Enel487 – Fall 2023 – Assign 3
Real-Time Analysis
Handed Out: [2023-10-19 Thu]
Due: [2023-10-24 Tue] 23h55
Dillan Zurowski
200431334

1. (10 marks) Consider a system that has three tasks with periods: 10 millisecond, 39 millisecond, and 1 second. If the WCETs have been estimated at 4 milliseconds, 12 milliseconds, and 98 milliseconds, respectively, what is the total time-loading of the system? (We are ignoring context switch time) Is the task set guaranteed to have a feasible schedule, by the RMS criterion? If not, what would be the easiest rewrite that would make the three tasks schedulable? Explain your answer

Task 1: Utilization = 4 ms / 10 ms = 0.4
Task 2: Utilization = 12 ms / 39 ms ≈ 0.3077 (rounded to four decimal places)
Task 3: Utilization = 98 ms / 1000 ms = 0.098

Total Utilization = 0.4 + 0.3077 + 0.098 ≈ 0.8057 (rounded to four decimal places)
RMS criterion for utilization with 3 tasks: n(2^(1/n)-1) , n=3 tasks
Max utilization = 0.7798
0.8057 >= 0.7798, therefore the task set is not guaranteed a feasible schedule by RMS criterion

The easiest rewrite to make the three tasks schedulable would be to increase the periods to reduce utilization, or optimize the code to reduce the worst-case execution times.

2. (20 marks) A preemptive system has three concurrent tasks, described by the table below (context switch time is ignored). The background, or idle task is assumed to be nonessential and is fully preemptable by all higher priority tasks.

Task	Cycle	Execution Time	Priority
TaskA	10ms	4ms	3 (highest)
TaskB	20ms	5ms	1
TaskC	40ms	10ms	2
Idle	(continuous)	5ms	—

(a) Answer the following:
i. What is the system utilization?

Utilization = execution time/cycle
TaskA: = 4ms / 10ms = 0.4
TaskB: = 5ms / 20ms = 0.25
TaskC: = 10ms / 40ms = 0.25

System Utilization = 0.4 + 0.25 + 0.25 = 0.9

ii. Is this task set RMS scheduled?
0.9 > 0.7798 so the task set is not RMS scheduled.

iii. What is the response time for each task?

Task	Cycle	Execution Time	Priority
TaskA	10ms	4ms	3 (highest)
TaskB	20ms	5ms	1
TaskC	40ms	10ms	2
Idle	(continuous)	5ms	—

Ri = Ci+(Ri/Pj)*Cj

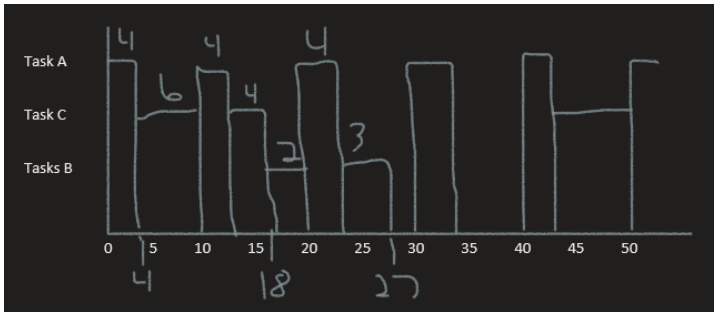
Task A (highest priority) - Response time = **4ms**

Task C
R1 = 10 + (0/10)*4 = 10+0=10
R1 = 10 + (10/10)*4 = 10+1*4) = 14ms
R1 = 10 + (14/10)*4 = 10 + 2*4) = 18ms
R1 = 10 + (18/10)*4 = 10 + 2*4) = **18ms**

Task B (lowest priority)
R2 = 5 + (5/40)*10+(5/10)*4 = 5 + 1*10 + 1*4 = 19
R2 = 5 + (19/40)*10+(19/10)*4 = 5 + 1*10 + 2*4 = 23
R2 = 5 + (23/40)*10+(23/10)*4 = 5 + 1*10 + 3*4 = 27
R2 = 5 + (27/40)*10+(27/10)*4 = 5 + 1*10 + 3*4 = **27ms**

iv. Do all the tasks meet their deadlines? By how much does each task beat, or miss, its deadline.
Does it meet -> response time <? Cycle time
Task A - 4ms < 10 so it meets its deadline by 6ms
Task B - 27 > 20ms so misses its deadline by 7ms
Task C - 19ms < 40ms so it meets its deadline by 21ms

v. Draw an execution time line for this system.



(b) Now suppose the priorities of Task B and C are interchanged, that is, TaskB has priority 2 and TaskC has priority 1. Answer the following:
i. What is the system utilization?

TaskA: = 4ms / 10ms = 0.4
TaskB: = 5ms / 20ms = 0.25
TaskC: = 10ms / 40ms = 0.25

System Utilization = 0.4 + 0.25 + 0.25 = 0.9

ii. What is the response time for each task?

Task	Cycle	Execution Time	Priority
TaskA	10ms	4ms	3 (highest)
TaskB	20ms	5ms	1
TaskC	40ms	10ms	2
Idle	(continuous)	5ms	—

Ri = Ci+(Ri/Pj)*Cj

Task A (highest priority) - Response time = **4ms**

Task B - Response time =
R1 = 5 + (0/10)*4 = 5+0=5
R1 = 5 + (5/10)*4 = 5+4=**9ms**

Task C (lowest priority)
R2 = 10 + (10/10)*4+(10/20)*5 = 10 + 1*4 + 1*5 = 19
R2 = 10 + (19/10)*4+(19/20)*5 = 10 + 2*4 + 1*5 = 23
R2 = 10 + (23/10)*4+(23/20)*5 = 10 + 3*4 + 2*5 = 32
R2 = 10 + (32/10)*4+(32/20)*5 = 10 + 4*4 + 2*5 = 36
R2 = 10 + (36/10)*4+(36/20)*5 = 10 + 4*4 + 2*5 = **36ms**

iii. Do all the tasks meet their deadlines? By how much does each task beat, or miss, its deadline.

Does it meet -> response time <? Cycle time

Task A - 4 < 10ms , meets deadline

Task B - 9 < 20, meets deadline

Task C - 36 < 40, meets deadline

iv. Draw an execution time line for this system.