzilish buyrug‘ini yozish imkonsiz bo‘lgani uchun, bu tizim chaqirig‘ida bitta protseduraga ega bo‘lgan kutubxon tomonidan amalga oshiriladi.



9.1- rasm. Linux operatsion tizimi darajalari

Ushbu protseduralar assemblerda yozilgan, ammo ularni C tilidan chaqirish mumkin. Har bir bunday protsedura avval argumentlarni kerakli joyga qo‘yadi va keyin emulyatsiya qilingan uzilish buyrug‘ini bajaradi. Shunday qilib, o‘qish (read) tizim chaqirig‘iga kirish uchun C dasturlash tilida protsedura

kutubxonasidan read ni chaqirish kerak. POSIX standarti tizim chaqiriqlarining interfeysini emas, balki kutubxona funksiyalarining interfeysini belgilaydi. Boshqacha qilib aytganda, POSIX standarti kutubxona protseduralari uning talablariga javob beradigan tizim nimani ta'minlashi kerakligini, ularning parametrlari qanday bo'lishi kerakligini, nima qilishi kerakligini va qanday natijalar qaytarilishini aniqlaydi. Standartda real tizim chaqiriqlari hatto esga olinmaydi.

Operatsion tizim va tizim chaqiriqlari kutubxonasi ga o'shimcharavi shda,

Linuxning barcha versiyalariko'plab standart dasturlarni taqdim etadi, ularning ba'zilar POSIX 1003.2 standartida ko'rsatilgan, boshqalari esa Linux tizimining turli xil versiyalarida farq qilish mumkin.

Ushbu dasturlarga buyruq protsessori (qobiq), kompilyatorlar, muharrirlar, matnlarni qayta ishlash dasturlari va fayllar bilan ishlash uchun yordamchi dasturlar kiradi. Aynan ushbu dasturlarni foydalanuvchi klaviaturadan ishga tushiradi. Shunday qilib, Linux operatsion tizimidagi uchta interfeys haqida gapirish mumkin: tizim chaqirig'i interfeysi, kutubxona funksiyalari interfeysi va standart yordamchi dasturlar to'plamidan tashkil topgan interfeys. Shaxsiy kompyuterlar uchun eng keng tarqalgan Linux distributivlarida ushbu klaviaturaga asoslangan foydalanuvchi interfeysi sichqonchaga yo'naltirilgan grafikali foydalanuvchi interfeysi bilan almashtirildi, buning uchun tizimning o'zida hech qanday o'zgarishlar talab qilinmadi. Aynan shu moslashuvchanlik Linuxni shunchalik mashhur qildiki, unga asosiy texnologiyadagi ko'plab o'zgarishlardan omon qolishga imkon berdi. Linux tizimining grafik foydalanuvchi interfeysi o'tgan asrning 1970 yillarida UNIX uchun yaratilgan birinchi grafik foydalanuvchi interfeyslariga o'xshaydi va Macintosh kompyuterlari va keyinchalik shaxsiy kompyuterlar uchun Windows tizimi tufayli mashhur bo'ldi. Grafik foydalanuvchi interfeysi ish stoli muhitini yaratadi - bizga tanish bo'lgan oynalar, belgilar, kataloglar, qurilmalar paneli va o'tkazish imkoniyati. To'liq ish stoli muhiti, oynalarning tartibi va tashqi ko'rinishini, shuningdek, turli xil dasturlarni boshqaradigan va izchil grafik interfeysni yaratadigan ko'p oynali ma'murni o'z ichiga oladi. Linux uchun mashhur ish stollari GNOME (GNU Network Object Model Environment) va KDE (K Desktop Environment). Linux grafik foydalanuvchi interfeyslari odatda X11 (yoki oddiygina X) deb nomlanadigan X Windowing

System tomonidan qo'llab-quvvatlanadi. X-server klaviatura, sichqoncha va ekran kabi qurilmalarni boshqaruvchi asosiy komponent bo'lib, kirish dasturini qayta yo'naltirish yoki mijoz dasturidan natijani olish uchun javobgardir. Haqiqiy grafik foydalanuvchi interfeysi muhiti odatda past darajadagi kutubxonaning (xlib) tepasida qurilgan bo'lib, unda X server bilan o'zaro ishlash funksiyalari mavjud. Grafik interfeys X11-ning asosiy funksiyalarini kengaytiradi, oynalarning ko'rishini yaxshilaydi, tugmalar, menyular, belgilar va boshqalarni taqdim etadi. Grafik interfeysdan foydalangan holda Linux tizimlarida foydalanuvchi dasturni ishga tushirish yoki faylni ochish uchun sichqoncha tugmachasini bosishi, fayllarni bir joydan ikkinchi joyga ko'chirish mumkin va hokazo. Bundan tashqari, foydalanuvchilar xterm terminal emulyatsiya dasturini ishga tushirishlari mumkin, bu ularni asosiy operatsion tizim buyruq satrining interfeysi bilan ta'minlaydi.

### ***Qobiq***

Linuxda grafik foydalanuvchi interfeysi mavjud bo'lsada, aksariyat dasturchilar va ilg'or foydalanuvchilar hali ham qobiq deb nomlangan buyruq satri interfeysini afzal ko'rishadi. Ular ko'pincha grafik foydalanuvchi interfeysidan qobiqli bir yoki bir nechta oynalarni ishga tushiradilar va ularda ishlaydilar. Qobiq buyruqlar satri interfeysi foydalanish uchun tezroq, ancha kuchli, kengaytirilishi oson va sichqonchani doimiy ravishda ishlatish zaruratini tug'dirmaydi. U UNIX qobig'iga asoslangan, Bourne qobig'i deb nomlangan (Bourne Shell, Stiv Burne tomonidan yozilgan va keyin Bell Labs tomonidan yozilgan) va aslida Bourne Again SHell uchun qisqartma hisoblanadi. Boshqa ko'plab qobiqlar (ksh, csh va boshqalar) ishlatiladi, ammo bash Linux tizimlarining ko'pchiligida mavjud standart qobiqdir. Qobiq ishga tushirilganda, u ishga tushadi va keyin kiritish uchun so'rovni (odatda foiz yoki dollar belgisi) ko'rsatadi va foydalanuvchidan buyruq satriga kirishni kutadi. Foydalanuvchi buyruq satriga kirgandan so'ng, qobiq undan birinchi so'zni chiqarib tashlaydi, bu bo'shliqli yoki yorliqli belgi bilan ajratilgan belgilar qatorini anglatadi. Qobiq ushbu so'zni ishga tushiriladigan dasturning nomi deb taxmin qiladi, ushbu dasturni qidiradi va agar uni topsa, uni bajarish uchun ishga tushiradi. Bunday holda, qobiq ishga tushirilgan dastur muddati davomida to'xtatiladi.

Dastur ishi tugashi bilan qobiq keyingi buyruqni o‘qishga harakat qiladi. Bu yerda ta’kidlash kerakki, qobiq - bu oddiy foydalanuvchi dasturidir. Unga kerak bo‘lgan narsa – bu klaviaturadan o‘qish va monitorga chiqarish, shuningdek, boshqa dasturlarni ishga tushirish. Buyruqlar argumentlar bo‘lishi mumkin, ular ishga tushiriladigan dasturga matn satrlari ko‘rinishida uzatiladi. Masalan, buyruq satri

```
cp src dest
```

cp dasturini ikkita argument bilan ishlaydi, src va dest. Ushbu dastur birinchi argumentni mavjud faylning nomi sifatida sharhlaydi. U ushbu fayldan nusxa oladi va ushbu nusxani dest deb ataydi. Hamma argumentlar ham fayl nomlari hisoblanmaydi. Satrda

```
head -20 file
```

birinchi argument - 20 **head** dasturga **file** faylining dastlabki 20 satrini chop etishni buyuradi (standart 10 qator o‘rniga). Buyruqlar ishlashini boshqarish yoki qo‘shimcha qiymatlarni ko‘rsatadigan argumentlar bayroqlar deb nomlanadi va kelishuvda chiziqcha (-) belgisi bilan ko‘rsatilgan. Noaniqliklarga yo‘l qo‘ymaslik uchun chiziqcha (-) talab qilinadi, chunki, **head 20file** buyrug‘i qonuniydir. U **head** dasturiga **20** nomli faylning dastlabki 10 satrini chop etishni va keyin ikkinchi **file** faylining dastlabki 10 satrini chop etishga ko‘rsatma beradi. Linux tizimining ko‘pgina buyruqlari bir nechta bayroq va argumentlarni qabul qilishi mumkin. Fayllar guruhini aniqlashni osonlashtirish uchun, qobiq sehrli belgilarni (magic charecters), ba’zida guruh (wild cards) deb ham ataladi. Masalan, “yulduzcha” belgisi, barcha bo‘lishi mumkin bo‘lgan matn satrlarini bildiradi, shuning uchun

```
ls*.c
```

satri **ls** dasturiga ismlari **.c** bilan tugaydigan barcha fayllarni ro‘yxatga olishni buyuradi. Agar x.c, y.c va z.c fayllari bo‘lsa, bu buyruqqa tengdir.

```
ls x.c y.c z.c
```

Boshqa har qanday bir belgini o‘rnini bosadigan guruh belgisi bu so‘rov belgisidir. Bunga qo‘shimcha ravishda, kvadrat qavs ichida siz dasturni tanlashi kerak bo‘lgan juda ko‘p belgilarni belgilashingiz mumkin. Masalan,

```
ls [ape]*
```

buyrug‘i "a", "p" yoki "e" harflari bilan boshlanadigan barcha fayllarni ko‘rsatadi. Qobiq kabi dastur undan o‘qish yoki xulosa qilish uchun terminalni (klaviatura va monitor) ochmasligi kerak. Buning

o‘rniga, ishga tushirilgan dasturlar avtomatik ravishda faylni o‘qish, bu standart kirish (standard input) qurilmasi deb nomlanadi, faylga yozish uchun – standart chiqarish (standard output) qurilmasi, va standart xatoliklar (standard error) haqida xabar berishda faylga yozish huquqiga ega bo‘ladi. Odatda, terminal ushbu uchta qurilmaning barchasiga mos keladi, ya’ni standart kirishdan o‘qish klaviaturadan amalga oshiriladi va standart chiqishga (yoki xatoliklar uchun chiqish uchun) yozish ekranga tushadi. Ko‘p Linux dasturlari standart kirish moslamasidan ma’lumotlarni o‘qiydi va standart chiqish moslamasiga yozadi. Masalan,

```
sort
```

buyrug‘i saralash dasturini ishga tushiradi, u terminaldan satrlarni o‘qiydi (foydalanuvchi fayl oxirini belgilash uchun Ctrl+D tugmalar birikmasini bosmaguncha) va keyin ularni alifbo tartibida tartiblaydi va natijani ekranda namoyish etadi. Standart kirish va chiqish yo‘naltirilishi mumkin, bu juda foydali xususiyatdir. Buning uchun mos ravishda "<" va ">" belgilaridan foydalaniladi. Ularni bir vaqtning o‘zida bitta buyruq satrida ishlatishga ruxsat beriladi. Masalan,

```
sort <in> out
```

buyrug‘i **sort** dasturini **in** faylini kirish sifatida qabul qilishga va chiqishni **out** faylga yo‘naltirishga majbur qiladi. Standart xatoliklar xabari chiqishi yo‘naltirilmagani uchun, keyin barcha xatolik xabarlarini ekranga tushadi. Standart kirish moslamasidan ma’lumotlarni o‘qiydigan dastur ushbu ma’lumotlarga ma’lum bir ishlov berishni amalga oshiradi va natijani filtr deb nomlangan standart chiqish oqimiga yozadi. Uchta alohida buyruqdan iborat quyidagi buyruq satrini ko‘rib chiqamiz:

```
sort <in> temp; head -30 <temp; rm temp
```

Birinchi, **sort** dasturi ishga tushadi, u fayldan ma’lumotlarni oladi va natijani **temp** fayliga yozadi. U o‘z ishini tugatgandan so‘ng, qobiq **head** dasturni ishga tushiradi, unga **temp** fayldan standart terminal bo‘lgan standart chiqish moslamasiga dastlabki 30 qatorni chiqarishni buyuradi. Nihoyat, vaqtinchalik **temp** fayli o‘chiriladi. Shu bilan birga, u butunlay yo‘q qilinadi va endi qayta tiklanmaydi. Buyruqlar satrlari ko‘pincha ishlatiladi, bunda buyruqlar satridagi birinchi dastur ikkinchi dastur kirish sifatida foydalanadigan natijani

yaratadi. Oldingi misolda buning uchun vaqtinchalik **temp** fayli ishlatilgan. Ammo, Linux tizimi buning oson yo'lini ta'minlaydi.

Linux universal ko'p vazifali tizim hisoblanadi. Bitta foydalanuvchi bir vaqtning o'zida bir nechta dasturlardan ularning har biri alohida jarayon sifatida foydalanishi mumkin.

### ***Linux Utilitalari***

Linux foydalanuvchi buyruqlar qatori interfeysi (qobiq) ko'plab yordamchi standart dasturlardan tashkil topgan. Ushbu dasturlarni quyidagi olti toifaga bo'lish mumkin:

1. Fayllar va kataloglarni boshqarish buyruqlari;
2. Filtrlar;
3. Matn muharrirlari va kompilyatorlar kabi dasturlarni ishlab chiqish vositalari;
4. Matn protsessorlari;
5. Tizim boshqaruvi;
6. Har xil.

9.1- jadval

<b>Buyruq</b>	<b>Funksiya</b>
cat	standart fayla bir nechta fayllarni ulash
chmod	faylni himoya qilish rejimini o'zgartirish
cp	faylni nusxalash
cut	fayldan matn ustunlarini kesish
grep	ma'lum bir shablonni fayldan qidirish
head	fayldan birinchi qatorni ajratib olish
ls	katalog ro'yhati
make	ikkilik faylni yaratish uchun fayllarni kompilatsiya qilish
mkdir	katalog yaratish
paste	matnlarni faylga joylashtirish
pr	chop etish uchun faylni formatlash
rm	faylni o'chirish
rmdir	katalogni o'chirish
sort	fayl satrlarini alifbo bo'yicha tartiblash
tail	fayldan ohirgi satrlarni chiqarish
tr	Belgilarni bir to'plamdan boshqasiga o'zgartirish

POSIX 1003.1-2008 standarti ushbu dasturlarning taxminan 150 tasining sintaksisi va semantikasini aniqlaydi, asosan birinchi uchta toifaga kiradi. Ushbu dasturlarni standartlashtirish g'oyasining maqsadi barcha Linux tizimlarida ishlaydigan qobiq skriptlarini yozish imkoniyatiga ega bo'lishdir. Ushbu standart yordamchi dasturlarga qo'shimcha ravishda veb-brauzerlar, multimedia fayllarini ishga tushuruvchilar, rasmlarni ochish dasturlari, ofis to'plamlari va boshqa shu kabi ko'plab dasturlar mavjud. Fayllar va kataloglarni boshqarish buyruqlari 9.1- jadvalda keltirilgan. Masalan, quyidagi buyruqni ko'rib chiqamiz:

```
cp a b
```

Ushbu buyruq asl faylni o'zgartirmasdan, a faylni b fayliga ko'chirib o'tkazadi.

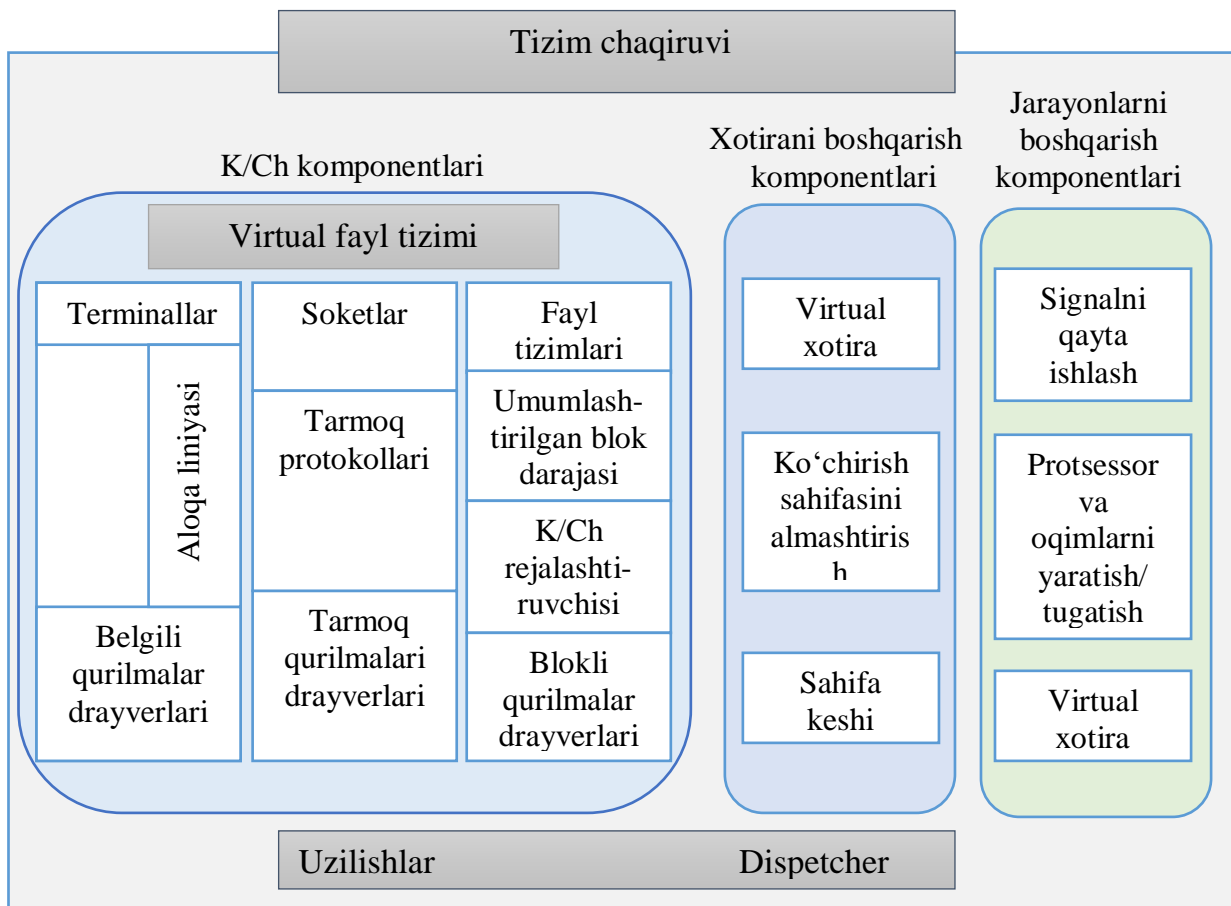
### ***Yadro tuzilishi***

9.1- rasmda Linux operatsion tizimining umumiy tuzilishi ko'rsatilgan. Endi tizimning yadrosini batafsil ko'rib chiqaylik (jarayonlarni va fayl tizimini rejalashtirishni o'rganishni boshlashdan oldin). Yadro to'g'ridan-to'g'ri qurilma ta'minoti bilan ishlaydi va K/Ch qurilmalari va xotira boshqaruv bloki bilan o'zaro aloqani ta'minlaydi, shuningdek, protsessorning ularga kirishini nazorat qiladi. Yadroning pastki darajasi (9.2- rasm) past darajadagi uzilishlarni ishlov beruvchi va dispetcherlash mexanizmidan iborat. Dispetcherlash uzilish sodir bo'lganda amalga oshiriladi. Shu bilan birga, past darajadagi kod jarayonning bajarilishini to'xtatadi, holatini yadro jarayonlari tarkibida saqlaydi va tegishli drayverni ishga tushiradi. Jarayonni dispetcherlash yadro ma'lum bir operatsiyani tugatgandan so'ng amalga oshiriladi va foydalanuvchi jarayonini yana ishga tushurish vaqti keladi.

Keyin turli xil yadro quyi tizimlarini uchta asosiy komponentlarga ajratilgan. 9.2- rasmdagi kiritish/chiqarish komponentlari yadroning barcha qismlarini, qurilmalar bilan o'zaro ishlashni, shuningdek, tarmoq operatsiyalari va tashqi qurilmalarga kiritish/chiqarish operatsiyalarini bajarishni o'z ichiga oladi. Eng yuqori darajada, barcha kiritish/chiqarish operatsiyalari Virtual fayl tizimi (VFS- Virtual File System) darajasida birlashtirilgan. Eng yuqori darajada faylni o'qish jarayoni (u xotirada yoki diskda bo'ladimi) - bu terminal kirishidan belgilar o'qish operatsiyasini



bajarish bilan bir xil. Eng past darajada, barcha kiritish/chiqarish operatsiyalari biron bir qurilma drayveridan o'tadi. Linux ning barcha drayverlari belgili qurilma drayverlari yoki blokli qurilma drayverlari deb tasniflanadi, asosiy farqi shundaki qidirish va tasodifiy kirishga faqat blokli qurilmalar uchun ruxsat beriladi. Texnik nuqtai nazardan, tarmoq qurilmalari belgili qurilmalar, ammo ular bilan ishlash biroz farq qiladi, shuning uchun ularni ajratib ko'rsatish yaxshiroq (9.2-rasmda ham ajratib ko'rsatilgan).



9.2- rasm. Linux operatsion tizimi yadrosi tuzilishi

### ***Linux tizimida jarayonlar***

Linux tizimidagi asosiy faol obyektlar - bu jarayonlar. Har bir jarayon bitta dasturni bajaradi va dastlab bitta boshqaruv oqimini oladi. Boshqacha qilib aytganda, jarayonda keyingi bajariladigan ko'rsatmalarni kuzatib boradigan bitta ko'rsatma hisoblagichi mavjud. Linux qo'shimcha jarayonlarni yaratishga imkon beradi (bajarishni boshlaganidan keyin). Linux - ko'p vazifali tizim va bir vaqtning

o'zida bir nechta mustaqil jarayonlar ishlashi mumkin. Bundan tashqari, har bir foydalanuvchi bir vaqtning o'zida bir nechta faol jarayonlarga ega bo'lishi mumkin, shuning uchun bir vaqtning o'zida yuzlab va hatto minglab jarayonlar katta tizimda ishlashi mumkin. Aslida, bitta foydalanuvchi ishchi stansiyalarida o'nlab fon jarayonlari mavjud, ular demonlar (daemons) deb nomlanadi. Ular tizim qobig'idan ishga tushganda boshlanadi. Oddiy demon - bu cron. Ushbu demon sizga Linux tizimingizni bir necha daqiqa, soat, kun va hatto oylar davomida rejalashtirishga imkon beradi. Masalan, foydalanuvchi keyingi seshanba kuni soat 3 da tish shifokoriga borishi rejalashtirilgan deb tasavvur qilamiz. Belgilangan kun va vaqt kelganda, cron demon o'z ishini ko'radi va o'z vaqtida ovozli signal dasturini (yangi jarayon shaklida) boshlaydi. Cron demon shuningdek, vaqti-vaqti bilan vazifalarni bajarish uchun ishlatiladi, masalan, har kuni soat 4.00 da diskdan nusxa olish. Demonlar Linux tizimida juda sodda tarzda amalga oshiriladi, chunki ularning har biri boshqa jarayonlarga qaramasdan alohida jarayon hisoblanadi. Linux operatsion tizimida jarayonlar yaratish juda oddiydir. Fork tizim chaqiruvchi asl jarayon nusxasini yaratadi, ona jarayoni (parent process) deb ataladi. Yangi jarayon farzand jarayoni (child process) deb ataladi. Ona va farzand jarayonlari o'zlarining shaxsiy xotira maydoniga ega. Agar ona jarayoni keyinchalik biron bir o'zgaruvchini o'zgartirsa, unda bu o'zgarishlar farzand jarayoniga ko'rinmaydi (va aksincha). Ochiq fayllar ona va farzand jarayonlari o'rtasida almashiladi. Bu shuni anglatadiki, agar biron-bir fayl fork tizim chaqirig'idan oldin ona jarayonida ochilgan bo'lsa, kelajakda ikkala jarayonda ham ochiq qoladi. Ushbu faylga biron bir jarayon tomonidan kiritilgan o'zgartirishlar boshqasiga ko'rinadigan bo'ladi. Bu xatti-harakatlar yagona oqilona usuldir, chunki bu o'zgarishlar ushbu faylni ochadigan har qanday boshqa jarayon uchun ham ko'rinadi.

```
Linux operatsion tizimida jarayonni yaratish
pid = fork( ); /* agar fork muvaffaqiyatli tugatilsa, pid > 0
ona jarayonida */
if (pid < 0) {
    handle_error(); /* fork muvaffaqiyatsiz tugadi (masalan, xotira
yoki qandaydir jadval to'lib qolsa) */
} else if (pid > 0) { /* bu yerda ona kodi joylashgan */
```

```
} else {          /* bu yerda farzand kodi joylashgan */  
}
```

Jarayonlar o‘zlarining PID-lariga ega bo‘ladilar (PID-jarayon identifikatori). Yuqorida aytib o‘tilganidek, jarayon yaratilganda, uning PID-si yangi jarayonning onasiga beriladi. Jarayon identifikatorlari turli xil usullarda qo‘llaniladi. Masalan, farzand jarayoni tugaganda, uning onasi yangi tugatilgan farzand jarayonining PID-ni oladi. Bu juda muhim bo‘lishi mumkin, chunki ona jarayonida ko‘plab farzand jarayonlari bo‘lishi mumkin. Farzand jarayonning ham farzand jarayonlari bo‘lishi mumkin. Linux tizimida jarayonlar bir-birlari bilan xabarlarining biron bir shaklidan foydalangan holda aloqa o‘rnatishi mumkin. Siz ikkita jarayon o‘rtasida kanal yaratishingiz mumkin, bunda bitta jarayon baytlar oqimini yozishi mumkin, boshqa jarayon esa o‘qishi mumkin. Ushbu kanallar ba‘zan kanallar (pipes) deb ataladi. Jarayonni sinxronizatsiya qilish bo‘sh kanaldan ma‘lumotlarni o‘qiyotganda jarayonni blokirovka qilish orqali amalga oshiriladi. Kanalda ma‘lumotlar paydo bo‘lganda, jarayon blokdan chiqariladi. Jarayonlar boshqa usulda - dasturiy uzilishlardan foydalangan holda aloqa o‘rnatishi mumkin. Jarayonlar tizimga kiruvchi signal kelganida qanday harakatlar qilish kerakligini aytishi mumkin.

### ***Linuxda jarayonlarni boshqarish tizim chaqiruvi***

Qobiqdan foydalanganda, farzand jarayoni foydalanuvchi tomonidan kiritilgan buyruqni bajarishi kerak. U buni exec tizim chaqiruvi bilan amalga oshiradi, bu butun xotira nusxasini birinchi parametrdan ko‘rsatilgan fayl tarkibiga almashtiradi.

Umumiy holatda, exec tizim chaqiruvi uchta parametrga ega: bajariladigan fayl nomi, massiv argumentlari ko‘rsatgichi va massiv qatorlari ko‘rsatgichi. Execl, execv, execl va execl kabi turli xil kutubxona protseduralari sizga ba‘zi parametrlarni o‘tkazib yuborish yoki ularni boshqa yo‘llar bilan belgilashga imkon beradi. Ushbu protseduralarning barchasi bir xil tizim chaqirig‘iga murojaat qiladi.

### ***Linux da jarayonlarni boshqarish***

UNIX ning klassik tizimda jarayonlarni boshqarish vositalari jarayonni yaratish va yangi dasturni ishga tushurishni ikki turdagi amallarga ajratadi:

- ❖ fork tizim chaqiruvchi yangi jarayonni yaratadi;
- ❖ yangi dastur exec tizim chaqiruvchi yordamida ishga tushiriladi.

UNIX da jarayon har bir alohida dastur konsepsiyasini amalga oshirishni qo‘llab-quvvatlash uchun zarur bo‘lgan barcha axborot bor. Linux tizimida jarayonning xususiyatlari uch guruhga bo‘linadi: jarayonni, uni o‘rab turgan muhit va kontekstni identifikatsiyalash. Jarayon identifikatori (PID) – bu jarayonning yagona identifikatori (son); ilova boshqa jarayon uchun signal, modify yoki wait tizim chaqiruvchi bajarilayotganida operatsion tizimida jarayonlarga buyruqlar berish uchun foydalaniladi.

**Vakolatlar (Credentials).** Har bir jarayon unga bog‘liq foydalanuvchi identifikatori va bir yoki undan ko‘p identifikatorlar guruhiga ega bo‘lishi kerak.

