CAS CS 350 HW8

Andrea Lopez

TOTAL POINTS

86 / 100

QUESTION 1

1 Q1 **30 / 30**

part a)

✓ - **0 pts** correct

- **8 pts** incomplete/missing
- **5 pts** came to conclusion by only testing the formula
- **4 pts** incorrect conclusion with work shown
- **4 pts** drew graph incorrectly

part b)

✓ - **0 pts** correct

- **10 pts** incomplete/missing
- **5 pts** incorrect with work shown
- **3 pts** minor errors in graph
- **5 pts** no graph

part c)

✓ - **0 pts** correct

- **7 pts** incomplete/missing
- **3.5 pts** incorrect with work shown
- **2 pts** minor errors in graph
- **3.5 pts** no graph

part d)

✓ - **0 pts** correct

- **5 pts** incomplete/missing
- **2.5 pts** attempted unsuccessfully to find modification that makes the system schedulable

QUESTION 2

2 Q2 **40 / 40**

✓ - **0 pts** Correct

- **40 pts** Incorrect

part a)

- **5 pts** Incorrect
- **4 pts** No reasoning

part b)

- **5 pts** Incorrect
- **4 pts** no reasoning

part c)

- **7 pts** Incorrect
- **3.5 pts** Partially correct
- **1 pts** miss for part a

part d)

- **8 pts** Incorrect

part e)

- **5 pts** Incorrect

part f)

- **2 pts** Mostly correct
- **5 pts** Partially correct
- **8 pts** Mostly incorrect
- **10 pts** Incorrect

QUESTION 3

3 Q3 **16 / 30**

- **0 pts** Correct

part a)

- **2 pts** One line incorrect
- **10 pts** Incorrect

part b)

✓ - **8 pts** **Incorrect or no explanation**

- **12 pts** Incorrect

part c)

- **2 pts** One line incorrect
- **4 pts** Mostly incorrect

✓ - **6 pts** **Incorrect or no conclusion**

- **8 pts** Incorrect

Problem 1

a) taskset is schedulable under RM if
$U \leq 0.69$ & $m(2^{\frac{1}{m}} - 1)$

① find utilization $\left(U = \sum_{i=1}^{m} \frac{C_i}{T_i}\right)$

$U = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{C_3}$) plug in!

$= \frac{5}{10} + \frac{2}{12} + \frac{5}{19}$

$= 0.5 + 0.167 + 0.263$

$\underline{U = 0.93}$

→ not ≤ 0.69!

② solve other side of equation: $m(2^{1/m} - 1)$

$m(2^{1/m} - 1)$) plug in!

$3(2^{1/3} - 1)$

$3(1.260 - 1)$

$3(0.260)$

$\underline{0.78}$

③ plug in to equation & compare!

$U \leq m(2^{\frac{1}{m}} - 1)$) plug in!

$0.93 \leq 0.78$

This is NOT true, so we have to graph to see if it is schedulable.

↳ see graph!

- once we see the graph, you can see that this taskset is schedulable under RM.

Problem 1 continued.

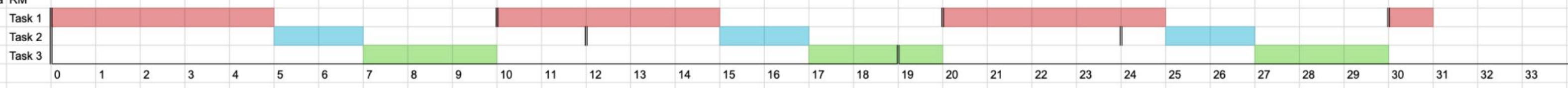
b) look at graph!
↳ NOT schedulable (deadline miss)

c) look at graph!

d) If you break up the critical section into 2, it'll have 2 critical sections so processes can be run in between.

