CHALMERS

EDA222/DIT161: Real time systems, 2012

Collection of examples

Contents

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg 2012

Worst Case Execution Time estimation

1. Consider a processor clocked at 100 MHz. Assume that there are instructions that can be executed during a clock cycle. State the least possible”time unit” that can be expressed as an integer and also represent execution of every instruction.

2. Consider the following code fragment. Time is supposed to be an integer of type ”time unit” and the numbers within parenthesizes denote the calculated WCET for the line.

```
...
1:    (4)   :     a := b + c;
2:    (5)   :     if( a < 9 ) then
3:    (64)  :           S1;
4:    (5)   :     else if a < 100 then
5:    (112) :           S2;
6:    (2)   :     else
7:    (112) :           S3;
8:    (1)   :     end;
...
```

   a) State the possible paths (using line indexes) through the code.
   b) Using Shaw’s method, estimate WCET for the possible paths.

3. Consider the procedure `Main` below. Assume that:
   - The cost for *assignment*, *return* and *comparison* is one time unit.
   - A *function call* overhead is two time unit.
   - *Addition* and *subtraction* costs are four time units.
   - Other language constructs will not generate any code so they are “null” cost.

   a) Using Shaw’s method, estimate WCET for the function Calculate in terms of z.
   b) The main() function calls Calculate(4) using parameter 4. Using Shaw’s method, estimate WCET for the main function.
   c) The deadline for the main() function is 95 time units. Determine whether the deadline is met.

```
int Calculate (int z){
            int R;

            if(z == 0)
                  R = 1;
            else if (z == 1)
                  R = 1;
            else
                  R = Calculate(z-1) + Calculate(z-2);
   return R;

         }

int main(){
int ans;
ans=Calculate(4);
}
```

Processor utilisation analysis

4. Consider the following task set:

Task	Period *T* [ms]	Deadline *D* [ms]	Execution time *C* [ms]
A	100	100	22
B	50	50	10
C	25	25	8

a) Calculate LCM (least common multiple) for the set.
b) Calculate the processor utilization factor.

5. Suppose conditions for RMSA is met by the following task set. Show if or if possibly not, the task set is schedulable.

Task	Period *T* [ms]	Deadline *D* [ms]	Execution time *C* [ms]	priority
A	7	7	1	0
B	14	14	1	1
C	18	18	4	2

Response time analysis

6. Suppose conditions for RMSA is met by the following task set. Show that:
a) This set is NOT necesserily schedulable due to RMSA?
b) It is, in fact, schedulable, due to response time analysis.

Task	Period *T* [ms]	Deadline *D* [ms]	Execution time *C* [ms]
A	30	30	10
B	40	40	10
C	60	60	14

7. Consider the following task set:

Task	Period *T* [ms]	Deadline *D* [ms]	Execution time *C* [ms]
A	10	7	3
B	12	6	4

a) Determine if the set is schedulable due to Rate Monotonic priority assignment.
b) Determine if the set is schedulable due to Deadline Monotonic priority assignment.
A full motivation is required.

8. Consider the following task set:

Task	Period *T* [ms]	Deadline *D* [ms]	Execution time *C* [ms]
A	70	65	15
B	40	40	10
C	30	12	10

Assign priorities according to deadline monotonic and then, does every task within this set meet its deadline?

9. Consider the following task set:

Task	Period *T* [ms]	Deadline *D* [ms]	Execution time *C* [ms]
A	1000	20	3
B	100	100	10
C	50	50	20
D	57	10	5
E	33	33	1
F	7	7	1

a) Calculate processor utilization for the set.
b) Give the execution order for “rate monotonic” priority assignment.
c) Give the execution order for “deadline monotonic” priority assignment.
d) State maximum response times for “deadline monotonic” priority assignment.
e) Will all deadlines be met?
f) State maximum response times for “rate monotonic” priority assignment.
g) Will all deadlines be met?
h) Add another task “FT” with period 30, deadline 5, execution time (WCET) 2. Assuming “deadline monotonic” priority assignment, will all deadlines be met?

10. Consider the following three tasks:

Task	Period T [ms]	Deadline D [ms]	Execution time C [ms]	priority
A	50	10	5	1
B	500	500	240	2
C	3000	3000	1000	3

The tasks utilize semaphores, s_1, s_2 and s_3 as follows:

Task	***Usage***
A	*lock(s_1); unlock(s_1);*
B	*lock(s_2); lock(s_3); unlock(s_3); unlock(s_2);*
C	*lock(s_3); lock(s_2); unlock(s_2); unlock(s_3);*

The critical sections lengths are as follows:

A uses s_1 max 5 ms	$cs_{P1,S1}$ = 5 ms
B uses s_2 max 10 ms	$cs_{P2,S2}$ = 10 ms
B uses s_3 max 5 ms	$cs_{P2,S3}$ = 5 ms
C uses s_2 max 10 ms	$cs_{P3,S2}$ = 10 ms
C uses s_3 max 25 ms	$cs_{P3,S3}$ = 25 ms