**Shaxsni identifikatsiyalash (Personality).** UNIX turidagi tizimlarda bu an’anaviy bo‘lmasa ham, Linux da har bir jarayon tizim chaqiruvchilarning semantik qatorini ma’lum tarzda modifikatsiya qilishga imkon beruvchi yagona shaxsiy identifikatorga ega. U asosan emulyatsiya kutubxonalarida tizim chaqiruvchilarni u yoki bu maxsus UNIX versiyalari (dialekti) bilan mosligi bo‘yicha so‘rov uchun foydalaniladi. Jarayonning muhiti ona jarayondan olinadi; nollar bilan tugatiladigan ikki vektordan iborat:

- ❖ argumentlar vektori – bajarilayotgan dasturni chaqirishda foydalaniladigan buyruqlar qatorining argumentlari ro‘yxatidan iborat; an’anaviy tarzda dasturning o‘zini nomi bilan boshlanadi;

- ❖ muhit vektori – o‘zgaruvchan muhitlarni berilgan nomlar va ularning ixtiyoriy matn qiymatlarini bog‘lovchi "name=value" juftlar ro‘yxati.

Muhitlar o‘zgaruvchilarini jarayonlar o‘rtasida uzatish va bu o‘zgaruvchilarni farzand jarayonlar tomonidan qabul qilish (meros qilib olish) – bu imtiyozsiz rejimda ishlaydigan tizim dasturiy ta’minotining komponentlariga ma’lumotlarni uzatishning moslashuvchan vositalaridir. Muhit o‘zgaruvchilarining mexanizmi butun tizimni sozlash orqali emas, balki, har bir jarayon uchun alohida o‘rnatilishi mumkin bo‘lgan OTni sozlash vositalarini taqdim etadi.

Jarayon konteksti – bu bajarilayotgan dasturning vaqtning har bir onida doimiy o‘zgaruvchan holatidir. Rejalashtirish konteksti – bu jarayon kontekstining eng muhim qismi; bu rejalashtiruvchi jarayonni

vaqtincha uzilish va ishga tushurish uchun foydalanadigan ma'lumot. Yadro har bir jarayon tomonidan har bir onda foydalanadigan resurslar va har bir jarayon tomonidan uni yaratilishdan boshlab to hozirgi vaqtigacha foydalanilgan resurslarning umumiy hajmi to'g'risida statik ma'lumotlar saqlanishini qo'llab-quvvatlaydi. Fayllar jadvali – bu tizim fayl tuzilmalariga ko'rsatkichlar vektori. Kiritish/chiqarish uchun tizimli chaqirishlar bajarilganida jarayonlar fayllar jadvalidagi indekslar yordamida bu tuzilmalarga havola etadi. Fayllar jadvalida ochiq fayllar ro'yxati saqlanadigan bo'lsa, fayl tizimining konteksti yangi fayllarni ochish uchun so'rovlar uchun qo'llaniladi. Bu yerda fayllarni izlash uchun joriy ildiz (root) katalog va ishchi (default) kataloglarga havolalar saqlanadi. Signallarga ishlov beruvchilar jadvali tegishli signallar paydo bo'lganida chaqirilishi mumkin bo'lgan, jarayonning manzil maydonidagi kichik dasturlarni belgilaydi. Jarayonning virtual xotirasi konteksti uning shaxsiy manzil maydonining barcha tarkiblarini belgilaydi.

### ***Jarayonlar va oqimlar***

Linuxda jarayon va oqimlar uchun bitta umumiy tasavvurdan foydalaniladi. Linux dagi oqim – bu umumiy manzil maydonini ona jarayon bilan birga foydalanadigan yangi jarayon. Farqi faqat clone tizim chaqiruvi bilan yangi oqim yaratilishida namoyon bo'ladi:

- ❖ fork klassik tizim chaqiruvi batamom yangi kontekstli, yangi jarayonni yaratadi;

- ❖ clone tizim chaqiruvi o'zining yangi shaxsiy identifikatoriga ega yangi jarayonni tashkil etadi, lekin unga o'zining ona jarayoni bilan birga ma'lumotlar tuzilmasidan birgalikda foydalanishga ruxsat berilgan.

Clone tizim chaqiruidan foydalanish jarayonlarga oqimlar tomonidan qanday resurslardan birgalikda foydalanishi ustidan ochiqdan-ochiq nazorat qilishga imkon beradi.

### ***Yadro vazifalarini rejalashtirish va yadroda sinxronlash***

Eslatib o'tamiz, rejalashtirish – bu operatsion tizim protsessori vaqtining turli vazifalar o'rtasida taqsimlash. OT larning aksariyatida rejalashtirish deyilganida jarayonlarni ishga tushurish va vaqtinchalik uzilishlar nazarda tutilsa, Linux da rejarashtirishga yadroning turli vazifalarini bajarish ham kiritiladi. Yadro vazifalarini bajarish o'z

ichiga ham mazkur jarayon tomonidan soʻralgan maʼlumotlar boʻyicha vazifalar, hamda drayver ishlashi jarayonida bajariladigan vazifalar ham kiradi. Yadro rejimida bajarish boʻyicha soʻrov ikki vaziyatda paydo boʻlishi mumkin:

- ❖ bajarilayotgan dastur ochiqdan-ochiq tizim soʻrovi yordamida yoki pinhoniyl, masalan, sahifa buzilishida OT ning xizmatini soʻrashi mumkin;

- ❖ qurilma drayveri apparat uzilishlarini ishga tushurishi mumkin, natijada protsessor yadro rejimida mazkur uzilishning ishlov beruvchisi vazifasini bajarishni boshlashi mumkin.

Yadroda sinxronlash yadroning muhim boʻlimlari boshqa muhim boʻlimlar tomonidan uzilishlarsiz bajarilishini talab qiladi.

Linux muhim boʻlimlarini himoyalashning ikki usulidan foydalanadi:

1. Yadroning oddiy kodi – toʻxtatilmaydi. Agar jarayon yadroning tizimli xizmat koʻrsatish kichik dasturi bajarilayotgan paytida vaqt boʻyicha uzilish qabul qilingan boʻlsa, need\_resched bayrogʻi tizim chaqiruvi tugashi va boshqaruv imtiyozsiz kodga uzatilishi kerak boʻlganida rejalashtiruvchi ishga tushishi uchun koʻrsatma sifatida xizmat qiladi.

2. Ikkinchi usul uzilishlarga ishlov berish servislarida bajariladigan yadroning muhim boʻlimlariga nisbatan qoʻllaniladi. Muhim boʻlimni bajarilishi paytida uzilishlarni oʻchirish uchun protsessorning uzilishlarni boshqaradigan apparaturadan foydalanib, yadro maʼlumotlarning umumiy tuzilmalariga bir vaqtda murojaat qilishining xavfsiz bajarilishini kafolatlaydi.

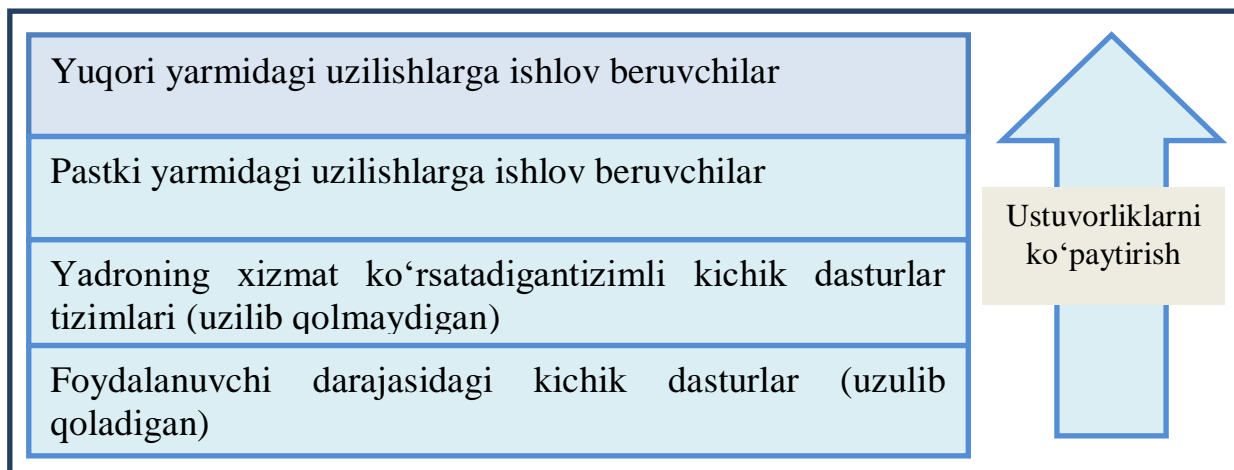
Unumdorlikda yoʻqotishlarga yoʻl qoʻymaslik uchun Linux yadrosi muhim boʻlimlar koʻpchiligini muhim boʻlim bajarilishini toʻla vaqtiga uzilishlarni oʻchirish zarurligi boʻlmasdan, bajarilishiga imkon beruvchi sinxronlash arxitekturasidan foydalaniladi.

Uzilishlarga ishlov berish xizmatlari yuqori yarmiga (top half) va pastki yarmiga (bottom half) boʻlinadi:

- ❖ yuqori yarmi – bu rekursiv uzilishlarni oʻchirish tarzda bajariladigan uzilishlarga ishlov berishning oddiy protsedurasi;

- ❖ pastki yarmi – uzilishlar rejimi yoqilgan holda, kichik yarimlar bir-birini toʻxtatmasligini taʼminlovchi mini-rejalashtiruvchidan foydalanib bajariladi.

Bu arxitektura yadroning oddiy kodi bajarilishida pastki yarimlarni tanlash mexanizmi bilan to'ldiriladi. 9.3- rasmda uzilishlarni himoyalashning darajalari ko'rsatilgan.



9.3- rasm. Uzilishlarni himoyalashning darajalari

Har bir darajaning kodi yuqoriroq darajali kod bilan to'xtatilishi mumkin, lekin teng yoki pastroq darajali kod bilan hech qachon to'xtatilmaydi.

Foydalanuvchi jarayoni doimo boshqa jarayon bilan to'xtalishi mumkin, agar uzilish vaqtni taqsimlash rejimida rejalashtiruvchi tomonidan amalga oshirilsa.

Linux jarayonlar rejalashtirishning ikki algoritmidan foydalanadi:

- ❖ jarayonlar o'rtasida uzilishlarni teng huquqli ravishda rejalashtirish uchun vaqtni taqsimlash algoritmi;

- ❖ teng huquqlikka nisbatan absolyut ustuvorlik muhimroq bo'lishi hollari uchun real vaqtli algoritmi.

Jarayonni rejalashtirish sinfi aynan qaysi algoritm qo'llanilishini belgilaydi. Vaqtni taqsimlash jarayonlari uchun Linux ishonch (credits) asosidagi ustuvorliklar (priority) qo'llaniladigan algoritmdan foydalanadi.  $credits := credits / 2 + priority$  qoidasi ham jarayon tarixini, ham uning ustuvorligini inobatga oladi. Bunday tizim bo'yicha interaktiv yoki kiritish/chiqarishni bajaruvchi jarayonlarning ustuvorliklari avtomatik ravishda belgilanadi. Linux rejalashtirishda quyidagi algoritmlardan foydalanadi: FIFO va Round Robin; ikkala vaziyatda faqat ayrim rejalashtirish klassi emas, balki har bir jarayon

ustuvorlikka ega. Rejalashtiruvchi jarayonni eng yuqori ustuvorlik bilan ishga tushiradi; teng ustuvorlikka ega jarayonlaridan eng ko'p vaqt kutib qolgan jarayon bajariladi. FIFO – jarayonlar tugaguncha yoki blokirovka qilinguniga qadar bajariladi. Round-robin – jarayon ma'lum vaqt o'tgach to'xtatiladi va rejalashtirish navbatining oxiriga joylashtiriladi; shuning uchun teng ustuvorlikka ega. RR-jarayonlar vaqtni avtomatik tarzda o'zaro taqsimlaydi. Linux 2.0 versiyasi SMP qurilmalarni qo'llab-quvvatlovchi Linux ning birinchi yadrosi bo'lgan; turli jarayonlar yoki oqimlar bir nechta protsessorlarda parallel ravishda bajarilishi mumkin. Yadroning uzilishsiz bajarish talabiga rioya qilish uchun SMP quyidagi cheklashni o'rnatadi: har bir onda bittadan ko'p bo'lmagan jarayon yadro rejimida kodni bajarishi mumkin.

### ***Linux da fizik xotirani boshqarish***

Linux da fizik xotirani taqsimlash tizimi xotiraning sahifalari, sahifalar guruhleri va uncha katta bo'lmagan bloklarini joylashtirish va bo'shatish bilan shug'ullanadi. U bajarilayotgan jarayonlarining manzil maydonida aks ettiriladigan virtual xotira uchun qo'shimcha mexanizmlarga ega. Sahifalarni taqsimlovchi fizik sahifalarni joylashtiradi va bo'shatadi; u so'rov bo'yicha yondosh sahifalar guruhlarini ham joylashtirishi mumkin. Linux ning sahifalar taqsimlovchisi buddy-heap (hamkorlik to'plami) algoritmini foydalanidigan fizik sahifalarni kuzatish uchun ishlatadi; uning tamoyillari quyidagicha:

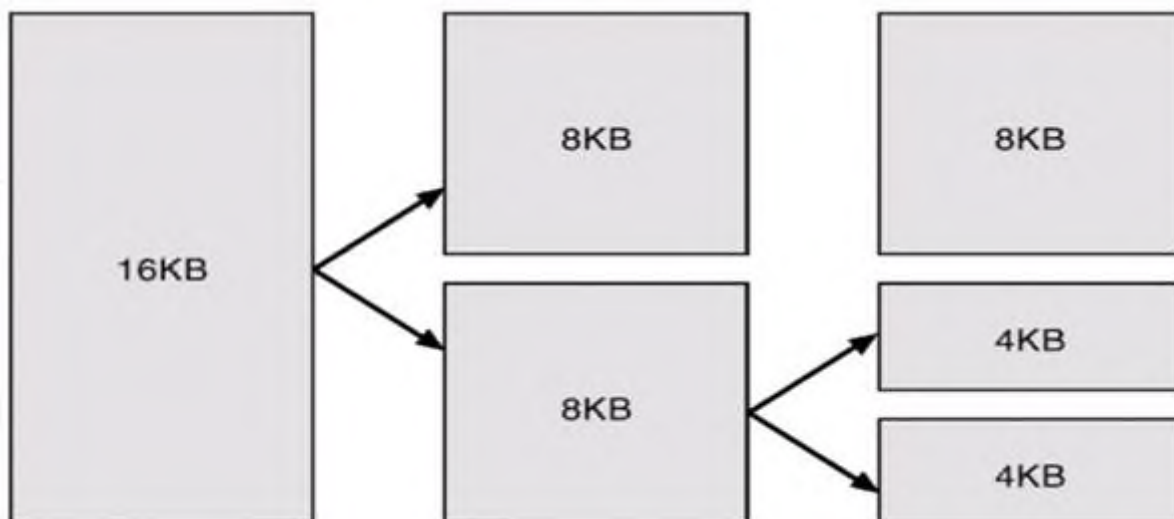
- ❖ taqsimlanishi lozim bo'lgan xotiraning har bir qismi uning yondosh "hamkori" bilan birga juftlikni tashkil etadi;

- ❖ ikkala qism-hamkorlar bo'shatilganida ular birlashadi va ikki barobar kattaroq yondosh hududni tashkil etadi.

Agar xotiraga kichik so'rovni qondirishga xotiraning kichik maydoni bo'lmasa, unda mazkur so'rovni qondirish maqsadida xotiraning kattaroq maydoni ikkita hamkor-maydonga bo'linadi.

Xotirani Linux ning yadrosi uchun bo'linish ham statik (drayver tizimni yuklash paytida xotiraning statik hududini zahiralaydi) ham dinamik (sahifalarni taqsimlash yordamida) tarzda amalga oshirish mumkin. Xotirani bo'linishi 9.4- rasmda ko'rsatilgan.





9.4- rasm. Linux tizimida xotirani bo‘linishi

### *Linux da virtual xotira*

Linux dagi virtual xotira tizimi har bir jarayonga ko‘rinadigan manzil maydonni qo‘llab-quvvatlaydi: u talab bo‘yicha virtual xotiraning sahifalarini tashkil etadi va bu sahifalarni diskka yuklashni, hamda kerak bo‘lganda diskdan qayta chiqarishni boshqaradi. Virtual xotiraning menejeri har bir jarayonning manzil maydoniga ikki nuqtai nazarni qo‘llab-quvvatlaydi:

- ❖ mantiqiy – manzil maydonini boshqarish buyruqlarini qo‘llab-quvvatlash; manzil maydoni kesishmaydigan yondosh hududlar jamlamasi sifatida qaraladi;

- ❖ fizik – har bir jarayonning sahifalar jadvali yordamida.

Virtual xotirani boshqarish uchun quyidagilardan foydalaniladi:

- ❖ mazkur hududga qayerdan sahifalar olinishini tavsiflovchi qayta chiqarish fayli (backing store);

- ❖ hududlar odatda fayl tomonidan qo‘llab-quvvatlanadi, yoki umuman qo‘llab-quvvatlanmaydi (xotira talabiga ko‘ra bekor qilinadi);

- ❖ hududni yozuvga javob harakati (sahifalardan birgalikda foydalanish yoki yozib olish vaqtida nusxa olish - cow).

Yadro ikki vaziyatda yangi virtual manzil maydonni yaratadi:

- ❖ jarayon exec tizim chaqiruvi bilan yangi dasturni ishga tushiradi;

- ❖ fork tizim chaqiruvi bilan yangi jarayonni yaratadi.

Yangi dastur bajarilishida jarayonga yangi, bo'sh manzil maydoni tadqim etiladi; dasturlarni yuklash protseduralari bu manzil maydoni virtual xotiraning hududlari bilan to'ldiradi. Fork yordamida yangi jarayonni yaratish amaldagi manzil maydonining to'liq nusxasini yaratishni nazarda tutadi. Yadro ona jarayonning virtual xotirasidan foydalanish deskriptorlaridan nusxa oladi, keyin farzand jarayon uchun sahifalar jadvallarining yangi jamlamasini yaratadi. Ona jarayonning sahifalar jadvallaridan bevosita farzand jarayonning sahifalar jadvaliga nusxalar olinadi. fork bajarilganidan keyin ona va farzand jarayonlar o'z virtual manzili maydonlarida bir xil fizik sahifalaridan foydalanadi. Agar fizik xotiraning sahifalarini birorta boshqa maqsad uchun kerak bo'lsa, sahifalarni boshqarish tizimi ularni diskka ko'chiradi (o'tkazadi). Sahifalarni boshqarish tizimi ikki qismga bo'linadi:

- ❖ qaysi sahifalarni va qachon diskka ko'chirishni belgilaydigan algoritmi;

- ❖ almashtirish mexanizmi, kerak bo'lgan hollarda ma'lumotlarni uzatish va fizik xotiraga qayta ko'chirishni amalda bajaradi.

Linux yadrosi har bir jarayonga, uni o'ziga tegishli ichki foydalanish uchun doimiy, hudud arxitekturasiga bog'liq manzil maydonni zahiralaydi. Virtual xotiraning bu hududi ikki hududdan iborat:

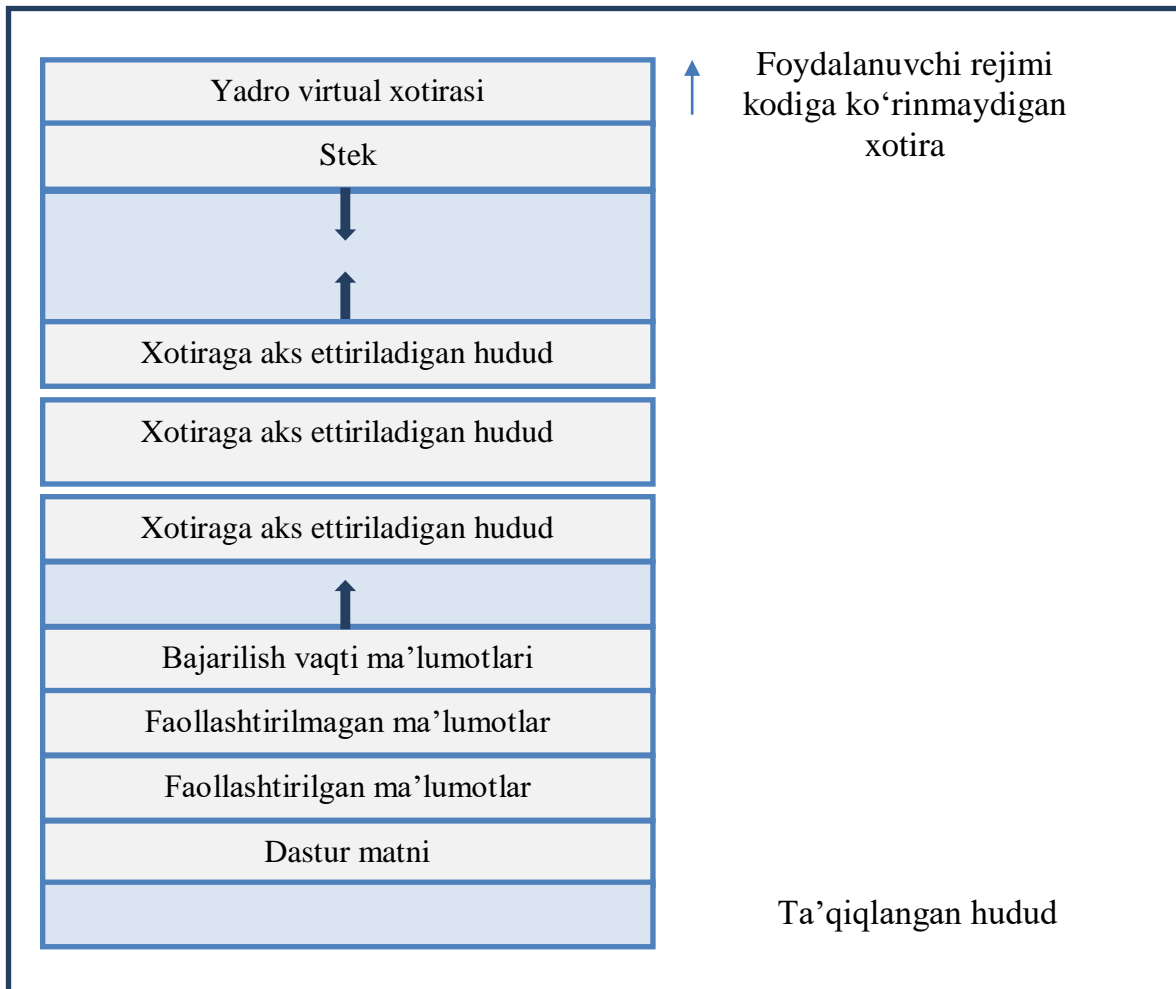
- ❖ statik hudud – tizimdagi xotiraning har bir foydalanadigan fizik sahifa uchun sahifalar jadvalida havolani saqlaydi, shuning uchun yadro kodi bajarilishida fizik manzilni virtual manzilga translyatsiya qilishda ishlatiladi;

- ❖ zahiralangan qismning qoldig'i – birorta boshqa maqsadda ishlatilmaydi; uning sahifalar jadvalining elementlari o'zgartirilishi va xotiraning istalgan sahifalariga yo'naltirilishi mumkin.

### ***Linux da foydalanuvchi dasturlarini bajarish va yuklash***

Linux dasturlarni yuklash funksiyalar jadvalini qo'llab-quvvatlaydi. Exec tizim chaqiruvi bajarilganida, har bir funksiyaga mazkur faylni yana bir marta yuklashga imkon beriladi. Ko'p marotaba dasturlarni ro'yhatdan o'tkazish Linux ga ham ELF, ham a.out kodlar formatlarini qo'llab-quvvatlashga imkon beradi. Dastlab ikkilik fayl sahifalari virtual xotiraga yuklanadi; agar dastur mazkur

sahifaga kirishga urinayotgan bo'lsa va sahifa buzilishi ro'y bersa (sahifa ishlamas), natijada mazkur sahifa fizik xotiraga yuklanadi. ELF (Executable and Linkage Format) formatli ikkilik fayl sarlavhadan iborat; undan keyin bir nechta sahifagacha tekislangan bo'limlar joylashadi. ELF yuklovchi sarlavhani o'qiydi va faylning bo'limlarini virtual xotiraning ma'lum hududlarida aks ettiradi. Linux da ELF-dasturlar uchun xotiraning taqsimlanishi 9.5- rasmda ko'rsatilgan.



9.5- rasm. ELF-dasturlar uchun xotiraning taqsimlanishi

Linux yadrosi moduli uchun ELF fayl bo'limining tuzilishi 9.2- jadvalda keltirilgan.

9.2- jadval

ELF fayl bo'limining tuzilishi	
.text	yo'riqnomalar
.fixup	bajarish vaqtining o'zgarishlari
.init.text	modulni initsializatsiya qilishning yo'riqnomasi
exit.text	modulning yakunlovchi yo'riqnomalari

.rodata.etrl.l	faqat o‘qish uchun qatorlar
.modinfo	modul makroslarining matni
__versions	modul versiyasi to‘g‘risidagi ma’lumotlar
.data	faollashtirilgan ma’lumotlar
.bss	faollashtirilmagan ma’lumotlar
other	boshqalar

### ***Statik va dinamik bog‘lanish***

Kutubxonaning funksiyalari bevosita ikkilik bajariladigan kodga o‘rnatilgan dasturlar o‘zining kutubxonalari bilan birga statik bog‘lanadi. Statik bog‘lanishning asosiy kamchiligi shundaki, barcha yaratilgan dastur bir xil umumtizim kutubxona funksiyalarini nusxalarini saqlashi kerak.

Dinamik bog‘lanish terminallarda ham fizik, ham diskli xotiradan foydalanishda samaraliroq, chunki u tizim kutubxonalarini xotiraga faqat bir marta yuklaydi.

### ***Linux fayl tizimlari***

Foydalanuvchi nuqtai nazaridan Linux fayl tizimi UNIX semantikasiga bo‘ysunadigan katalogning ierarxik daraxti ko‘rinishda ifodalanadi. Ichki nuqtai nazardan yadro bajarilish tafsilotlarini yashiradi va ko‘p turli fayl tizimlarini abstraksiyaning umumiy darajasi, ya’ni virtual fayl tizimi (VFS) orqali boshqaradi.

Linux VFS obyektga yo‘naltirilgan tamoyil bo‘yicha loyihalashtirilgan va fayllar tuzilishini belgilaydigan ta’riflar to‘plamidan foydalanadi. inode-object va file-object tizim tuzilishlari alohida fayllarni namoyon etadi. File system object obyektini butun fayl tizimini namoyon qiladi. Bunday obyektlarni manipulyatsiya qilish uchun mavhumlik (abstraksiya) darajasi mavjud. Ext2fs fayl tizimi – bu Linux ning asosiy fayl tizimi. U ma’lum faylga tegishli ma’lumotlar blokini izlash uchun UNIX BSD Fast File System (ffs) ga o‘xshash mexanizmdan foydalanadi. Ext2fs va ffs o‘rtasidagi asosiy farqlar ularning diskli xotiraning taqsimlash siyosati bilan bog‘liq. ffs tizimida disk 8Kb li bloklardan iborat fayllarga bo‘linadi, bloklar esa kichik fayl yoki faylning oxirida qisman to‘ldirilgan bloklarni saqlash uchun 1Kb hajmli qismlarga (fragmentlarga) bo‘linadi. Ext2fs tizimi bo‘linishlardan foydalanmaydi; u xotirani maydaroq birliklar bilan

taqsimlaydi. Ext2fs da blokning hajmi standart 1Kb ga teng, lekin 2Kb va 4Kb bloklar ham qo‘llab-quvvatlanadi.

Ext2fs tizimi diskdagi fizik ulashgan yondosh bloklarga faylning mantiqiy ulashgan yondosh bloklarini joylashtirish maqsadida, loyihalangan taqsimlash siyosatlaridan foydalanadi, bunda bitta kiritish/chiqarish amalini bir nechta ulashgan yondosh bloklarda amalga oshirish mumkin bo‘ladi. Ext2fs fayllar tizimining tuzilish sxemasi 9.3- jadvalda ko‘rsatilgan.

9.3- jadval

Superblok (Superblock)
Guruh deskriptorlari (Group Descriptors)
Bloklarning bitli xaritasi (Block Bitmap)
Indeksli deskriptorlarning bitli xaritasi (Inode Bitmap)
Indeksli deskriptorlarning jadvali (Inode Table)
Ma’lumotlar (Data)

**Ext2fs da bloklar guruhi.** Ext2 turidagi barcha bloklar, bloklar guruhlariga bo‘linadi. Har bir bloklar guruhiga global deskriptor jadvalida alohida yozuv shakllantiriladi; bu yozuvda quyidagi asosiy parametrlar saqlanadi:

- ❖ bloklar bitli xarita blokining raqami;
- ❖ inode bitli xarita blokining raqami;
- ❖ inode jadvali blokining raqami;
- ❖ gruppadagi bo‘sh bloklar soni;
- ❖ katalogdagi inode lar soni.

