

**Министерство цифровых технологий Республики Узбекистан
Ташкентский университет информационных технологий
имени Мухаммада аль-Хоразми**

Факультет Компьютерный Инжиниринг

Кафедра "Компьютерные системы"

**Методические указания к выполнению практических
работ по предмету
"АРХИТЕКТУРА КОМПЬЮТЕРА"
для студентов заочной формы обучения и второго
высшего образование**

Ташкент-2023

UDK: 004.72 (075.8)

Авторы: Ражабов Ф.Ф., Ирмухамедова Н.А. “Архитектура компьютера”
Методические указания ТУИТ. 33 листов. Ташкент- 2023.

Настоящая методическая указания направлена на закрепление теоретических знаний, полученных на лекциях и формирование практических навыков, включает в себя темы, цель, задачи, относящиеся к теме, порядок выполнения, а также структуру отчета и список использованной литературы.

Методическое указание к практическим занятиям по предмету “Архитектура компьютера” составлен в соответствии учебной программы и предназначено для студентов заочной формы обучения и второго высшего образования направления “60610500 - Компьютер инжиниринг («Вычислительная инженерия»).

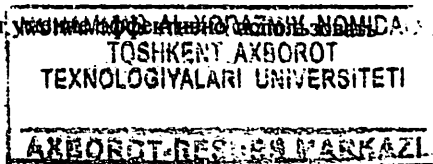
Рецензенты:

- Джуманов Ж.Х. – Профессор кафедры компьютерных систем, доктор технических наук Ташкентского университета информационных технологий имени Мухаммада Аль-Хоразми;
- Эшмуродов Д. – ТГТУ, к.т.н. доцент кафедры «Метрология стандартизация и сертификация»

ВВЕДЕНИЕ

Сегодня с развитием компьютерных технологий возрастает потребность в создании быстрых и эффективных программ. В настоящее время для создания конкурентоспособной программы требуется, чтобы программа была не только функциональной, но и эффективной. Эффективность означает, что программа сокращает временные затраты за счет использования всех ресурсов компьютера. В этом отношении практическое обучение науке об архитектуре компьютера служит эффективным руководством. В данном руководстве по эксплуатации рассмотрены функциональная и структурная организация современных компьютеров и вычислительных систем, программирование параллельной обработки информации, а также параллельное программирование с использованием мультиплексорных, многоядерных, кластерных и конвейерных вычислительных архитектур, а также наиболее популярные и эффективные методы распараллеливания. представлены библиотеки и технологии, представлены принципы и алгоритмы. В начале темы освещаются понятия организации ЭВМ, теоретическая и практическая части многоядерных и суперкомпьютеров и алгоритмы их работы, в конце практической работы приводятся теоретические и практические примеры современных библиотек распараллеливания, и представлены алгоритмы работы каждого инструмента распараллеливания, а информация об их возможностях полностью проиллюстрирована с помощью примеров и полученных результатов с помощью рисунков, а последовательность работы с примерами представлена на уровне понятности учащихся.

Предмет архитектура компьютера и считается предметом, преподаваемым в 5-м семестре на уровне бакалавриата, и к этому времени студенты будут иметь знания о принципах программирования и организации компьютера. В рамках данного предмета объединяются два вышеуказанных предмета и формируется предмет «Архитектура компьютерных систем».



внутренние ресурсы компьютера. В ходе практических занятий у студента разовьется умение использовать наиболее популярные и эффективные библиотеки и технологии распараллеливания, а также эффективное использование компьютерных процессоров.

Практическая работа №1

Организация общей структуры компьютерной системы.

Цель работы: изучить общую структуру и функции компьютерной системы, получить информацию о характеристиках системы персонального компьютера.

1. Теоретическая часть

История компьютера насчитывает несколько 10 лет. Существует пять популярных поколений компьютеров. Каждое поколение стало свидетелем нескольких технологических достижений, которые изменили функциональность компьютеров. Компьютеры играют важную роль во всех сферах жизни. Архитектура компьютера обычно определяется набором функций, необходимых пользователю.

1.1. Компьютер и его основные компоненты

Компьютер (англ. Computer) — это ЭВМ (Электронная вычислительная машина), что в переводе с английского означает «вычислитель» или «вычислять».

Персональный компьютер (англ. personal computer) — это компьютер, предназначенный для личного пользования, цена, размеры и возможности которого удовлетворяют потребности обычных людей. Это компьютер, созданный как вычислительная машина, но в настоящее время все чаще используемый как средство доступ к компьютерным сетям.

Ноутбук(англ. laptop) — это портативный персональный компьютер, сочетающий в себе обычные компоненты персонального компьютера, включая дисплей, клавиатуру и указывающее устройство (обычно сенсорную панель), а также аккумуляторы. Ноутбуки имеют небольшие размеры и вес, а время автономной работы ноутбуков варьируется от 1 часа до 15 часов.

Персональные компьютеры содержат все компоненты, необходимые для полного функционирования стандартного компьютера.

Системный блок — это основной компонент компьютера, который содержит основные компоненты системы. Сюда входят такие устройства, как основная логическая плата, процессор, память, один или несколько дисководов различных типов, импульсный источник питания, а также соединительные провода и кабели. Системный блок также имеет платы расширения для обеспечения аудио, видео, сетевых и других функций.

Клавиатура — это компьютерное устройство ввода, которое используется для ввода символов и команд в систему.

Мышь — это устройство ввода, используемое с графическим интерфейсом пользователя для отображения, выбора или активации изображений на мониторе. Перемещая мышь по гладкой поверхности, пользователь заставляет курсор синхронно перемещаться по дисплею компьютера.

Монитор — это стандартное устройство отображения для персонального компьютера, которое используется для отображения символов и графики в действительных цветах(true color).