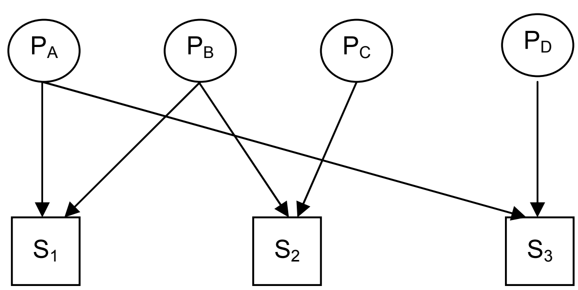
Note that time $cs_{P2,S3}$ is included in $cs_{P2,S2}$.

Assume that ICPP (Immediate Ceiling Priority Protocol) is used to lock and unlock the semaphores. Is this task set then schedulable?

11. The following task set should be scheduled due to *deadline monotonic*.
Three semaphores; S_1, S_2 and S_3 are used to synchronize the tasks.
H_{Si} denotes the maximum locking time when a task locks semaphore *i*.
P_A, P_B, P_C and P_D denote the tasks A,B,C and D.

Task	T [ms]	D [ms]	C [ms]	Priority	H_{S1} [ms]	H_{S2} [ms]	H_{S3} [ms]
A	5	4	2	1	1	-	1
B	16	12	3	2	1	2	-
C	20	16	3	3	-	3	-
D	28	28	4	4	-	-	2

The following access graph illustrates use of the semaphores.



Assume that ICPP (Immediate Ceiling Priority Protocol) is used to lock and unlock the semaphores. Is this task set then schedulable?

12. Consider the following task set:

Task	Period *T* [ms]	Deadline *D* [ms]	Execution time *C* [ms]
A	7	7	2
B	8	8	1
C	10	10	3
D	25	25	2

Priorities are assigned due to *deadline monotonic*.

a) Show that a processor utilization test cannot guarantee that this set is schedulable.
b) Determine response times due to response time analysis.
c) Add two semaphores *S* and *Q*. Assume ICPP and the following:
Task B and task C uses *S*, task C and task D uses *Q*, *critical section S*=1 ms and *critical section Q* = 2ms
Determine response times due to response time analysis considering these semaphores.

13. Determine the maximum number of bits in the following CAN-messages:
    a) 2 bytes in standard CAN
    b) 8 bytes in standard CAN
    c) 3 bytes in EXTENDED CAN
    d) 7 bytes in EXTENDED CAN.

14. The following message set shall be scheduled on a CAN-bus at bitrate 50 Kbit/s:

Message	*T* [ms]	*D* [ms]	*# of bytes*
M1	10	4	1
M2	20	6	5
M3	15	10	6

Assign priorities due to deadline monotonic and:
    a) Determine response times.
    b) Calculate the bus utilization factor.

Processor demand analysis

15. The following task set should be scheduled due to *earliest deadline first* (EDF).

Task	*T* [ms]	*D* [ms]	*C* [ms]
A	3	2	1
B	4	2	1
C	5	4	2

a) Calculate processor utilization factor.
b) Draw a timing diagram showing a possible scenario for execution order. (“simulation”).
c) Determine if the task set can be scheduled by performing processor demand analysis.

16. A real-time system with three periodic tasks is scheduled due to EDF. The run-time system executes all tasks with preemption and due to task priorities. The following table details periods (*T*), deadlines (*D*) and worst case execution times (*C*). All tasks arrives at $t = 0$.

Task	*T* [ms]	*D* [ms]	*C* [ms]
A	4	4	3
B	10	4	1
C	20	16	3

a) Show that Liu & Layland's simple utilization based test is inapplicable in this case.
b) Use *processor demand analysis* to determine whether the task set is schedulable or not

17. A real-time system with five periodic tasks is scheduled due to EDF. The run-time system executes all tasks with preemption and due to task priorities. The following table details periods (*T*), deadlines (*D*) and worst case execution times (*C*). All tasks arrives at $t = 0$.

Task(P)	*T* [ms]	*D* [ms]	*C* [ms]
1	6	6	2
2	15	15	2
3	16	16	4
4	10	10	2
5	15	15	1

a) Use utilization based test to show that this task set is schedulable due to EDF.
b) Draw a timing diagram ($t = 0 – 32$) showing how the tasks will be executed due to EDF.

Solutions: Worst Case Execution Time estimation

1. Duration (period) of a 100 MHz frequency is 10 ns. Instruction execution time is stated in 'clock cycles' by manufacturers. Every instruction execution time must thus be a multiple of the 10 nanoseconds. Thus 'time unit' = 10 ns is an obvious choice.