Bloklar bitli xaritasi – bu har biti unga tegishli blok birorta faylga ajratilganligi ko‘rsatuvchi tuzilma. Agar bir 1 ga teng bo‘lsa, demak blok band. Shunga o‘xshash funksiyani indeksli deskriptorlarning bitli xaritasi bajaradi; u aynan qaysi indeksli deskriptorlar band, qaysilari esa band emasligini ko‘rsatadi. Linux ning yadrosi o‘z ichiga kataloglarni qamrab olgan inode sonidan foydalanib, inode kataloglarni guruhlariga bir tekisda taqsimlashga urinadi; inode fayllarni, agar bunga imkon bo‘lsa, ota katalog guruhiga joylashtiradi. Jadvalda ma’lumotlar deb belgilangan barcha qolgan joy fayllarni saqlash uchun ajratiladi.

**Ext2fs da fayllarni manzillash.** Ma’lumotlarni manzillash tizimi – bu fayl tizimining eng muhim komponentlaridan biridir.

Aynan manzillash tizimi diskdagi bo'sh va band bloklarning ko'pchiligadan kerakli faylni topishga imkon beradi. Ext2 fayl tizimi fayl bloklarini manzillashning quyidagi sxemasidan foydalanadi. Faylning manzilini saqlash uchun har bir 4 baytdan iborat 15 maydonlar ajratilgan. Agar faylning kattaligi 12 blok yoki undan kamroq bo'lsa, bu klasterlar raqamlari manzilning bevosita birinchi 12 maydonida sanab o'tiladi. Agar faylning kattaligi 12 blokdan ko'proq bo'lsa, u holda 13- maydonda klaster manzili joylashadi va unda faylning quyidagi bloklar raqamlari joylashtirilishi mumkin. Shunday qilib manzilning 13-chi elementi bilvosita manzillash uchun foydalaniladi. Blokning maksimal, ya'ni 4096 baytga teng, kattaligida 13-chi elementda fayl ma'lumotlari quyidagi klasterlarining 1024 raqamigacha joylashtirilgan bo'lishi mumkin. Agar faylning kattaligi  $12+1024$  blokdan ko'proq bo'lsa, 14-chi maydon blokdan foydalaniladi; unda har birida 1024 raqam mavjud fayl ma'lumotlari bloklarining 1024 raqami saqlandigan bloklar joylashgan blokning raqami bor. Bu yerda endi ikkilangan bilvosita manzillash qo'llaniladi. Va nihoyat agar fayl o'z ichiga  $12+1024+1048576 = 1049612$  dan ko'proq bloklarni qamrab olgan bo'lsa, unda 15-chi maydon uch marta katta bilvosita manzillash uchun foydalanadi. Shunday qilib yuqori ta'riflangan manzillash tizimi blokning maksimal kattaligi 4 Kb bo'lganida faylning kattaligi 2 terabayt va undan ko'proq bo'lishi mumkin.

**Ext3** (Third Extended Filesystem) – ext2 fayl tizimining yangilangan ko'rinishi, lekin bu tizimda yangi texnologiya (jurnallar bilan ishlash) qo'llanilgan. Diskda sodir bo'lgan o'zgarishlarning barchasi “jurnallarga” yozib boriladi va ma'lumotlarga zarar yetganda shu jurnal orqali fayllarni tiklash mumkin bo'ladi. Jurnal bilan ishlashning quyidagi rejimlari mavjud:

❖ **Ordered** – bu rejimda fayl tizimida sodir bo'lgan xizmat fayllarini o'zgarishlari yozib boriladi. Rejim odatiy tarzda ishlatiladi;

❖ **Journal** – buzilish sodir bo'lganda ma'lumotlarni minimal yo'qotishga erishmoqchi bo'lsangiz shu rejimdan foydalanishingiz mumkin. Bu rejimda xizmat fayllaridan tashqari foydalanuvchining ma'lumotlari ham jurnalga yozilishi mumkin. Shuning uchun bu rejim eng sekin ishlaydigan rejim hisoblanadi;

❖ **Writeback** – eng tez va eng foydasiz bo'lgan rejim hisoblanadi.

Jurnalning ishlash rejimini faqat Ext3 fayl tizimi uchun o'rnatish mumkin va u quyidagicha o'rnatiladi: **/etc/fstab** fayli o'zgartiriladi.

`/dev/sda5/ext3 defaults, data=journal 1 1`

Ext3 fayl tizimida disk bo'limlarining maksimal qiymati 4 Tbayt, lekin Linuks yadrosining 2.6 versiyasida 16 Tbayt qilib belgilangan. Faylning maksimal xajmi 1 Tbayt bo'lishi mumkin. Journallar bilan ishlash texnologiyasini o'chirib qo'yish ham mumkin.

**Ext4** – Linuksning yangi ishlab chiqilgan fayl tizimlaridan biri hisoblanadi. Bu fayl tizimi Linuks yadrosining 2.6.28 versiyasida paydo bo'lgan. Ext3 fayl tizimi bilan solishtiradigan bo'lsak, Ext4 fayl tizimi ishlashi va ishonchliligi 2 barobar oshganligini ko'rishimiz mumkin. Bu tizimda disk bo'limining maksimal qiymati 1024Pbayt (1Ebayt), fayl xajmining maksimal qiymati 2 Tbayt qilib belgilangan.

**Linux Proc fayl tizimi.** Proc fayl tizimi ma'lumotlarni saqlamaydi; buni o'rniga ularning ichidagilari talab bo'yicha, kiritish/chiqarishga foydalanuvchi so'roviga muvofiq hisoblab chiqiladi. Proc tizimi katalog va uning ichida joylashgan fayllar tuzilmalarini amalga oshiradi; keyin u o'zining ichidagi har bir katalog va fayl uchun yagona va saqlanadigan inode raqamini belgilaydi. U bu inode raqamini identifikatsiya uchun ishlatadi, ya'ni foydalanuvchi faylning aniq inode ga murojaat qilganda yoki katalogning aniq inode da izlashni bajarishga uringanda qanday amal zarurligini belgilaydi. Ma'lumotlar shu fayllarning birortasida o'qilganida proc tizimi tegishli axborotni yig'adi, uni matn shakliga aylantiradi va tegishli jarayonning o'qish buferiga joylashtiradi.

### ***Linux da kiritish/chiqarish***

Qurilmalarga yo'naltirilgan Linux fayllar tizimi disk xotirasiga ikki kesh yordamida kiradi:

- ❖ ma'lumotlar virtual xotira tizimi bilan birlashgan sahifalar keshida saqlanadi;

- ❖ metama'lumotlar bufer keshida saqlanadi, shu bilan birga har bir kesh diskning bloki bilan indekslanadi.

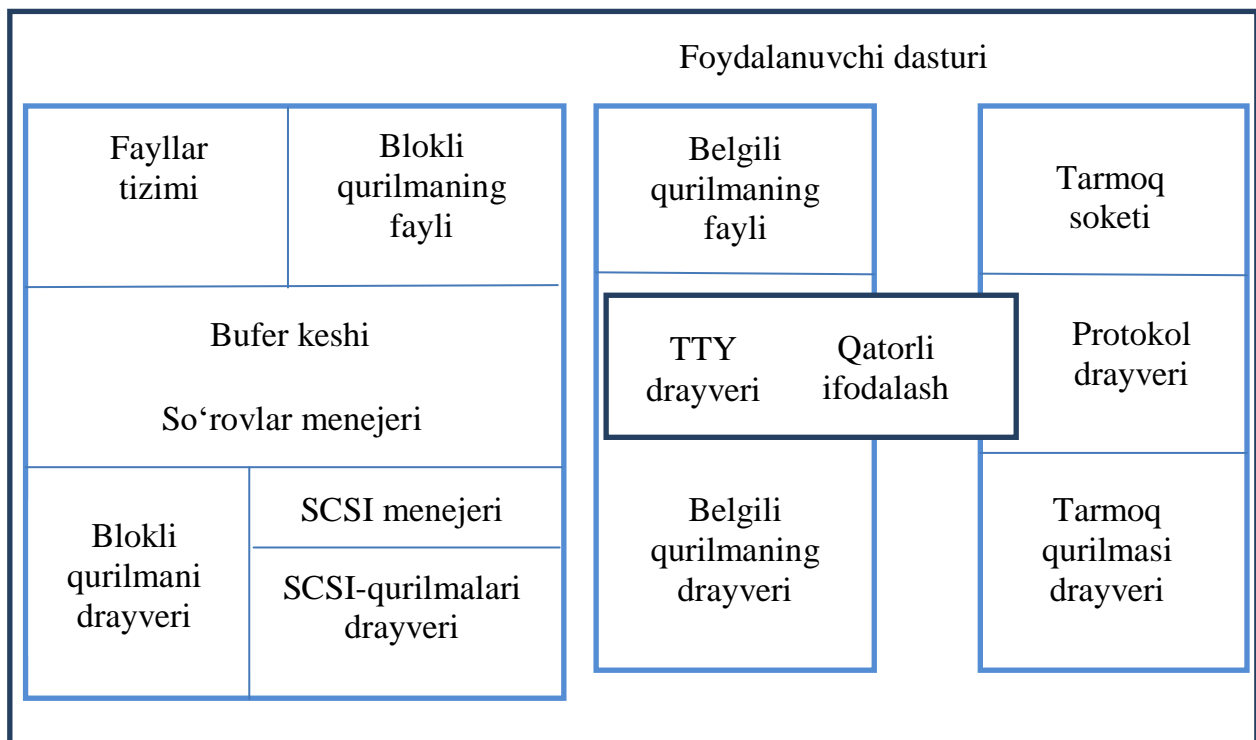
Linux qurilmalarni quyidagi uchta sinflarga ajratadi:

- ❖ blokli qurilmalar – to'la mustaqil, ma'lumotlar o'zgarimas kattalikdagi bloklardan ixtiyoriy foydalanishga ruxsat beradi;

❖ belgili qurilmalar – ular qatoriga barcha boshqa qurilmalarning aksariyat qismi kiradi; ular oddiy faylarning funkcionalligini qo‘llab-quvvatlashi shart emas;

❖ tarmoq qurilmalari – yadro tarmoq tizimi bilan o‘zaro harakat qiladi.

Linuxda qurilmalar drayverlari modulli tuzilishi 9.6- rasmda ko‘rsatilgan.



9.6- rasm. Qurilmalar drayverlarining modulli tuzilishi

Blokli qurilmalar tizimdagi barcha disk qurilmalariga asosiy interfeysni ta‘minlaydi. Blokli bufer keshi ikki maqsadda xizmat qiladi:

- ❖ faol kiritish/chiqarish uchun bufer keshi sifatida;
- ❖ tugallangan kiritish/chiqarish uchun kesh sifatida.

So‘rovlar menejeri blokli qurilmaning drayveri yordamida bufer tarkibini o‘qish va yozishni boshqaradi.

Belgili qurilmaning drayveri o‘zgarmas bloklarning ma‘lumotlaridan ixtiyoriy foydalanishni qo‘llab-quvvatlamaydi. Belgili qurilmaning drayveri turli talab qilinayotgan kiritish/chiqarish amallarni bajaradigan funksiyalar to‘plamini ro‘yhatdan o‘tkazadi. Yadro belgili qurilmadan faylni o‘qish yoki yozish so‘roviga deyarli



xech qanday oldindan ishlov berishni amalga oshirmaydi, lekin oddiy so‘rovni qurilmaning drayveriga uzatadi. Bundan asosiy mustasno – bu terminal qurilmalardan (TTY) foydalanishni amalga oshiruvchi belgili qurilmalar drayverlarining alohida to‘plami; ular uchun yadro standart interfeysni qo‘llab-quvvatlaydi.

### ***Linux da jarayonlarning o‘zaro harakati***

UNIX ga o‘xshab, Linux jarayonlarga - hodisalar ro‘y berganligi to‘g‘risida signallar yordamida xabar beradi. Signallarning cheklangan to‘plami mavjud va ular birorta axborotni olib bormaydi: jarayon faqat signal borligi haqidagi faktdan foydalanishi mumkin. Linux yadrosi, yadro rejimida bajarilayotgan jarayonlar o‘rtasida aloqa qilish uchun signallardan foydalanmaydi. Yadro ichidagi aloqa tuzilmalarini rejalashtiruvchi - states (holatlar) va wait.queue (pipe) (kutish navbati) - yordamida amalga oshiriladi. Konveyer mexanizmi (pipe) farzand jarayonga ona jarayondan aloqa kanalini meros qilib olishga imkon beradi. Konveyerning bir uchida yozilayotgan ma‘lumotlar uning ikkinchi uchida o‘qilishi mumkin. Umumiy xotira aloqa qilishning juda tez usulini ta‘minlaydi; bir jarayon bilan umumiy xotiraning hududiga yozilgan barcha ma‘lumotlar mazkur hududni o‘zining manzil maydonida aks etgan boshqa jarayon tomonidan o‘sha zahoti o‘qilishi mumkin. Lekin sinxronlash maqsadida, umumiy xotira boshqa birorta aloqa mexanizmlari bilan birgalikda foydalanilishi kerak. Umumiy xotiradagi obyekt umumiy xotiradagi hududlar uchun faylni chiqarish maqsadida foydalaniladi; xuddi shunday fayl xotirada aks ettiriladigan hududdan ma‘lumotlarni chiqarish uchun ishlatiladi. Umumiy xotiraga aks ettirishlar sahifalar buzilishlarini ajratiladigan obyekt band qilingan xotiraning hududiga qayta yo‘naltiradi. Ajratiladigan obyektlar o‘z ichidagilarni esida qoldiradi, xattoki mazkur onda xech qanday jarayonlar ularni o‘zining virtual xotira maydonida aks ettirilmayotgan bo‘lsa ham.

### ***Linux da tarmoqning tuzilishi***

Tarmoqda ishlash – buLinux da funkcionallikning eng muhim sohasidir. Linux ning tarmoq tizimi UNIX – UNIX aloqalari uchun asosiy Internet-protokollarni qo‘llab-quvvatlaydi. Shuningdek, u UNIX bo‘lmagan operatsion tizimlarga xos protokollarni, xususan, Appletalk va IPX kabi kompyuter tarmoqlarida ishlatiladigan

protokollarni qoʻllaydi. Linux tarmoq tizimi abstraksiyaning uch darajasida amalga oshiriladi:

- ❖ soket interfeysi;
- ❖ drayverlar protokollari;
- ❖ tarmoq qurilmalari drayverlari.

Linux da tarmoq protokollarining eng muhim toʻplami – bu Internet protokollari toʻplami. Linux tarmoq tizimi tarmoqning barcha qismlarida turli mashinalar oʻrtasida mashrutlashni taʼminlaydi. Mashrutlash protokolini yuqori darajasida UDP, TCP va ICMP protokollar qoʻllaniladi.

### ***Linux da xavfsizlik***

Linux tizimida ulanadigan autentifikatsiya modullarini (pluggable authentication modules - PAM) umumiy kutubxonaga asoslangan holda foydalanuvchini tasdiqlashni talab qiladigan har qanday komponentlar tomonidan ishlatishga asoslangan. UNIX turdagi tizimlarda, jumladan Linux da ham foydalanishni boshqarish foydalanuvchi va guruhning yagona raqamli identifikatorlari (uid va gid) yordamida amalga oshiriladi. Foydalanishga ruxsat berilishini boshqarish obyektlarga qanday amallardan (oʻqish, yozish, bajarish) egasi, guruh va barcha boshqa foydalanuvchilar foydalanishi mumkinligini koʻrsatuvchi himoyalash niqoblarini berish yoʻli bilan amalga oshiriladi. Linux standart UNIX (set uid) mexanizmini quyidagi ikki usul bilan toʻldiradi:

- ❖ bu mexanizm POSIX spetsifikatsiyasi orqali bajariladi, bu har bir jarayonga oʻzining amaldagi uid ni koʻp marotaba boʻshatish va takror olishga imkon beradi;

- ❖ amaldagi uid ga muvofiq faqat vakolatlar kichik toʻplami taqdim etuvchi jarayonning tavsifi toʻldiriladi.

Linux mijozga ayrim server jarayoniga, birorta boshqa imtiyozlar bermagan holda, maʼlum fayldan foydalanishga tanlab ruxsat berishni uzatish mexanizmini taʼminlaydi.

Linux OT oilasi xavfsizligining asosi bu tizim foydalanuvchilaridir. Foydalanuvchilar muayyan huquqlarga ega, ular oʻzlarining fayllari, tizimi va dasturlarini oʻzlari uchun ishga tushurishadi. Shuningdek, funksiyalar toʻplami, tizim chaqiruvlari va operatsiyalari, qaysi jarayonni amalga oshirish jarayonni boshlagan foydalanuvchi huquqlariga bogʻliq.

Tizimda ro'yxatdan o'tgan barcha foydalanuvchilar /etc/passwd faylida saqlanadi. Tizimning har bir foydalanuvchisining login nomi mavjud. Ammo, tizim foydalanuvchilarni nomidan emas, balki ularning noyob foydalanuvchi identifikatorlari yoki UID (foydalanuvchi identifikatori) tomonidan ajratib turadi. Bundan tashqari, har bir foydalanuvchi bir yoki bir nechta guruh a'zosi. Guruhlar /etc/group faylida ko'rsatilgan. Har bir guruhning o'ziga xos nomi va yagona GID (Group Identifier) guruh identifikatori mavjud. /Etc/passwd faylida foydalanuvchi asosiy guruh identifikatori ko'rsatilgan. Shunday qilib, UID va GID foydalanuvchi tizimda qanday huquqlarga ega ekanligini aniqlaydi. Barcha foydalanuvchilar orasida maxsus pozitsiya UID = 0 yoki super foydalanuvchi deb ataladigan foydalanuvchi tomonidan amalga oshiriladi. Bu foydalanuvchi odatda kirish nomi root ga ega, tizimda UID = 0 va turli nomlar bilan bir nechta foydalanuvchi bo'lishi mumkin. Super foydalanuvchi cheklovsiz huquqlarga ega va fayllar va jarayonlar bo'yicha har qanday amaliyotni amalga oshirishi mumkin. Bunga qo'shimcha ravishda, ba'zi tizimdagi chaqiruvlar va operatsiyalar faqat super foydalanuvchi tomonidan amalga oshirilishi mumkin. Misol uchun, faqat super foydalanuvchi uchun quyidagi operatsiyalar mavjud: tizim fayllarini yaratish, tizim vaqtini o'zgartirish, resurslardan foydalanish cheklovlarini oshirish, jarayon ustuvorligini oshirish, tarmoq nomini o'rnatish, tarmoq interfeyslarini sozlash, to'xtatish va tizimni qayta ishga tushirish. Super foydalanuvchi nomi bilan ishlashda ma'mur ehtiyotkorlik bilan harakat qilishi kerak, chunki ehtiyotsiz harakatlarni qaytarib bo'lmas oqibatlariga olib kelishi mumkin.

Parolni shifrlash algoritmi tizimning o'ziga xos versiyasiga bog'liq va bir tomonlama shifrlash algoritmidir. Masalan, FreeBSD tizimi MD5 algoritmidan foydalanadi. Shuni esda tutish kerakki, parol ochiq ko'rinishda hech qachon va hech joyda saqlanmaydi. Agar foydalanuvchi tizimga kirsam, u bilan kiritilgan parol shifrlanib, ikkita shifrlangan variantni taqqoslanadi. Hatto super foydalanuvchi har qanday foydalanuvchi parolini topa olmaydi, lekin u passwd utilitasi yordamida har qanday foydalanuvchi parolini o'zgartirishi mumkin. Ushbu xizmat oddiy foydalanuvchilarga ularning parollarini o'zgartirish imkonini beradi. Agar "\*" belgisi shifrlangan parol o'rniga parol faylida bo'lsa, u foydalanuvchi tizimga kira olmaydi.

Agar parol maydoni bo'sh bo'lsa, foydalanuvchi bo'sh parolni kiritib kirishi mumkin.

Hatto shifrlangan shaklda parol faylida xavfsizlikning ba'zi bir xavfi mavjud, chunki parollarni tanlash mumkin, masalan, lug'at yordamida. Shuning uchun ko'plab tizimlarda parollar /etc/passwd faylidan boshqa fayllarga o'tkaziladi. Masalan, FreeBSD-da parollar /etc/master.passwd faylida va Solaris va Linuxda, /etc/shadow-faylida. Ushbu fayllarning ikkalasi ham super foydalanuvchi uchun mavjud bo'lib, ular hatto o'qish uchun ham tizimning boshqa foydalanuvchilarida mavjud emas. Agar foydalanuvchi tizimga kirsa, /etc/passwd faylida ko'rsatilgan buyruq tarjimoni ishga tushiriladi. Odatda bu standart buyruq tarjimonlari /bin/sh (Bourne shell), /bin/csh (C shell), /bin/ksh (murch qobig'i) yoki /bin/bash (yana Bourne shell) foydalanuvchining buyruqlarini kiritish va vazifalarni bajarishga imkon beradi. Ushbu maydonda tarjimonga ehtiyoj sezmaydigan foydalanuvchilar uchun, masalan, /sbin/nologin/ mavjud bo'lmagan yoki /bin/date (uzoq foydalanuvchilar uchun) ni belgilashingiz mumkin. Umumiy holatda, ushbu sohada har qanday dasturni ko'rsatish mumkin, shuning uchun bu vazifani bajarib bo'lgach, foydalanuvchi avtomatik tarzda tizimdan chiqadi. /etc/shell faylida buyruq tarjimoni sifatida ishlatilishi mumkin bo'lgan dasturlarning ro'yxati mavjud.

Yangi yaratilgan fayl egasi-foydalanuvchisi faylni yaratgan foydalanuvchi hisoblanadi. Yangi yaratilgan faylning guruh egasi turli tizim versiyalarida turlicha belgilanadi. Linuxda uchta asosiy fayllarga kirish klasslari mavjud:

- ❖ u (user) – foydalanuvchi egasi (siz yoki ma'mur) uchun kirish klassi;

- ❖ g (group) – guruh a'zolari uchun kirish klassi;

- ❖ o (others) - boshqa foydalanuvchilar uchun kirish klassi.

Unix har bir klass uchun uch turdagi kirishlarni qo'llab-quvvatlaydi:

- ❖ r (read) – o'qish huquqi;

- ❖ w (write) – yozish huquqi;

- ❖ x (execute) – bajarish huquqi.

Kirish huquqi ro'yxati ls -l yordamchi dasturi yordamida olinishi mumkin.

Masalan:

- rwx rw- r- - 1 root bek 1077 Apr 1 18:05 /usr/home/ma'ruza/1

Yuqoridagi misoldan, /usr/home/ma'ruza/1 fayli uchun foydalanuvchi egasi root ekanligini ko'rish mumkin, guruh egasi bek, ma'mur huquqi rwx, guruh a'zolari huquqi rw, boshqa foydalanuvchilar huquqi r.

Agar foydalanuvchilarni huquqini chmod buyrug'i orqali o'zgartirish mumkin. Masalan, chmod 0 – w file1.

### ***Audit***

Linux ning Audit tizimi xavfsizlikka tegishli ma'lumotlarni kuzatib borish usulini taqdim etadi. Oldindan tuzilgan qoidalarga asoslanib, Audit tizimda sodir bo'layotgan hodisalar haqida iloji boricha ko'proq ma'lumotlarni yozib olish uchun jurnal yozuvlarini yaratadi. Ushbu ma'lumot muhim siyosat muhitlari uchun xavfsizlik siyosati va ular tomonidan qilingan harakatlarning buzilishini aniqlash uchun juda muhimdir. Audit tizimni qo'shimcha xavfsizlik bilan ta'minlamaydi; aksincha, undan tizimda ishlatiladigan xavfsizlik siyosatining buzilishini aniqlash uchun foydalanish mumkin. Ushbu qoidabuzarliklarni SELinux kabi qo'shimcha xavfsizlik choralari bilan oldini olish mumkin. Quyidagi ro'yhat Audit tizimi o'zining jurnal fayllariga yozishi mumkin bo'lgan ba'zi ma'lumotlarni taqdim etadi:

- ❖ Hodisaning sana va vaqti, turi va natijasi;
- ❖ Obyektlar va subyektlar uchun sezgirlik yorliqlari;
- ❖ Audit sozlamalarini barcha o'zgarishlari va audit jurnali fayllariga kirishga bo'lgan urinishlar;
- ❖ Barcha turdagi foydalanilgan autentifikatsiya mexanizmlar, masalan, SSH, Kerberos va boshqalar;
- ❖ /etc/passwd kabi ishonchli ma'lumotlar bazasidagi o'zgarishlar;
- ❖ Tizimga ma'lumotlarni yuklash yoki tizimdan ma'lumotlarni ko'chirib olishga bo'lgan urinishlar;
- ❖ Foydalanuvchi identifikatori, obyekt yoki subyekt yorlig'i va boshqa atributlarga asoslangan hodisalarni qo'shish yoki olib tashlash.

## **9.2. Windows operatsion tizimi, qurilish tamoyili va ahamiyati**

Hozirgi kunda shaxsiy kompyuterdan foydalanuvchilar asosan qulay grafik interfeysi uchun Windows operatsion tizimidan

foydalanishadi. Bundan tashqari, Windows operatsion tizimining server versiyalari ichki lokal tarmoqlarda ham ko'plab korxonalar va tashkilotlarda foydalanib kelinmoqda. Shaxsiy kompyuterlardan foydalanadigan foydalanuvchilarning 85% asosan Windows operatsion tizimidan foydalanadi. Microsoft Windows Microsoft tomonidan ishlab chiqilgan bir qator operatsion tizimlar va ish stoli muhitlarini ifodalaydi. Windows operatsion tizimlarining birinchi versiyasi - MS-DOS, 1985yilda kiritilgan, grafik interfeysga ega edi. U bir nechta hujjatlarni, kompyuter sichqonchasini qo'llab-quvvatlashni, ochiladigan menyuni taqdim etdi va buning hammasi ranglarda ko'rish mumkin edi. Windowsning keyingi versiyalari asta-sekin MS-DOSning apparat va dasturiy ta'minotning funksiyalarini o'rnatgan. MS-DOSga kiritilgan barcha o'zgarishlarni keyinchalik integratsiyalashgan holda ishlab chiqilgan operatsion tizimni yaratishga yordam berdi. Hozirgi kunda Microsoft Windows nafaqat odatdagi qulaylik va ta'sirchan funktsionallik tufayli, balki uning yadrosi va Microsoft Office kabi boshqa dasturiy ta'minot imkoniyatlari bilan integratsiyalashuv darajasi yuqori bo'lgani sababli eng mashhur OTdir. Garchi bunday dasturiy integratsiyani dastlab xavfsizlik muammolari bo'lgan bo'lsa-da, bugungi kunda Microsoft Windows nafaqat shaxsiy kompyuterlarda, balki hisoblash qurilmasining server versiyalarida ham keng qo'llaniladi. Eng so'nggi Windows 10 operatsion tizimining ishlab chiqarish zanjirida yakuniy aloqa bo'ldi, ammo bu uning rivojlanish davrining oxirigacha mo'ljallanmagan. Integratsiyalangan operatsion tizim sifatida, Microsoft Windows-ning barcha versiyalari o'rnatishdan so'ng darhol ishlatilishi mumkin bo'lgan oldindan o'rnatilgan dasturiy ta'minot bilan ta'minlangan. Asosiy matn tahrirlovchisi va kalkulyatori Windowsning birinchi versiyasida foydalanish uchun taqdim etildi. Windows 98 Media Payer, Internet Explorer va Outlook Expressni taqdim etdi. Windows Vista'dan boshlab, DVD Maker foto galereyasi va Windows 7, 8, 10-da informer gadget'lari bilan tasvirlangan yon panel paydo bo'ldi. Ikkinchi Windows XP xizmat paketidan va Windows Vista-ning ishlab chiqarilishidan boshlab, o'rnatilgan xavfsizlik devori xavfsizligi xususiyati (xavfsizlik devori) paydo bo'ldi. Zamonaviy Windows operatsion tizimlarida ushbu funktsiya yadro darajasida amalga oshiriladi. Sandbox orqali zararli xatti-harakatlar tizimdagi beqarorlikka yoki xavfsizlik buzilmasligiga olib

kelmaydi, barcha dasturiy ta'minot tizim yadrosidan tashqari, uchinchi tomon virusga qarshi tizimlarini ham o'z ichiga oladi. Kompyuterlar uchun Windows operatsion tizimlarining umumiy foydalanuvchi versiyalariga qo'shimcha ravishda, Microsoft operatsion tizimining server versiyalari, mobil va o'rnatilgan versiyalarini chiqazdi. Windows Server tizimi raqobatchilik boshqaruv paketiga ega, yangilangan xizmatlar, saqlash serverlari, veb-server, media server va yangi o'yinchi o'ylashdan ko'ra ko'proq ta'sirchan funksiyalarni o'z ichiga oladi. Ushbu paket asosan korporativ foydalanish uchun, mahalliy tarmoqni boshqarish va turli xil kompyuterlarni o'z maqsadlari uchun birlashtiruvchi tashkilotlar uchun mo'ljallangan. Windows Mobile PDA va smartfonlar kabi shaxsiy qurilmalar uchun mo'ljallangan operatsion tizimlarning bir qatoridir. Microsoft Windows har kuni yangi xususiyatlar va xavfsizlik yangilanishlarini chiqaradi. Foydalanuvchilarning ko'pchiligi ushbu mantiqan tuzilgan Windows operatsion tizimidagi oilada ishlash imkoniyatidan juda mamnun. Bundan tashqari, Windows 10-ning eng so'nggi versiyasi Microsoft-ga ko'ra texnologiya dunyosida yangi pog'ona bo'ladi.