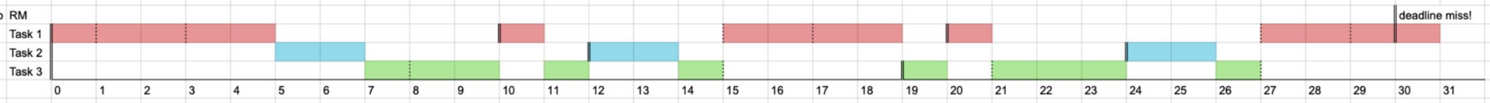
Problem 1)

1a RM



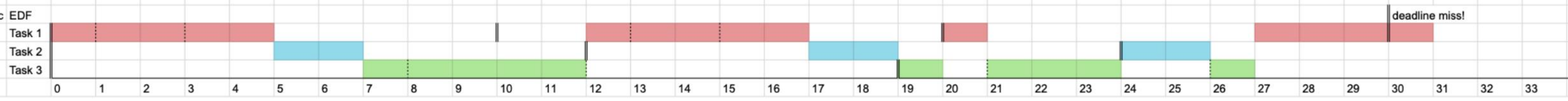
Task ID	Period	WCET	Util.
1	10	5	0.5
2	12	2	0.16666
3	19	5	0.26315
		∑Util.	0.92982

1b RM



Task ID	Period	Rem. 1	Critical	Rem. 2	WCET
1	10	2	2	1	5
2	12	-	-	-	2
3	19	0.5	4.5	0	5

1c EDF



1 Q1 **30 / 30**

part a)

✓ **- 0 pts** correct

- 8 pts incomplete/missing

- 5 pts came to conclusion by only testing the formula

- 4 pts incorrect conclusion with work shown

- 4 pts drew graph incorrectly

part b)

✓ **- 0 pts** correct

- 10 pts incomplete/missing

- 5 pts incorrect with work shown

- 3 pts minor errors in graph

- 5 pts no graph

part c)

✓ **- 0 pts** correct

- 7 pts incomplete/missing

- 3.5 pts incorrect with work shown

- 2 pts minor errors in graph

- 3.5 pts no graph

part d)

✓ **- 0 pts** correct

- 5 pts incomplete/missing

- 2.5 pts attempted unsuccessfully to find modification that makes the system schedulable

Problem 2.

a) NO, if there are 2 roomies in the house, then it is NOT possible that the roomies end up starving. This is because there are at least 2 of each type of utensil (chop-pairs, forks, spoons). This, combined with the fact that the second dinner will never be the same as the first, ensures that each roomie always has access to a utensil for whichever meal they eat. They also have enough plates & glasses to eat.

b) No, if there are 4 roomies in the house, then there is still no deadlock that occurs. Since they only have 3 glasses, 1 waits and 3 of them will be able to do the first die roll (1st meal). From there, assuming the worst case, there is still at least 1 roommate who is able to roll for the second meal, acquire the utensils needed for both meals, and eat the meal. In other words, there is still progress for at least 1 roomie. Once that roomie is done eating, they will release the utensils they had & the other roomies will be able to make progress once again. therefore, there is NO deadlock!

example:

Roomies
1 = 1st = c, 2nd = f (wait)
2 = 1st = c, 2nd = f (wait)
3 = 1st = f, 2nd = c (wait)
4 = 1st = f, 2nd = c (wait)

spoon = 2
fork = 2
chopsticks = 2

c) Part A:
↳ same answer!

Part B:
↳ answer changes!

Since there are now 4 glasses, then all 4 roomies would be able to roll the die for the first meal choice (we can assume that the first 2 roomies picked sushi & the last 2 picked pasta for the 1st meal). If in the second choice the first 2 roomies pick pasta & the last 2 pick sushi, then there would be a deadlock, since none of the roomies would be able to make any progress (since they'd be waiting for the utensils necessary for their second meal forever.