2. a) Possible paths are:
   1,2,3,8
   1,2,4,5,8
   1,2,4,6,7,8
   b)
   path: 1,2,3,8 = 4+5+64+1 = 71.
   path: 1,2,4,5,8 = 4+5+5+112+1=127.
   path: 1,2,4,6,7,8=4+5+5+2+112+1=129

3.

a) Case1: If z is 0

WCET("Calculate", 0)= ("Dec", R) + ("Com", z==0)+ ("Assign", R)+ "return"= 4

Case2: If z is 1

WCET("Calculate", 1)= ("Dec", R)+ ("Com", z==0)+ ("Com", z==1)+ ("Assign", R)+ "return"
= 5

Case3: If z is greater than 1

WCET("Calculate", z)= ("Dec", R)+ ("Com", z==0)+ ("Com", z==1)
+ ("sub", z-1)+ ("call", Calculate(z-1))+ WCET("Calculate", z-1))
+ ("sub", z-2)+ ("call", Calculate(z-2))+ WCET("Calculate", z-2))
+ ("add", Calculate(z-1) and Calculate(z-2))+ ("assign",R)+ "return"
=1+1+1+4+2+WCET("Calculate", z-1))+4+2
+WCET("Calculate", z-2))+4+1+1
=21+WCET("Calculate", z-1))+WCET("Calculate", z-2))

b) WCET("Calculate", 4) =21+WCET("Calculate", 3)+WCET("Calculate", 2)
=21+ (21+WCET("Calculate", 2) +WCET("Calculate", 1))
+ (21 +WCET("Calculate", 1)+WCET("Calculate", 0))
=21+21+(21+WCET("Calculate", 1)+WCET("Calculate", 0)))
+WCET("Calculate", 1))
+ (21 +WCET("Calculate", 1)+WCET("Calculate", 0))
=21+21+21+5+4+5+21+5+4=107

WCET("main")=("Dec", ans) + ("call", Calculate(4))+ WCET("Calculate", 4)+ "assign"
= 1+2+107+1 = 111

c) Since the deadline 95 is smaller than 111, the deadline of "main" is missed.

4. a) It's straight forward that's LCM is 100, a "formal" check yields:
   $\frac{100}{100}=1, \quad \frac{100}{50}=2, \frac{100}{25}=4$
   i.e there is no smaller divisor, so *Least Common Multiple* must be 100.

   b) Utilization factor (sum up c/p):
   $\frac{22}{100}+\frac{10}{50}+\frac{8}{25}\approx 0{,}74=74\%$

5. RMSA requires the following condition to be fulfilled:
   $\sum_{i=1}^{n}\frac{c_i}{p_i}\leq n\,(2^{1/n}-1)$
   Evaluation of left hand yields:
   $\frac{1}{7}+\frac{1}{14}+\frac{4}{18}\approx 0{,}43$

   Evaluation of right hand (n=3) yields:
   $3\,(2^{1/3}-1)\approx 0{,}78$

   I.e. inequality is true and thereby the task set is schedulable.

Solutions: Response time analysis

6. a) Utilization factor (left hand) for this task set is approx. 0,817.
   Calculating right hand yields approx. 0780.
   Thus LH > RH and the required condition for schedulability according to RMSA is NOT met.

   b) Applying response time analysis:

$$R_i^{n+1} = C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i^n}{T_j} \right\rceil C_j$$

Response time for R_a:

$$R_a^0 = 10$$

Response time for R_b:

$$R_b^0 = 10$$

$$R_b^1 = 10 + \left\lceil \frac{10}{30} \right\rceil 10 = 10 + 10 = 20$$

$$R_b^2 = 10 + \left\lceil \frac{20}{30} \right\rceil 10 = 10 + 10 = 20$$

Response time for R_c:

$$R_c^0 = 14$$

$$R_c^1 = 14 + \left\lceil \frac{14}{30} \right\rceil 10 + \left\lceil \frac{14}{40} \right\rceil 10 = 14 + 10 + 10 = 34$$

$$R_c^2 = 14 + \left\lceil \frac{34}{30} \right\rceil 10 + \left\lceil \frac{34}{40} \right\rceil 10 = 14 + 20 + 10 = 44$$

$$R_c^3 = 14 + \left\lceil \frac{44}{30} \right\rceil 10 + \left\lceil \frac{44}{40} \right\rceil 10 = 14 + 20 + 20 = 54$$

$$R_c^4 = 14 + \left\lceil \frac{54}{30} \right\rceil 10 + \left\lceil \frac{54}{40} \right\rceil 10 = 14 + 20 + 20 = 54$$

Gives us:

Task	Period *T* [ms]	Deadline *D* [ms]	Execution time *C* [ms]	***R***
A	30	30	10	10
B	40	40	10	20
C	60	60	14	54

I.e: $R \leq D$ for tasks A,B and C, which proves that the task set is in fact schedulable according to response time analysis.

7. a) Order is PA, PB and the response time for PB becomes: 3+4 =7, FAIL.

   b) Order is PB, PA and the response time for PA becomes: 4+3 =7, OK.