Windows operatsion tizimi quyidagi imkoniyatlarga ega:

- ❖ Ko'p vazifalilik, ya'ni Windows muhitiga o'tgandan keyin bir vaqtning o'zida bir necha dasturlarda ishlash imkoniyati paydo bo'ladi. Masalan, hujjatlarni tahrirlash, rasm chizish, hujjatlarni chop etishga berish, musiqa eshitish va hokazo;

- ❖ Ma'lumotlar buferidan foydalanish imkoniyati, ya'ni bimalol grafik muharriridagi yoki elektron jadvaldagi ma'lumotlarni boshqa dasturlarga qo'yish yoki aksincha;

- ❖ Windows muhitida ishlaydigan barcha dasturlarning shriftlari, drayverlarining printerdan foydalanish va boshqarish tugmalarining bir xilligi foydalanuvchiga bitta dasturda ishlaganda ikkinchisida ishlashga oson ko'nikish hosil qilish imkonini beradi;

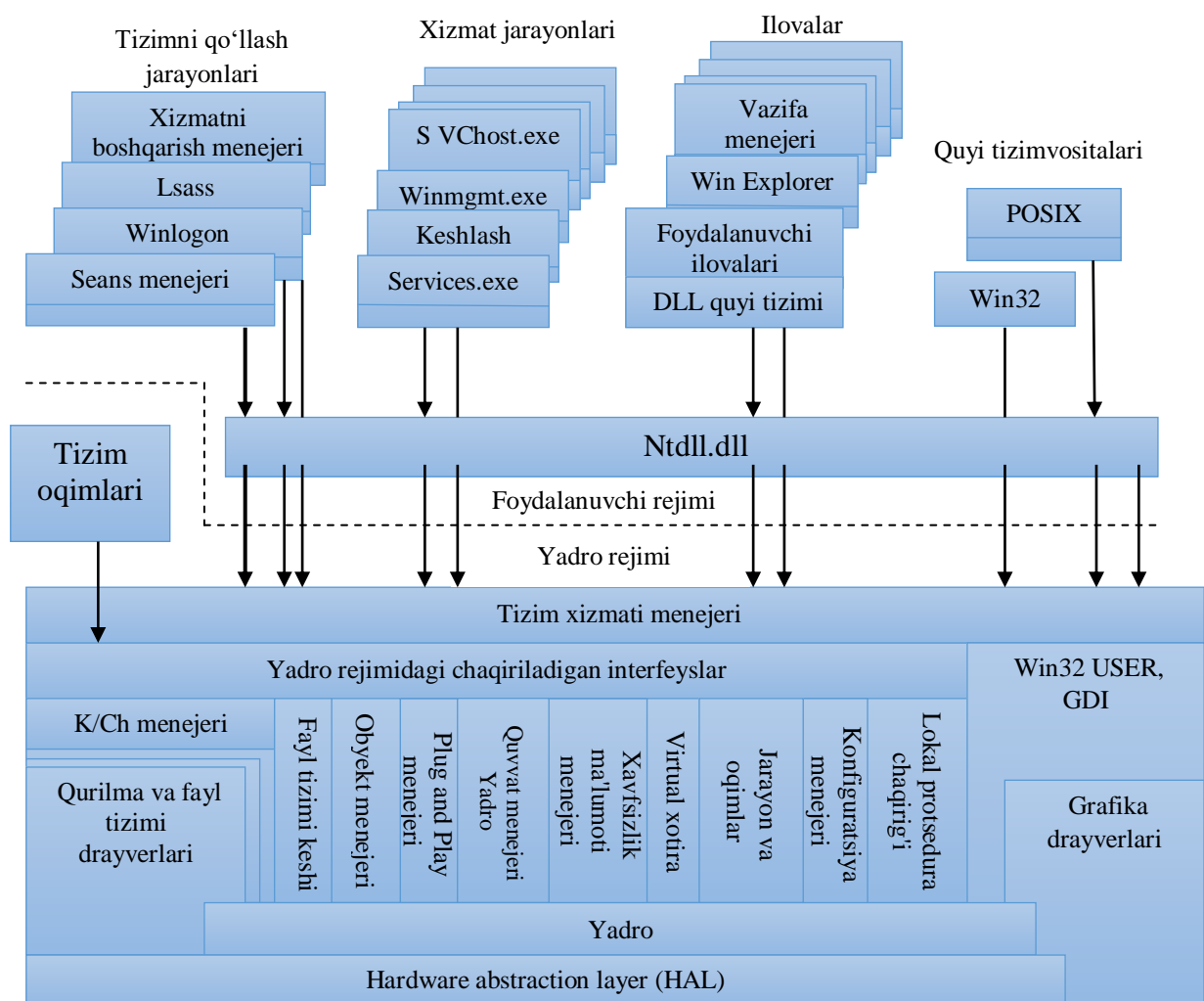
- ❖ Windows operatsion tizimida kompyuter tarmoqlaridan foydalanish juda qulay, masalan lokal kompyuter tarmog'ida ishlash uchun tarmoqli muhit yorlig'ini ishlatish yetarli;

- ❖ Foydalanuvchi interfeysining yagonaligi, ya'ni Windows muhitida oddiydan tortib murakkabgacha dasturlarning barchasida muloqat oynalarining, menyularining, vositalar panelining va boshqalarning yagona ko'rinishga ega ekanligi va hokozo imkoniyatlarni aytish mumkin;

- ❖ Universal grafika - Windows dasturlarining qurilmalarga va dastur ta'minotiga bog'liqsizligini ta'minlaydi;
- ❖ Ma'lumotlar almashuvi - Windows dasturlararo ma'lumotlar almashish imkoniyatiga ega;
- ❖ Windows muhitida elektron pochta va faksdan foydalanish mumkin va hokazo.

Bundan tashqari Windows operatsion tizimining ko'plab server versiyalari mavjud. Ohirgi ishlab chiqilgan versiya Windows Server 2019 hisoblanadi.

### Windows 7 operatsion tizimi tuzilishi



9.7- rasm. Windows 7 operatsion tizimining umumiy tuzilishi

9.7- rasmda Windows 7 operatsion tizimining umumiy tuzilishi ko'rsatilgan, Windows ning NT ga asoslangan barcha versiyalari ushbu ma'lumot darajasida bir xil tuzilishga ega. Deyarli barcha



operatsion tizimlarda bo'lgani kabi, Windows amaliy dasturlarni asosiy OT dasturiy ta'minotidan ajratadi. Bajaruvchi yadro, qurilma drayverlari va apparat abstraksiyasi qatlamini o'z ichiga olgan xat yadro rejimida ishlaydi. Yadro rejimi dasturi tizim ma'lumotlari va qo'shimcha qurilmalarga kirish huquqiga ega. Foydalanuvchi rejimida ishlaydigan qolgan dasturiy ta'minotlar, tizim ma'lumotlariga cheklangan kirish huquqiga ega.

Windows yuqori modulli arxitekturaga ega. Har bir fayl tizimining funksiyasi faqat bitta OT komponenti tomonidan boshqariladi. Qolgan OT va barcha dasturlar ushbu funksiyaga standart interfeyslardan foydalangan holda javobgar komponent orqali kirishadi. Tizimning kalit ma'lumotlariga faqat tegishli funksiyalar orqali kirish mumkin. Umuman olganda, har qanday modul, butun tizimni yoki uning standart dasturiy interfeyslarini qayta yozmasdan olib tashlanishi, yangilanishi yoki o'zgartirilishi mumkin. Windows yadro rejimining komponentlari quyidagilardan iborat:

❖ **Bajaruvchi** (executive): Xotirani boshqarish, jarayonlar va oqimlarni boshqarish, xavfsizlik, kiritish/chiqarish, va jarayonlararo aloqa kabi operatsion tizimning asosiy xizmatlarini o'z ichiga oladi.

❖ **Yadro** (kernel): Protsessorlarning ishlashini boshqaradi. Yadro oqimlarni rejalashtirishni, jarayonlarni almashtirishni, istisnolar va uzilishlarni qayta ishlashni, hamda ko'pprotsessorli sinxronlashni boshqaradi. Qolgan bajaruvchi va foydalanuvchi darajalaridan farqli ravishda, yadro o'z kodi oqimlarda ishlamaydi.

❖ **Qurilma abstraksiyasi darajasi** (Hardware abstraction layer - HAL): Umumiy qurilma buyruqlari va javoblar va ma'lum bir platformaga xos bo'lgan xaritalar. U OTni platformadagi maxsus qurilma farqlaridan ajratib turadi. HAL tufayli har bir kompyuterning tizim shinasi, xotiraga to'g'ridan-to'g'ri kirish (DMA) kontrolleri, uzilishlar kontrolleri, tizim taymerlari va xotira kontrolleri, bajaruvchi (Execution) va yadro komponentlari uchun bir xil ko'rinishga ega bo'ladi. Bundan tashqari, SMP uchun zarur bo'lgan yordamni taqdim etadi.

❖ **Qurilma drayverlari** (device drivers): bajarish funksiyasini kengaytiradigan dinamik kutubxonalar. Bunga foydalanuvchi kiritish/chiqarish funksiyasi chaqirig'ini ma'lum qurilma vositalariga kiritish/chiqarish so'rovlari va fayl tizimlarini, tarmoq protokollarini va yadro rejimida ishlashi kerak bo'lgan boshqa har qanday tizim

kengaytmalarini amalga oshirish uchun dasturiy ta'minot qismlarini tarjima qiladigan qurilma vositalari drayverlari kiradi.

❖ **Oynalar va grafik tizim** (Windowing and graphic system): GUI funksiyalarini bajaradi, masalan, oynalar bilan ishlash, foydalanuvchi interfeysini boshqarish va chizish.

Windows bajaruvchisi maxsus tizim komponentlarini o'z ichiga oladi va foydalanuvchi rejimi dasturi uchun API ni taqdim etadi. Quyida har bir boshqaruv modulining qisqacha tavsifi keltirilgan:

❖ **Kiritish/chiqarish menejeri** (I/O manager): kiritish/chiqarish qurilmalarini dasturga kirish imkoniga ega bo'lishini ta'minlaydi va keyingi qayta ishlash qurilmasi uchun mos keladigan drayverlarni yuborish uchun javobgardir.

❖ **Kesh menejeri** (cache manager): fayllarga asoslangan kiritish/chiqarish operatsiyalarini samaradorligini oshiradi.

❖ **Obyekt menejeri** (object manager): jarayonlar, oqimlar va sinxronizatsiya obyektlari kabi resurslarni ifodalash uchun foydalaniladigan Windows bajaruvchi obyektlarini yaratadi, boshqaradi va yo'q qiladi. Obyektlarni saqlash, nomlash va xavfsizlikni ta'minlash uchun yagona qoidalarni qo'llaydi. Obyekt menejeri shuningdek har bir jarayonni tavsiflovchi jadvalida yozuvlarni yaratadi, bular kirishni boshqarish ma'lumotlari va obyekt ko'rsatgichlaridan iborat.

❖ **Plug and Play drayveri** (Plug and Play driver): ma'lum bir qurilmani qo'llab-quvvatlash uchun qaysi drayverlar talab qilinishini aniqlaydi va ushbu drayverlarni yuklaydi.

❖ **Quvvat menejeri** (power manager): turli xil qurilmalar o'rtasida quvvatni boshqarishni muvofiqlashtiradi va ishlamayotgan qurilmalarni o'chirish, protsessorni uyqu rejimiga o'tkazish va hatto butun xotirani diskka yozish va butun tizimni o'chirish orqali quvvat sarfini kamaytirish uchun sozlanishi mumkin.

❖ **Xavfsizlik ma'lumotnoma monitori** (security reference monitor): kirishni boshqarish qoidalarini va auditni shakllantirishni ta'minlaydi. Windows barcha himoyalangan obyektlar fayllar, jarayonlar, xotira maydonlari, kiritish/chiqarish qurilmalari uchun kirishni boshqarish va tekshirish uchun bir xil usullardan foydalanadi.

❖ **Virtual xotirani boshqarish** (virtual memory management): virtual manzil, fizik xotira va diskdagi fayl sahifalarini boshqaradi. Jarayonning virtual manzillarini kompyuter xotirasidagi fizik

sahifalarga joylashtiradigan qurilmalar va xotirani boshqarish tuzilmalarini boshqaradi.

❖ **Jarayon/oqim menejeri** (process/thread manager): jarayon va oqim obyektlarini yaratadi, boshqaradi va o'chiradi.

❖ **Konfiguratsiya menejeri** (configuration manager): u tizim ro'yhatga olishini joriy qilish va boshqarish uchun javobgardir, bu tizim parametrlari va har xil parametrlarning individual sozlamalari saqlanadigan ombor hisoblanadi.

❖ **Kengaytirilgan mahalliy protsedura chaqirig'i vositasi** (Advanced local procedure call (ALPC) facility): u xizmatlarni va quyi tizimlarni amalga oshiradigan mahalliy jarayonlar o'rtasidagi aloqa uchun samarali jarayonlar protsedurasini chaqirish mexanizmini joriy etadi.

❖ **Foydalanuvchi rejimidagi jarayonlar** (user mode processes). Windows foydalanuvchi rejimining to'rtta asosiy turini qo'llab-quvvatlaydi:

❖ **Tizimning maxsus jarayonlari** (special system processes): seans menejeri, autentifikatsiya quyi tizimi, servis menejeri va tizimga kirish kabi tizimni boshqarish uchun zarur bo'lgan foydalanuvchi rejimlari xizmatlari.

❖ **Xizmat ko'rsatish jarayonlari** (service processes): chop etish menejeri, hodisalarni ro'yxatga olish, qurilma drayverlari bilan o'zaro aloqada bo'lgan foydalanuvchi rejimlari, turli xil tarmoq xizmatlari va boshqalar. Xizmatlar Microsoft va boshqa dasturiy ta'minot ishlab chiquvchilari tomonidan tizimning funktsional imkoniyatlarini kengaytirish uchun ishlatiladi, chunki ular Windows tizimida fondagi foydalanuvchi rejimini boshqarishning yagona usuli hisoblanadi.

❖ **Quyi tizimlar muhiti** (Environment subsystems): turli xil OT muhitlarini taqdim etadi. Win32 va POSIX quyi tizimlarini qo'llab-quvvatlaydi. Har bir muhit quyi tizimi o'z ichiga quyi tizim jarayonida va/yoki Windows tizimidagi foydalanuvchi chaqiriqlarini ALPC chaqiriqlariga o'zgartiradigan quyi tizim va dinamik bog'lanish kutubxonalaridan (DLL) foydalanadigan barcha dasturlar tomonidan almashinadigan quyi tizim jarayonlarini oladi.

❖ **Foydalanuvchi ilovalari** (user applications): bajariladigan fayllar (exe) va DLL kutubxonalar, ya'ni foydalanuvchilar tizimni yaratishda ishlatadigan funktsional imkoniyatlarni ta'minlaydi. Exe va

DLL odatda muhitning ma'lum bir quyi tizimiga qaratilgan, lekin OT tarkibiga kiruvchi ba'zi dasturlar o'zlarining tizim interfeyslaridan (NT API) foydalanadilar. Shuningdek, 64 razryadli tizimlarda 32 razryadli dasturlarni ishga tushirishni qo'llab-quvvatlaydi.

Boshqa versiyalarda interfeysni o'zgarishi, muhandislik yaxshilanishi, ishlash unumdorligini yaxshilanishi, ishonchlilik, xavfsizlik, oqimlarni qo'llashni yaxshilanishi, energiya samaradorligi yaxshilanishlarini keltirib o'tish mumkin.

### ***Windows 7 da oqim***

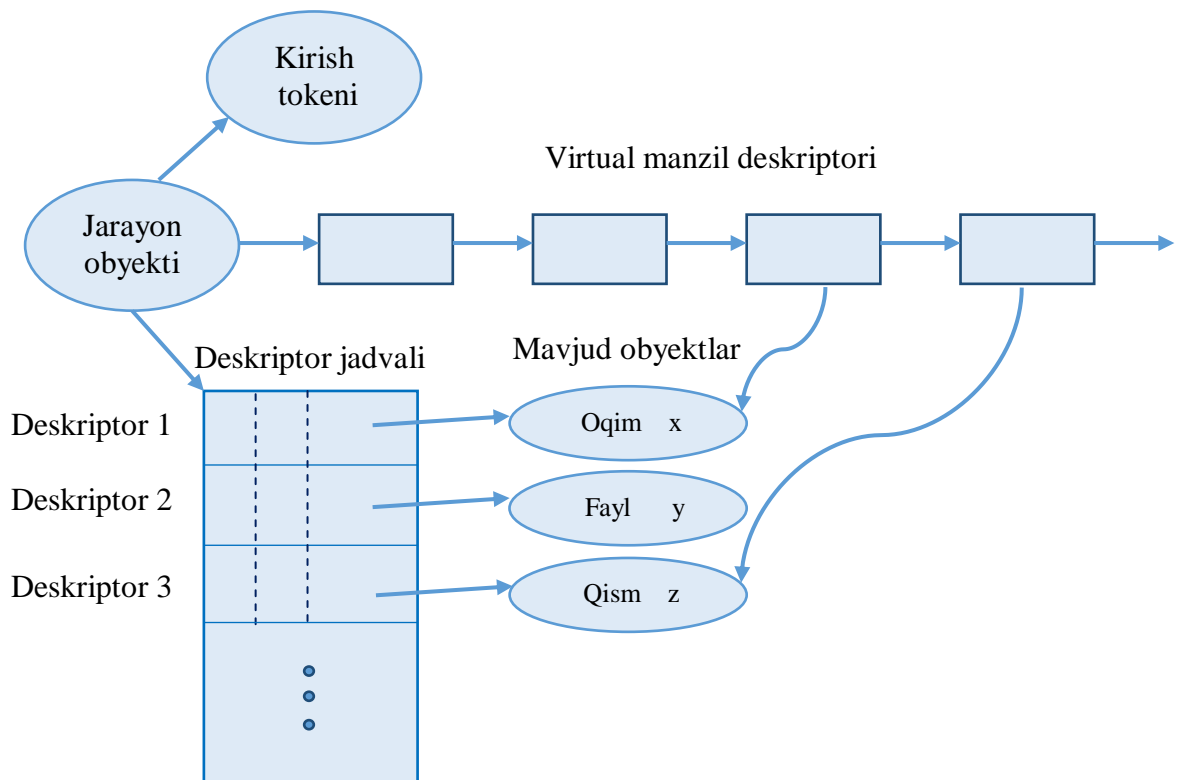
Windowsni ishlab chiqish jarayoni turli xil OT muhitlarini qo'llab-quvvatlash zarurati bilan bog'liq. Turli xil OT muhitlari tomonidan qo'llab-quvvatlanadigan jarayonlar bir qator sabablarga ko'ra farq qiladi, jumladan:

- ❖ jarayonlar qanday nomlanadi;
- ❖ jarayonlar ichida oqimlar taqdim qilinadimi;
- ❖ jarayonlar qanday taqdim etiladi;
- ❖ jarayon resurslari qanday himoyalangan;
- ❖ jarayonlararo aloqa va sinxronizatsiya uchun qanday mexanizmlardan foy-dalaniladi;
- ❖ jarayonlar bir-biri bilan qanday bog'liq.

Shunga ko'ra, Windows yadrosi tomonidan taqdim etiladigan shaxsiy jarayon tuzilishi va xizmatlari nisbatan sodda va universaldir, bu har bir OT quyi tizimiga ma'lum bir jarayon tuzilishi va funkcionallikni taqlid qilishga imkon beradi. Windows jarayonlarining muhim xususiyatlari quyidagilar:

- ❖ Windows jarayonlari obyekt sifatida bajariladi;
- ❖ Jarayon yangi jarayon sifatida yoki mavjud jarayonning nusxasi sifatida yaratilishi mumkin;
- ❖ Bajariladigan jarayon bir yoki bir nechta oqimlarni o'z ichiga olishi mumkin.
- ❖ Jarayon va oqim obyektlari o'rnatilgan (ajralmas) sinxronizatsiya imkoniyatlariga ega.

Jarayon tomonidan boshqariladigan yoki foydalaniladigan resurslar va ular bilan qanday bog'liqligi 9.8- rasmda ko'rsatilgan.



9.8- rasm. Windows jarayoni va uning resurslari

Har bir jarayonga protseduraning asosiy tokeni deb nomlangan, xavfsizlik kirish tokenlari tayinlanadi. Foydalanuvchi birinchi marta tizimga kirganda, Windows foydalanuvchi uchun xavfsizlik identifikatorini o‘z ichiga olgan aniq bo‘lmagan tokenni yaratadi.

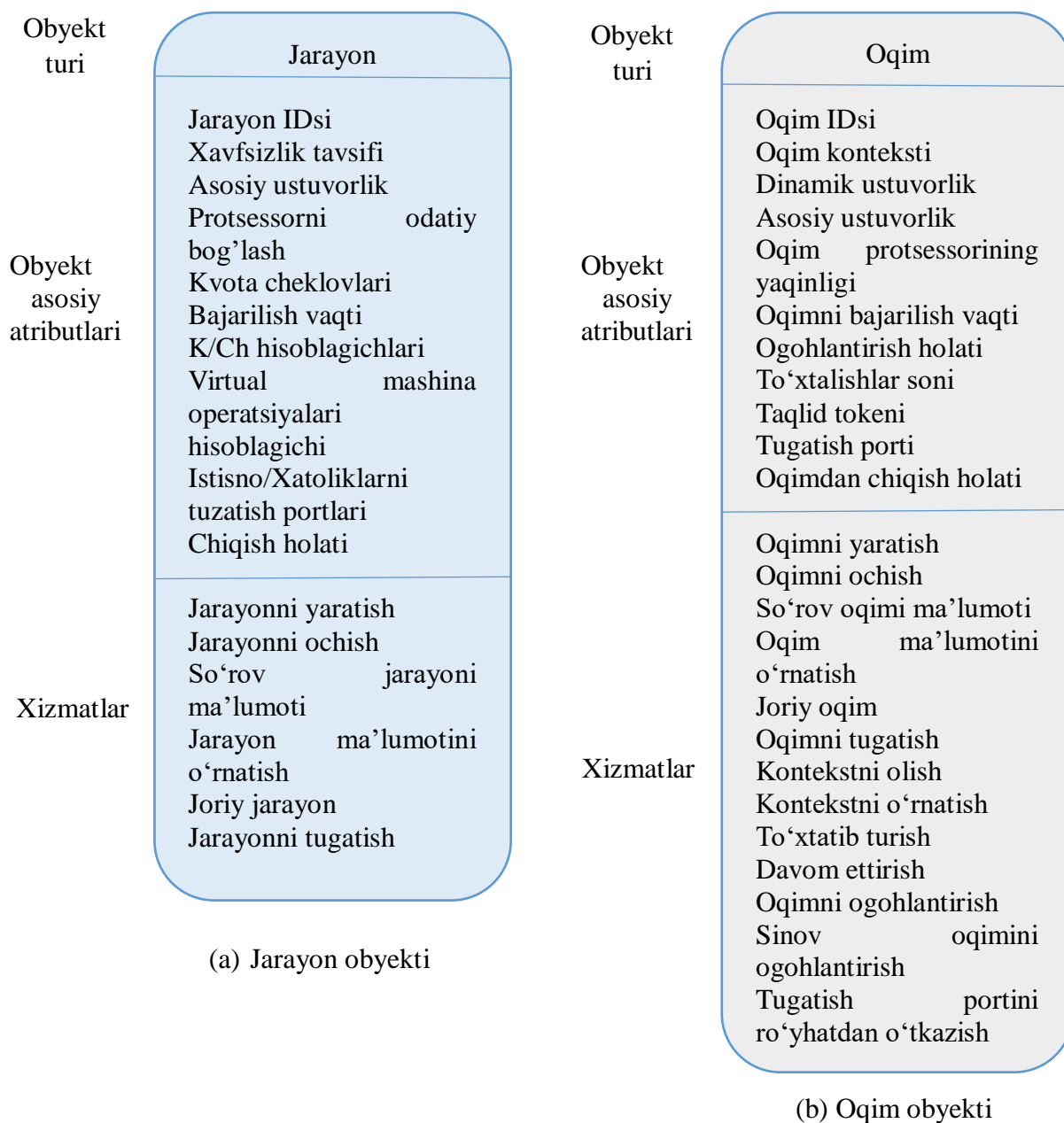
Foydalanuvchi birinchi marta tizimga kirganda, Windows foydalanuvchi uchun xavfsizlik identifikatorini o‘z ichiga olgan kirish tokenini yaratadi.

Ushbu foydalanuvchi nomidan yaratilgan yoki bajariladigan har bir jarayon ushbu kirish tokenining nusxasiga ega. Windows foydalanuvchini himoyalangan obyektlarga kirish yoki tizimda va himoyalangan obyektlarda cheklangan funksiyalarni bajarish imkoniyatini tekshirish uchun tokendan foydalanadi.

### ***Jarayon va oqim obyektlari***

Windows jarayon bilan bog‘liq obyektlarning ikki turidan foydalanadi: jarayonlar va oqimlar. Jarayon - bu foydalanuvchi vazifasi yoki xotira va ochiq fayllar kabi resurslarga ega bo‘lgan dasturga mos keladigan obyekt.

Oqim bu ketma-ket ishlaydigan va uzilib qolgan vazifaning taqsimlangan birligi, shuning uchun protsessor boshqa oqimga o'tishi mumkin.



(a) Jarayon obyekti

(b) Oqim obyekti

### 9.9- rasm. Windowsda jarayon (a) va oqim (b) obyektlari

Windows ning har bir jarayoni 9.9(a)- rasmda ko'rsatilgan umumiy tuzilishi ko'rsatilgan obyekt bilan ifodalanadi. Har bir jarayon bir qator atributlar bilan belgilanadi va bajarilishi mumkin bo'lgan harakatlar yoki xizmatlar soni bilan belgilanadi. Jarayon interfeysning e'lon qilingan usullari to'plami orqali chaqirilganda xizmatni amalga oshiradi. Windows yangi jarayonni yaratganda, u

yangi obyekt namunasini yaratish uchun shablon sifatida Windows jarayoni uchun belgilangan obyekt sinfini yoki turini ishlatadi. Yaratish vaqtida atribut qiymatlari belgilanadi. 9.1- jadvalda jarayon obyekti uchun har bir obyekt atributlarining qisqacha ta'rifi berilgan. Windows jarayonida bajarish uchun kamida bitta oqim bo'lishi kerak. Ushbu oqim keyinchalik boshqa oqimlarni yaratishi mumkin. Ko'p protsessorli tizimda bir xil jarayonning bir nechta oqimlari parallel ravishda bajarilishi mumkin. 9.2- jadvalda oqim obyekti atributi qiymati ko'rsatilgan. Masalan, oqim protsessorining o'xshashligi - bu oqimni bajaradigan ko'p protsessorli tizimdagi protsessorlar to'plamidir; ushbu to'plam jarayon protsessorining yaqinligiga teng yoki pastki qismidir. Oqim obyektlarining atributlaridan biri, oqimning so'nggi boshlanishida protsessor registrlari qiymatlarini o'z ichiga olgan kontekst ekanligiga e'tibor bering. Ushbu ma'lumot oqimlarni to'xtatib turish va davom ettirishga imkon beradi. Bundan tashqari, oqim to'xtatilganda uning kontekstini o'zgartirib, oqim holatini o'zgartirish mumkin.

