Homework 5, due May 2nd 5pm CST

Handin at 1102 DCL. Slide under door if TA not present.

Important: Please type or *neatly* write your solutions. Anything we can't read will receive no credit. You must show work to receive full credit.

1. (20 points) There are 5 periodic tasks with the following computation times and periods.

   - τ_1: $C_1 = 3$, $T_1 = 70$
   - τ_2: $C_2 = 7$, $T_2 = 80$
   - τ_3: $C_3 = 18$, $T_3 = 95$
   - τ_4: $C_4 = 20$, $T_4 = 100$
   - τ_5: $C_5 = 30$, $T_5 = 160$

   In addition, assume the following are true.

   - Each context switch requires exactly 1 unit of time.
   - τ_3 can block all higher-priority tasks for a duration of 3.
   - τ_4 can block all higher-priority tasks for a duration of 5.
   - τ_4 has a pre-period deadline of 15.
   - τ_5 has a pre-period deadline of 18.

   Use the UB and Exact Schedulability tests to see if all tasks are schedulable.

   Solution:
   Here is a summary of the problem.

Task	C	T	S	B	D
τ_1	3	70	2	8	
τ_2	7	80	2	8	
τ_3	18	95	2	5	
τ_4	20	100	2		15
τ_5	30	160	2		18

Start with the UB test and apply iteratively.

Utilization of τ_1 equals $(3 + 2 \times 1 + 8) \div 70 = 0.186$ which is less than $UB(1)$. Thus τ_1 is schedulable.

Similarly, utilization of τ_1 and τ_2 equals $5 \div 70 + (7 + 2 + 8) \div 80 = 0.284$ which is less than $UB(2)$. Thus τ_1 and τ_2 are schedulable.

Utilization of τ_1, τ_2, and τ_3 equals $5 \div 70 + 9 \div 80 + (18 + 2 + 5) \div 95 = 0.448$ which is less than $UB(3)$. Thus τ_1, τ_2, and τ_3 are schedulable.

Utilization of τ_1, τ_2, τ_3, and τ_4 equals $5 \div 70 + 9 \div 80 + 20 \div 95 + (20 + 2 + 15) \div 100 = 0.764$, which is more than $UB(4)$. Need to apply the Exact Schedulability test. Use the iterative process described in class to apply the Exact Schedulability test. Take into account of the the context switching times.

$$a_0 = 5 + 9 + 20 + 22 = 56$$

$$a_1 = 22 + \left\lceil \frac{56}{70} \right\rceil 5 + \left\lceil \frac{56}{80} \right\rceil 9 + \left\lceil \frac{56}{95} \right\rceil 20 = 56 = a_0 < (T_4 - D_4)$$

We see that $a_0 = a_1$ and can stop. $a_1 < (T_4 - D_4)$; therefore, these 4 tasks is schedulable.

Finally, we check the utilization of all 5 tasks. By UB, it is equal to $5 \div 70 + 9 \div 80 + 20 \div 95 + 22 \div 100 + 50 \div 160 = 0.927$. Need to apply the Exact test.

$$a_0 = 5 + 9 + 20 + 22 + 32 = 88$$

$$a_1 = 32 + \left\lceil \frac{88}{70} \right\rceil 5 + \left\lceil \frac{88}{80} \right\rceil 9 + \left\lceil \frac{88}{95} \right\rceil 20 + \left\lceil \frac{88}{100} \right\rceil 22 = 102$$

$$a_2 = 32 + \left\lceil \frac{102}{70} \right\rceil 5 + \left\lceil \frac{102}{80} \right\rceil 9 + \left\lceil \frac{102}{95} \right\rceil 20 + \left\lceil \frac{102}{100} \right\rceil 22 = 144$$

$$a_3 = 32 + \left\lceil \frac{144}{70} \right\rceil 5 + \left\lceil \frac{144}{80} \right\rceil 9 + \left\lceil \frac{144}{95} \right\rceil 20 + \left\lceil \frac{144}{100} \right\rceil 22 = 149$$

$$a_4 = 32 + \left\lceil \frac{149}{70} \right\rceil 5 + \left\lceil \frac{149}{80} \right\rceil 9 + \left\lceil \frac{149}{95} \right\rceil 20 + \left\lceil \frac{149}{100} \right\rceil 22 = 149$$

We see that $a_4 = a_3$ and can stop; however, $a_4 > T_5 - D_5$. This means that the set of all 5 tasks is *not* schedulable. Alternatively, we could've just stopped at a_2 because that was already greater than $T_5 - D_5$.

2. (30 points) There are 3 periodic tasks, τ_1, τ_2, and τ_3 and 2 shared data structures among the tasks. First, τ_1 and τ_2 share DS_1 where both τ_1's and τ_2's critical sections are 5 (units of time). Second, τ_2 and τ_3 share DS_2 where τ_2's critical section is 7 and τ_3's critical section is 10.

   (a) Assuming that semaphores *cannot* be nested, what are the worst case blocking times of each task if the Basic Priority Inheritance protocol is used for synchronization? (5 points)

   Solution:

   τ_3 cannot be blocked since it has the lowest priority. τ_1 can be blocked by τ_2 on DS_1 for 5. τ_2 can be blocked by τ_3 on DS_2 for 10.