8. Task execution order is: C,B,A

Response time for R_A	$R_A^{i+1} = \lceil R_A^i/T_B \rceil * C_B + \lceil R_A^i/T_C \rceil * C_C + C_A = \lceil R_A^i/40 \rceil * 10 + \lceil R_A^i/30 \rceil * 10 + C_A$

i=0	0+0+15=15
i = 1	10+10+15=35
i = 2	10+20+15=45
i = 3	20+20+15=**55**
i = 4	20+20+15=**55**

Response time for R_B	$R_B^{i+1} = \lceil R_B^i/T_C \rceil * C_C + C_B = \lceil R_B^i/30 \rceil * 10 + 10$

1	0+10=10
2	10+10 =**20**
3	10+10 =**20**

$R_C = C_C =$ **10**

Check: 55< 65, 20 < 40, 10 < 12, OK.

9.

a) Utilisation:
(3 + 100 + 400 + 5*1000/57 + 1 *1000/37 + 1*1000/7)/1000=0.76

b) Rate monotonic priorities:
F, E, C, D, B, A

c) Deadline monotonic priorities:
F, D, A, E, C, B

d) R_B

$$R_B^{i+1} = \lceil R_B^i/T_C \rceil * C_C + \lceil R_B^i/T_E \rceil * C_E + \lceil R_B^i/T_A \rceil * C_A + \lceil R_B^i/T_D \rceil * C_D + \lceil R_B^i/T_F \rceil * C_F + C_B = \lceil R_B^i/50 \rceil * 20 + \lceil R_B^i/33 \rceil * 1 + \lceil R_B^i/1000 \rceil * 3 + \lceil R_B^i/57 \rceil * 5 + \lceil R_B^i/7 \rceil * 1 + 10$$

1	0+10=10
2	20+1+3+5+2+10=41
3	20+2+3+5+6+10=46
4	20+2+3+5+7+10=47
5	20+2+3+5+7+10=47

R_C

$$R_C^{i+1} = \lceil R_C^i/T_E \rceil * C_E + \lceil R_C^i/T_A \rceil * C_A + \lceil R_C^i/T_D \rceil * C_D + \lceil R_C^i/T_F \rceil * C_F + C_C = \lceil R_C^i/33 \rceil * 1 + \lceil R_C^i/1000 \rceil * 3 + \lceil R_C^i/57 \rceil * 5 + \lceil R_C^i/7 \rceil * 1 + 10$$

1	0+20=20
2	1+3+5+3+20=32
3	2+3+5+5+20=35
4	1+3+5+5+20=35

R_E

$$R_E^{i+1} = \lceil R_E^i/T_A \rceil * C_A + \lceil R_E^i/T_D \rceil * C_D + \lceil R_E^i/T_F \rceil * C_F + C_E = \lceil R_E^i/1000 \rceil * 3 + \lceil R_E^i/57 \rceil * 5 + \lceil R_E^i/7 \rceil * 1 + 1$$

1	0+1=1
2	3+5+1+1=10
3	3+5+2+1=11
4	3+5+2+1=11

R_A

$$R_A^{i+1} = \lceil R_A^i/T_D \rceil * C_D + \lceil R_A^i/T_F \rceil * C_F + C_A = \lceil R_A^i/57 \rceil * 5 + \lceil R_A^i/7 \rceil * 1 + 3$$

1	0+3=3
2	5+1+3=9
3	5+2+3=10
4	5+2+3=10

R_D

$$R_A^{i+1} = \lceil R_A^i/T_F \rceil * C_F + C_A = \lceil R_A^i/7 \rceil * 1 + 3$$

1	0+5=5
2	1+5=6
3	1+5=6

$R_F = 1$

e) Ia all $R \le D$? => Yes, schedulable!

f) Response times for Rate monotonic priority assignments:

$$R_x^{i+1} = C_X + \Sigma \lceil R^i/T_j \rceil * C_j$$

R_A

$$R_A^{i+1} = \lceil R_A^i/T_B \rceil * C_B + ... + \lceil R_A^i/T_F \rceil * C_F + C_A =$$
$$\lceil R_A^i/100 \rceil * 10 + \lceil R_A^i/57 \rceil * 5 + \lceil R_A^i/50 \rceil * 20 + \lceil R_A^i/33 \rceil * 1 + \lceil R_A^i/7 \rceil * 1 + C_3$$

1	0+3=3
2	10+20+5+1+1+3=40
3	10+20+5+2+7+3=47
4	10+20+5+2+7+3=47

R_B

$$R_B^{i+1} = \lceil R_A^i/T_C \rceil * C_C + ... + \lceil R_A^i/T_F \rceil * C_F + C_B = \lceil R_B^i/57 \rceil * 5 + \lceil R_B^i/50 \rceil * 20 + \lceil R_B^i/33 \rceil * 1 + \lceil R_B^i/7 \rceil * 1 + 10$$