9.1- jadval. Windowsda jarayon obyekti atributlari

Jarayon IDsi	Operatsion tizim uchun jarayonni aniqlaydigan noyob qiymat
Xavfsizlik tavsifi	Obyektni kim yaratganligini, unga kim kirishi yoki undan foydalanishi mumkinligini va kimga obyektga kirish huquqi berilmaganligini tavsiflab beradi
Asosiy ustuvorlik	Jarayon oqimlari uchun asosiy ishga tushirish ustuvorligi
Protsessorni standart bog'lash	Jarayon ishga tushiradigan standart protsessorlar to'plami
Kvota cheklovlari	Tizimning sahifali va sahifali bo'lmagan maksimal xotirasi, foydalanuvchi fayllarini ishlatishi mumkin bo'lgan fayl maydoni va protsessor vaqti
Bajarilish vaqti	Jarayonda barcha oqimlarning bajarilish umumiy vaqti
K/Ch hisoblagichlari	Jarayon oqimlari bajargan kiritish/chiqarish operatsiyalari soni va turini yozadigan o'zgaruvchilar

Virtual mashina operatsiyalari hisoblagichi	Jarayon oqimlari bajargan virtual xotira operatsiyalarining soni va turlarini yozadigan o'zgaruvchilar
Istisno/Xatoliklarni tuzatish portlari	Jarayon menejeri jarayon oqimlaridan biri istisno qilsa, xabar yuboradigan jarayonlararo aloqa kanallari. Odatda ular mos ravishda muhit quyi tizimi va xatoliklarni tuzatish jarayonlari bilan bog'liq
Chiqish holati	Jarayonni tugatish uchun sabab

## 9.2- jadval. Windowsda oqim obykti atributlari

Oqim IDsi	serverga murojaat qilganda oqimni aniqlaydigan noyob qiymat
Oqim konteksti	Tizimning ishlash holatini belgilaydigan registr qiymatlari va boshqa o'zgaruvchan ma'lumotlar to'plami
Dinamik ustuvorlik	Har qanday vaqtda oqimning bajarilish ustuvorligi
Asosiy ustuvorlik	Oqimning dinamik ustuvorligining pastki chegarasi
Oqim protsessorining yaqinligi	Oqim bajarilishi mumkin bo'lgan protsessorlar to'plami, bu oqimlar to'plami yoki protsessorning oqim jarayoni uchun barcha yaqinligi
Oqimni bajarilish vaqti	Foydalanuvchi rejimida va yadro rejimida bajarilgan umumiy vaqt
Ogohlantirish holati	Kutayotgan oqimning asinxron protsedura chaqirig'ini bajarilayotganini ko'rsatuvchi bayroq
To'xtalishlar soni	Qayta tiklanmagan holda, ushbu oqim bajarilishi necha marta to'xtatilganligi soni
Taqlid tokeni	Tizimga boshqa jarayon nomidan operatsiyalarni bajarishga imkon beradigan vaqtinchalik kirish tokeni (quyi tizimlar tomonidan ishlatiladi)
Tugatish porti	Jarayonlar menejeri oqim tugashi bilan xabar yuboradigan jarayonlararo aloqa kanali (quyi tizimlar tomonidan ishlatiladi)
Oqimdan chiqish holati	Oqimni tugatish uchun sabab



**Ko‘p oqimlilik.** Windows jarayonlar o‘rtasidagi parallellashni qo‘llab-quvvatlaydi, chunki turli xil jarayonlardagi oqimlar bir vaqtning o‘zida ishlashi mumkin (ular bir vaqtning o‘zida ishlaydi). Bundan tashqari, bir xil jarayonda bir nechta oqimlarni alohida protsessorlarga ajratish mumkin va ular bir vaqtning o‘zida bajariladi (aslida bir vaqtning o‘zida ishlaydi). Ko‘p oqimli jarayon bir nechta jarayonlarni ishlatmasdan parallellashga erishadi. Xuddi shu jarayon ichidagi oqimlar umumiy manzillar oralig‘i orqali ma’lumot almashishlari va jarayonning umumiy resurslaridan foydalanishlari mumkin. Turli jarayonlardagi oqimlar ushbu jarayonning umumiy resurslari orqali ma’lumot almashishlari mumkin. Turli jarayonlardagi oqimlar ikki jarayon o‘rtasida o‘rnatilgan umumiy xotira orqali ma’lumot almashishi mumkin.

Obyektga yo‘naltirilgan ko‘p oqimli jarayon server dasturini amalga oshirishning samarali vositasidir. Masalan, bitta server jarayoni bir nechta mijozlarga bir vaqtning o‘zida xizmat ko‘rsatishi mumkin.

**Oqim holatlari.** Windowsda oltita holatlardan biri bo‘lishi mumkin.

**Tayyorlilik (Ready):** tayyor oqimni bajarish uchun rejalashtirish mumkin. Yadro menejeri barcha tayyor oqimlarni kuzatadi va ularni ustuvorlik tartibida rejalashtiradi.

**Zaxira (Standby):** keyingi belgilangan protsessorida ishga tushirish uchun zaxira oqim tanlangan. Ushbu protsessor mavjud bo‘lmaguncha oqim bu holatda kutadi. Agar zaxira oqimning ustuvorligi yetarlicha yuqori bo‘lsa, ushbu protsessorida ishlaydigan oqim zaxira oqim foydasiga ishlatilishi mumkin. Aks holda, zaxira oqimi ishlayotgan oqim bloklanmaguncha yoki o‘zining vaqt oralig‘ida tugamaguncha kutadi.

**Bajarilish (Running):** yadro menejeri oqimni almashtirishni tugatgandan so‘ng, zaxira oqimi bajarilish holatiga o‘tadi va bajarishni boshlaydi va bajarishni o‘zining vaqt oralig‘i tugamaguncha, bloklanmaguncha yoki vazifa tugatilmaguncha davom ettiradi. Dastlabki ikki holatda u tayyorlilik holatiga o‘tadi.

**Kutish (Waiting):** Oqim qachon kutish holatiga o‘tadi, (1) hodisa uchun bloklangan bo‘lsa (masalan, kiritish/chiqarish), (2) u ixtiyoriy ravishda sinxronizatsiya maqsadalarini kutsa yoki (3) muhitning quyi tizimi oqimning o‘zini to‘xtatib turishga yo‘naltirsa.

**O'tish (Transition):** Oqim ishga tushurilish uchun tayyor bo'lsa, lekin resurslar mavjud bo'lmasa, u kutish holatidan so'ng ushbu holatga o'tadi. Resurslar mavjud bo'lganda, oqim tayyorlik holatiga o'tadi.

**Tugatish (Terminated):** Oqim o'zini-o'zi tugatishi mumkin, boshqa oqim tomonidan, yoki uning ona jarayoni yakunlansa tugatilishi mumkin. Oqimni bajarish tugallangandan so'ng, oqim tizimdan chiqariladi yoki bajaruvchi tomonidan qayta ishga tushirilishi uchun saqlab qo'yilishi mumkin.

### ***OT quyi tizimini qo'llab-quvvatlash***

Jarayonlar va oqimlarning universal vositalari turli xil OT muhitlarining aniq jarayonlari va oqim tuzilmalarini qo'llab-quvvatlashi kerak. Har bir OT quyi tizimi tegishli operatsion tizimning jarayonlari va oqimlari imkoniyatlarini taqlid qilish uchun Windows jarayonlari va oqimlarining imkoniyatlaridan foydalanish uchun javobgardir. Jarayon/oqimlarni boshqarishning ushbu sohasi juda murakkab va bu yerda biz faqat umumiy sharhni beramiz.

Jarayonni yaratish dasturdan yangi jarayonni talab qilish bilan boshlanadi. Ilova jarayonni yaratish uchun so'rovni tegishli himoyalangan quyi tizimga yuboradi va so'rovni bajaruvchi organga yuboradi. Bajaruvchi jarayon obyektini yaratadi va quyi tizimda ushbu obyektga deskriptorni qaytaradi. Windows jarayonni yaratganda, u avtomatik ravishda oqimni yaratmaydi. Win32 holatida har doim boshlang'ich oqim bilan yangi jarayon yaratilishi kerak. Shu sababli, Win32 quyi tizimi uchun Windows jarayon menejeri yangi jarayon uchun oqimni Windows-dan qaytarib olish uchun yana chaqiriladi. Tegishli oqim va jarayon haqida ma'lumot dasturga qaytariladi. POSIX holatida oqimlar qo'llab-quvvatlanmaydi. Shuning uchun, POSIX quyi tizimi Windows-dan yangi jarayon uchun oqimni oladi, va shunda jarayon faollashtirilishi mumkin, faqat jarayonga oid ma'lumotni dasturga qaytaradi. Bajaruvchi tomonidan yangi jarayon yaratilganda, yangi jarayon uning yaratilish jarayonidan ko'plab o'ziga xos xususiyatlarni meros qilib oladi. Biroq, Win32 muhitida bu jarayonni yaratish noto'g'ri bajariladi. Ilova mijoz jarayoni Win32 quyi tizimiga jarayonni yaratish so'rovini yuboradi; keyin quyi tizim Windows-ning bajaruvchisiga jarayon haqida so'rov yuboradi. Kutilayotgan natija shundan iboratki, yangi jarayon server jarayoniga

emas, balki mijoz jarayonining xususiyatlariga ega bo‘ladi, Windows quyi tizimga yangi jarayonning onasini belgilashga imkon beradi. Keyin yangi jarayon onaning kirish tokenini, kvota chegaralarini, asosiy ustuvorlikni va standart protsessor bilan bog‘lanishni meros qilib oladi.

### ***Windowsda xotirani boshqarish***

Windows virtual xotira menejeri xotira qanday taqsimlanishi va almashtirish (paging) qanday bajarilishini boshqaradi. Xotira menejeri turli platformalarda ishlash va 4 Kbaytdan 64 Kbaytgacha bo‘lgan sahifalar hajmidan foydalanish uchun mo‘ljallangan. Intel va AMD64 platformalarida har bir sahifada 4 Kbayt, Intel Itanium platformalarida esa har bir sahifa 8 Kbayt dan tashkil topadi.

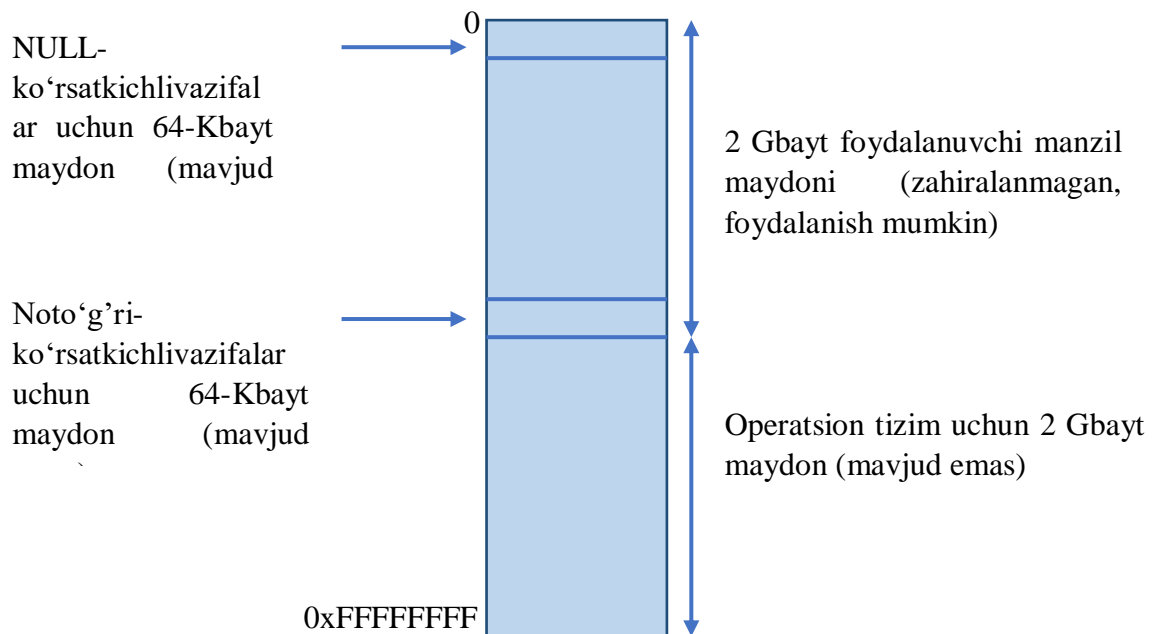
**Windowsda virtual manzil xaritasi.** 32-razryadli platformalarda, har bir Windows foydalanuvchi jarayoni alohida 32-razryadli manzil maydonini ko‘radi va har bir jarayon uchun 4 Gb virtual xotirani ta‘minlaydi. Odatda, ushbu xotiraning yarmi OT uchun ajratilgan (zaxiralangan), shuning uchun har bir foydalanuvchi amalda 2 Gb virtual manzil maydoniga ega va barcha jarayonlar yadro rejimida ishlaganda yuqori 2 Gb tizim maydonining ko‘p qismini birgalikda ishlatadi. Mijozlar va serverlarda katta hajmdagi xotiraga ega ilovalar Windowsning 64-razryadli versiyasidan foydalangan holda yanada samarali ishlashi mumkin. Netbuklarga qo‘shimcha ravishda, zamonaviy kompyuterlarning aksariyati 32-razryadli yoki 64-razryadli tizim sifatida ishlashi mumkin bo‘lgan AMD64 protsessor arxitekturasidan foydalanadi. 9.10- rasmda odatiy 32-razryadli foydalanuvchi jarayoniga ko‘rinadigan standart virtual manzil maydoni ko‘rsatilgan. U to‘rtta maydondan iborat:

**0x00000000 dan 0x0000FFFF:** Dasturchilarga NULL-ko‘rsatkich vazifalarini aniqlashga yordam berish uchun ajratilgan.

**0x00010000 dan 0x7FFEFFFF:** Mavjud foydalanuvchi manzil maydoni. Bu maydon asosiy xotiraga yuklanishi mumkin bo‘lgan sahifalarga bo‘lingan.

**0x7FFF0000 dan 0x7FFFFFFF:** foydalanuvchi kirishi mumkin bo‘lma-gan himoya sahifasi. Ushbu sahifa OT uchun mavjud ko‘rsatkichdan (chegaradan) tashqaridagi havolalarni tekshirishni osonlashtiradi.

**0x80000000 dan 0xFFFFFFFF:** tizim manzillari maydoni. Ushbu 2 Gbayt jarayon Windows bajaruvchisi (Executive), yadro, HAL (hardware abstraction layer) va qurilma drayverlari uchun qoʻllaniladi. 64-razryadli platformlarda, Windows 7 da 8 Tbayt foydalanuvchi manzil maydoni mavjud.



9.10- rasm. Windowsda standart 32-razryadli virtual manzillar maydoni

**Windowsda almashtirish (paging).** Jarayon yaratilganda, u odatda, deyarli butun 2 Gb (yoki 64-razryadli Windowsda 8 TB) foydalanuvchi maydonidan foydalanishi mumkin. Ushbu maydon belgilangan oʻlchamdagi sahifalarga boʻlinadi, ularning har birini asosiy xotiraga oʻtkazish mumkin, ammo OT 64 Kbayt chegaralarida belgilangan qoʻshni maydonlardagi manzilni boshqaradi. Maydon uchta holatdan bittasida boʻlishi mumkin:

- ❖ Mavjud (available): hozirda ushbu jarayon tomonidan ishlatilmaydigan manzillar;

- ❖ Zahiralangan (reserved): virtual xotira menejeri tomonidan jarayon uchun ajratilgan manzil, boshqa maqsadlarda ajratilmasligi uchun zahiralangan (masalan, stek oʻsishi uchun qoʻshni joyni tejash);

- ❖ Ajratilgan (committed): virtual xotira menejeri tomonidan virtual xotira sahifalariga kirish uchun jarayon tomonidan foydalanish

uchun ajratilgan manzillar. Ushbu sahifalar diskda ham, fizik xotirada ham joylashishi mumkin. Diskda bo'lganda ularni fayllarda (ko'rsatilgan sahifalarda) saqlash mumkin yoki almashtirish faylidan joy egallashi mumkin (ya'ni, asosiy xotiradan o'chirilganda, sahifalar yoziladigan disk fayli). Zahiralangan va ajratilgan xotira o'rtasidagi farq foydalidir, chunki u (1) tizim tomonidan talab qilinadigan virtual xotira maydonining hajmini kamaytiradi va sahifa faylini kichikroq qilish imkonini beradi; va (2) dasturlarga manzillarni dasturga kiritmasdan yoki ularni resurs kvotalarini to'ldirmasdan zahiralash imkonini beradi.

Jarayon birinchi marta faollashtirilganda, uning ishchi to'plamini boshqarish uchun ma'lumotlar tuzilmalari tayinlanadi. Jarayon uchun zarur bo'lgan sahifalar fizik xotirada saqlanganda, xotira menejeri jarayonga tayinlangan sahifalarni kuzatish uchun ma'lumotlar tuzilmalaridan foydalanadi. Faol jarayonning ishchi to'plamlari quyidagi umumiy kelishuvlar yordamida tuzilgan:

Asosiy xotira ko'p bo'lganda, virtual xotira menejeri faol jarayonlarning ishchi to'plamlari ko'payishiga yordam beradi. Buning uchun, sahifada xatolik bo'lganda, jarayonga yangi fizik sahifa qo'shiladi, ammo eski sahifalar yuklanmaydi, bu esa ushbu jarayonning doimiy to'plamini bitta sahifaga ko'paytirishga olib keladi.

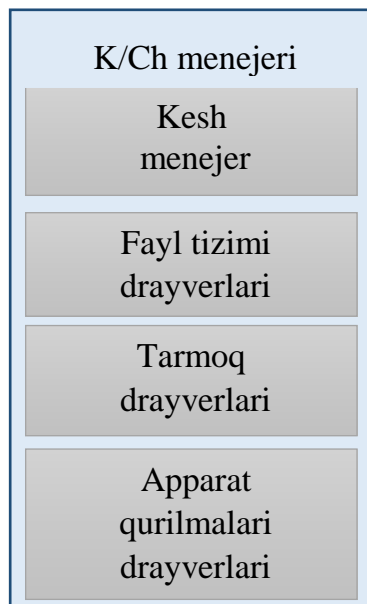
Xotira yetishmovchiligi yuzaga kelganda (kamayganda), virtual xotira menejeri tizim uchun xotirani tiklaydi, yaqinda ishlatilgan sahifalarni faol jarayonlarning ishchi to'plamlaridan chiqarib tashlaydi va shu to'plamlarning hajmini kamaytiradi.

Xotira ko'p bo'lsa ham, Windows xotiradan foydalanish sarfini oshiradigan katta jarayonlarni kuzatadi. Tizim yaqin orada foydalanilmagan sahifalarni jarayondan olib tashlashni (o'chirishni) boshlaydi. Ushbu siyosat tizimni yanada sezgir qiladi, chunki yangi dastur to'satdan xotira yetishmasligini keltirib chiqarmaydi va tizim allaqachon ishlayotgan jarayonlarning doimiy to'plamlarini kamaytirishga harakat qilganda foydalanuvchini kutishga majbur qiladi.

### ***Windows da kiritish/chiqarish qurilmalari***

9.11- rasmda Windows K/Ch menejeri bilan bog'liq bo'lgan yadro rejimi-ning asosiy komponentlari ko'rsatilgan. K/Ch menejeri

operatsion tizim uchun barcha K/Ch uchun javobgar va barcha turdagi drayverlarga chaqiriqlarni amalga oshirishlari uchun yagona ko‘rinishdagi interfeysni ta’minlaydi.



9.11- rasm. Windows K/Ch menejeri

### ***Kiritish/chiqarish qurilmalarining asosiy imkoniyatlari***

K/Ch menejeri to‘rt turdagi yadro komponentlari bilan yaqindan ishlaydi.

**Kesh menejeri:** Kesh menejeri barcha fayl tizimlari uchun fayllarni keshlashni boshqaradi. U mavjud fizik xotira hajmiga qarab ma’lum bir fayl uchun taqsimlangan kesh hajmini dinamik ravishda oshirishi va kamaytirishi mumkin. Tizim yangilanishlarni diskda emas, balki, faqatgina keshda saqlaydi. Yadro oqimi diskka yozish uchun vaqti-vaqti bilan yangilanishlarni paketlaydi. Yangilanishlarni paketlarga yozish orqali K/Ch samaradorligini yanada oshirish imkonini beradi.

**Fayl tizimi drayverlari:** K/Ch menejeri fayl tizimi drayverini boshqa qurilma drayveri kabi qabul qiladi va fayl tizimining qismini (volume) kiritish/chiqarish so‘rovlarini ushbu qism uchun tegishli dastur drayveriga yo‘naltiradi. Fayl tizimi, o‘z navbatida, qurilma vositasi adapterini boshqaruvchi dastur drayverlariga kiritish/chiqarish so‘rovlarini yuboradi.

**Tarmoq drayverlari:** Windows birlashtirilgan tarmoq imkoniyatlari va masofaviy fayl tizimlarini qo‘llab-quvvatlashni o‘z ichiga oladi. Vositalar Windows bajaruvchisining bir qismi emas, balki dasturiy ta‘minot drayveri sifatida amalga oshiriladi.

**Qurilma drayverlari:** Ushbu dastur drayverlari qurilma abstraktsiyasi qatlami (HAL) orqali kirish joylaridan foydalanib, yotdamchi qurilma registrlariga kirishadi. Ushbu protseduralar to‘plami Windows tomonidan qo‘llab-quvvatlanadigan har bir platforma uchun mavjud.

### ***Asinxron va Sinxron K/Ch***

Windows K/Ch operatsiyalarining ikkita modelini taklif qiladi: asinxron va sinxron. Ilovaning ishlashini optimallashtirish uchun iloji boricha asinxron rejim ishlatiladi. Asinxron K/Ch bilan dastur K/Ch operatsiyasini boshlaydi va keyin K/Ch so‘rovi bajarilguncha qayta ishlashni (bajarishni) davom ettirishi mumkin. Sinxron kiritish/chiqarishda dastur kiritish/chiqarish operatsiyasi tugaguncha bloklanadi. Asinxron K/Ch chaqiriq oqimi nuqtai nazaridan samaraliroq hisoblanadi, chunki u K/Ch operatsiyasini K/Ch menejeri tomonidan navbatga qo‘yilgan va keyinchalik bajariladigan oqimni bajarishni davom ettirishga imkon beradi. Shu bilan birga, asinxron K/Ch operatsiyasini chaqirgan dastur operatsiya qachon tugashini aniqlash uchun ba‘zi usullarni talab qiladi. Windows K/Ch ni tugaganligini bildirish uchun quyidagi besh xil signal usulini taqdim etadi:

- ❖ Fayl obykti signali (Signaling the file object);
- ❖ Hodisa obykti signali (Signaling an event object);
- ❖ Asinxron protsedura chaqirig‘i (Asynchronous procedure call);
- ❖ K/Ch ni tugatish portlari (I/O completion ports);
- ❖ So‘rov (Polling).

### ***RAID (Redundant Arrays of Independent Disks) dasturi***

Windows ikki xil RAID konfiguratsiyasini qo‘llab-quvvatlaydi, va ular quyidagicha aniqlanadi:

**Qurilma RAID:** diskni saqlash qurilmasi shkafi yoki kontroller qurilmasi orqali bir yoki bir nechta mantiqiy disklarga birlashtirilgan fizik disklar.

Dasturiy ta'minot RAID: xatoliklarga bardoshli FTDISK diskklari dasturiy ta'minot drayveri tomonidan bir yoki bir nechta mantiqiy qismlarga birlashtirilgan uzluksiz disk maydoni.

### ***Qismlarni shifrlash (Volume Encryption)***

Windows BitLocker deb nomlangan xususiyatdan foydalanib, butun qismlarni shifrlashni qo'llab-quvvatlaydi. Bu alohida fayllarni shifrlashdan ko'ra xavfsizroqdir, chunki butun tizim ma'lumotlar xavfsizligini ta'minlash uchun ishlaydi. Kriptografik kalitni taqdim etishning uchta turli xil usullari ta'minlanishi mumkin, bu bir-biriga bog'liq bo'lgan xavfsizlik darajalaridan foydalanishga imkon beradi.

### ***Windowsda fayllarni boshqarish***

Windows NT ni ishlab chiquvchilari ishchi stansiyalar va serverlarning yuqori talablarini qondirish uchun mo'ljallangan yangi fayl tizimini, yangi fayl tizimi texnologiyasini (NTFS – New Technology File System) ishlab chiqdilar. Yuqori darajadagi ilovalarga quyidagi misollarni keltirishimiz mumkin: klient-server dasturlari, masalan, fayl serverlari, hisoblash serverlari va ma'lumotlar bazasi serverlari. Windows operatsion tizimida hozirgi kunda NTFS (NT oilasi operatsion tizimlarida) fayl tizimlaridan foydalaniladi. Bundan oldingi fayl tizimi esa bu FAT (bir nechta versiyalari mavjud) fayl tizimi hisoblanadi. Biz Windows operatsion tizimlarida qo'llaniladigan ushbu fayl tizimlarini ko'rib chiqamiz.

### ***FAT fayl tizimi***

FAT fayl tizimi osti formatlangandagi mantiqiy qismlar tarkibi 9.12- rasmda keltirilgan.

- ❖ Yuklaydigan sektor operatsion tizimni yuklaydigan dasturni o'z ichiga oladi. Bu dasturning ko'rinishi operatsion tizim turiga bog'liq, ya'ni ushbu bo'limda yuklanadigan OT.

- ❖ FAT jadvalining asosiy nusxasi diskga fayl va katalogni joylashtirish to'g'risida axborotni o'z ichiga oladi.

- ❖ FAT jadvalining zahira nusxasi.

- ❖ Ildiz katalog katalogdagi har bir yozuv 32 baytdan iborat bo'lgan fayllar va kataloglar to'g'risida 512 ta yozuvni saqlashga yordam beradigan 32 sektorda (16 Kbait) o'lchamli belgilangan soha bilan shug'ullanadi.



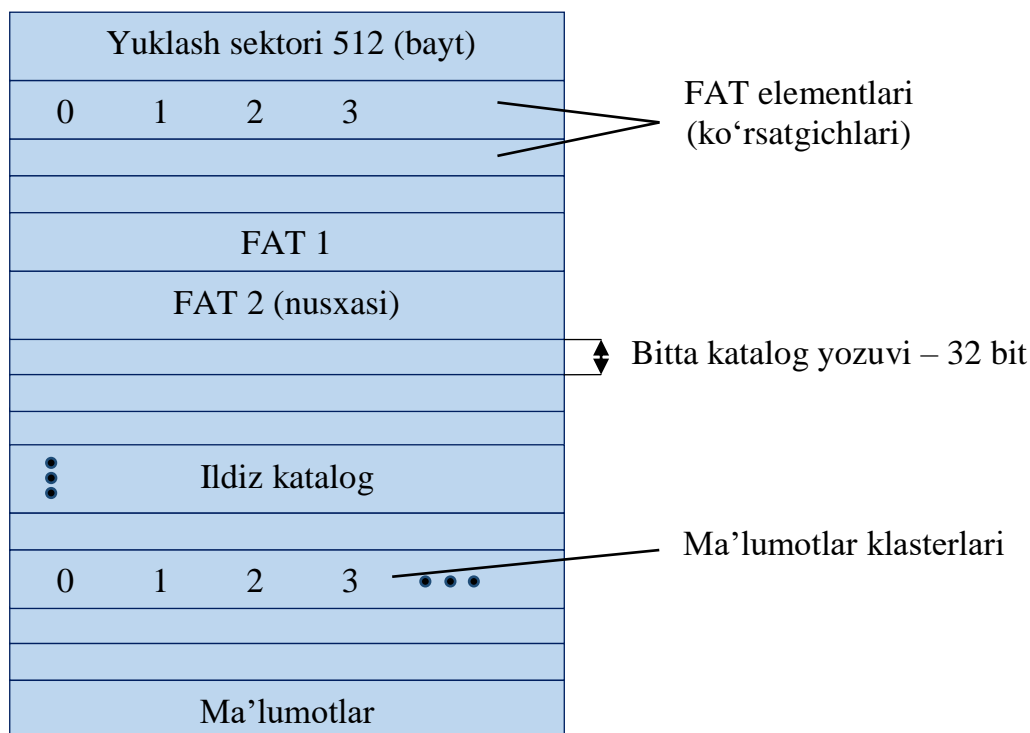
❖ Ma'lumotlar joylashgan joy ildizli katalogdan tashqari barcha fayl va barcha kataloglarni joylashtirish uchun mo'ljallangan.

FAT fayl tizimi ikki turdagi faylni qo'llab-quvvatlaydi: oddiy fayl va katalog. Fayl tizimi faqat ma'lumotlarni xotirada taqsimlaydi.

FAT jadvali(asosiy va zahira nusxalari) ma'lumotlar klasteri miqdoriga teng miqdorda indeksni ko'rsatuvchi massivdan iborat. Klaster va indeksli ko'rsatkichlar bir xil ma'noga ega bo'ladi – nolinch klaster nolinch ko'rsatkichga mos keladi va boshqalar.

Indeksli ko'rsatkich klaster bilan bog'liq holatni tavsiflovchi quyidagi belgilarni qabul qilishi mumkin:

- ❖ Bo'sh klaster (foydalanilmaydigan);
- ❖ Faylda foydalanadigan klaster va faylning so'nggi klasteri bo'lmaydi – bu holatda indeksli ko'rsatkich faylning keyingi klaster raqamini o'z ichiga oladi;
- ❖ Faylning so'ngi klasteri;
- ❖ Nuqsonli klaster;
- ❖ Zahira klasteri.



9.12- rasm. FAT fayl tizimining fizik tuzilishi

FAT jadvali bo'limning barcha fayllari uchun umumiy bo'ladi. Boshlang'ich holatda bo'limlardagi barcha klasterlar bo'sh bo'ladi, va

barcha indeksli ko'rsatkichlar (zahira va nuqsonli bloklarga mos keluvchilardan tashqari) "klaster bo'sh" qiymatini qabul qiladi. OT faylni joylashtirishda FAT boshlanish boshini va birinchi bo'sh indeksli ko'rsatkichni izlashni ko'rib chiqadi. Uning raqami aniqlangandan so'ng bu ko'rsatkich katalog yozilgan birinchi klaster raqami maydonida belgilanadi. Bu raqam bilan klasterda fayl ma'lumotlari yoziladi, u faylning birinchi klasteri bo'ladi. Agar fayl bitta klasterda joylashsa u holda ushbu klasterga mos keluvchi faylning so'nggi klasterini identifikatsiyalovchi maxsus qiymatlarini oladi.