Принтер — это устройство для печати копии оцифрованного изображения, которое переносит данные на бумажный или пленочный носитель. Обычно для вывода информации на бумагу используют матричные, струйные и лазерные принтеры.

Общую структуру компьютерной системы составляют устройства, которые необходимы для нормального функционирования компьютера. Эта составляющие для ввода любых данных в центральный процессор с помощью устройств ввода, таких как мышь, сканер, клавиатура и т. д., вывода на экран устройством вывода которые составляют базовую структуру и функциональные возможности компьютера (рис. 1.1).



Рисунок 1.1 - Базовые составляющие компоненты компьютера

Оперативная память (ОЗУ- англ. Random Access Memory, RAM) — это энергозависимый тип компьютерной памяти, который обеспечивает оперативный доступ к любой ячейке по ее адресу для чтения или записи. Любая процессором выполняемая часто используемые процессы - машинная программа(код) за короткое время загружает в оперативную память, что служит для ускорения работы компьютера. Основной единицей измерения ОЗУ является ее объем памяти, который измеряется в килобайтах, мегабайтах, гигабайтах. Частота работы ОЗУ определяется как количество потоков данных, проходящих через шину ОЗУ за заданный период времени.

1.2. Центральный процессор.

ЦП (центральный процессор- англ. CPU (Central Processing Unit)) — это интегральная микросхема (или часть его), которая выполняет инструкции, содержащие компьютерную программу, считающуюся центральным процессором или основным процессором компьютера. ЦП является основным блоком любого компьютера, и этот часть компьютерной системы является мозгом компьютера. Центральный процессор управляет работой всех частей компьютера.

1.3. Основная память

Основная память — одно из основных устройств компьютера, предназначенное для хранения операционной системы, программ и данных. Чтобы выполнить любую задачу на компьютере, вы должны сначала ввести информацию или инструкции в компьютер.

1.4. Устройства ввода и вывода.

Устройства ввода-вывода (анг. Input/Output unit) - устройства, которые помогают получать всю необходимую информацию от компьютера и наоборот. Блок ввода и вывода действует как связующее звено между пользователями и компьютером.

Поскольку компьютер становится важным в нашей жизни в настоящее время, нам необходимо иметь информацию об этом компьютере и его архитектуре.

2. Практическая часть

Для того, чтобы иметь полную характеристику персонального компьютера, нам необходимо иметь информацию об основных устройствах нашего компьютера, процессоре, процессорных ядрах, частотах и т.д. Существует несколько способов определения характеристик нашего персонального компьютера, в качестве примера можно привести программу AIDA64. Программа AIDA64 - работает на операционных системах Windows, iOS, Android, и эта программа отображает подробную информацию о компонентах компьютера.

В данной практической работе используется программу AIDA64.exe для определения характеристик персонального компьютера. Для определения основных характеристик персонального компьютера выполняются следующие действия в указанной последовательности:

1-шаг. На этом шаге скачивается AIDA64.exe по ссылке <https://www.aida64.com/> и устанавливается программа на персональный компьютер (рис. 1.2).

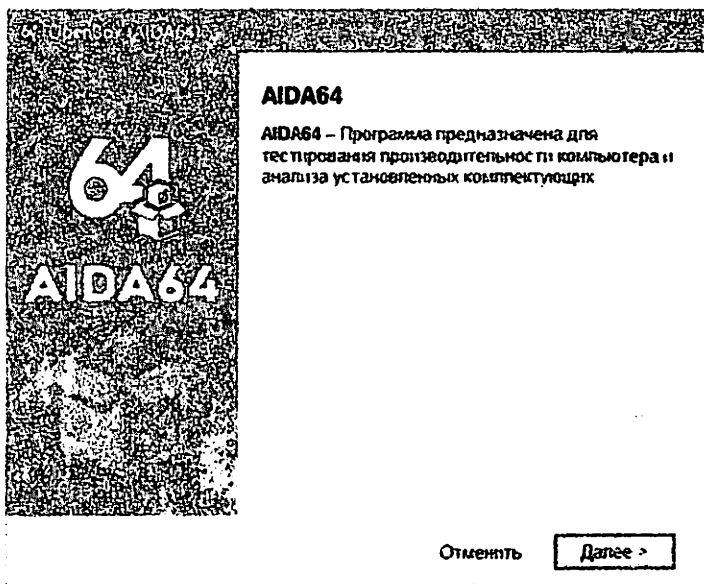


Рисунок 1.2 - Последовательность установки AIDA64.exe

2-шаг. После установки программы запускается и появляется главный экран программы (рис. 1.3).