1	0+10=10
2	20+5+1+2+10=38
3	20+5+2+6+10=43
4	20+5+2+7+10=44
5	20+5+2+7+10=44

R_D

$$R_D^{i+1} = \lceil R_D^i/T_E \rceil * C_E + \lceil R_D^i/T_F \rceil * C_F + C_D = \lceil R_D^i/50 \rceil * 20 + \lceil R_D^i/33 \rceil * 1 + \lceil R_D^i/7 \rceil * 1 + 5$$

1	0+5=5
2	20+1+2+5=28
3	20+1+4+5=30
4	20+1+5+5=31
5	20+1+5+5=31

R_C

$$R_C^{i+1} = \lceil R_C^i/T_E \rceil * C_E + \lceil R_C^i/T_F \rceil * C_F + C_C = \lceil R_C^i/33 \rceil * 1 + \lceil R_C^i/7 \rceil * 1 + 20$$

1	0+20=20
2	1+3+20=24
3	1+4+20=25
4	1+4+20=25

R_E

$R_E^{i+1} = \lceil R_E^i/T_F \rceil * C_F + C_E = \lceil R_E^i/7 \rceil * 1 + 1$

1	0+1=1
2	1+1=2
3	1+1=2

R_A 47
R_B 44
R_C 25
R_D 31
R_E 2
R_F 1

g) $R_A=47 > D_A=20$ i.e. FAIL!
$R_D=31 > D_D=10$ ie. FAIL!

h) Response times Deadline monotonic priority assignments

New task set:

	T	D	C
a	1000	20	3
b	100	100	10
c	50	50	20
d	57	10	5
e	33	33	1
f	7	7	1
ft	30	5	2

$R_x^{i+1} = C_X + \Sigma \lceil R^i/T_j \rceil * C_j$

R_B
(start with 47)

$R_B^{i+1} = \lceil R_B^i/T_C \rceil * C_C + \lceil R_B^i/T_E \rceil * C_E + \lceil R_B^i/T_A \rceil * C_A + \lceil R_B^i/T_D \rceil * C_D + \lceil R_B^i/T_F \rceil * C_F + \lceil R_B^i/T_{FT} \rceil * C_{FT} + C_B =$
$\lceil R_B^i/50 \rceil * 20 + \lceil R_B^i/33 \rceil * 1 + \lceil R_B^i/1000 \rceil * 3 + \lceil R_B^i/57 \rceil * 5 + \lceil R_B^i/7 \rceil * 1 + \lceil R_B^i/30 \rceil * 2 + 10$

1	20+2+3+5+7+4+10=51
2	40+2+3+5+8+4+10=72
3	40+3+3+10+11+6+10=83
4	40+3+3+10+12+6+10=84
5	40+3+3+10+12+6+10=84

R_C
(start with 35)

$R_C^{i+1} = \lceil R_C^i/T_E \rceil * C_E + \lceil R_C^i/T_A \rceil * C_A + \lceil R_C^i/T_D \rceil * C_D + \lceil R_C^i/T_F \rceil * C_F + \lceil R_C^i/T_{FT} \rceil * C_{FT} + C_C =$
$\lceil R_C^i/33 \rceil * 1 + \lceil R_C^i/1000 \rceil * 3 + \lceil R_C^i/57 \rceil * 5 + \lceil R_C^i/7 \rceil * 1 + \lceil R_C^i/30 \rceil * 2 + 10$

1	39
2	40
3	40

R_E
(start with 11)

$R_E^{i+1} = \lceil R_E^i/T_A \rceil * C_A + \lceil R_E^i/T_D \rceil * C_D + \lceil R_E^i/T_F \rceil * C_F + \lceil R_E^i/T_{FT} \rceil * C_{FT} + C_E =$
$\lceil R_E^i/1000 \rceil * 3 + \lceil R_E^i/57 \rceil * 5 + \lceil R_E^i/7 \rceil * 1 + \lceil R_E^i/30 \rceil * 2 + 1$

1	13
2	13

R_A
(start with 10)

$R_A^{i+1} = \lceil R_A^i/T_D \rceil * C_D + \lceil R_A^i/T_F \rceil * C_F + \lceil R_A^i/T_{FT} \rceil * C_{FT} + C_A = \lceil R_A^i/57 \rceil *5 + \lceil R_A^i/7 \rceil *1 + \lceil R_A^i/30 \rceil *2+3$

1	12
2	12

R_D
(start with 6)

$R_D^{i+1} = \lceil R_D^i/T_F \rceil * C_F + \lceil R_D^i/T_F \rceil * C_F + C_D = \lceil R_D^i/7 \rceil *1 + \lceil R_{FT}^i/30 \rceil *2+3$

1	8
2	9
3	9

R_F
(start with 1)

$R_F^{i+1} = \lceil R_F^i/T_F \rceil * C_F + \lceil R_F^i/T_F \rceil * C_F + C_F = \lceil R_{FT}^i/30 \rceil *2+3$

1	3
2	3

$R_{FT} = 2$

Task	R	D	
FT	2	5	OK
F	3	7	OK
D	9	10	OK
A	12	20	OK
E	13	33	OK
C	40	50	OK
B	84	100	OK

10.