Agar faylning o'lchami bitta klasterdan katta bo'lsa, u holda OT FATni ko'rib chiqishni davom ettiradi va bo'sh klasterda keyingi ko'rsatkichni izlaydi. Uni aniqlagandan so'ng, avvalgi ko'rsatkich ushbu klaster raqamini oladi, ya'ni endilikda faylning navbatdagi klasteri bo'lib qoladi. Jarayon faylning barcha ma'lumotlari joylashmagunga qadar davom etadi. Bunday ko'rinishda faylning barcha klasterlari bilan bog'liqlik ro'yxati yaratiladi. Fayl formatlangandan (shakllantirilgandan) so'ng dastlab ma'lumotlar qismiga klasterlar ketma-ketlik bilan joylashtiriladi. Biroq bitta faylda olis masofada joylashgan fayl klasterlari miqdori aniqlangandan so'ng, boshqa fayl klasteri bilan almashadi.

FAT jadvali o'lchami va uning indeksli ko'rsatkichlarida foydalaniladigan razryadlar ma'lumotlar qismida klasterlar miqdorini belgilaydi. Fragmentatsiyada yo'qotishlarni kamaytirish uchun klasterlar unchalik katta bo'lmagani ma'qulroq, manzilli axborot hajmini qisqartirish va almashish tezligini oshirish uchun esa teskarisini qilish ma'qulroq bo'ladi. FAT fayl tizimi ostida diskni formatlash uchun odatda murosali yechim tanlaniladi va klaster o'lchami 1 dan 128 sektorgacha yoki 512 dan 64 Kbaytgacha oraliqdan tanlaniladi.

Shakl shubhasiz, indeksli ko'rsatkich razryadi ma'lum bir hajmli disk uchun klasterga maksimal raqamni berish mumkin. Indeksli ko'rsatkich razryadlariga taa'luqli bir nechta FAT mavjud va ular quyidagicha ifodalanadi: FAT12, FAT16 va FAT32. FAT 12 fayl tizimida 12 – razryadli ko'rsatkich disk ma'lumotlar qismida 4096 klasterni qo'llab-quvvatlashga yordam beradi, FAT16 da – 65536 klasterni qo'llab-quvvatlash uchun 16 razryadli va FAT32 – 4 milliarddan ortiq klasterlar uchun 32 razryadlilar ishlatiladi.

Nomi	Fayl deskriptori raqami
Prog 1	23
firelights	126
doc_23.txt	51
glazing.txt	17
lambda_good	

### 9.13- rasm. FATda fayllarni ko'rsatuvchi ro'yxat

FAT12 fayl tizimi odatda 16 Mbayt dan katta bo'lmagan hajmli diskarga to'g'ri keladi, 4 kbaytdan ko'proq klasterlarni foydalanmaslikka imkon beradi. Bu sabab tufayli FAT16 512 Mbaytdan katta bo'lmagan hajmli disklar uchun mo'ljallangan. Katta disklar uchun esa 8 Gbayt gacha hajmli disklar bilan ishlaydigan 4 Kbaytli klasterlardan foydalanadigan FAT32 ma'qul hisoblanadi va faqat katta hajmli disklar uchun 8, 16 va 32 kbaytlilar ishlatiladi. FAT16 bo'limining maksimal o'lchami 4 Gbayt bilan chegaralanadi bunday hajm har biri 64 Kbayt bo'yicha 65536 klasterni beradi, FAT32 bo'limi maksimal o'lchami chegaralanmagan – 32 Kbayt bo'yicha  $2^{32}$  klaster.

Indeksli ko'rsatkichli razryadda belgilangan FAT jadvali ma'lumotlar qismi hajmiga bog'liq o'lchamni qo'llaydi: diskda FAT fayl tizimidan olis masofada joylashgan faylda birinchi bayt maxsus belgilar kiritilgan katalog yozuviga mos keladi. Bu yozuv bo'sh bo'ladi, faylning barcha indeksli ko'rsatkichi "bo'sh klasterga" qo'shiladi. Katalog yozuvidagi qolgan ma'lumotlar ya'ni faylning birinchi klasteri raqami soni olis masofada joylashgan faylni yanglishib qayta tiklash uchun imkoniyat sifatida qoldirish uchun tegilmaydi. Olis masofada joylashgan FAT faylni qayta tiklash uchun ko'plab utilitalar mavjud bo'ladi.

FAT zahira nusxasi fayl bilan bog'liq bo'lgan barcha operatsiyalarda asosiy nusxa bilan sinxronizatsiyalanadi. Shu sababli zahira nusxa foydalanuvchi ish jarayonida yanglish rad etib bo'lmaydi. Zahira nusxa xotiraning asosiy sektorlari shikastlanganda yoki o'qiy olmagan holatda foydalaniladi.

Fayl to‘g‘risida manzilli axborotni saqlash usulidan FAT da foydalanish yuqori ishonchlikni bermaydi – bir joyda indeksli ko‘rsatkich ro‘yxati buzilishi mumkin. Masalan, tashqi elektromagnit halaqiti sababi bo‘yicha OT dasturiy kodi ishlashni rad etadi, faylni so‘ng barcha klasterlari to‘g‘risida axborot yo‘qoladi.

FAT12 va FAT16 fayl tizimlari 12 ta belgidan iborat fayl nomli operatsiyalarni bajaradi. FAT16 versiyada Windows NT operatsion tizimi yangi turdagi katalog yozuvini kiritadi – “uzun nom”, ya’ni 255 ta belgigacha uzunlikdagi nomdan foydalanishga imkon beradi, chunki nom uzunligidagi har bir belgi ikki baytli Unicode shaklida saqlanadi.

FAT12 va FAT16 fayl tizimlari shaxsiy kompyuterlar davrining birinchi o‘n yilligida keng tarqalgan operatsion tizimlarda - MS-DOS va Windows 3.x operatsion tizimlarida qo‘llanilishi tufayli keng tarqalgan. Biroq qattiq disk hajmining doimiy ravishda o‘sib borishi, hamda bu fayl tizimlari ishonchliligiga talablarning o‘sib borishi boshqa turdagi fayl tizimlari va FAT32 tizimini yaratilishiga olib keldi.

### ***NTFS fayl tizimi***

NTFS (*NT File System*) fayl tizimi FAT va HPFS (OS/2 uchun fayl tizimi) va o‘sha davrdagi boshqa mavjud fayl tizimlarini ishlab chiqishda orttirilgan tajriba hisobiga 1990-yillar boshlarida Windows NT OT uchun asosiy fayl tizimi sifatida ishlab chiqildi. Bugungi kunda NTFS Windows NT OT oilasining barcha versiyalarida, ya’ni, Windows NT 3.1 versiyasidan boshlab hozirgi kundagi Windows 8.1, Windows 10, Windows Server 2019 operatsion tizimlarida ham qo‘llanadi.

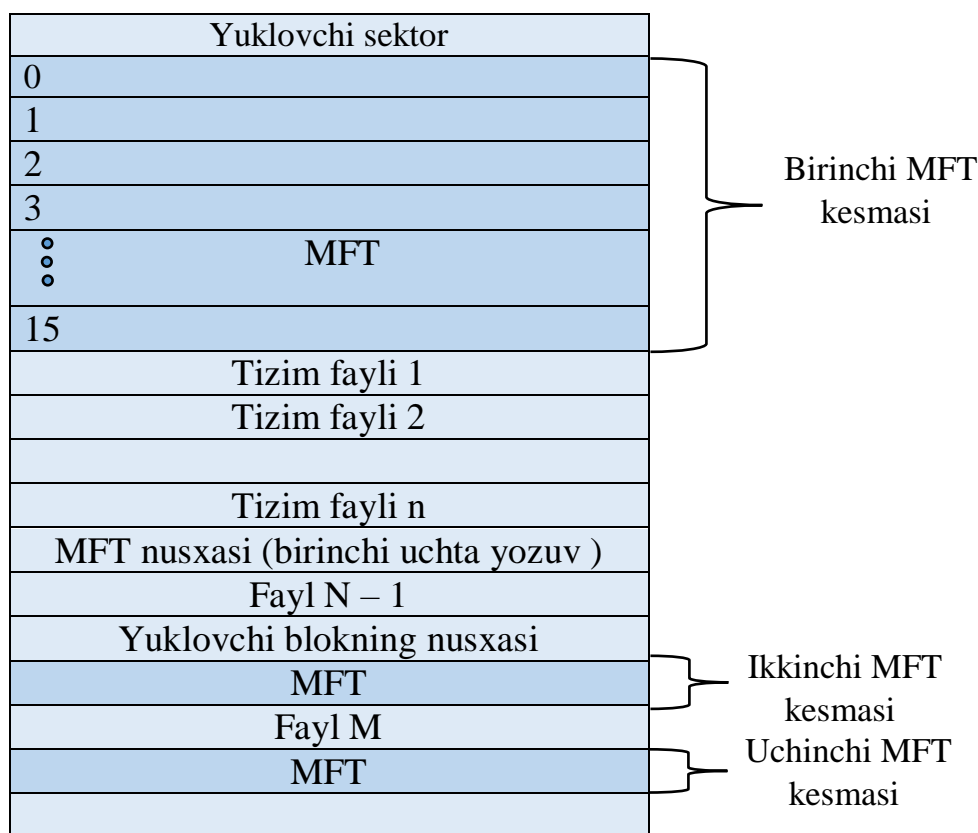
NTFS fayl tizimi xususiyatlari:

- ❖ Katta fayllarni va  $2^{64}$  baytgacha hajmdagi disklarni qo‘llab-quvvatlashi;
- ❖ Diskni boshqaruv apparaturasi va dasturi uzilish bo‘lganda va rad etilgandan so‘ng qayta tiklanishi;
- ❖ Katta o‘lchamli disklarda operatsiyalar tezligining yuqoriligi;
- ❖ Fragmentatsiyalashning past darajasi, ya’ni katta o‘lchamli disklarda;
- ❖ Yumshoq struktura, yangi yozuv turlarini qo‘shish hisobiga rivojlantirish va avvalgi FT versiyalari bilan muvofiq saqlaydigan fayl atributlari;

- ❖ Diskli tarqatgich rad etilishiga bardoshlilik;
- ❖ Alohida fayl va kataloglarga kirishni boshqarish.

### NTFS tuzilishi

FAT va s5/ufs bo‘limlaridan farqi NTFS yoxud faylni o‘zini yoxud fayl qismini taqdim etadi. NTFS asosiy tuzilishi bosh fayl jadvali (Master File Table, MFT) hisoblanadi. MFT har bir fayli disk hajmiga bog‘liq (1, 2 yoki 4 kbayt) belgilangan uzunlikka ega bo‘ladi. Katta o‘lchamli disklar uchun MFT yozuvi o‘lchami 2 Kbaytga teng, uni biz yozuv o‘lchami deb hisoblaymiz. NTFS fayllar MFT da fayllar holatini belgilovchi fayl raqamini identifikatsiyalaydi. Faylni identifikatsiya-lashning bu yo‘li s5 va ufs fayl tizimlarida foydalaniladigan yo‘lga yaqinroq. NTFS klasterlar ketma ketligidan iborat bo‘ladi. NTFS klasterlar tartib raqami klasterning mantiqiy raqami (Logical Cluster Number, LCN) deb ataladi. NTFS fayli ham klasterlar ketma-ketligidan iborat bo‘ladi, shu sababli faylning ichki klaster tartib raqami virtual klaster raqami (Virtual Cluster Number, VCN) deb ataladi.



9.14- rasm. NTFS fayl tizimi tuzilishi

NTFS da klaster tartib raqamlarini saqlash uchun 264 ta klastergacha fayl o'lchamini va diskni qo'llab-quvvatlash imkoniyatini beradigan 64 razryadli ko'rsatkichdan foydalanadi. Klasterning o'lchami 4 Kbayt - 64 milliard kilobaytdan iborat fayl va disklardan foydalanish imkonini beradi. NTFS ning tuzilishini 9.14-rasmda ko'rsatilgan. NTFS yuklovchi bloki bo'limlar boshida joylashgan, uni nusxasi esa – bo'lim o'rtasida joylashgan. Yuklovchi blok BIOS parametrlarining standart bloklarini, bo'limdagi bloklar miqdorini, shuningdek MFT asosiy nusxa boshlang'ich mantiqiy klaster raqami va MFT ochiq nusxasini o'z ichiga oladi.

So'ng birinchi bo'lakda MFT joylashadi u o'z ichiga NTFS fayl tizimi to'g'risida yozuvlarni shakllantirishga yordam beradigan 16 ta standart tashkil qiluvchilarni oladi.

#### NTFS da fayl tuzilishi

NTFS dagi har bir fayl va katalog atributlar to'plamidan iborat bo'ladi. Fayl nomi va uni ma'lumotlari fayl atributi sifatida ko'riladi. NTFS har bir atributi maydonlardan iborat bo'ladi: atribut turi, uzunligi, qiymati va imkoniyatlari. Atribut turi, uzunligi va nomi atribut sarlovhasida ko'rsatiladi. Tizimning mavjud atributlar to'plami NTFS tuzilishini ifodalaydi. Tizim atributlari belgilangan nomga va uning turi kodiga, hamda aniq bir formaga (shaklga) ega bo'ladi. Foydalanuvchilarni aniqlashda atributlarni qo'llash mumkin. Fayl atributlarini saqlashning ikki xil yo'li mavjud – MFT jadval yozuvida rezidentli saqlash va uni tashqi qismlarda norezidentli saqlash. Bunday ko'rinishda fayl rezidentli qismi rezidentli atributlardan iborat bo'ladi, norezidentli qismi esa – norezident atributlardan iborat bo'ladi. Saralash faqat rezidentli atributlar bo'yicha amalga oshirilishi mumkin.

Tizim to'plami quyidagi atributlarni o'z ichiga oladi:

❖ Atributlar ro'yxati (*Attribute List*) – har bir atribut joylashgan MFT yozuv raqamlari joyini ko'rsatuvchi manzilni o'z ichiga olgan fayldan iborat bo'ladi.

❖ Fayl nomi (*File Name*) – Unicode formatidagi fayl nomi uzunligi, hamda ona katalogi uchun MFT jadvaliga kiradigan raqam; agar bu fayl bir nechta kataloglardan iborat bo'lsa, u holda unda *File Name* turidagi bir nechta atributlar bo'ladi; bu atributlar har doim rezidentli bo'lishi kerak.

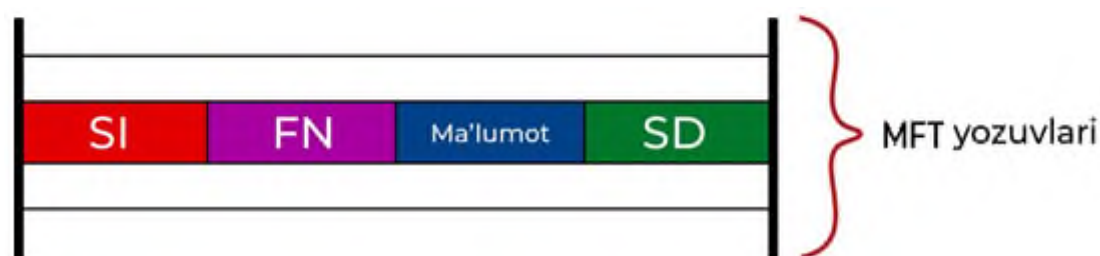
❖ MS-DOS nomi (*MS-DOS Name*) – fayl nomi;

- ❖ versiya (*Version*) – faylning so‘nggi versiya raqami;
- ❖ deskriptor xavfsizligi – kirish qoidalar ro‘yxatini (ACL) va ushbu faylda qanday xildagi operatsiyani ro‘yxatdan o‘tkazish kerakligini belgilovchi audit maydonini o‘z ichiga olgan himoya to‘g‘risidagi axborot;
- ❖ bo‘lim versiyasi (*Volume Version*) – faqat fayl bo‘limida foydalaniladi;
- ❖ bo‘lim nomi (*Volume Name*);
- ❖ ma‘lumotlar (*Data*) – odatiy fayl ma‘lumotlari;
- ❖ MFT bitli kartasi (*MFT bitmap*) – bo‘lim blokida foydalaniladigan xarita;
- ❖ Indeks ildizi (*Index Root*) – daraxt ildizi katalogdan faylni izlash uchun foydalaniladi;
- ❖ Indeksni joylashtirish (*Index Allocation*) – V daraxt indeksli ro‘yxat norezident qismi;
- ❖ Standart axborot (*Standard Information*) - bu atribut fayl to‘g‘risidagi qolgan barcha standart axborotlarni saqlaydi. Ya‘ni biron bir boshqa fayl atributi bilan bog‘lanish murakkab bo‘lgan holatlarda, masalan, faylni yaratish vaqti, yangilanish vaqti va boshqalar.

NTFS fayllari kichik, katta, juda katta va eng kata joylashtirish usullariga bog‘liq.

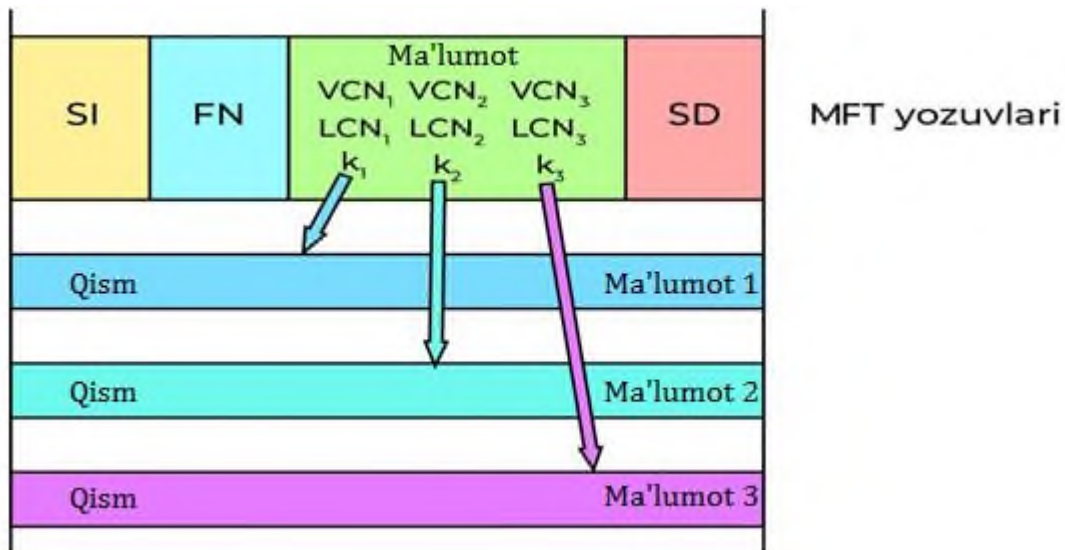
Kichik fayllar. Agar fayl kichik hajmga ega bo‘lsa, u holda u MFT ning bitta yozuvi ichida joylashtirilishi mumkin, masalan, 2 kbayt o‘lchamda. NTFS kichik fayllari quyidagi atributlardan iborat bo‘ladi (9.15- rasm).

- ❖ Standart axborot (Standard Information, SI);
- ❖ Fayl nomi (File Name, FN);
- ❖ Ma‘lumotlar (Data);
- ❖ Deskriptor xavfsizlik (Security Descriptor, SD).

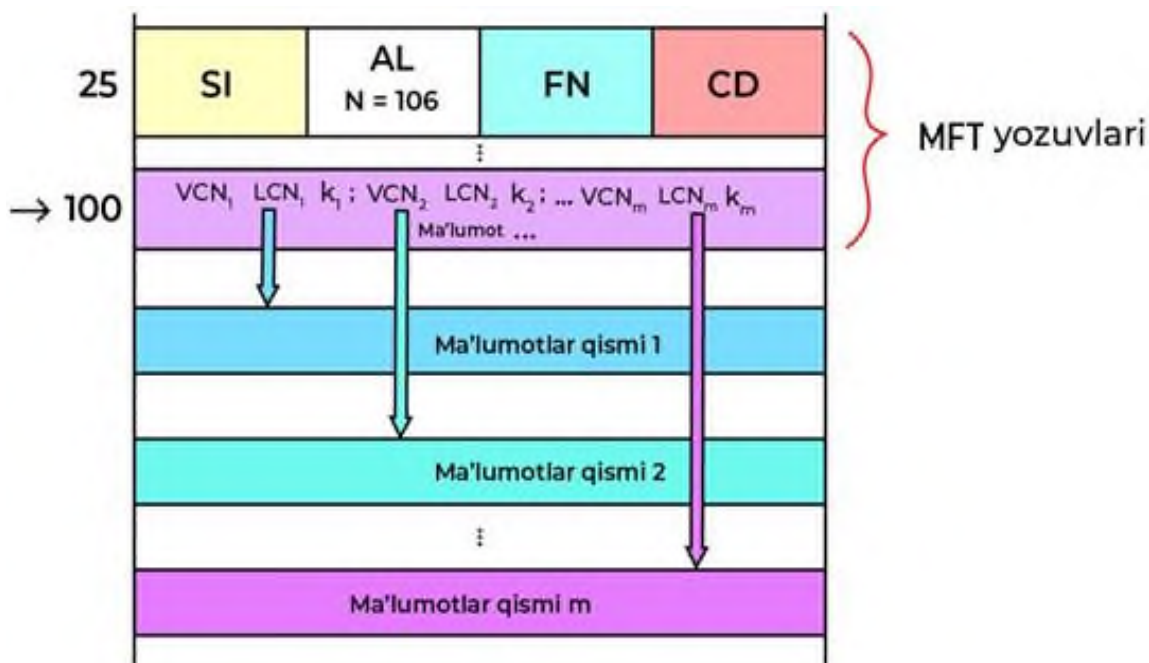


9.15- rasm. Katta bo‘lmagan NTFS fayli

Katta fayllar. Agar fayl ma'lumot MFT bitta yozuvida joylashmasa, u holda bu fakt ma'lumotlar atributi sarlavhasida ko'rsatiladi. Bu holatda ma'lumotlar atributi har bir ma'lumot qismida manzil axborotini o'z ichiga oladi (9.16- rasm).



9.16- rasm. Katta fayl



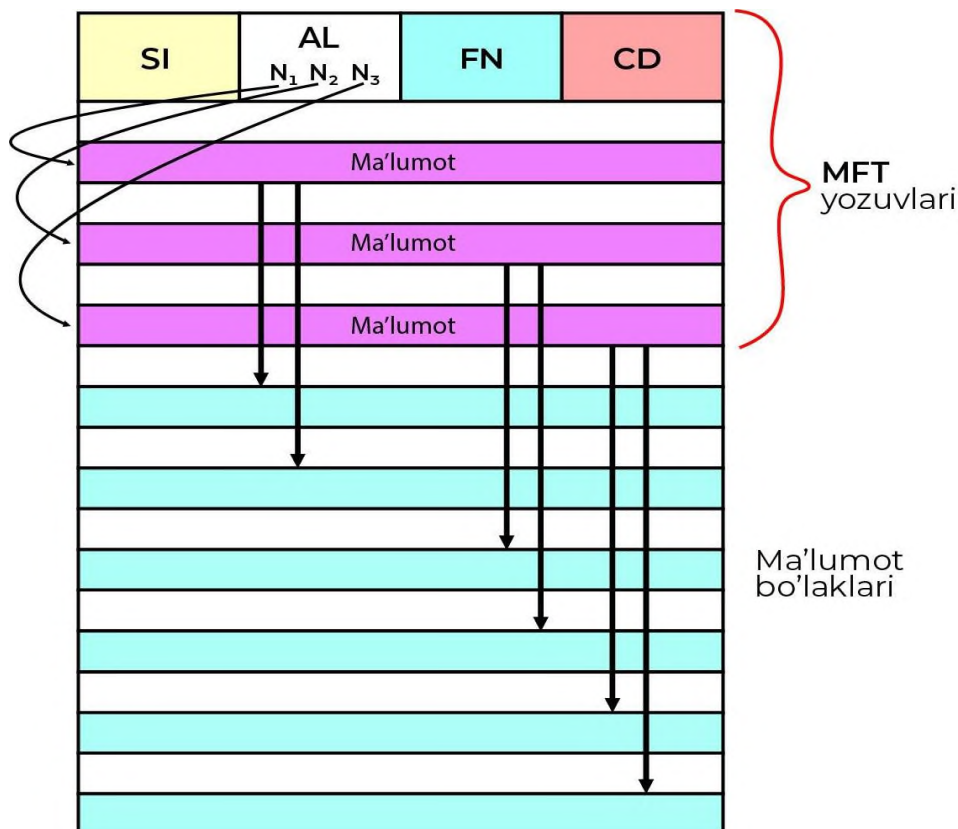
9.17- rasm. Juda katta fayl

Juda katta fayllar. Agar uning ma'lumotlar atributlari bir necha marotaba katta bo'lsa va bitta yozuvga joylashtirib bo'lmasa, u holda



bu atribut MFT ning boshqa yozuviga joylashtiriladi. Bunday atribut havola faylning asosiy yozuviga joylashtiriladi (9.17- rasm). Bu havola **Attribute List** atributini o‘z ichiga oladi. Ma’lumot atributining o‘zi esa ma’lumot norezident qismi manzilini o‘z ichiga oladi.

Eng katta fayllar. Eng katta fayl uchun **Attribute List** atributida MFTning qo‘shimcha yozuvlari joylashgan bir nechta atributlarni ko‘rsatadi (9.18- rasm).



9.18- rasm. Eng katta fayl

Bundan tashqari ikkilik yordamchi manzillashdan foydalanish mumkin. U holda norezident atribut boshqa norezident atributga ko‘rsatiladi, natijada NTFS da uzunlik tizim uchun atribut eng katta bo‘lmasligi ham mumkin.

### **Windows da xavfsizlik**

Kirishni boshqarish konsepsiyalariga yaxshi misol, bu Windows ga kirishni boshqarish vositasi bo‘lib, u obyektga yo‘naltirilgan tushunchalarni foydalanishni boshqarishning kuchli va moslashuvchan imkoniyatlarini ta’minlash uchun ishlatadi. Windows platformasining xavfsizligi har bir foydalanuvchi yoki foydalanuvchilar guruhi uchun

xavfsizlik modeliga asoslangan. Windows jarayonlar, oqimlar, fayllar, semaforalar, oynalar va boshqa obyektlar uchun qo'llaniladigan kirishni boshqarishning yagona mexanizmini ta'minlaydi. Kirishni boshqarish ikki obyekt tomonidan tartibga solinadi: har bir jarayon bilan bog'liq bo'lgan kirish tokeni va protsessorga kirish mumkin bo'lgan har bir obyekt bilan bog'liq bo'lgan xavfsizlik deskriptori.

Tizimda ro'yxatdan o'tgan har bir foydalanuvchi o'z hisobiga ega, bu foydalanuvchi haqidagi shaxsiy ma'lumotlarni o'z ichiga oladi. Tizim ushbu ma'lumotlardan foydalanuvchini autentifikatsiya qilish va domen resurslariga kirishga ruxsat berish uchun foydalanadi. Bundan tashqari, har bir tizim obyekti, jumladan, fayllar, printerlar, tarmoq xizmatlari, Active Directory konteynerlari va boshqalarga xavfsizlik identifikatori beriladi. Obyektning xavfsizlik identifikatori, obyektga kirish huquqini belgilaydi va qaysi foydalanuvchilar ushbu obyekt bilan muayyan xatti-harakatlarni amalga oshirishga ruxsat berilganligini aniqlaydigan kirishni boshqarish ro'yxatini (ACL - Access Control List) o'z ichiga oladi. Obyekt xavfsizlik identifikatori auditni o'tkazish uchun qanday hodisalarni belgilaydi. Windows avtorizatsiyasi obyektning xavfsizlik identifikatori bilan obyektning kirish qurilmasini moslashtirishga asoslanadi. Obyekt xususiyatlarini boshqarish orqali ma'murlar ruxsatlarni o'rnatishlari, egalik huquqini belgilashlari va foydalanuvchi ruxsatini kuzatishi mumkin.

Har bir xavfsizlik identifikatori ikki turdagi ro'yxatlarni o'z ichiga olishi mumkin:

❖ Tizimga kirishni boshqarish ro'yxati (SACL – System Access Control List), tizim darajasida obyekt uchun o'rnatilgan huquqlar (o'qish, yozish, bajarish va boshqalar) va cheklovlarni kuzatishga imkon beradi. Faqat tizim darajasida ruxsat huquqiga ega bo'lgan foydalanuvchilar ushbu ro'yxatga o'zgartirishlar kiritishi mumkin.

❖ Foydalanuvchi uchun ixtiyoriy kirishni boshqarish ro'yxati (DACL –Discretionary Access Control List), ushbu obyektning egasi tomonidan o'rnatilgan huquqlar va cheklovlarni kuzatishga imkon beradi. DACL obyektning hozirgi egasi sifatida ro'yxatga olingan foydalanuvchi tomonidan o'zgartirilishi mumkin. Har ikkala ro'yxat ham bir xil tuzilishga ega.