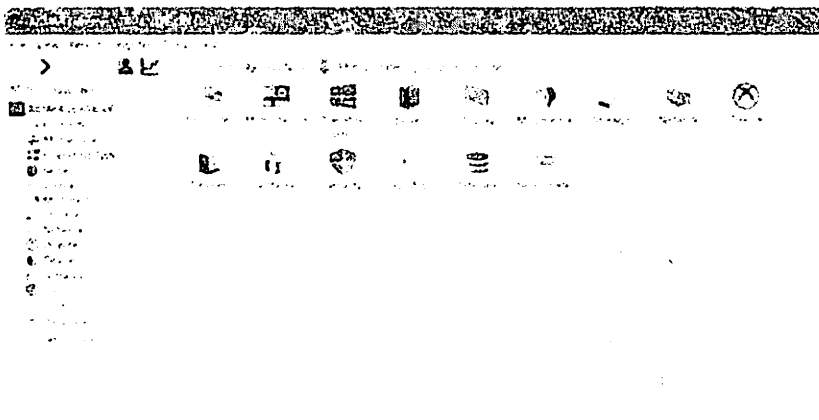


Рисунок 1.3 - Главное окно программы AIDA64.exe

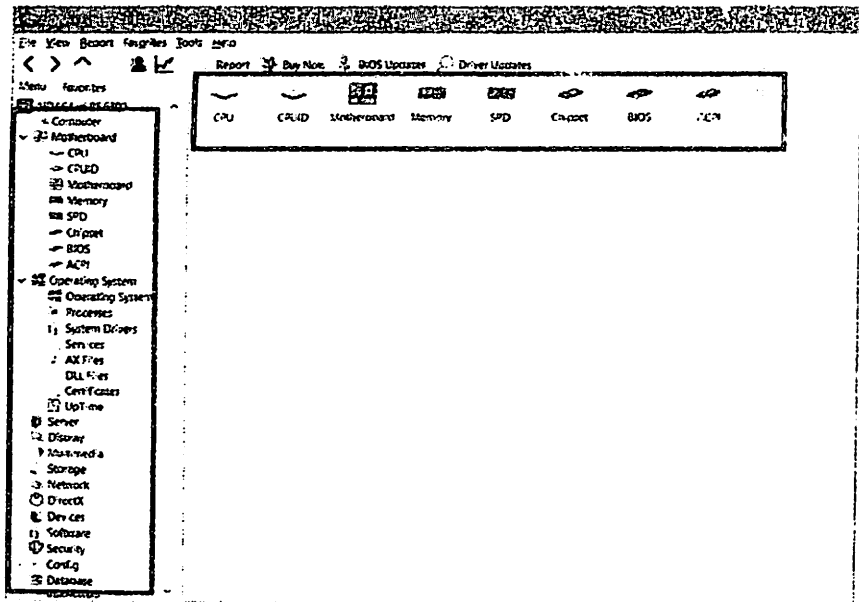


Рисунок 1.4 - Просмотр характеристик персонального компьютера с помощью программы AIDA64.exe

Таким образом с помощью программы AIDA64.exe, представленной в практической части выше, мы можем получить всю информацию о характеристиках нашего персонального компьютера и также тестировать составляющие части компьютера.

3. Задания

1. Представить и объяснить информацию об основных устройствах вашего персонального компьютера на основе таблицы(табл.1).
2. Это задание выполняется по подгруппам. Подгруппа выбирается в соответствии ниже приведенных требований к компьютеру:
 - 1-группа: для офисных программ, для обработки видео, аудио-данных.
 - 2- группа: для обработки видео и изображений.

3- группа: компьютер с дополнительным графическим процессором.

4- группа: для разработки игр и приложений.

5- группа: высокопроизводительный компьютер.

3. Перечислите основные возможности вашего персонального компьютера по группам.

1-таблица

Характеристики персонального компьютера

Модель компьютера	
Процессор	
Разрядность процессора	
RAM (ОЗУ)	
Video карта (GPU)	
...	

4. Включить результат выполнения задачи (снятие скриншота экрана компьютера во время работы программы) в отчет.

4.Содержание:

- а) Титульный лист;
- б) Полноценное выполнение вышеизложенных пунктов по примеру, который приведен ниже;
- в) Скриншоты поэтапного выполнения заданий к работе;
- г) Заключение по выполненной работе.

5. Форма отчета

- 1. Название практической работы.
- 2. Компьютер и его составные части.

3. Теоретическая информация, то есть описание функций, используемых для выполнения задачи.

4. Совместимость задач, отнесенных к разделу групп, приведенных в заданиях.

5. Сводная и использованная литература.

6. Требуется: Текстовый редактор - Microsoft Word, шрифт - Times New Roman, интервал - 1,5, размер - 14.

6. Контрольные вопросы

1. Из каких основных частей состоит компьютер?
2. Расскажите о составляющих компьютера.
3. Перечислите основные части компьютера.
4. Перечислите вспомогательные части компьютера.
5. Объясните основные характеристики современных компьютерных систем.

Практическая работа №2.

Виды архитектур многоядерных процессоров

Цель работы: Иметь общее понятие о многоядерных процессорах и рассмотреть методы определения ядер процессора персонального компьютера.

1. Теоретическая часть

Архитектура — это в широком смысле описание сложной системы, состоящей из множества взаимосвязанных элементов. Быстродействие компьютера во многом зависит от процессора, поэтому при выборе процессора необходимо иметь полную характеристики и информацию.

1.1. Многоядерные процессоры.

Многоядерный процессор — это центральный процессор, содержащий 2 или более вычислительных ядер в одном процессорном кристалле или в одном корпусе (рис. 2.1).

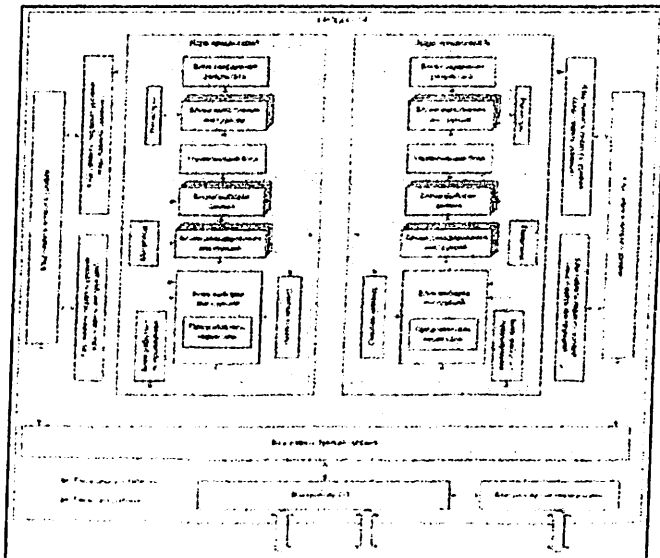


Рисунок 2.1- Функции каждого ядра многоядерного процессора

Многоядерный процессор — это интегрированный чип с двумя или более процессорными ядрами, подключенными для повышения производительности и снижения энергопотребления. Эти процессоры позволяют более эффективно работать с несколькими задачами одновременно, например, параллельная обработка и многопоточность. Первая архитектура многоядерных процессоров была реализована в модели Intel Core. На рисунке 2.1 показаны функции двух ядер процессора и описаны этапы выполнения одного цикла обработки, видно, что функции ядер одинаковы. Вот отличительные черты архитектуры:

- динамическое выполнение до четырех команд с использованием 14-ступенчатого конвейера;
- интеллектуальное управление внутренними регистрами и кэшем (использование кэша второго уровня совместно с обоими ядрами процессора);
- улучшена обработка мультимедийных команд, за один процессорный цикл;
- выполнение многих составных 128-битных команд обработки мультимедиа.

