ENSE452 Assignment 3

1. 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

   Time loading of the system: Ui = Ci / Pi where Ci is the WCET and Pi is the period
   - 4 ms / 10 ms = 0.4
   - 12 ms / 39 ms = 0.3077
   - 98 ms / 1000 ms = 0.098

   Total time loading of the system = 0.4 + 0.3077 + 0.098 = **0.8057**

   RMS criterion for utilization with n tasks: n (2^(1/n) - 1) = 3 (2^(⅓) - 1) = 0.7798 < 0.8057 therefore, the task set is not guaranteed to have a feasible schedule by the RMS criterion.

   The easiest rewrite would be to change the 39 millisecond period to be 40 milliseconds which would make this a harmonic set and U < 100% so it is now guaranteed to be a feasible schedule.

2. 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?

   Ui = Sum of (Ci/Pi) = (4 ms/ 10 ms) + (5 ms / 20 ms) + (10 ms / 40 ms) = 0.4 + 0.25 + 0.25 = 0.9 = **90%**

   ii. Is this task set RMS scheduled?

Max allowaable utilization = n (2^(1/n) - 1) = 3 (2^(⅓) - 1) = 0.7798 < 0.9 so the **task set is not RMS scheduled**

iii. What is the response time for each task?

WCET + (roundUp(previous result / period of previous task)) * previous WCET
Task A response time = **4 ms**

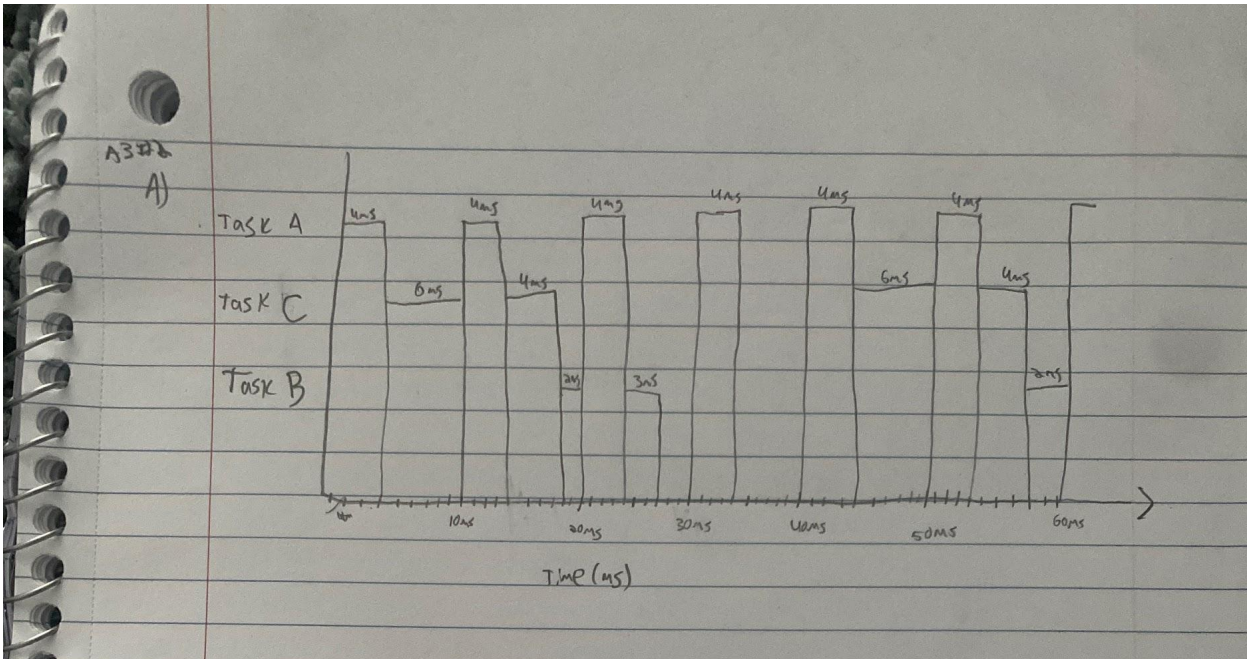
Task C response time 10ms + (0ms / 10ms) * 4 = 10 ms
10 ms + (10ms/10ms)*4ms = 14ms
10 ms + (14 ms/ 10ms) * 4 ms = 10 + 2 *4 = 18 ms
10 ms + (18/10) *4 = 10 + 2*4 = **18 ms Converged**

Task B response time = 5ms + (5ms / 40ms) * 10ms + (5/10) * 4 = 5 + 10 + 4 = 19ms
5ms + (19ms /40) * 10 + (19/10) * 4 = 5 + 10 + 8 = 23ms
5 + (18/40) * 10 + (23/10) * 4 = 5 + 10 + 12 = 27 ms
5 + (27/40) * 10 + (27/10) * 4 = 5 + 10 + 12 = **27 ms Converged**

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

Task A: 4 ms < 10ms Deadline beat by 6 ms
Task B: 27 ms > 20 ms Deadline missed by 7 ms
Task C: 18 ms < 40ms Deadliine beat by 22 ms

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?