Kirishni boshqaruv yozuvi (ACE - Access Control Entry) uch qismdan iborat: birinchisi, bu yozuv tegishli bo'lgan foydalanuvchilar

yoki guruhlarni bildiradi, ikkinchisi erkin foydalanish huquqi va uchinchisi, bu huquqlar beriladimi yoki olinadimi (cheklashlar) shuni bildiradi.

Windowsda fayl va papkalarga foydalanuvchilarni kirish huquqlarini belgilash NTFS fayl tizimidan foydalanganda mumkin bo‘ladi. Ushbu fayl tizimi, fayl tizimi elementlari uchun xavfsizlik identifikatorlarini qo‘llab-quvvatlaydi va quyidagi kirish huquqlarini belgilash imkonini beradi: o‘qish, yozish, bajarish, o‘chirish, kirish huquqlarini o‘zgartirish, admin huquqlarini olish, attributlarni yozish va o‘qish. Fayllarni shifrlash tizimi (EFS-File System Encrypting) - bu Windows yadrosida joylashgan NTFS ga integratsiyalangan xizmatdir. Uning asosiy vazifasi - diskda saqlangan ma’lumotlarni ruxsatsiz kirishdan shifrlash orqali himoya qilishdir. Fayllarni shifrlash tizimi Windows CryptoAPI arxitekturasidan foydalanadi. U ochiq kalit yordamida shifrlash texnologiyasiga asoslangan. Har bir faylni shifrlash uchun, faylni shifrlash kaliti tasodifiy tarzda yaratiladi.

### ***Windows OT da CryptoAPI***

CryptoAPI – bu Windows operatsion tizimida amaliy dasturlash interfeysi (API - Application Programming Interface) hisoblanib, uning asosiy vazifasi operatsion tizim va uning nazorati ostida ishlaydigan ilovalar uchun shifrlash xizmatlarini taqdim etish hisoblanadi. U ilovalarni, ma’lumotlarni shifrlash va turli xil yo‘llar bilan raqamli imzolash imkonini beradigan, xususiy kalitlarni himoyalashni ta’minlaydigan bir qator funksiyalarni o‘z ichiga oladi. Shu bilan birga, CryptoAPI vazifalari hech qanday kriptografik operatsiyalarni bajarmaydi, dastur va shifrlash xizmati provayderi (CSP - Cryptographic Service Provider) o‘rtasida vositachi sifatida xizmat qiladi. Shifrlash xizmati provayderning dasturiy qismi dll fayli hisoblanib, u shifrlash funksiyasini qo‘llab-quvvatlaydi.

### ***Audit***

Audit xavfsizlik tizimining juda muhim elementi bo‘lib, tizimda foydalanuvchi harakatlariga aloqador hodisalarni maxsus jurnallarda qayd etish imkonini beradi. Windows siz kuzatishni hohlagan hodisalar turlarini tanlab olish va ularni tizim jurnallarida qayd etish imkonini beruvchi audit tizimini qo‘llab-quvvatlaydi. Windows hodisalarni uchta jurnalda qayd qiladi:

❖ Tizim jurnali (System log) - operatsion tizim va uchinchi tomon komponentlaridan kelgan xatolik xabarlarini, ogohlantirishlarni va boshqa ma'lumotlarni o'z ichiga oladi. Ushbu jurnalga kiritilgan hodisalar ro'yxati operatsion tizim va uchinchi tomon komponentlari tomonidan oldindan belgilanadi va foydalanuvchi tomonidan o'zgartirilishi mumkin emas. Jurnal Sysevent.evt faylida joylashgan.

❖ Xavfsizlik jurnali (Security log) - audit tomonidan ro'yhatga olingan xatti-harakatlarning muvaffaqiyatli va muvaffaqiyatsiz urinishlari haqidagi ma'lumotlarni o'z ichiga oladi. Ushbu jurnalda qayd etilgan hodisalar ma'mur tomonidan belgilangan audit algoritmi tomonidan belgilanadi. Jurnal Secevent.evt faylida joylashgan.

❖ Ilova jurnali (Application log) - turli ilovalar bilan belgilangan xatolik xabarlarini, ogohlantirishlar va boshqa ma'lumotlarni o'z ichiga oladi. Ushbu jurnalda qayd etilgan hodisalar ro'yxati dasturni ishlab chiquvchilari tomonidan belgilanadi. Jurnal Appevent.evt faylida joylashgan.

Barcha jurnallar %SystemRoot%\System32\Config papkasida joylashgan bo'ladi. Har bir jurnalni sozlash uchun, Event Log Settings dialog oynasidan (jurnal xususiyatlari) foydalanish mumkin.

Ushbu oyna orqali quyidagilarni boshqarish mumkin:

❖ Arxivlangan jurnallar hajmi (standart hajmi 512 KB, hajmi 64 dan 4 194 240 KB gacha o'zgarishi mumkin);

❖ Journallarda eskirgan yozuvlarni almashtirish;

❖ Zarur bo'lgan hollarda eski hodisalarni o'chirish - yangi hodisalarni yozish paytida jurnal to'lgan bo'lsa, operatsion tizim eng eski hodisalarni o'chiradi;

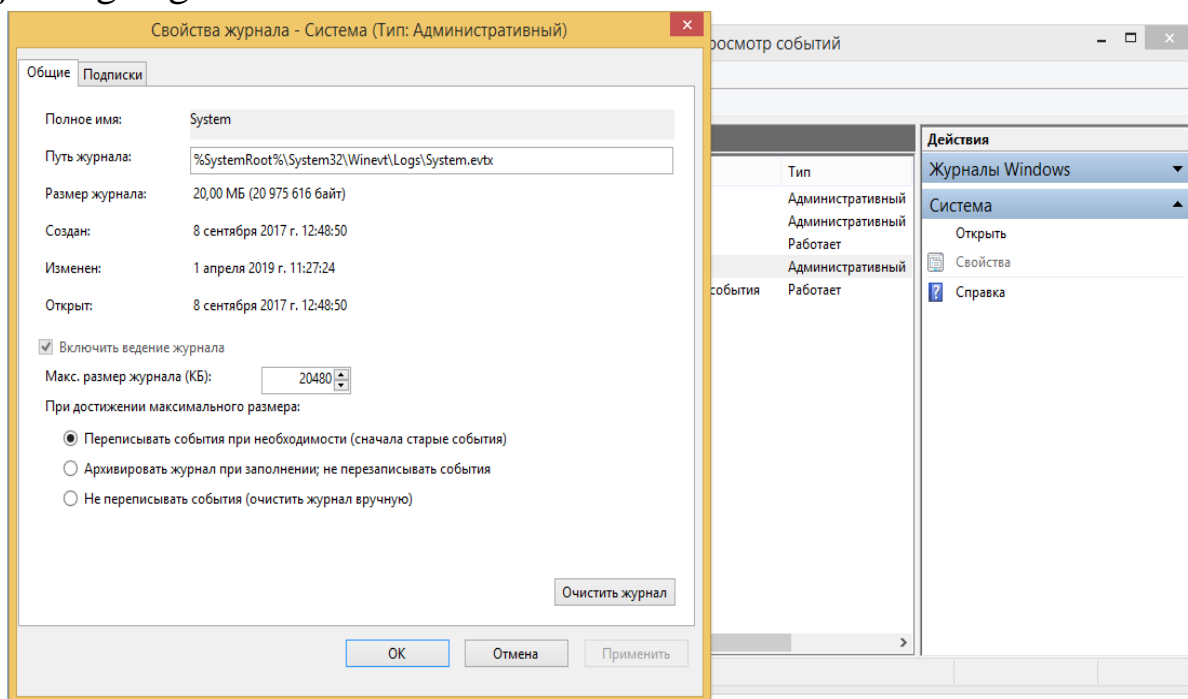
Hodisalarni yozishni taqiqlash - jurnal to'lganda, yangi hodisalar qayd qilinmaydi. Journallarni o'chirish qo'lda bajariladi.

Harbirjurnaldaamalgaoshirilganharakatlar, uniamalgaoshirganfoydalanuvchihaqida, shuningdek, hodisalarvaqtivasanasito'g'risidama'lumotmavjudbo'ladi.

Shuningdek, barcha muvaffaqiyatli va muvaffaqiyatsiz harakatlarni audit jurnali orqali ko'rishingiz mumkin. Jurnalga 5 xil turdagi yozuvlar yoziladi:

❖ *Xatolik.* Ma'lumotlarni yo'qotish yoki funksiyalarni yo'qotish kabi hodisalar. Masalan, boshlash paytida agar xizmat ishga tushurilmasa, xatolik to'g'risidagi xabar yoziladi.

❖ *Ogohlantirish.* Agar diskda bo'sh joy kam qolgan bo'lsa, jurnalga ogohlantirish kiritiladi.



9.19- rasm. Hodisalar jurnali xususiyatlarini belgilash

❖ *Ma'lumot.* Ilovalar yoki xizmatlarni muvaffaqiyatli bajarilganligini yozish hodisasi.

❖ *Muvaffaqiyatli Audit.* Tizim xavfsizligini ta'minlash bilan bog'liq bo'lgan muvaffaqiyatli bajarilgan harakatga mos keladigan hodisa. Misol uchun, foydalanuvchi muvaffaqiyatli tizimga kirsam, "Muvaffaqiyatli Audit" hodisasi jurnalga yoziladi.

❖ *Xatolik auditi.* Tizim xavfsizligini ta'minlash bilan bog'liq bo'lgan muvaffaqiyatsiz bajarilgan harakatga mos keladigan hodisa. Misol uchun, foydalanuvchi tarmoq diskiga kirishda muvaffaqiyatsizlikka uchrasa, "Xatolik audit" hodisasi jurnalga yoziladi.

Ma'mur audit hodisalarini boshqarish imkoniyatiga ega. Ular quyidagilar:

- ❖ Tizimga kirishni nazorat qilish;
- ❖ Hisobni boshqarish auditi;
- ❖ Katalog xizmatiga kirishni nazorat qilish (masalan, Active Directory);
- ❖ Tizimga kirishni tekshirish;

- ❖ Obyektga kirishni nazorat qilish;
- ❖ Audit siyosatining o'zgarishi;
- ❖ Audit imtiyozidan foydalanish;
- ❖ Jarayonni kuzatish auditi - dasturni ishga tushirish, jarayonni tugatish, identifikatorlarni takrorlash va obyektga bevosita kirish kabi hodisalar;
- ❖ Tizim holati auditi - kompyuter hodisalarini qayta tiklash yoki o'chirish, yana tizim xavfsizligiga yoki xavfsizlik jurnaliga ta'sir qiluvchi hodisalarni belgilaydi.

### **9.3. Android mobil operatsion tizimi, qurilish tamoyili va ahamiyati**

#### ***Mobil qurilmalarning operatsion tizimlari***

So'nggi bir necha yillardan beri, mobil qurilmalar sanoati shunchalik ulkanlashib kettiki, internetdan umumiy foydalanish va faollik bo'yicha hattoki kompyuterlar bozorini ham egallab bormoqda. Bu albatta mobil vositalardagi ustunlik, hamda ma'lumotlar tarmog'i bo'yicha muntazam aloqa tezligi sababidandir.

Huddi kompyuterlar o'tmishidagi singari, mobil operatsion tizimlari (OT) ishlab chiqaruvchilari orasida ham yakka liderlik uchun kurash boshlanib ketdi. Eng talabgorlar ro'yhatiga iOS, Android va Windows Mobile lar turli sanoat ishtirokchilari tomonidan kiritilgan. Ushbu holat nafaqat kim eng kuchli ekanligi to'g'risidagi savol, balki, siz uchun qaysi biri to'g'ri kelishini bilish uchun ham muhimdir. Ushbu bo'limda biz mobil qurilmalar uchun mavjud bo'lgan asosiy operatsion tizimlarni ko'rib chiqamiz. Bugungi kunda mobil qurilmalarda eng ko'p qo'llaniladigan asosiy operatsion tizimlar Android va iOS dir. Undan keyingi o'rinlarda birinchi bo'lib yaratilgan smartfonlardan Blackberry va Windows Mobile ni keltirishimiz mumkin. Biz mavzuimizni mobil qurilmalarda eng ko'p qo'llanilgan, dastlabki va hozirgi kundagi mobil qurilmalarning operatsion tizimlari tarixidan boshlaymiz.

Mobil qurilmalarning operatsion tizimlari ham juda xilma-xildir. Lekin hozirgi kunda ularning ko'pchiligi mobil OTlar bozorida o'z o'rnini yo'qotgan. Mobil operatsion tizimlariga misol qilib Blackberry (o'zining qurilmalari uchun ishlab chiqilgan), Palm (2001yildagi

mobil operatsion tizimlari orasida yetakchilardan bo'lgan), Symbian (2004-2010yillargacha birinchi o'rinlarda yurgan), iOS (2007yil) va Android (2008yil) operatsion tizimlari ishlab chiqarilgandan so'ng, Symbian operatsion tizimi o'z o'rnini sekin astalik bilan ushbu operatsion tizimlarga bo'shatib berdi. 2011yilning uchunchi choragida katta yetakchilik bilan Android operatsion tizimi boshqa mobil qurilmalarning operatsion tizimlarini ortda qoldirdi (50% ni tashkil qildi). 2013yilga kelib bu ko'rsatgich 80% ni tashkil qildi, va bu yetakchilik hozirgi kungacha saqlanib kelmoqda. Biz yuqorida mobil operatsion tizimlar bozorida o'z o'rnini egallagan mobil qurilmalar haqida gaplashdik, bulardan tashqari yana ko'plab mobil operatsion tizimlarni keltirish mumkin.

Smartfon yoki planshet universal kompyuterlardir. Ular fayl tizimi, xotira, operatsion tizim va dasturlarga ega. Operatsion tizim zararli dasturlardan himoyasiz. Ba'zi platformalar boshqalarga qaraganda zaifroq, ammo har bir platforma zararli dasturlarning hujumlariga duch kelishi mumkin. Mobil qurilma va noutbuk yoki ish stoli tizimi o'rtasidagi farq ma'lumotlar va interfeyslarni boshqarishdan iborat. Bu qurilmadagi ma'lumotlarga kirishni qiyinlashtiradi. Yaxshiyamki, ushbu ma'lumotlarni olish usullari mavjud. Biz turli xil mobil operatsion tizimlar haqida gaplashayotgan bo'lsakda, bu yerda asosiy maqsad ular haqida batafsil gaplashish emas. Biz ushbu mobil operatsion tizimlar nima, qanday tuzilganligi va ulardan ma'lumotlarni olishning ba'zi usullari haqida gaplashamiz. Eng muhimi shundaki, ushbu tizimlar mobil bo'lmagan tizim bilan qanday o'zaro bog'liq. Barcha smartfonlarda operativ xotira va doimiy xotira (saqlash) qurilmalari mavjud. Xotira energiyaga bog'liqdir, ya'ni tizim o'chirilishi bilan uning tarkibi o'chib ketadi. Xotira qisqa ma'lumotlarning joylashuvini nazarda tutmada, saqlash qurilmasi bu sizning kompyuteringizning komponentidir, bu sizga uzoq muddat ma'lumotlarni saqlash va kirishga imkon beradi. Ushbu xotira qurilmasi flash xotira deb ataladi.

### ***Shifrlash va masofadan boshqarish***

Odatda, mobil qurilmalar shifrlashni amalga oshirish uchun Advanced Encryption Standard (AES) standartidan foydalanadilar. Ma'lumotni shifrlashda sizga kalit kerak bo'ladi. AES simmetrik shifrlashdir, ya'ni bir xil kalit ham shifrlashda, ham deshifrlash uchun

ishlatiladi. Ikkala yo‘nalishda ham bir xil kalitdan foydalanayotganingiz uchun, uni xavfsiz saqlash juda muhimdir. Kalitlarni saqlash va boshqarish har qanday shifrlash sxemasining muhim qismidir. Kalitni himoya qilish uchun har bir qurilma turli xil vositalardan foydalanadi, ammo foydalanuvchidan parolni talab qilmasdan shifrlash odatiy hol emas. Agar qurilmadan foydalanish uchun autentifikatsiyadan o‘tishingiz talab qilinmasa, qurilmadagi ma’lumotlarni shifrlashning ahamiyati bo‘lmaydi, chunki har kim xohlagan vaqtda telefonni ko‘tarib undan foydalanishi mumkin.

### ***Android operatsion tizimi***

Ushbu tizimning eng o‘ziga xos xususiyatlaridan biri - unda o‘zlashtirish imkoniyatlari miqdoridir. Bu qulaylik telefonning butun telefonning barcha qismi, o‘rnatilgan fon, vidjetlar va ikonkalarni o‘z ichiga oladi. Android Google Play yoki nashr qiluvchining o‘zidan bevosita yuklab olinadigan, bir milliondan oshiq turli ilovalarga ega. Agar siz aynan o‘zingizga moslashtirilgan mobil xizmatini istasangiz, Android eng yuqori tanlovdur.

**Android** — Linux yadrosida yaratilgan portativ (tizimli) operatsion tizim bo‘lib, kommunikatorlar, planshetli kompyuterlar, elektron kitoblar, raqamli pleerlar, qo‘l soatlari, netbuklar va smartbuklarda ishlatiladi. Google tomonidan sotib olingan Android Inc. kompaniyasi tomonidan yaratilgan. Keyinchalik Google tizim rivojlanishi bilan shug‘ullanadigan Open Handset Alliance (OHA) ittifoqini tuzdi. Android yordamida Java dasturlarni tuzish mumkin. Android Native Development KitSida va boshqa tillarda yoziladigan dasturlarni yaratadi. Android ochiq kodli operatsion tizim hisoblanadi.





9.20- rasm. Android operatsion tizimi logotipi va undagi xizmatlar

U dastlab 2003yilda Android kompaniyasi tomonidan ishlab chiqilgan. Ikki yil o‘tib 2005yilning iyulida Google korporatsiyasi Android Inc. kompaniyasini sotib oldi. 2007yilning 5noyabrida Google Open Handset Alliance tuzilganligini e‘lon qilib, birinchi ochiq mobil Android tizimini namoyish etdi. Android – Google ga tegishli mobil operatsion platformadir.

2008yilning 23sentabrida operatsion tizimning birinchi versiyasini namoyish etdi. Birinchi versiya chiqqanidan so‘ng tizim bir necha marta yangilandi. Bu yangilashlar tizimda aniqlangan xatolarni tuzatib, yangi funksiyalar qo‘shdi.

2009yilda platformaning to‘rtta yangilanishlari chiqdi. Shunday qilib, fevralda turli xatolarni tuzatadigan 1.1 versiyasi chiqdi. Aprel va sentabrda mos ravishda 1.5 „SupSace“ va 1.6 „Donut“ updatelari chiqdi. „SupSake“ versiyasi bir qancha o‘zgartirishlar kiritdi: virtual klaviatura, video yozish va tomosha qilish, brauzer va boshqalar. „Donut“da birinchi marta CDMA tizimlarini qo‘llash paydo bo‘ldi. 2009yilning oktabrida bir nechta Google akkauntlarini, HTML5 tilidagi brauzerni qo‘llaydigan va bir nechta yangiliklar kiritilgan Android 2.0 „Eclair“ operatsion tizimi chiqdi. „Eclair“ (2.1) versiyasida „tirik rasmlar“ paydo bo‘ldi va ekran blokirovkasining ko‘rinishi o‘zgardi.

2010yilning o‘rtasida Google Android 2.2 „Froyo“ versiyasi chiqdi. 2010yilning ohirida esa Android 2.3 „Gingerbread“ namoyish etildi. Bu updatelar an’anaviy blokirovkani raqamli yoki xarf-raqamliga o‘zgartirdi, „Kesish, nusxa ko‘chirish“, qayta qo‘yish funksiyalarini soddalashtirdi va hokazo. 2011yilning 22fevralida Android 3.0 „Honeycomb“ namoyish etildi. Uning asosiy yangiligi planshetga moslashtirilgani edi. 2011yilning 19oktabrida chiqqan Android 4.0 „Ice Cream Sandwich“ — birinchi universal tizim (ham planshetlarga, ham smartfonlarga moslashtirilgan) chiqdi. 2012yilda tizimning yangi versiyasi — „Jelly Bean“ chiqdi. U iyunda 4.1 raqami ostida namoyish etilib, oktabr ohirida yangi update chiqqach 4.2 ga o‘zgardi.

Android ostida ishlaydigan birinchi moslama 2008 yilning 23 sentabrida HTC kompaniyasining HTC Dream smartfoni (T-Mobile aloqa operatori tomonidan T-Mobile G1 nomi ostida namoyish

etilgan) bo'ldi. Tezda boshqa smartfon ishlab chiqaruvchilari tomonidan Android OT ostida moslamalar chiqarish so'rovlari paydo bo'ldi. Androidning planshetlarga moslashtirilgan 3- versiyasi (Honeysomb) chiqqach planshet ishlab chiqaruvchilariga ham Android yoqib qoldi.



9.21- rasm. HTC Dream (T-Mobile G1) — Android OT ostidagi birinchi smartfon



## 9.22- rasm. Android 10 OTning ishchi stoli

Google kompaniyasi ham boshqa ishlab chiqaruvchilar bilan hamkorlikda o‘zining “Google Nexus” moslamalarini Android OT ostida chiqara boshladi. Aynan shu moslamalar birinchi bo‘lib yangi versiyalarda chiqadi.

Smartfonlar va planshetlardan tashqari Android AT boshqa moslamalarga ham o‘rnatiladi. Masalan, 2009yilda sotuvga Androidda ishlovchi fotoramka sotuvga chiqdi. 2011yilda Blue Sky kompaniyasi Android OT ostidagi “aqli” iWatch soatlarini chiqardi. 2012yil avgustda Nikon dunyodagi birinchi Androidda ishlovchi fotokamerani namoyish etdi. “Google Nexus” seriyasida smartfon va planshetlardan tashqari Android ostidagi Nexus mediapleeri mavjud. Shu bilan birga Android rusumidagi televizorlar ham mavjuddir.

2012yil oktabrda Google boshqaruvchi direktori Larri Peyj Android ostidagi moslamalar soni 500 mln.dan ortgani va har kuni 1,3 mln.ga ko‘payayotgani haqida xabar berdi.

### ***Google Play***

2008yil 22oktabrda Google Android OT dasturlarining onlayn do‘koni — Android Market ochilishi haqida e‘lon qildi. Kelishivga ko‘ra yaratuvchilar foydaning 70 %ini, operatorlar esa qolgan 30 %ni olishadi. 2009yil fevralda AQSHvaBuyuk Britaniya yaratuvchilarida Android Marketdagi o‘z dasturlari uchun pul olish imkoniyati paydo bo‘ldi. Sony Ericson kompaniyasi dasturlarning onlayn do‘koni Android Marketda shaxsiy kanalini ishga tushirdi. U kanalda kompaniya tomonidan tavsiya etilgan dasturlar va o‘yinlar berilgan. 2011yil dekabrda Android Market tuzilganidan beri 10 mlrd. dastur ko‘chirib olindi. 2012yil martda Google kompaniyasi „Kitoblar“, „Android Market“, „Musiqqa“ va boshqa servislarni yagona Google Play servisiga birlashtirdi. Google Play internet do‘koni 190ta davlatda ishlaydi, 700 mingdan ortiq dasturga ega, ochilganidan beri 25 mlrd.ta dastur ko‘chirib olingan. Android tizimi uchun maxsus Droid va Roboto shriftlari yaratilgan. 1.5-versiyadan boshlab Android versiyalarining nomlari biror-bir desert nomidan olinadi. Nomlarning birinchi xarflari versiyalarning yangilanishi bo‘yicha alifbo xarflari tartibida ketadi: 1.5 Cupcake („keks“), 1.6 Donut (pishiriq turi), 2.0/2.1 Eclair („ekler“), 2.2 Froyo („muzlagan yogurt“ qisqartmasi),

2.3 Gingerbread („zanjabilli shirinkulcha“), 3.0 Honeycomb („asalari ini“), 4.0 Ice Cream Sandwich („muzqaymoq briketi“), 4.1/4.2 Jelly Bean (shirinlik turi), 4.5 KitKat (*shokolad turi*), 5.0 Lollipop (chupa chups), 6.0 Marshmallow, 7.0 Nougat, 2017yil 21avgustda 8.0 Oreo, 2018yil 6avgustda 9 Pie, 2019yil 3sentyabrda Android 10 versiyasi yaratilgan. 2011-yil fevralda birinchi bo‘lib Android OT moslamalari Buyuk Britaniyada iOS operatsion tizimi ostidagi iPhone4 ni ortda qoldirdi. Hozirgi kunda Android operatsion tizimida ishlaydigan qurilmalar soni iOSga nisbatan ancha ko‘p. 2014yilning 3-choragida sotilgan smartfonlarning 86 foizida Android operatsion tizimi o‘rnatilgan edi.

Android ning maqsadi mobil qurilmalarni yanada aqlli, shaxsiylashtirilgan va joylashuvga yo‘naltirilganroq qilish edi. Android qurilmada ishlatiladigan barcha jihozlar bilan o‘zaro ishlash uchun mo‘ljallangan maxsus Linux yadrosiga asoslangan. Mobil qurilmalarda ular bilan bog‘liq cheklangan qurilmalar mavjud. Linux ni ish stolida yoki noutbukda ishlatsangiz, foydalanuvchi istalgan miqdordagi kengaytirish kartalari yoki USB-qurilmalarni ulashi mumkin. Buning uchun drayverlar talab qilinadi va Linuxda, odatda, ushbu drayverlar to‘g‘ridan-to‘g‘ri yadroga o‘rnatiladi. Mobil qurilmada qurilmaning o‘zida o‘rnatilgan juda oz miqdordagi qurilmalar mavjud bo‘lganligi sababli, ishlab chiqaruvchi yadroni juda kichik va ma‘lum bir qurilma uchun moslashtirishi mumkin. Mobil protsessorlar ish stoli protsessorlari bilan bir xil emas. Ilovalar ham farq qiladi. Android qurilmalarida ilovalar himoyalangan joylarda ishlaydi. Ko‘pincha dasturlar Java-da yoziladi. Eslatib o‘tamiz, Java to‘g‘ridan-to‘g‘ri protsessorida ishlaydigan mashina kodiga kirmaydi. U bitta dastur uchun maxsus mo‘ljallangan virtual mashinada ishlaydi. Android 4.4 (KitKat) ga qadar Google o‘zining virtual mashinalari uchun Dalvik dasturidan foydalangan. Dalvik dasturi kerak bo‘lganda bayt kodni mashina kodiga o‘zgartiradi. Androidning yangi versiyalari Lollipop va Marshmallow kabi ishlanmalari, Google oldindan kompilyatsiyalangan fayllardan foydalanadigan virtual mashinaga o‘tdi. Bu shuni anglatadiki, dastur o‘rnatilganda, operatsion tizim mashina kodiga o‘tkazishni amalga oshiradi, shuning uchun dastur ishga tushgandan so‘ng darhol bajarishga tayyor bo‘ladi. Hech qanday kechikishlar bo‘lmaydi, chunki dastur ishlayotgan vaqtda bayt kod mashina kodiga o‘zgartiriladi. Xususan, Android yordamida siz asosiy

operatsion tizimga kirishingiz mumkin. Bu qurilmaga va undan fayllarni nusxalash uchun fayl tizimiga to'g'ridan-to'g'ri kirishga imkon beradigan Android Debug Bridge (ADB) yordamida amalga oshiriladi. ADB boshqa vazifalarni ham bajaradi, shuning uchun bu siz ko'proq ma'lumot olishni xohlashingiz mumkin bo'lgan yordamchi dastur hisoblanadi. Android bilan bog'liq muammo shundaki, u ko'pchilik uchun ochiqdir. Google Applega o'xshash cheklovlarga ega emas, shuning uchun ba'zi ishlab chiqaruvchilar o'zlarining shaxsiy do'konlarini yaratishlari mumkin. Foydalanuvchilar shuningdek, ilovalar uchun uchinchi tomon manbalariga ruxsat berishlari mumkin. Bu qurilmaning ishonchliligi va xavfsizligi uchun muammo tug'dirishi mumkin, chunki ushbu boshqa manbalar tizimga zararli dasturlarni kiritish usuli bo'lishi mumkin.

### ***Android operatsion tizimi tuzilishi***

Android dasturiy steki, qoida tariqasida, Linux yadrosi va xizmatlarni taqdim etadigan ilova muhiti, shuningdek dastur va ish vaqtini boshqarish orqali ta'minlangan C/C++ kutubxonalaridan iborat. 9.23- rasmda Android operatsion tizimi tuzilishi keltirilgan. U quyidagi asosiy qismlardan tashkil topadi.