1.1. Архитектура многопроцессорных компьютеров

Многоядерный процессор многопроцессорных компьютерах обеспечивает связь между всеми доступными ядрами, и они соответствующим образом распределяют и назначают все задачи обработки. Обработанные данные с каждого ядра передаются на основную плату компьютера (материнскую плату) по общей шине после завершения всех операций обработки. Такой способ организации компьютера, относительно компьютера одно процессорной организации по общей, превосходит по производительности (рис. 2.2).

Поскольку у процессоров есть общая память, необходимо разумно использовать ее и согласовывать данные. В многопроцессорной системе

несколько кэшей работают для общего ресурса памяти. Когерентность кэша — это свойство кэша, которое обеспечивает целостность данных, хранящихся в отдельных кэшах для общего ресурса. Эта концепция является частным случаем концепции когерентности памяти, когда несколько ядер имеют доступ к общей памяти (что повсеместно распространено в современных многоядерных системах). Если описать эти понятия в общем смысле, то картина следующая: один и тот же блок данных может загружаться в разные кэши, где данные обрабатываются по-разному.

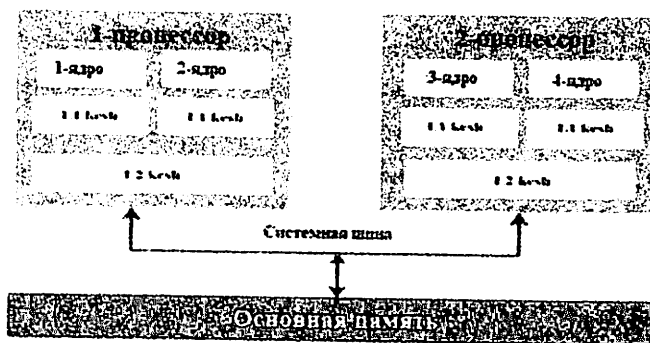


Рисунок 2.2-Архитектура многопроцессорных компьютеров.

Каждый многоядерный процессор состоит из двух или более ядер, а также нескольких кэшей. Кэши — это относительно небольшие области очень быстрой памяти, которые хранят часто используемые инструкции или данные и делают это содержимое легкодоступным для ядра без обращения к системной памяти. Процессор сначала проверяет кэш, и если требуемый контент доступен, ядро извлекает этот контент из кэша и тем самым добывается высокая производительность системы.

Если содержимое недоступно, ядро обращается к системной памяти за необходимым содержимым. Некоторые многоядерные архитектуры ЦП могут разделять кэши L1 и L2. Основываясь на предполагаемых преимуществах архитектуры многоядерных процессоров, производители

сосредотачиваются на ней. Реализация этой технологии оказалась очень дешевой и универсальной, что позволило вывести ее на широкий рынок.

В настоящее время на современном рынке компьютерной техники, то можно увидеть широкое распространение получили компьютеры с четырехъядерными и восьмиядерными процессорами.

2. Практическая часть

В этом практическом руководстве показаны несколько способов ознакомления с ядрами ПК:

1-способ. Для этого щелкнут правой кнопкой мыши на иконку «Мой компьютер» на рабочем столе и нажать на пункт «Управление», в левой части получившегося окна найти пункт «Диспетчер устройств», а в правой части откроется список процессоров. Нажать на пункт «Процессоры» и можно увидеть количество ядер данного персонального компьютера (рис. 2.3).

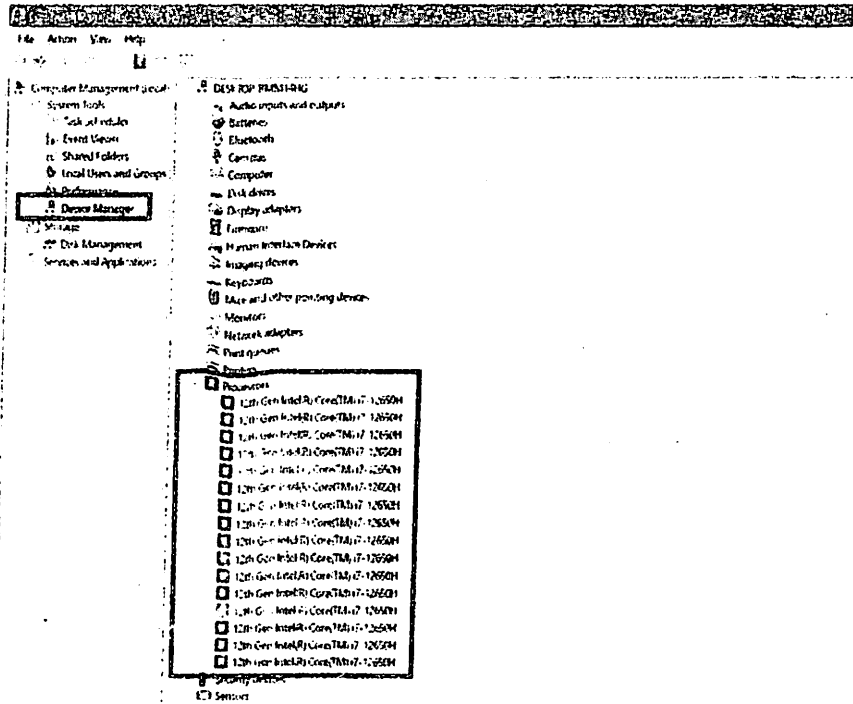


Рисунок 2.3- Просмотр ядер ПК

2-способ. Командную строку ПК (черное окно) ввести команду: *"WMIC CPU Get DeviceID, NumberOfCores"* (рис. 2.4).