System utilization is still **90%**

ii. What is the response time for each task?

Task A response time = **4 ms**

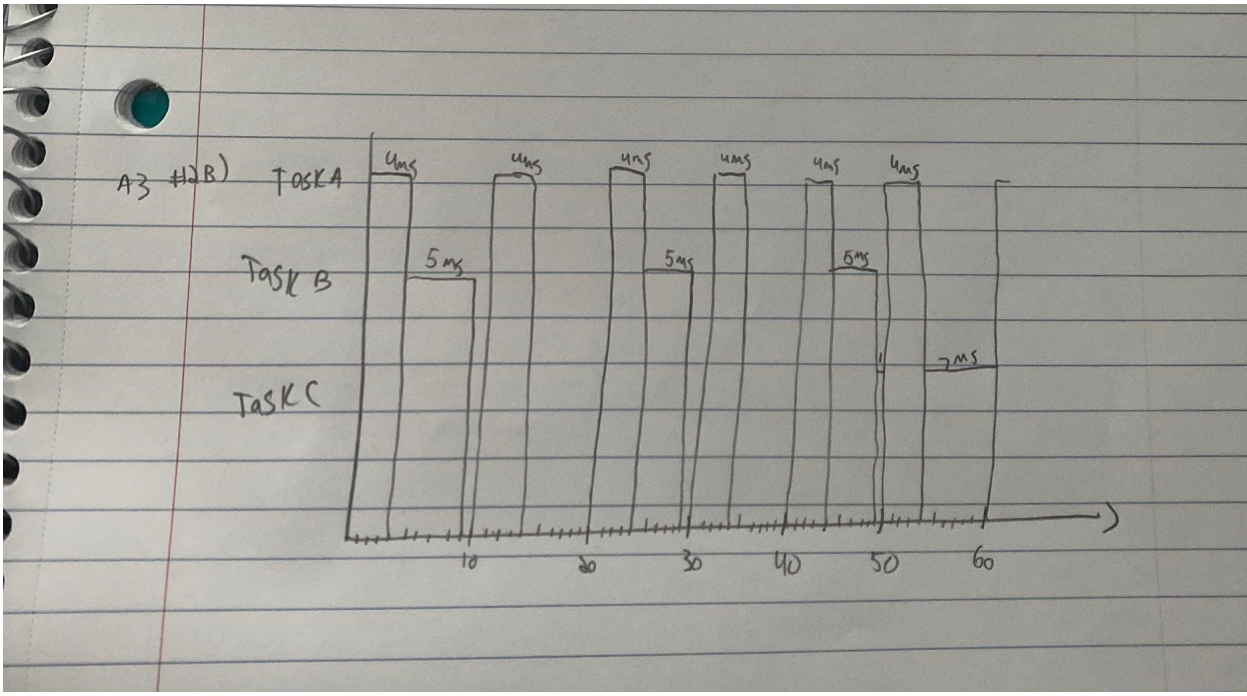
Task B response time = 5 ms + (0 /10)* 4 = 5ms
5 ms + (5/ 10) * 4 = 5 + 1*4 = 9ms
5 ms + (9/10) * 4 = 5 + (1*4) = **9ms Converged**

Task C response time = 10ms + (9/20) * 5 = 10 + (1*5) = 15ms
10ms + (10/10) * 4 + (10/20) * 5 = 10 + 4 + 5 = 19ms
10ms + (19/10)*4 + (19/20) * 5 = 10 + 8 + 5 = 23 ms
10ms + (23/10)*4 + (23/20)*5 = 10 + 12 + 10 = 32 ms
10 ms + (32/10)*4 + (32/20)*5 = 10 + 16 + 10 = 36ms
10ms + (36/10) *4 + (36/20)*5 = 10 + 16 + 10 = **36ms Converged**

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

Task A: 4 ms < 10 ms Deadline beat by 6 ms
Task B: 9ms < 20 ms Deadline beat by 11 ms
Task C: 36 ms < 40 ms Deadline beat by 4 ms

iv. Draw an execution time line for this system.


A3 #2B)
TASK A
TASK B
TASK C
4ms
4ms
4ms
4ms
4ms
4ms
5ms
5ms
5ms
7ms
10
20
30
40
50
60