Task	Blocked Time
τ_1	5
τ_2	10
τ_3	0

   (b) Assuming that semaphores *can* be nested, what are the worst case blocking times of each task if the Basic Priority Inheritance protocol is used for synchronization? (10 points)

   Solution:

   τ_3 cannot be blocked since it has the lowest priority. τ_2 can be blocked by τ_3 on DS_2 for 10.

   As for τ_1, the worst case would be the following sequence of events:

   i. τ_3 locks on DS_2.

   ii. τ_2 locks on DS_1 and tries to lock on DS_2 but is blocked. τ_3 inherits τ_2's priority.

   iii. τ_1 tries to lock on DS_1 but is blocked. τ_2 inherits τ_1's priority, because τ_1 is the highest priority task that τ_2 is currently blocking. In addition, τ_3 now inherits the new τ_2 priority, which is the same as τ_1. In other words, all 3 tasks have the same priority.

   iv. τ_3 finishes its CS for DS_2 in 10 and returns to normal priority.

   v. τ_2 can now lock on DS_2 and finish its CS in 7. Afterwards, it unlocks DS_2 and finish its CS for DS_1 in 5.

   This entire sequence of events will block τ_1 for $10 + 7 + 5 = 22$.

Task	Blocked Time
τ_1	22
τ_2	10
τ_3	0

(c) Assuming that semaphores *cannot* be nested, what are the worst case blocking times of each task if the Priority-Ceiling Protocol is used for synchronization? (5 points)

Solution:
Same as (a).

(d) Assuming that semaphores *can* be nested, what are the worst case blocking times of each task if the Priority-Ceiling Protocol is used for synchronization? (10 points)

Solution:
τ_3 cannot be blocked since it has the lowest priority. τ_2 can be blocked by τ_3 on DS_2 for 10. PCP has the property that, in the worst case, a high-priority task can be blocked atmost once by a low-priority task. Therefore, τ_1 has a worst case blocking time of $5 + 7$. The exact sequence of events that will allow this is as below.

i. τ_2 locks on DS_1.
ii. τ_1 tries to lock on DS_1 but cannot because DS_1 has a priority ceiling of τ_1 which τ_2 currently owns.
iii. τ_2 locks on DS_2 successfully, because no other tasks currently own any semaphores.
iv. τ_2 finishes its CS for DS_2 and DS_1.

Task	Blocked Time
τ_1	12
τ_2	10
τ_3	0

3. (30 points) There are three periodic tasks with the following computation times (C), periods (T), blocking times (B), and pre-period deadlines (D). The context switching time is $S = \frac{1}{2}$.

Task	C	T	B	D	S
τ_1	5	35	5	14	1
τ_2	20	90	10	28	1
τ_3	15	120	0	35	1

(a) (10 points) Use the Utilization Bound and Exact Schedulability tests as needed to show that the task set is schedulable. Show your work.

Solution:

- Use the UB test on the first task:

$$\frac{5+1+5+14}{35} \approx 0.7143 \le U(1) = 1$$

$\Rightarrow$ The first task is schedulable.

- Use the UB test on the first two tasks:

$$\frac{5+1}{35} + \frac{20+1+10+28}{90} \approx 0.8270 \le U(2) \approx 0.8284$$

$\Rightarrow$ The first two tasks are schedulable.

- Use the UB test on all three tasks:

$$\frac{5+1}{35} + \frac{20+1}{90} + \frac{15+1+35}{120} \approx 0.8298 > U(3) \approx 0.7797$$

$\Rightarrow$ The UB test is inconclusive. Use the exact test on all three tasks:

$$\begin{aligned} a_0 &= 6 + 21 + 16 = 43 \\ a_1 &= 16 + \left\lceil \frac{43}{35} \right\rceil 6 + \left\lceil \frac{43}{90} \right\rceil 21 = 49 \\ a_2 &= 16 + \left\lceil \frac{49}{35} \right\rceil 6 + \left\lceil \frac{49}{90} \right\rceil 21 = 49 \end{aligned}$$

$$a_1 = a_2 = 49 \le (120 - 35) = 85$$

$\Rightarrow$ All three tasks are schedulable.

(b) (20 points) A fourth aperiodic server will now be added to the task set to maximally utilize the leftover CPU cycles. The new server τ_a has computation time C_a, period $T_a = 34$, blocking time $B_a = 0$, and pre-period deadline $D_a = 0$. What is the largest integer value for C_a that will allow all four tasks to meet their deadlines? Hint: you might want to write a simple program to calculate this. Do not include your program in the solutions, but you must show that your solution is indeed the largest integer possible, i.e., (your answer + 1) does not work.