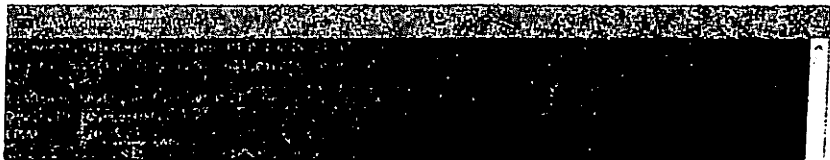
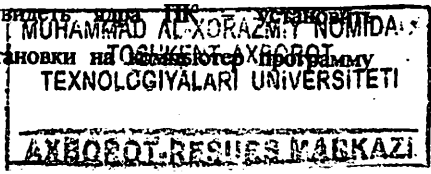


Рисунок 2.4- Просмотр ядер ПК через cmd (черное окно).

3-способ. Еще один способ увидеть ядра ПК - установить дополнительную программу. После установки на компьютер программы



CPU-Z.exe (рис. 2.5) будет доступно полная информация о ядрах процессора данного компьютера.

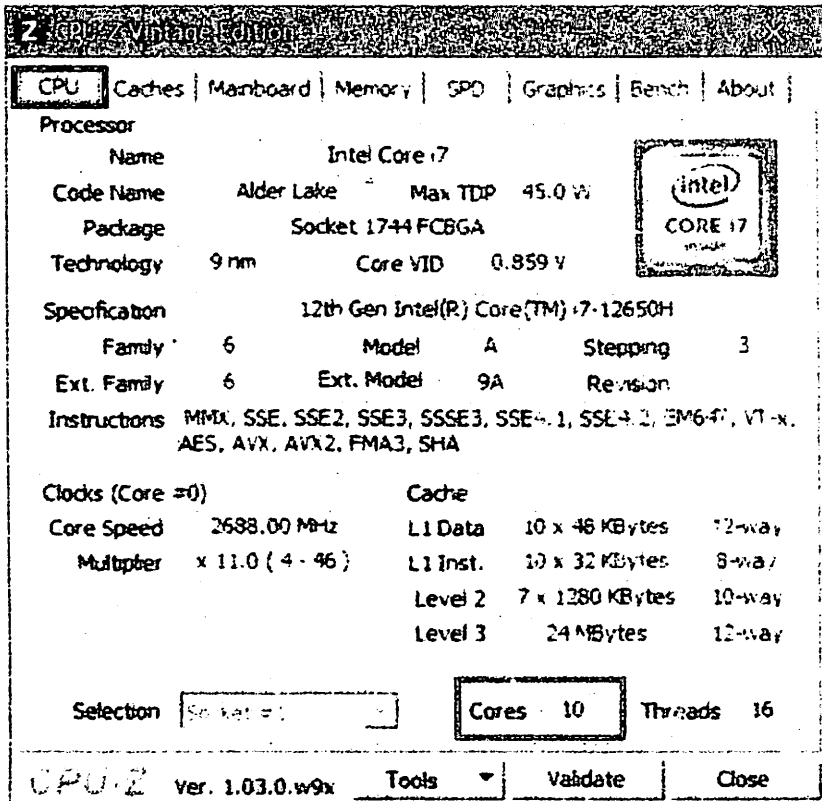


Рисунок 2.5- Просмотр ядер персонального компьютера через программу CPU-Z.exe

2. Задания

1. Сведите в таблицу и аннотируйте все данные ядра ПК.
2. Задание выполняется по подгруппам. Разделите группы для определения:

1-группа: многоядерные процессоры и их классификация;

- 2- группа: 2 (двухъядерных) процессора и их классификация;
 - 3- группа: 4 (четыре) ядерный процессор;
 - 4- группа: 8 (восемь) ядерный процессор;
 - 5- группа: Современные многоядерные процессоры.
3. Определите задачи, по группам и процессору также его количеству ядер, используя методы, указанные в практической части вашего персонального компьютера, и определите все характеристики.
4. Включите результат выполнения задания (сфотографировав экран компьютера во время практической работы) в отчет

3. Содержание:

- а) Титульный лист;
- б) Полноценное выполнение вышеизложенных пунктов по примеру, который приведен ниже;
- в) Скрин-шоты поэтапного выполнения заданий к работе;
- г) Заключение по выполненной работе.

4. Форма отчета

- 1. Название практической работы.
- 2. Многоядерные процессоры и их типы.
- 3. Теоретическая информация, то есть описание характеристик, используемых для выполнения задачи.
- 4. Совместимость задач, отнесенных к разделу групп, приведенных в заданиях.
- 5. Сводная и использованная литература.
- 6. Требуется: Текстовый редактор - Microsoft Word, шрифт - Times New Roman, интервал - 1,5, размер - 14.

5. Контрольные вопросы.

- 1. Расскажите про процессор и его ядра?**
- 2. Объясните типы многоядерных процессоров и их отличия?**
- 3. Объясните разницу между алгоритмами производительности двухъядерных и четырехъядерных процессоров?**
- 4. Каковы характеристики многоядерных процессоров?**
- 5. Объясните тенденции развития современных многоядерных процессоров?**

Практическая работа №3

Принципы работы MPI в кластерных компьютерах

Цель работы: получить представление о применении MPI-программ в кластерных компьютерах. Получить навыки конфигурирования программы MPICH2.