On the basis of how processes use semaphores we will determine the priority ceilings of semaphores:

$uses(P1) = s_1$	$ceil(s_1) = pri(P1) = 1$
$uses(P2) = s_2, s_3$	$ceil(s_2) = max\{ pri(P2), pri(P3)\} = 2$
$uses(P3) = s_3, s_2$	$ceil(s_3) = max\{ pri(P2), pri(P3)\} = 2$

We can now determine b_{P1}:
- "Examine all processes with lower priority", i.e, P2 and P3
- "Determine which of these processes use semaphores", i.e, s2 and s3.
- "Pick out the semaphores that have higher (or the same) ceiling priorities as the current process". (no such semaphores are here ...)

Accordingly, $b_{P1} = 0$.

We can now determine b_{P2}:
- processes with lower priority than P2, i.e P3
- which semaphores these processes use ", i.e., s2 and s3.
- semaphores which have a higher (or the same) ceiling priority as P2 i.e, s2 and s3

Maximum blocking is $\max\{cs_{P3,S2}, cs_{P3,S3}\} = 25$ ms.

There is no lower prioritry process than P3. So, $b_{P3} = 0$.

The result can be summarized in the following table:

Task	*pri*	*T*	*D*	*C*	*B*
A	1	50	10	5	0
B	2	500	500	240	25
C	3	3000	3000	1000	0

Now response time analysis with consideration of the blocking yields the following table:

Task	*pri*	*T*	*D*	*C*	*B*	*R*
A	1	50	10	5	0	5
B	2	500	500	240	25	295
C	3	3000	3000	1000	0	2445

I.e. all $R_i < D_i$, schedulable.

11.

$$R_i^{n+1} = C_i + B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i^n}{T_j} \right\rceil C_j$$

We first have to determine each blocking factor B_i. To do this we need the ceiling priorities.
Let 1 be high priority, then we have P_A (pri) = 1, P_B (pri) = 2, P_C (pri) = 3, P_D (pri) = 4.
Ceiling priorities: For each semaphore, find the highest priority among the tasks that uses this semaphore, this is the semaphores ceiling priority:

ceil{S_1} = max {P_A (pri), P_B (pri)} = max {1,2} = 1
ceil{S_2} = max {P_B (pri), P_C (pri)} = max {2,3} = 2
ceil{S_3} = max {P_A (pri), P_D (pri)} = max {1,4} = 1

Now we identify for each task P_i, which lower priority tasks that may interfere (block P_i).

P_A can be blocked by P_B and P_D since they use semaphores with a ceiling priority that is higher or equal to the priority of P_A.
Blocking factor for P_A becomes:

B_A = max { P_B uses S_1, P_B uses S_2, P_D uses S_3 } =
max { $P_B(H_{S1})$,$P_B(H_{S2})$, $P_D(H_{S3})$ } =
max { 1 , 2 , 2 } = **2**

P_B can be blocked by P_C and P_D.
Blocking factor for P_B becomes:

B_B = max { P_C uses S_2, P_D uses S_3 } =
max { $P_C(H_{S2})$, $P_D(H_{S3})$ } =
max { 3 , 2 } = **3**

P_C can be blocked by P_D.
Blocking factor for P_C becomes:

B_C = max { P_D uses S_3 } =
max { $P_D(H_{S3})$ } =
max { 2 } = **2**

Now, P_D cannot be blocked since there are no lower priority tasks, thus we conclude:

B_A = max{1,2} = 2
B_B = max{3,2} = 3
B_C = max{2}= 2
B_D= 0

Finally we calculate the response times and check all deadlines:

Response time for P_A.

$R_A = C_A + B_A = 2 + 2 = 4 \le D_A = 4$: *OK!*

Response time for P_B.

Iteration	$R_B^{i+1} = C_B + B_B + \lceil R_B^i / T_A \rceil * C_A = C_B + B_B + \lceil R_B^i / 5 \rceil *2$

Setting start value $R_B^0 = C_B = 3$ yuilds:

1	3 + 3 + 2 = 8
2	3 + 3 + 4 = 10
3	$3 + 3 + 4 = 10 \le D_B = 12$: *OK!*

Response time for P_C.

Iteration	$R_C^{i+1} = C_C + B_C + \lceil R_C^i / T_B \rceil * C_B + \lceil R_C^i / T_A \rceil * C_A =$ $C_C + B_C + \lceil R_C^i / 16 \rceil *3 + \lceil R_C^i / 5 \rceil *2$

Setting start value $R_C^0 = C_C = 3$ yuilds:

1	3 + 2 + 3 + 2 =10
2	3 + 2 + 3 + 4 = 12
3	3 + 2 + 3 + 6 = 14
4	$3 + 2 + 3 + 6 = 14 \le D_C = 16$: *OK!*

Response time for P_D.

