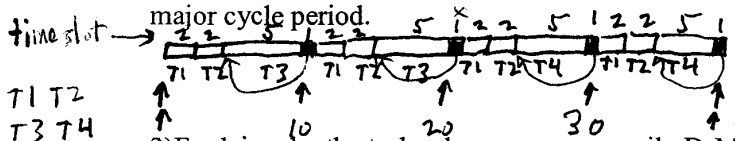
**Real-Time Systems ECSE 421**
**Assignment 10: exam** (answers)

1)The following tasks are scheduled by the ukernel scheduler task (units are clock ticks):

| Task | execution time | period | deadline |
|---|---|---|---|
| T1 | 2 | 10 | 10 |
| T2 | 2 | 10 | 10 |
| T3 | 10 | 40 | 40 |
| T4 | 10 | 40 | 40. |

T3 and T4 are delayed by one time unit every time there is a context switch from T1 or T2. Is it possible to make this set of tasks R-M schedulable? Show a Gantt chart over 1 major cycle period.




yes, just barely

2)Explain why the tasks above are necessarily D-M schedulable but are not necessarily R-M schedulable.

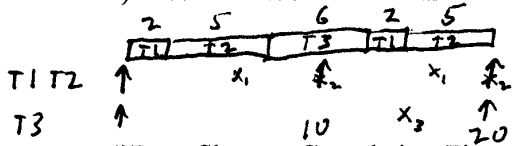
$U_p = 2/10 + 2/10 + \frac{10+2}{40} + \frac{10+2}{40} = 1$

$4(2^{1/4} - 1) = .76 < U_p$ => R-M scheduling will not always work, but could sometimes.

The following tasks are scheduled by the ukernel scheduler task (units are clock ticks) for questions 3-5; show the correct Gantt chart over 1 major cycle period in each question.

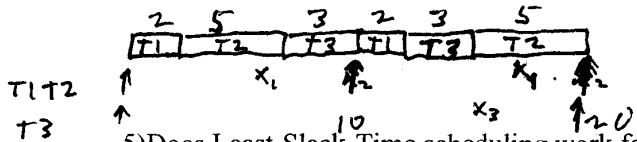
| Task | execution time | period | deadline |
|---|---|---|---|
| T1 | 2 | 10 | 6 |
| T2 | 5 | 10 | 10 |
| T3 | 6 | 20 | 15. |

3)Does Earliest-Deadline-First scheduling work for these 3 tasks?



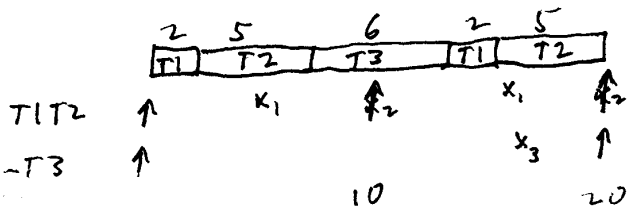

yes, T2 just barely makes its deadline

4)Does Shortest-Completion-Time scheduling work for these 3 tasks?




yes, T3 just barely makes its deadline along with T2.

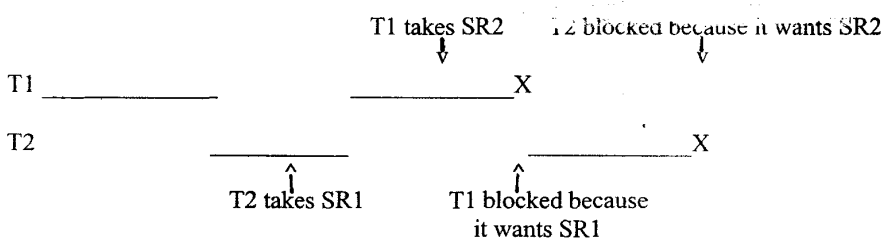
5)Does Least-Slack-Time scheduling work for these 3 tasks?




yes, same as EDF diagram.

**Real-Time Systems ECSE 421**
**Assignment 10: exam** (answers)

For Questions 6 and 7, priorites for tasks T1 and T2 are $P_{T1} > P_{T2}$ for the following timing diagram: T1 and T2 make use of shared resources SR1 and SR2.




6)Explain why Priority Ceiling Protocol solves the above situation whereas Priority Inheritance Protocol does not. Priority inheritance protocol only adjusts task priorities based on what resources are owned by each task. But Priority ceiling protocol will remove the deadlock when it assigns higher priority to T2 (because of the resource SR1 that it owns that T1 wants) and suspends T1 thereby releasing SR2.

7)If Priority Ceiling Protocol is applied in question 6, does priority inversion still occur? Why? yes, priority inversion still occurs because T2 is running while T1 (the highest priority task, desired by the embedded system) is suspended at the point T1 wants SR1.

The following tasks are scheduled by the ukernel scheduler task (units are clock ticks) for questions 8~~9, show the correct Gantt chart over 1 major cycle period in each question~~.

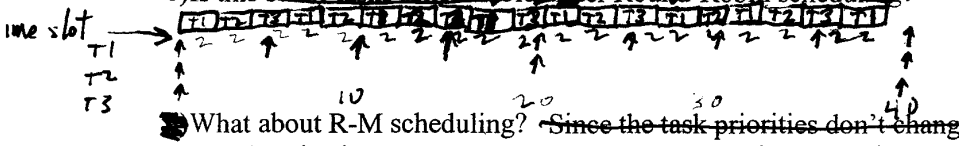
| Task | execution time | period | deadline | priority |
|---|---|---|---|---|
| T1 | 2 | 5 | 5 | 1 |
| T2 | 6 | 20 | 20 | 2 |
| T3 | 10 | 40 | 40 | 3. |

T3's response time

| | |
|---|---|
| R-M | 38 |
| R-R | 36 |

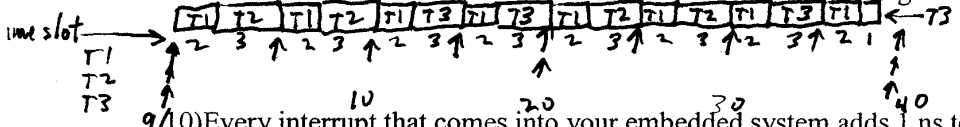
What is T3's response time under R-R scheduling?

8)~~Is this set of tasks schedulable under Round Robin scheduling?~~




yes, it works with 2 spare units at the end of the cycle period

What about R-M scheduling? ~~Since the task priorities don't change, show why the Gantt chart is different from the case of Round Robin scheduling?~~




yes it works. Its different because R-M scheduling is pre-emptive, round-robin is not.

9/10)Every interrupt that comes into your embedded system adds 1 ns to the normal execution time of a task. If two *preemptable* interrupts are being used, one which is occurring 1% of the time and the other 2% of the time, what is the task's running time (t') in the presence of these interrupts in terms of its normal running time (t) when no interrupts are present? If the two interrupts were not preemptable and masked each other instead, show how the answer would be different.

pmptive case: $t'(s) = t(s) + \sum_I (t(I) * f_I) * t'(s) = t_s + (.01*1 + .02*1)\,t'(s) \Rightarrow \boxed{t'(s) = \frac{t(s)}{.97}}$

asked case: $t'(s) = t(s) + [(.01 - .01*.02)*1 + (.02 - .02*.01)*1]\,t'(s) = t(s) + (.03 - .0004)*t'(s)$

$\Rightarrow \boxed{t'(s) = t(s)/.9704}$