1. Краткая теория

MPI (Message Passing Interface) – интерфейс обмена сообщениями (информацией) между одновременно работающими вычислительными процессами. Он широко используется для создания параллельных программ для вычислительных систем с распределённой памятью (кластеров).

MPICH — самая известная реализация MPI, созданная в Арагонской национальной лаборатории (США). Существуют версии этой библиотеки для всех популярных операционных систем. К тому же, она бесплатна. Перечисленные факторы делают MPICH идеальным вариантом для того, чтобы начать практическое освоение MPI.

В данной практической работе речь пойдёт об MPICH2. Двойка в названии — это не версия программного обеспечения, а номер того стандарта MPI, который реализован в библиотеке. MPICH2 соответствует стандарту MPI 2.0, отсюда и название.

MPICH2 — это быстродействующая и широко портируемая реализация стандарта MPI (реализованы оба стандарта MPI-1 и MPI-2). Цели создания MPICH2 следующие:

1. Предоставить реализацию MPI, которая эффективно поддерживает различные вычислительные и коммуникационные платформы, включая общедоступные кластеры (настольные системы, системы с общей памятью, многоядерные архитектуры), высокоскоростные сети (Ethernet 10 Гбит/с, InfiniBand, Myrinet, Quadrics) и эксклюзивные вычислительные системы (Blue Gene, Cray, SiCortex).

2. Сделать возможными передовые исследования технологии MPI с помощью легко расширяемой модульной структуры для создания производных реализаций.

В данном случае имеется кластер из несколько машин (*вычислительных узлов*), работающих под управлением Microsoft Windows. Для учебных целей можно запускать все вычислительные процессы и на одном компьютере. Если компьютер одноядерный, то, естественно, прироста быстродействия вы не получите, — только замедление.

В качестве среды разработки можно использовать Visual Studio 2019 ... 2022 или Dev C++. Для удобства изложения созданная программа, использующую MPI, и предназначенную для запуска на нескольких вычислительных узлах, называется *MPI-программой*.

1.1. Принципы работы MPICH

Начнем изложение материала о программе MPICH2, с принципов её работы.

MPICH для Microsoft Windows состоит из следующих компонентов:

- Менеджер процессов `smprd.exe`, который представляет собой системную службу (сервисное приложение). Менеджер процессов ведёт список вычислительных узлов системы, и запускает на этих узлах MPI-программы, предоставляя им необходимую информацию для работы и обмена сообщениями.
- Заголовочные файлы (`.h`) и библиотеки стадии компиляции (`.lib`), необходимые для разработки MPI-программ.
- Библиотеки времени выполнения (`.dll`), необходимые для работы MPI-программ.
- Дополнительные утилиты (`.exe`), необходимые для настройки MPICH и запуска MPI-программ.

Все компоненты, кроме библиотек времени выполнения, устанавливаются по умолчанию в папку C:\Program Files\MPICH2; dll-библиотеки устанавливаются в C:\Windows\System32.

Менеджер процессов является основным компонентом, который должен быть установлен и настроен на всех компьютерах сети (библиотеки времени выполнения можно, в крайнем случае, копировать вместе с MPI-программой). Остальные файлы требуются для разработки MPI-программ и настройки некоторого «головного» компьютера, с которого будет производиться их запуск.

Менеджер работает в фоновом режиме и ждёт запросов к нему из сети со стороны «головного» менеджера процессов (по умолчанию используется сетевой порт 8676). Чтобы как-то обезопасить себя от хакеров и вирусов, менеджер требует пароль при обращении к нему. Когда один менеджер процессов обращается к другому менеджеру процессов, он передаёт ему свой пароль.

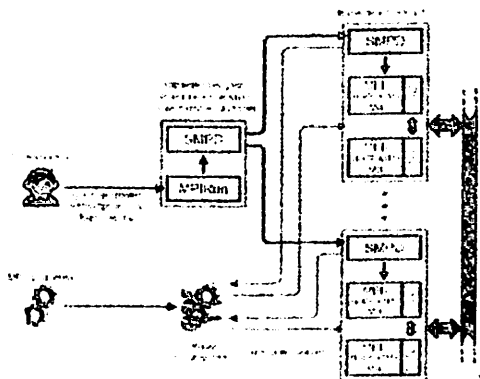


Рисунок 3.1- Схема работы MPICH на кластере.

В современных кластерах «сеть передачи данных» обычно отделяется от «управляющей сети»

Запуск MPI-программы производится следующим образом (смотрите рис. 3.1.):

1. Пользователь с помощью программы Mpirun (или Mpiexec, при использовании MPICH2 под Microsoft Windows) указывает имя исполняемого файла MPI-программы и требуемое число процессов. Кроме того, можно указать имя пользователя и пароль: процессы MPI-программы будут запускаться от имени этого пользователя.
2. Mpiexec передаёт сведения о запуске локальному менеджеру процессов, у которого имеется список доступных вычислительных узлов.
3. Менеджер процессов обращается к вычислительным узлам по списку, передавая запущенным на них менеджерам процессов указания по запуску MPI-программы.
4. Менеджеры процессов запускают на вычислительных узлах несколько копий MPI-программы (возможно, по несколько копий на каждом узле), передавая программам необходимую информацию для связи друг с другом.