Iteration	$R_C^{i+1} = C_D + \lceil R_D^i / T_C \rceil * C_C + \lceil R_D^i / T_B \rceil * C_B + \lceil R_D^i / T_A \rceil * C_A =$ $C_D + \lceil R_D^i / 20 \rceil *3 + \lceil R_B^i / 16 \rceil *3 + \lceil R_B^i / 5 \rceil *2$

Setting start value $R_D^0 = C_D = 4$ yuilds:

1	4 + 3 + 3 + 2 =12
2	4 + 3 + 3 + 6 = 16
3	4 + 3 + 3 + 8 = 18
4	4 + 3 + 6 + 8 = 21
5	4 + 6 + 6 + 10 = 26
6	4 + 6 + 6 + 12 = 28
7	$4 + 6 + 6 + 12 = 28 \le D_D = 28$: *OK!*

I.e. All scheduled tasks will meet their deadline.

12. a) RMSA *requires* the following condition to be fulfilled:

$$\sum_{i=1}^{n} \frac{c_i}{p_i} \le n\,(2^{1/n} - 1)$$

Evaluation of left hand yields:

$$\frac{2}{7} + \frac{1}{8} + \frac{3}{10} + \frac{2}{25} \approx 0{,}79$$

Evaluation of right hand (n=4) yields:

$$4(2^{1/4} - 1) \approx 0{,}76$$

I.e. LH > RH, *not* schedulable according to this condition, QED...

b), c)

Task	*T*	*D*	*C*	*pri*	*B*	*R(b)*	*R(c)*
A	7	7	2	0	0	2	2
B	8	8	1	1	1	3	4
C	10	10	3	2	2	6	**11**
D	25	25	2	3	0	14	14

13.

Standard CAN: $\left\lfloor \frac{34 + 8i}{4} \right\rfloor + 47 + 8i$, where i is the number of data bytes in the message

Extended CAN: $\left\lfloor \frac{54 + 8i}{4} \right\rfloor + 67 + 8i$, where i is the number of data bytes in the message

a) 75
b) 135
c) 110
d) 150

14.
a)

Message	*D(us)*	*T(us)*	*bits*	*C(us)*	*B(us)*	*R(us)*
M1	4000	10000	65	1300	2300	3600
M2	6000	20000	105	2100	2300	5700
M3	10000	15000	115	2300	0	5700

$$R_i^{n+1} = C_i + B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i^n}{T_j} \right\rceil C_j$$

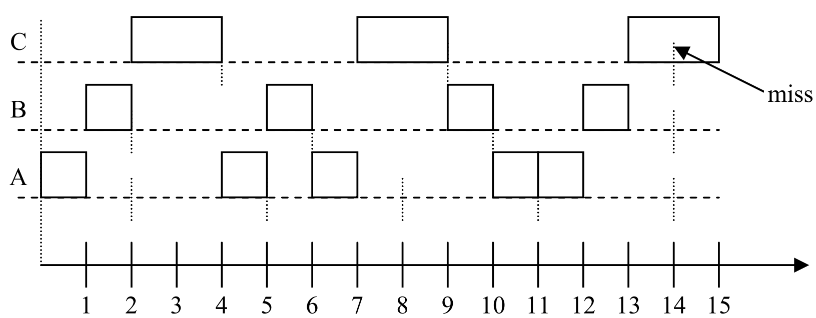
b) Bus utilisation ≈ 39%

Solutions: Processor demand analysis

15. a) Utilization factor:

$$U = \sum_{i=1}^{i=n} \left(\frac{C_i}{T_i} \right) = \frac{1}{3} + \frac{1}{4} + \frac{2}{5} = \frac{20}{60} + \frac{15}{60} + \frac{24}{60} = \frac{59}{60}$$

b) Timing diagram:



c) Processor demand calculation.
LCM { T_A , T_B , T_C } = LCM { 3, 4, 5 } = 60

Determine checkpoints K within interval [0,60].

For T_A we obtain the following checkpoints
K_A = { D^k_A = k*T_A + D_A , k = 0, 1, 2, 3, ...} = { 2, 5, 8, 11, 14, ...}

For T_B we obtain the following checkpoints
K_B = { D^k_B = k*T_B + D_B , k = 0, 1, 2, ...} = { 2, 6, 10, 14, ...}

For T_C we obtain the following checkpoints
K_C = { D^k_C = k*T_C + D_C , k = 0, 1, 2, ...} = { 4, 9, 14, 19, ...}

Processor demand C_P (0, L) thus must be checked for the following checkpoints.
$\boldsymbol{K} = K_A \cup K_B \cup K_C$ = { 2, 4, 5, 6, 8, 9, 10, 11, 14, 19, ...}