Problem 2 continued...

d) We know a plate is dirty if a roomie is done eating. In other words, if they're done eating then that means they were able to pick 2 meals & get the utensils they needed for both. Therefore, it would be safe to say that in order to find the maximum number of dirty plates, then its the same as finding the maximum number of friends that can eat at the same time.

ex:

	m1	m2
1	f	s
2	s	c
3	c	f
4		

} 3 roomies can eat at the same time

Therefore, since when these 3 roomies finish eating they have to wash their plate, the maximum possible number of dirty plates is $\boxed{3}$.

e) We can make the roomies eat the first meal first and wash the utensils that they used and then move on to picking the second meal to be able to eat. Also have to change the semaphore for plates

Semaphore plates := 6

Starting on line 38 (after first choice is made)

```
eat_1st_course();      // eat!

if (choice_1st_course ==PASTA){     // wash utensils
    signal (forks); }               // to make them
if (choice_1st_course == SUSHI){    // available
    signal (chop_pair); }
if (choice_1st_course == SOUP){
    signal (spoons);}
```

e) lines 60 - 73 :
↳ remove the "first part of the "or"
(choice_1st_course == PASTA , etc.)

<u>should look like:</u>

```
if(choice_2nd_course == PASTA){
    signal(forks); }
if (choice_2nd_course == SUSHI){
    signal(chop_pair);}
if(choice_2nd_course ==SOUP){
    signal(spoon);}
```

f)

		Parameter	Resources				
			plates	glasses	forks	spoons	pair_chops
		R(K)	4	6	2	2	2
Roomies	ro1	C1(K)	1	1	1	1	1
	ro2	C2(K)	1	1	1	1	1
	ro3	C3(K)	1	1	1	1	1
	ro4	C4(K)	1	1	1	1	1

2 Q2 **40 / 40**

✓ **- 0 pts** Correct

- 40 pts Incorrect

part a)

- 5 pts Incorrect

- 4 pts No reasoning

part b)

- 5 pts Incorrect

- 4 pts no reasoning

part c)

- 7 pts Incorrect

- 3.5 pts Partially correct

- 1 pts miss for part a

part d)

- 8 pts Incorrect

part e)

- 5 pts Incorrect

part f)

- 2 pts Mostly correct

- 5 pts Partially correct

- 8 pts Mostly incorrect

- 10 pts Incorrect

Problem 3

a)

Parameter	Resources			Parameter	Resources		
	R1	R2	R3	Parameter	R1	R2	R3
V(K)	3	3	2				
$A_1(K)$	0	1	0	$N_1(K)$	7	4	3
$A_2(K)$	2	0	0	$N_2(K)$	1	2	2
$A_3(K)$	3	0	2	$N_3(K)$	6	0	0
$A_4(K)$	2	1	1	$N_4(K)$	0	1	1
$A_5(K)$	0	0	2	$N_5(K)$	4	3	1

b) Yes, the state reported in the table is safe!
order of completion:
1. P_2
2. P_4
3. P_5
4. P_1
5. P_3

c)

			Resources				Resources		
		Parameter	R1	R2	R3	Parameter	R1	R2	R3
		$V(K)$	0	0	2				
Processes	P_1	$A_1(K)$	0	1	0	$N_1(K)$	7	4	3
	P_2	$A_2(K)$	2	0	0	$N_2(K)$	1	2	2
	P_3	$A_3(K)$	3	0	2	$N_3(K)$	6	0	0
	P_4	$A_4(K)$	2	1	1	$N_4(K)$	0	1	1
	P_5	$A_5(K)$	3	3	2	$N_5(K)$	1	0	1

3 Q3 **16 / 30**

- **0 pts** Correct

part a)

- **2 pts** One line incorrect
- **10 pts** Incorrect

part b)

✓ **- 8 pts Incorrect or no explanation**

- **12 pts** Incorrect

part c)

- **2 pts** One line incorrect
- **4 pts** Mostly incorrect

✓ **- 6 pts Incorrect or no conclusion**

- **8 pts** Incorrect