Очень важным моментом здесь является то, что перед запуском MPI-программа не копируется автоматически на вычислительные узлы кластера. Вместо этого менеджер процессов передаёт узлам путь к исполняемому файлу программы точно в том виде, в котором пользователь указал этот путь программе Mpiexec. Это означает, что если, например, запустить программу C:\prog.exe, то все менеджеры процессов на вычислительных узлах будут пытаться запустить файл C:\prog.exe. Если хотя бы на одном из узлов такого файла не окажется, произойдёт ошибка запуска MPI-программы.

Чтобы каждый раз не копировать вручную программу и все необходимые для её работы файлы на вычислительные узлы кластера, обычно используют *объект сетевой ресурс*. В этом случае пользователь копирует программу и дополнительные файлы на сетевой ресурс, видимый всеми узлами кластера, и указывает путь к файлу программы на этом ресурсе. Дополнительным удобством такого подхода является то, что при

наличии возможности записи на общий сетевой ресурс запущенные копии программы могут записывать туда результаты своей работы.

Работа MPI-программы происходит следующим образом:

1. Программа запускается и инициализирует библиотеку времени выполнения MPICH путём вызова функции `MPI_Init`.
2. Библиотека получает от менеджера процессов информацию о количестве и местоположении других процессов программы, и устанавливает с ними связь.
3. После этого запущенные копии программы могут обмениваться друг с другом информацией посредством библиотеки MPICH. С точки зрения операционной системы библиотека является частью программы (работает в том же процессе), поэтому можно считать, что запущенные копии MPI-программы обмениваются данными напрямую друг с другом, как любые другие приложения, передающие данные по сети.
4. Консольный ввод-вывод всех процессов MPI-программы перенаправляется на консоль, на которой запущена `Mpicxes`. Перенаправлением ввода-вывода занимаются менеджеры процессов, так как именно они запустили копии MPI-программы, и поэтому могут получить доступ к потокам ввода-вывода программ.
5. Перед завершением все процессы вызывают функцию `MPI_Finalize`, которая корректно завершает передачу и приём всех сообщений, и отключает MPICH.

Все описанные выше принципы действуют, даже если запустить MPI-программу на одном компьютере.

1.2. Запуск MPI-программ

Для запуска MPI-программ в комплект MPICH2 входит программа с графическим интерфейсом `Wmpicxes`, которая представляет собой оболочку вокруг соответствующей утилиты командной строки `Mpicxes`. К сожалению, `Wmpicxes` не всегда работает корректно, поэтому самый

подходящий способ запускать MPI-программы — это пользоваться непосредственно Mpiexec. Однако, рассмотрим Wmpirxcs, так как эта программа интуитивно понятнее.

Окно программы Wmpirxcs показано на рис. 3.2. (обратите внимание, что включён флажок «more options»).

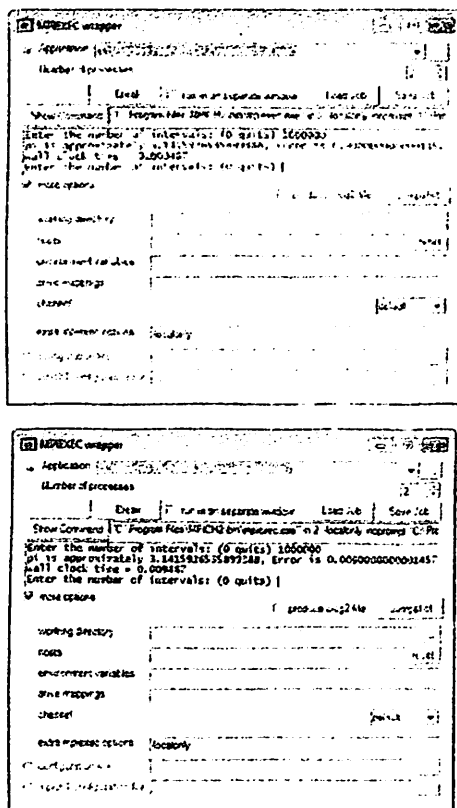


Рисунок 3.2- Программа Wmpirxcs

Элементы управления окна имеют следующий смысл:

Поле ввода «Application»: сюда вводится путь к MPI-программе. Как уже было сказано ранее, путь передаётся в неизменном виде на все

компьютеры сети, поэтому желательно, чтобы программа располагалась в общей сетевой папке.

- «Number of processes»: число запускаемых процессов. По умолчанию процессы распределяются поровну между компьютерами сети, однако это поведение можно изменить при помощи конфигурационного файла.
- Кнопка «Execute» запускает программу; кнопка «Break» принудительно завершает все запущенные экземпляры.
- Флажок «run in a separate window» перенаправляет вывод всех экземпляров MPI-программы в отдельное консольное окно.
- Кнопка «Show Command» показывает в поле справа командную строку, которая используется для запуска MPI-программы (Wmpiexec — оболочка над Mpiexec). Командная строка собирается из всех настроек, введенных в остальных полях окна.
- Далее идёт большое текстовое поле, в которое попадает ввод-вывод всех экземпляров MPI-программы, если не установлен флажок «run in a separate window».
- Флажок «more options» показывает дополнительные параметры.
- «working directory»: сюда можно ввести рабочий каталог программы. Опять же, этот путь должен быть верен на всех вычислительных узлах. Если путь не указан, то в качестве рабочего каталога будет использоваться место нахождения MPI-программы.
- «hosts»: здесь можно указать через пробел список вычислительных узлов, используемых для запуска MPI-программы. Если это поле пустое, то используется список, хранящийся в настройках менеджера процессов текущего узла.
- «environment variables»: в этом поле можно указать значения дополнительных переменных окружения, устанавливаемых на всех узлах на время запуска MPI-программы. Синтаксис следующий: имя_1=значение_1, имя_2=значение_2.