Consider the general expression for processor demand analysis:

$$C_p(0,L) = \sum N_i^L \times C_i = \sum \left(\left\lfloor \frac{L - D_i}{T_i} \right\rfloor + 1 \right) \times C_i$$

Each checkpoint is analysed in the following table:

L	$N^L_A * C_A$	$N^L_B * C_B$	$N^L_C * C_C$	$C_P(0, L)$	$C_P(0, L) \leq L$
2	(⌊(2-2) / 3⌋)+1)*1 = 1	(⌊(2-2) / 4⌋)+1)*1 = 1	(⌊(2-4) / 5⌋)+1)*2 = 0	2	OK
4	(⌊(4-2) / 3⌋)+1)*1 = 1	(⌊(4-2) / 4⌋)+1)*1 = 1	(⌊(4-4) / 5⌋)+1)*2 = 2	4	OK
5	(⌊(5-2) / 3⌋)+1)*1 = 2	(⌊(5-2) / 4⌋)+1)*1 = 1	(⌊(5-4) / 5⌋)+1)*2 = 2	5	OK
6	(⌊(6-2) / 3⌋)+1)*1 = 2	(⌊(6-2) / 4⌋)+1)*1 = 2	(⌊(6-4) / 5⌋)+1)*2 = 2	6	OK
8	(⌊(8-2) / 3⌋)+1)*1 = 3	(⌊(8-2) / 4⌋)+1)*1 = 2	(⌊(8-4) / 5⌋)+1)*2 = 2	7	OK
9	(⌊(9-2) / 3⌋)+1)*1 = 3	(⌊(9-2) / 4⌋)+1)*1 = 2	(⌊(9-4) / 5⌋)+1)*2 = 4	9	OK
10	(⌊(10-2) / 3⌋)+1)*1 = 3	(⌊(10-2) / 4⌋)+1)*1 = 3	(⌊(10-4) / 5⌋)+1)*2 = 4	10	OK
11	(⌊(11-2) / 3⌋)+1)*1 = 4	(⌊(11-2) / 4⌋)+1)*1 = 3	(⌊(11-4) / 5⌋)+1)*2 = 4	11	OK
14	**(⌊(14-2) / 3⌋)+1)*1 = 5**	**(⌊(14-2) / 4⌋)+1)*1 = 4**	**(⌊(14-4) / 5⌋)+1)*2 = 6**	**15**	**NOT OK!**

I.e. NOT schedulable since $C_P(0, 14) = 15$ exceeds length of the interval.

16.

a) Since $D_i < T_i$ for *at least* on task in the set, the simple L&L test don't apply.

b)

LCM{A,B,C} = LCM{4,10,20} = 20.

Checkpoints ***K***:

K_A = {4,8,12,16,20}

K_B = {4,14}

K_C = {16}

Now: $\boldsymbol{K} = K_A \cup K_B \cup K_C$ = { 4,8,12,14,16,20}

L	$N^L_A * C_A$	$N^L_B * C_B$	$N^L_C * C_C$	$C_P(0, L)$	$C_P(0, L) \leq L$
4	$(\lfloor(4-4) / 4\rfloor+1)*3 = 3$	$(\lfloor(4-4) / 10\rfloor+1)*1 = 1$	$(\lfloor(4-16) / 20\rfloor+1)*3 = 0$	4	OK
8	$(\lfloor(8-4) / 4\rfloor+1)*3 = 6$	$(\lfloor(8-4) / 10\rfloor+1)*1 = 1$	$(\lfloor(8-16) / 20\rfloor+1)*3 = 0$	7	OK
12	$(\lfloor(12-4) / 4\rfloor+1)*3 = 9$	$(\lfloor(12-4) / 10\rfloor+1)*1 = 1$	$(\lfloor(12-16) / 20\rfloor+1)*3 = 0$	10	OK
14	$(\lfloor(14-4) / 4\rfloor+1)*3 = 9$	$(\lfloor(14-4) / 10\rfloor+1)*1 = 2$	$(\lfloor(14-16) / 20\rfloor+1)*3 = 0$	11	OK
16	$\boldsymbol{(\lfloor(16-4) / 4\rfloor+1)*3 = 12}$	$\boldsymbol{(\lfloor(16-4) / 10\rfloor+1)*1 = 2}$	$\boldsymbol{(\lfloor(16-16) / 20\rfloor+1)*3 = 3}$	**17**	**NOT OK!**
20	$(\lfloor(20-4) / 4\rfloor+1)*3 = 15$	$(\lfloor(20-4) / 10\rfloor+1)*1 = 2$	$(\lfloor(20-16) / 20\rfloor+1)*3 = 3$	20	OK

I.e. NOT schedulable since $C_P(0, 16) = 17$ exceeds length of the interval.

17.

a) U =59/60. Since U not greater than 1 and Di=Ti, the task set is schedulable using EDF.

b)