**Linux yadrosi.** Android ochiq manbali Linux yadrosida yaratilgan. Ushbu yadroni tanlashning asosiy sabablaridan biri shundaki, u Android operatsion tizimini rivojlantirish uchun tasdiqlangan asosiy funksiyalarni taqdim etgan. Linux yadrosining xususiyatlari:

- ❖ Xavfsizlik – Linux yadrosi dastur va tizim o'rtasida xavfsizlikni ta'minlaydi;
- ❖ Xotirani boshqarish – u dasturlarimizni ishlab chiqish erkinligini ta'minlab, xotira boshqaruvini samarali boshqaradi;
- ❖ Jarayonni boshqarish – u jarayonni yaxshi boshqaradi, kerak bo'lganda jarayonlar o'rtasida resurslarni taqsimlaydi;
- ❖ Tarmoq – u tarmoq aloqasini samarali boshqaradi;
- ❖ Drayver modeli – Ilovaning ishlashini ta'minlaydi. Qurilma ishlab chiqaruvchilari o'zlarining drayverlarini Linux ga asoslanib qurishlari mumkin.

**Kutubxonalar.** Yadro ustida ishlaydigan Android platformasi turli funksiyalar bilan ishlab chiqilgan. U ko'plab ochiq manbali

vositalarga ega bo'lgan turli xil C/C ++ yadro kutubxonalaridan iborat. Ulardan ba'zilar:

- ❖ Android runtime – Android bajarish vositasi asosan Java va ART (Android Run Time) kutubxonalaridan iborat. Android ning eski versiyalari (4.x va oldingi versiyalari) Dalvik ga asoslangan;

- ❖ Open GL (graphics library) – ushbu kross-tilli, kross-platformali dasturiy interfeys 2D va 3D kompyuter grafikalarini yaratish uchun ishlatiladi;

- ❖ WebKit – ushbu ochiq kodli veb-brauzer mexanizmi veb-tarkibni namoyish qilish va sahifani yuklashni soddalashtirish uchun barcha xususiyatlarni taqdim etadi;

- ❖ Media frameworks – ushbu kutubxonalar audio va video ni ishga tushurish va yozish uchun imkon beradi;

- ❖ Secure Socket Layer (SSL) – ushbu kutubxonalar Internet xavfsizligini ta'minlash uchun ishlatiladi.

**Android Runtime.** Bu arxitekturaning uchinchi qismidir. Dalvik virtual mashinasi deb nomlangan asosiy tarkibiy qismlardan birini taqdim etadi. Bu Java uchun maxsus ishlab chiqarilgan Java virtual mashinasi sifatida ishlaydi. Android bitta moslamada bir nechta nusxalarning samarali ishlashini ta'minlash uchun mo'ljallangan o'zining maxsus virtual mashinasidan foydalanadi. Dalvik virtual mashinasi past darajadagi funktsiyalarni qo'llab-quvvatlash uchun xavfsizlik, ko'p tarmoqli va xotirani boshqarish uchun Linux yadrosidan foydalanadi.

**Ilovalar platformasi.** Android jamoasi orqa fonda qurilgan ma'lum bir tasdiqlangan kutubxonalariga asos solgan va ularning barchasi Android interfeyslari orqali taqdim etiladi. Ushbu interfeyslar barcha turli kutubxonalarni birlashtiradi va bu orqali ularni dasturchi uchun foydaliligi va qulayligini ta'minlaydi. Ushbu interfeyslardan ba'zilar quyidagilarni o'z ichiga oladi:

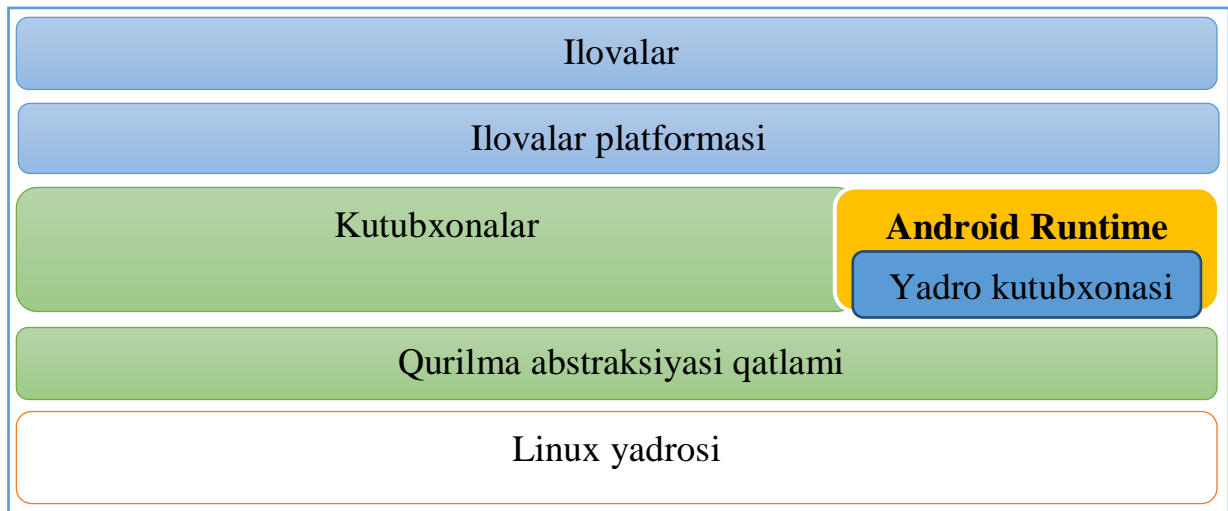
- ❖ Aktivlik menejeri – hayot sikli aktivligi va stek aktivligini boshqaradi;

- ❖ Telefoniya menejeri – u telefoniya xizmatlariga abonent raqamlari kabi tegishli ma'lumotlar ko'rinishida kirishni ta'minlaydi;

- ❖ Tizimni ko'rish –bu ko'rinishlar va tartiblarni qayta ishlash orqali foydalanuvchi interfeysini yaratadi;

- ❖ Joylashuv menejeri – u qurilmalarning geografik joylashuvini aniqlaydi.

**Ilovalar.** Android ilovalarini yuqori qatlamdan topish mumkin. Ilova qatlamida biz faqatgina ushbu qatlamga o‘rnatilishi uchun o‘z dasturimizni yozamiz. Ilovalarga misol qilib o‘yinlar, xabarlar, kontaktlar va boshqalarni keltirishimiz mumkin.



### 9.23- rasm. Android operatsion tizimi tuzilishi

#### *Fayl tizimi tuzilishi*

Android uchun ishlatiladigan yadro Linux yadrosida sozlangan bo‘lsada, muhitning qolgan qismi Linuxdan farq qiladi. Buni bir joyda, fayl tizimining tuzilishida ko‘rishingiz mumkin. Sizda /etc, /usr, /var, /lib va boshqalar bo‘ladi. Agar siz Android tizimidagi ildiz fayl tizimiga qaragan bo‘lsangiz, siz bundan juda boshqacha narsani ko‘rasiz. Agar siz to‘liq fayl tizimiga ega bo‘lsangiz, Android tizimini tadqiq qilishni qiyinlashtirishi mumkin, chunki hamma narsa qaerda saqlanganligini bilmasligingiz mumkin. Odatda Android-da quyidagi kataloglar mavjud. Esda tuting, ularning hammasi ham haqiqiy kataloglar emas. Ular ulanish nuqtalari bo‘lishi mumkin, ya’ni katalog boshqa joydan bo‘limni biriktirish uchun joy sifatida mavjudligini anglatadi. Bu tashqi saqlash moslamasi yoki soxta fayl tizimi bo‘lishi mumkin.

**acct:** hisob ma’lumotlari saqlanadigan acct katalogi. Ushbu hisob ma’lumotlari boshida protsessorni ishlatishni hisobga oladi.

**cache:** tez-tez ishlatiladigan ma’lumotlar, shuningdek, tizimning ishdan chiqishi natijasida yo‘qotilib qayta tiklangan ma’lumotlar kesh katalogida saqlanadi.

**d:** ushbu katalog yadroni tuzatish uchun ishlatiladi.

**data:** foydalanuvchi ma'lumotlari, shuningdek kontaktlar va matnli xabarlar saqlanadigan ma'lumotlar katalogi.

**default.prop:** bu standart xususiyatlarni o'z ichiga olgan fayl.

**dev:** bu Linux tizimidagi /dev katalogi bilan bir xil. Bu telefon yoki planshetdagi qo'shimcha qurilmalarga tegishli barcha maxsus fayllarni o'z ichiga oladi.

**etc:** Linux tizimi kabi etc katalogi sozlash fayllarini o'z ichiga oladi.

**init:** bu telefon yoki planshet ishga tushganda barcha boshqa xizmatlar va dasturlarni ishga tushirish uchun ishlatiladigan asosiy jarayon. Ushbu dastur ushbu katalogda saqlanadi.

**mnt:** Linux tizimi kabi, bu tashqi qurilmalar uchun o'rnatish nuqtasi.

**proc:** proc katalogi soxta fayllar tizimi, u ishga tushirilgan jarayonlar bilan bog'liq kataloglar va fayllarni o'z ichiga oladi.

**root:** bu super foydalanuvchi uchun uy katalogidir.

**sdcard:** sdcard katalogi, ma'lumotlarni saqlash mumkin bo'lgan SD-kartani ulash joyidir.

**sys:** bu sysfs soxta fayl tizimi uchun ulanish nuqtasi, yadro ichki tuzilishi bilan bog'liq ma'lumotlarni o'z ichiga oladi.

**system:** tizim katalogi, unda Linux va Unix tizimlarida bo'lgani kabi barcha umumiy kataloglar saqlanadi.

**uventd.rc:** bu/dev katalogiga yozish uchun konfiguratsiyasi fayli.

**vendor:** bu /system/vendor katalogiga havola

Android qurilmasida bir nechta bo'limlar mavjud. Tizim bo'limi Android platformasini, kutubxonalarini, ilovalar, standart dasturlarni va tizim ikkilik fayllarini o'z ichiga oladi. Yuklanuvchi qism (boot loader) past darajadagi ishga tushirish dasturlarini o'z ichiga oladi. Yadro va unga tegishli fayllar yuklash bo'limida joylashgan. Agar qurilmangizda texnik xizmat ko'rsatishingiz kerak bo'lsa, siz kichikroq operatsion tizim nusxasini o'z ichiga olgan tiklash bo'limidan foydalanasiz. Ushbu rasm ishlamayotgan Android qurilmasiga zarur o'zgartirishlarni kiritish uchun ishlatilishi mumkin. Yangilanayotgan qurilma tiklash bo'limi va yuklash rasmidan foydalanishi mumkin. userdata bo'limi - bu barcha foydalanuvchi ma'lumotlari saqlanadigan joy. Bunga foydalanuvchi saqlashi mumkin bo'lgan har qanday rasmlar, musiqa, videolar va ilova



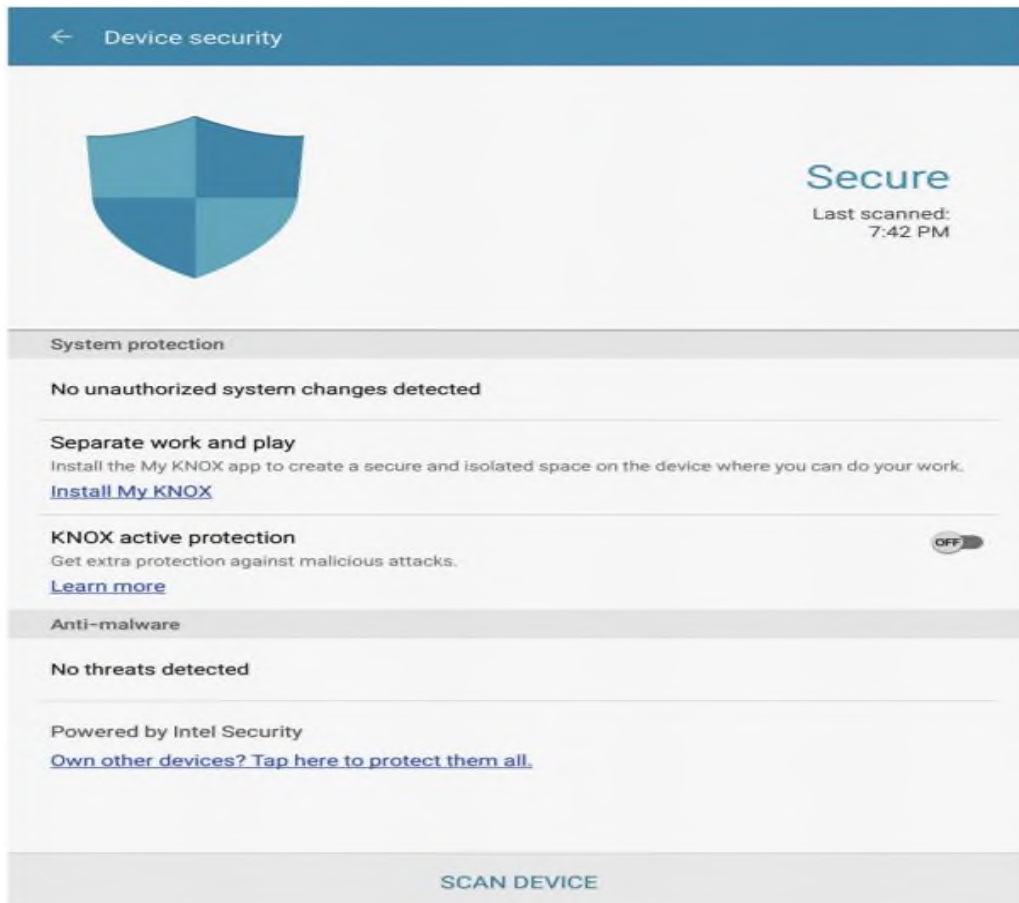
ma'lumotlari kiradi. Ushbu qism juda muhimdir. Ohirgi bo'lim bu kesh. Doimiy ravishda ishlatiladigan ma'lumotlar ushbu bo'limda saqlanadi. U yerda tiklash ma'lumotlari, simsiz tarmoq orqali yuklab olingan har qanday xizmat paketlari saqlanadi.

### ***Zararli dasturlar***

Google foydalanuvchilarga ilovalarni tarqatishda Play Store dan foydalaniladi. Taqdim etilgan ilovalarni Play Store-ga qo'shish va sotishdan oldin tekshirish jarayonidan o'tishi kerak. Ammo, bu tekshirish jarayoni mukammal emas. Ba'zan, zararli dasturlar Play Store orqali qurilmalarga kiritilishi mumkin. Google tekshirish jarayonini takomillashtirdi, ammo zararli dastur har doim buni amalga oshirishi mumkin. Bundan tashqari, albatta, foydalanuvchilar uchinchi tomon manbalaridan o'rnatilgan ilovalarni tekshirishni o'chirib qo'yishlari mumkin. Aslida, Android mobil zararli dasturlarning asosiy maqsadiga aylangan bo'lsa-da, haqiqat shundaki, ushbu zararli dasturlarning katta qismi asosan Osiyo va Yaqin Sharqda joylashgan uchinchi tomon ilovalar do'konlaridan keladi (Forbes, 2014). Google Play Store-ga yuborilgan ilovalarning zararli harakatlarni sinchkovlik bilan tekshirilishini ta'minlash uchun ko'p vaqt va kuch sarflaydi. An'anaviy kompyuter qurilmalari o'rniga mobil qurilmalardan tobora ko'proq foydalanilayotganligi sababli, mobil qurilmalarga zararli dasturlarning hujumlari ko'proq uchramoqda. Shu sababli, ba'zi Android qurilmalarini ishlab chiqaruvchilar o'zlarining qurilmalariga zararli dasturlardan himoya qilish dasturlarini o'rnatmoqdalar. Shunday qilib, foydalanuvchilar o'zlari uchun antivirus dasturlarini izlashlari va o'rnatishlari shart emas. Agar foydalanuvchilar uchinchi tomon manbalaridan dasturlarni o'rnatish usullarini topa olsalar ham, antivirus dasturi uni aniqlaydi. Android qurilmalaridagi sozlamalar ilovasida "Ekranni bloklash va xavfsizlik" bo'limidan "qurilma xavfsizligi"ni topasiz. Bundan foydalanish orqali qurilmangizni tahdidlar uchun tekshirishingiz mumkin. Siz turli xil parametrlarni 9.24- rasmda Samsung plan-shetida "Qurilma xavfsizligi" bo'limida ko'rishingiz mumkin.

Ushbu planshet tekshirishdan o'tkazildi. Dastur odatda zararli dasturlarni tekshirishni rejalashtirmaydi. Foydalanuvchilarni ushbu parametrga o'tishini va qo'lda tekshirishni boshlashni talab qiladi. Ammo, siz Samsung himoyani ta'minlash uchun taklif qiladigan ba'zi

qo'shimcha funksiyalarni ko'rishingiz mumkin. Albatta, bu barcha Android qurilmalarida keng tarqalmagan, chunki Knox - bu Samsungning xavfsizlik funksiyasidir.



9.24- rasm. Knox xavfsizlik funksiyasi

### ***Android Debug Bridge (ADB) dan foydalanish***

Ishlab chiquvchilar ADBdan Android qurilmalari bilan o'zaro aloqa qilish uchun foydalanadilar. U Google tomonidan taqdim etilgan, Android dasturlarini ishlab chiqish uchun foydalaniladigan, dasturiy ta'minotni ishlab chiqarish to'plamining bir qismi sifatida o'rnatilishi mumkin yoki agar siz uni mobil qurilmalar bilan o'zaro bog'lashda ishlatmoqchi bo'lsangiz, uni oflayn rejimda o'rnatish mumkin. ADB juda ko'p xususiyatlarga ega. Birinchisi, tortish funksiyasi. Agar siz tortish (pull) parametri bilan ADBni chaqirsangiz va keyin nimani olishni aniqlasangiz, ma'lumotni to'g'ridan-to'g'ri qurilmadan chiqarib olishingiz mumkin. Buni qurilma ishlayotgan

paytda qilish mumkin, ammo ADBni tiklash rejimida yuklangan qurilmada ham ishlatish mumkin. Yaxshiyamki, tiklash rejimiga o'tish uchun tizim yoqilganda tugmalar bosish kombinatsiyasini bilish kerak. Odatda, bu qurilma o'chirilgan paytda quvvat tugmachasini va ovozi balandligini pasaytirish tugmachasini ushlab turish kabi narsa. Qurilma tiklash rejimida ishga tushadi, shundan so'ng siz tizimga ulanish uchun ADBdan foydalanishingiz mumkin. ADB Linux, Windows va Mac OS X-larda ishlaydi, shuning uchun siz qaysi Android qurilmadan va qanday operatsion tizimdan ma'lumotlarni yo'q qilmoqchi bo'lsangiz ham xuddi shu vazifani bajarishingiz mumkin. ADBdan foydalanib, buyruq satri yordamida tizimni o'rganishingiz mumkin. ADB qobig'i buyrug'i tizim qobig'iga kirish huquqini beradi. Android Linux yadrosiga asoslangan va qurilmada ko'pgina Unix/Linux larning umumiy yordamchi dasturlari mavjud. ADB qobig'idan foydalanib, ilgari taklif qilinganidek, SSH serverini o'rnatish jarayonidan o'tmasdan to'g'ridan-to'g'ri tizimga kirishingiz mumkin.

### **Nazorat savollari**

1. Linux operatsion tizimi yaratilish tarixini tushuntiring.
2. Linux operatsion tizimi imkoniyatlarini keltiring.
3. Linux operatsion tizimi tuzilishini tushuntiring.
4. Linux operatsion tizimida jarayonlarni boshqarishni tushuntiring.
5. Linuxda jarayonlar va oqimlarni tushuntiring.
6. Linuxda fizik xotirani boshqarish.
7. Linuxda virtual xotira.
8. Linux fayl tizimlarini tushuntiring.
9. Linuxda kiritish/chiqarishni tushuntiring.
10. Linuxda xavfsizlik vositalarini tushuntiring.
11. Windows operatsion tizimi imkoniyatlari?
12. Windows 7 operatsion tizimi tuzilishini tushuntiring.
13. Windows operatsion tizimida jarayon va oqimlar.
14. Windowsda jarayon va oqim obyektlarini tushuntiring.
15. Windowsda ko'p oqimlilik.
16. Windowsda oqim holatlari turlarini keltiring.
17. Windowsda xotirani boshqarishni tushuntiring.
18. Windowsda almashtirishni tushuntiring.

19. Windowsda kiritish/chiqarish qurilmalarining asosiy imkoniyatlari.
20. Windowsda asinxron va sinxron kiritish/chiqarishni tushuntiring.
21. RAID dasturi qanday maqsadda ishlatiladi?
22. Windowsda fayllarni boshqarishni tushuntiring.
23. Windows operatsion tizimida qo'llaniladigan fayl tizimini tushuntiring.
24. Windowsda xavfsizlikni tushuntiring.
25. Windowsda audit tizimi nima va uning asosiy vazifalari.
26. Windowsda CryptoAPI nima?
27. Mobil qurilmalarning operatsion tizimlari turlari.
28. Android operatsion tizimi afzalliklari.
29. Android OTning fayl tizimini tushuntiring.
30. Android Debug Bridge dan foydalanish nima uchun kerak?

## FOYDALANILGAN ADABIYOTLAR

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne. Operating system concepts Ninth edition, 2013. – 944 p. ISBN: 9781118063330, ISBNBRV: 9781118129388.

2. Ann McIver McHoes Ida M. Flynn. Understanding operating systems. Sixth edition, 2011. – 591 p. Library of Congress Control Number: 2010920344.

3. Э.Таненбаум, Х.Бос. Современные операционные системы 4-е издание — СПб.: Питер, 2015. — 1120 с.: ил. — (Серия «Классика computer science»).

4. В.Г.Олифер, Н.А.Олифер. Сетевые операционные системы: Учебник для вузов. 2-е изд. — СПб.: Питер, 2009. — 669 с.: ил.

5. С.В. Назаров, А.И. Широков. Современные операционные системы Учебное пособие — М.: Национальный Открытый Университет «ИНТУИТ», 2012. — 367 с.: ил., табл. — (Основы информационных технологий), 2012.

6. Бабамухамедова М.З. “Операцион тизим ва офис иловалари” фанидан маърузалар матни, 2012. – 172 б.

7. В.О. Сафонов. Основы современных операционных систем. СПб.: СПбГУ, 2009. — 468с.

8. William Stallings. Operating systems internals and design principles. Seventh edition, 2012. – 820 p. ISBN-13: 978-0-13-230998-1 (alk. paper).

9. Behrouz A. Forouzan. TCP/IP Protocol Suite. Fourth Edition, 2010. – 1029 p. ISBN 978-0-07-337604-2 — ISBN 0-07-337604-3 (hard copy : alk. paper).

10. Sh.M.Mirziyoyev. 2017-2021 yillarda O‘zbekiston Respublikasini rivojlantirishning beshta ustuvor yo‘nalishlari bo‘yicha HARAKATLAR STRATEGIYASI.

11. O‘zbekiston Respublikasi Prezidentining “Oliy ta’lim tizimini yanada rivojlantirish chora-tadbirlari to‘g‘risida”gi qarori (2017 yil 20 aprel, PQ-2909-son).



## MUNDARIJA

<b>KIRISH</b> .....	3
<b>I BOB OPERATSION TIZIMLAR VA ULARNING AHAMIYATI</b>	
1.1. Operatsion tizim tushunchasi va ularning ahamiyati..	7
1.2. Operatsion tizimlarning rivojlanish davrlari.....	9
1.3. Operatsion tizimlarning tuzilishi va asosiy komponentlari.....	15
1.4. Kompyuter tizimlari, ularning tuzilishi va ahamiyati. ....	18
<b>II BOB OPERATSION TIZIMDA JARAYONLARNI VA PROTSessorNI BOSHQARISH</b>	
2.1. Operatsion tizimda jarayonlar va ularning holatlari ...	36
2.2. Operatsion tizimdagi oqimlar va ularning turlari .....	43
2.3. Operatsion tizimda vazifalar va jarayonlarni rejalashtirish.....	51
2.4. Operatsion tizimda protsessorni boshqarish .....	63
2.5. Jarayonlarni rejalashtirish algoritmlari.....	71
2.6. Berkliklar, Berkliklarni aniqlash va bartaraf etish usullari .....	78
<b>III BOB XOTIRANI TASHKIL ETISH VA BOSHQARISH USULLARI</b>	
3.1. Kompyuter xotirasi turlari va ularning tashkil etilishi .....	86
3.2. Xotirani boshqarish va undan umumiy foydalanish mexanizmi .....	93
3.3. Xotirani taqsimlash, qayta taqsimlash va jamlash usullari .....	96
3.4. Xotirani tartiblash va kengaytirish, virtual xotira.....	108
<b>IV BOB KOMPYUTER TASHQI QURILMALARINI BOSHQARISH</b>	
4.1. Kompyuter tizimlarining kiritish/chiqarish qurilmalari .....	126
4.2. Operatsion tizimning qurilmalarni boshqarish vazifalari .....	134
4.3. Ketma-ket yozishga asoslangan axborot saqlash qurilmalari .....	141

	4.4. To‘g‘ridan to‘g‘ri yozishga asoslangan axborot saqlash qurilmalari .....	145
	4.5. Qurilmalarni boshqarish drayveri va dasturiy ta‘minoti .....	154
	4.6. Ma‘lumotlarni kiritish/chiqarish menejeri va dasturiy ta‘minoti.....	158
<b>V BOB</b>	<b>FAYL TIZIMI VA FAYLLARNI BOSHQARISH JARAYONI</b>	
	5.1. Operatsion tizimlar fayl tizimi, tuzilishi va tashkil etilishi .....	173
	5.2. Fayllarni fizik tashkil etish, ularni o‘qish va yozish usullari .....	178
	5.3. Fayllar, ularning turlari va atributlari .....	186
	5.4. Fayllarni boshqarish va ular ustida amallar bajarish ..	190
	5.5. Fayllarni himoyalash va ularga murojaatlarni boshqarish.....	194
<b>VI BOB</b>	<b>KOMPYUTER TARMOQLARINI QURISH VA BOSHQARISH</b>	
	6.1. Kompyuter tarmoqlari, ularning turlari va asosiy tushunchalar.....	202
	6.2. Kompyuter tarmoqlarining tuzilishi va topologiyalari.....	207
	6.3. Ma‘lumotlar kommutatsiyasi va marshrutlash mexanizmi .....	211
	6.4. Kompyuter tarmoqlari arxitekturasi va protokollari...	216
	6.5. Operatsion tizim tarmoq funksiyalari va turlari.....	225
<b>VII BOB</b>	<b>OPERATSION TIZIMLARNING TARMOQ XIZMATLARI</b>	
	7.1. Mijoz-server arxitekturasi, tuzilishi va asosiy tushunchalari.....	231
	7.2. Tarmoq operatsion tizimlari va ularning asosiy tarmoq xizmatlari.....	239
	7.3. Tarmoq kompyuterlarini tashkil etish va boshqarish mexanizmlari .....	249
	7.4. Tarmoq fayllarini tashkil etish va boshqarish mexanizmlari .....	258
	7.5. Tarmoq dasturlarini tashkil etish va boshqarish mexanizmlari .....	269



	7.6. Domen tizimi va internet xizmatlarini tashkil etish....	276
<b>VIII BOB</b>	<b>OPERATSION TIZIMLARDA XAVFSIZLIK MASALALARI</b>	
	8.1. Operatsion tizimning asosiy xavfsizlik masalalari .....	289
	8.2. Operatsion tizimga bo‘ladigan tahdidlar va ularning turlari .....	295
	8.3. Operatsion tizimning himoya vositalari va dasturlari.	301
	8.4. Operatsion tizimning autentifikatsiya usullari.....	305
	8.5. Axborot xavfsizligini ta’minlashda kriptografiya masalalari.....	311
	8.6. Xavfsizlik sertifikatlari va elektron raqamli imzo .....	317
<b>IX BOB</b>	<b>AMALIYOTDA OPERATSION TIZIMLARDAN FOYDALANISH</b>	
	9.1. Linux operatsion tizimi, qurilish tamoyili va ahamiyati .....	326
	9.2. Windows operatsion tizimi, qurilish tamoyili va ahamiyati .....	359
	9.3. Android mobil operatsion tizimi, qurilish tamoyili va ahamiyati.....	392
	<b>FOYDALANILGAN ADABIYOTLAR.....</b>	<b>407</b>

**U.R.XAMDAMOV, DJ.B.SULTANOV,  
S.S.PARSIYEV, U.M.ABDULLAYEV**

## **OPERATION TIZIMLAR**

*(O'quv qo'llanma)*

**Toshkent – «NIHOL PRINT» OK – 2021**

Muharrir: A.Tog'ayev  
Tex. muharrir: F.Tog'ayeva  
Musavvir: B.Esanov  
Musahhiha: O.Muxammadiyeva  
Kompyuterda  
sahifalovchi: G.Tog'ayeva

9323



№ 7439-765f-47f1-7ea1-a683-4648-1314.  
Bosishga ruxsat etildi: . Bichimi 60x841 /16.  
Shartli bosma tabog'i 26,0. Nashr bosma tabog'i 25,75.  
Adadi100. Buyurtma № 66.

«Nihol print» Ok da chop etildi.  
Toshkent sh., M. Ashrafiy ko‘chasi, 99/101.