Task	C	T	B	D	S
τ_a	C_a	34	0	0	1
τ_1	5	35	5	14	1
τ_2	20	90	10	28	1
τ_3	15	120	0	35	1

Solution:

$C_a = 8$. Use the exact test on all four tasks:

$$
\begin{aligned}
a_0 &= 9 + 6 + 21 + 16 = 52 \\
a_1 &= 16 + \left\lceil \frac{52}{34} \right\rceil 9 + \left\lceil \frac{52}{35} \right\rceil 6 + \left\lceil \frac{52}{90} \right\rceil 21 = 67 \\
a_2 &= 16 + \left\lceil \frac{67}{34} \right\rceil 9 + \left\lceil \frac{67}{35} \right\rceil 6 + \left\lceil \frac{67}{90} \right\rceil 21 = 67
\end{aligned}
$$

$$a_1 = a_2 = 67 \leq (120 - 35) = 85$$

$\Rightarrow$ All four tasks are schedulable. Use the exact test on the first three tasks:

$$
\begin{aligned}
a_0 &= 9 + 6 + 21 = 36 \\
a_1 &= 21 + \left\lceil \frac{36}{34} \right\rceil 9 + \left\lceil \frac{36}{35} \right\rceil 6 = 51 \\
a_2 &= 21 + \left\lceil \frac{51}{34} \right\rceil 9 + \left\lceil \frac{51}{35} \right\rceil 6 = 51
\end{aligned}
$$

$$a_1 = a_2 = 51 \leq (90 - 10 - 28) = 52$$

$\Rightarrow$ The first three tasks are schedulable. Use the exact test on the first two tasks:

$$
\begin{aligned}
a_0 &= 9 + 6 = 15 \\
a_1 &= 6 + \left\lceil \frac{15}{34} \right\rceil 9 = 15
\end{aligned}
$$

$$a_0 = a_1 = 15 \leq (35 - 5 - 14) = 16$$

$\Rightarrow$ The first two tasks are schedulable. The first task cannot be blocked, has no pre-period deadline, and $9 \leq 34$ so it is also schedulable.

We must now show that the task set is not schedulable when $C_a = 9$. Use the exact test on all four tasks:

$$
\begin{aligned}
a_0 &= 10 + 6 + 21 + 16 = 53 \\
a_1 &= 16 + \left\lceil \frac{53}{34} \right\rceil 10 + \left\lceil \frac{53}{35} \right\rceil 6 + \left\lceil \frac{53}{90} \right\rceil 21 = 69 \\
a_2 &= 16 + \left\lceil \frac{69}{34} \right\rceil 10 + \left\lceil \frac{69}{35} \right\rceil 6 + \left\lceil \frac{69}{90} \right\rceil 21 = 79 \\
a_3 &= 16 + \left\lceil \frac{79}{34} \right\rceil 10 + \left\lceil \frac{79}{35} \right\rceil 6 + \left\lceil \frac{79}{90} \right\rceil 21 = 85 \\
a_4 &= 16 + \left\lceil \frac{85}{34} \right\rceil 10 + \left\lceil \frac{85}{35} \right\rceil 6 + \left\lceil \frac{85}{90} \right\rceil 21 = 85
\end{aligned}
$$

$$a_3 = a_4 = 85 \leq (120 - 35) = 85$$

$\Rightarrow$ All four tasks are schedulable. Use the exact test on test on the first three tasks:

$$
\begin{aligned}
a_0 &= 10 + 6 + 21 = 37 \\
a_1 &= 21 + \left\lceil \frac{37}{34} \right\rceil 10 + \left\lceil \frac{37}{35} \right\rceil 6 = 53
\end{aligned}
$$

$$a_1 = 53 > (90 - 10 - 28) = 52$$

$\Rightarrow$ The first three tasks are not schedulable.

4. (20 points) There are three stations S_1 through S_3 connected to a 1 Mbit/sec FDDI ring.

   Station S_1 transmits periodic sensor data stream to S_3: $\{(C_{13} = 10, T_{13} = 90)\}$.

   Station S_2 is attached to a 10 frames-per-second video camera that captures 256 by 128 pixel frames at 2 bits per pixel. It transmits them in an uncompressed format to station S_3.

   Station S_3 transmits two periodic sensor data streams. One stream goes to S_1: $\{(C_{31} = 5,\ T_{31} = 50)\}$. The other stream goes to S_2: $\{(C_{32} = 40,\ T_{32} = 300)\}$.

   S_1 holds the token for a maximum of $H_1 = 10$ msec. S_2 holds the token for a maximum of $H_2 = 15$ msec. S_3 holds the token for a maximum of $H_3 = 20$ msec. The walk time is $W = 25$ msec.

   For each of the three stations, show the equivalent periodic task set (include each task's computation time and period). You do **not** need to perform any schedulability analysis on the tasks.

   Solution:
   $TTRT = 10 + 15 + 20 + 25 = 70$ msec.

   Each station has a task τ_{ttrt} who's period is $TTRT$ and computation time is $TTRT$ minus the holding time of the station.

   The camera's period is $\frac{1}{10} = 100$ msec. Its computation time is $\frac{256\times128\times2}{2^{20}} = 62.5$ msec.

Station S_1

Task	C	T
τ_{ttrt}	60	70
τ_{13}	10	90

Station S_2

Task	C	T
τ_{ttrt}	55	70
τ_{camera}	62.5	100

Station S_3

Task	C	T
τ_{ttrt}	50	70
τ_{31}	5	50
τ_{32}	40	300