- «drive mappings»: здесь можно указать сетевой диск, подключаемый на каждом вычислительном узле на время работы MPI-программы. Синтаксис: Z:\\winsrv\wdir.
- «channel»: позволяет выбрать способ передачи данных между экземплярами MPI-программы.
- «extra prtexes options»: в это поле можно ввести дополнительные ключи для командной строки Mprtexes.

2. Задания

Для вариантов 1-5:

Протестировать MPI-приложение по адресу C:\Program Files\MPICH2\examples\cpi.exe.

В программе Wmpitexes указать число процессов i (номер варианта), и задать число интервалов $1350*i$.

Для вариантов 6-10:

Протестировать MPI-приложение по адресу C:\Program Files\MPICH2\examples\vittua.exe.

В программе Wmpitexes указать число процессов i (номер варианта), и задать число интервалов $750*i$.

Для вариантов 11-15:

Протестировать MPI-приложение по адресу C:\Program Files\MPICH2\examples\muna.exe.

В программе Wmpitexes указать число процессов i (номер варианта), и задать число интервалов $1700*i$.

3. Содержание:

- а) Титульный лист;
- б) Полноценное выполнение вышеизложенных пунктов по плану, который приведен ниже;
- в) Скриншоты поэтапного выполнения заданий к работе;

г) Заключение по выполненной работе.

4. Форма отчета

1. Название практической работы.
2. Многоядерные процессоры и их типы.
3. Теоретическая информация, то есть описание характеристик, используемых для выполнения задачи.
4. Совместимость задач, отнесенных к разделу групп, приведенных в заданиях.
5. Сводная и использованная литература.
6. Требуется: Текстовый редактор - Microsoft Word, шрифт - Times New Roman, интервал - 1,5, размер - 14.

5. Контрольные вопросы

- 1 – Что из себя представляет MPI-программа?
- 2 – В чем особенности программы MPICH2?
- 3 – Принципы работы MPICH2.
- 4 – Компоненты программы MPICH2.
- 5 – Запуск MPI-программы.
- 6 – Запуск MPI-программы с использованием Wmpirerex.exe.
- 7 – перечислите поля, используемые в интерфейсе программы Wmpirerex.exe, и выполняемые в них команды.

СПИСОК ЛИТЕРАТУРЫ

Основная литература

1. Rajabov F.F., Imuxamedova N.A., Atadjanova N.S. Raqamli axborotlarni qayta ishlash va yaratish texnologiyasi. Toshkent, O'zR FA "Fan" nashriyoti, 2021. 272 b
2. Мусаев М.М. "Процессоры современных компьютеров". Олий ўқув кўртлари учун қўлланма. Тошкент.: "Алоқачи" нашриёти, 2020 йил. 12 боб. – 512 бет.
3. Ражабов Ф.Ф., Ирмухамедова Н.А., Ахмадова К. Учебное пособие к выполнению практических работ по дисциплине «ПРОЕКТИРОВАНИЕ СИСТЕМ VLSI». ТУИТ. Ташкент, 2022 - 130 с.
4. A.Clements. "Computer Organization and Architecture" Cengage Learning . Australia – 2018. – 816 p.
5. Х.Э.Хужаматов, Н.М.Ахмедов, Ф.Ф. Ражабов, Э.Рейпназаров «Компьютерные сети. Методические указания по выполнению лабораторных работ». ТУИТ, Ташкент, 2021 -120с.
6. Таненбаум Э., Остин Т. Архитектура компьютера // 6-е издание. СПб.: Питер, 2013. — 811 с.
7. Буза М.К. Архитектура компьютеров. Учебник для вузов.- Минск. Новое знание. 2006. – 559с.
8. Grytsenko, V. I., Misuno, I. S., Rachkovskiy, D. A., "The concept and architecture of the software neurocomputer SNC," Control systems and machines, (3), 3 -14 (2004).
9. Барановская Т.П., Лойко В.И., Семенов М.И., Трубилин А.И. Архитектура компьютерных систем и сетей. Учебное пособие.- М., Финансы и статистика, 2003.- 256с.

Интернет-адреса

1. <https://www.gecksforgecks.org/multicomputer/>
2. <https://education.molssi.org/parallel-programming/>
3. <https://mitpress.mit.edu/neural-computing-architectures/>
4. <https://www.osp.m/osp/>
5. <https://www.openmp.org/resources/openmp-compilers-tools/>
6. <https://www.cs.nuim.ie/MPI.htm>
7. <https://ds.cs.buc.edu/mpi/mpi.html>

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
Практическая работа №1. Организация общей структуры компьютерной систем	5
Практическая работа №2. Виды архитектур многоядерных процессоров	13
Практическая работа №3. Принципы работы MPI в кластерных компьютерах	21
СПИСОК ЛИТЕРАТУРЫ	30

«Архитектура компьютера» Методические указания

Методическое указание предназначено

для студентов бакалавриата 4 курса

ТУИТ имени Мухаммада ал-Хоразмий по специальности
60610500 – Компьютерный инжиниринг (Компьютерный инжиниринг).

Рассмотрен и рекомендован к изданию
на заседании кафедры Компьютерные системы

Протокол № 12 от 22 февраля 2023 г.

Рассмотрен и рекомендован к изданию
учебно-методическим советом факультета Компьютерный инжиниринг

Протокол № 9 от 14 марта 2023 г.

Рассмотрен и рекомендован к изданию
учебно-методическим советом ТУИТ имени Мухаммада ал Хоразмий.

Протокол № 9 (166) от 27 июня 2023 г.

Составители:

Ф.Ф. Ражабов

Н.А. Ирмухамедова

Рецензенты:

Ж. Х. Джуманов

Д.Эшмуродов

Формат 60x84 1/16. Печ. лист 2,25

Заказ № 7. Тираж 20.

Отпечатано в «Редакционно издательском»
отделе при ТУИТ.

Ташкент ул. Амир Темур, 108.