CS376 W14 Assignment - Deadlocks

Directions: Edit this document to answer the following questions. If you need to draw a diagram or handwrite something, be sure to include it in this document also.

1. *(10 points)* Complete your SPTQ for this class. Submit a screenshot of your SPTQ to do list showing your name, and your list showing that the SPTQ for CS376 is completed.

Fillout Task List
Task Owner: Dalton Rothenberger
Project Title: Spring 2020 SPTQ Student Perceptions Teaching Questionnaire (QP)
Category: SPTQ
Subcategory: Spring 2020

Subject	Due date	Status
CS-376-A - Operating System	Wednesday, May 6, 2020	Completed

2. *(15 points)* Recall that hold and wait is one of the four necessary conditions for deadlock to occur. Give a scenario in which this would lead to under-utilization of resources. How might this lead to a process possibly never getting to run (ie starve)?

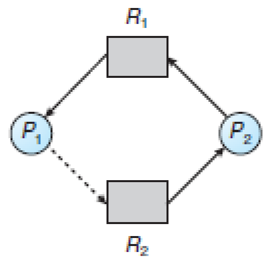
   Hold and wait occurs when a process is holding one resource and waiting for another. This is inherently a under utilization of resources. The process that is holding one resource is not using it because it is waiting for the other to be freed up but while the process is waiting the resource it is holding could have been used. This might lead to a process never getting to run if the resource it is waiting for is never freed or very popular.

3. *(10 points)* One way of assuring hold and wait does not occur is for a thread to wait to acquire its resources until all resources are available. How does a pre-emptive scheduling system cause problems with this scheme?

   A pre-emptive scheduling system causes problems with this because a process could be waiting for its resources and then when it finally gets its resources it gets preempted by another process and has to go back to waiting till all the resources are available.

4. *(15 points)* What does it mean for a system to be in an unsafe state? Is deadlock guaranteed to occur in an unsafe state? Justify your answer by showing a resource diagram (including resources and threads) in an unsafe state and show why a deadlock won't necessarily occur.

   An unsafe state means that deadlock may occur but it is not guaranteed to occur.



   As we can see from above, we have an unsafe state. There is a possibility of deadlock if P1 claims R2 while the rest stands. If P2 were to wait until some point R1 is relinquished from P1, then we do not have a cycle anymore so then we can grant R2 to P1 so the deadlock is avoided.

Use the banker's algorithm and the following system state:

	Allocation	*Max*	*Available*
	ABCD	*ABCD*	*ABCD*
T_0	0012	0012	1520
T_1	1000	1750	
T_2	1354	2356	
T_3	0632	0652	
T_4	0014	0656	

5. *(10 points)* What would the Need matrix look like?
6. *(15 points)* Run through the safety algorithm and determine if this system is in a safe state. Show and explain each step of the algorithm

5)

	Need
	A B C D
T_0	0 0 0 0
T_1	0 7 5 0
T_2	1 0 0 2
T_3	0 0 2 0
T_4	0 6 4 2

6)

	Allocation	Need	Available
	A B C D	A B C D	A B C D
T_0	0 0 1 2	0 0 0 0	1 5 2 0
T_1	1 0 0 0	0 7 5 0	
T_2	1 3 5 4	1 0 0 2	
T_3	0 6 3 2	0 0 2 0	
T_4	0 0 1 4	0 6 4 2	

Initial State

	A	B	C	D
Work	1	5	2	0

	0	1	2	3	4
Finish	F	F	F	F	F

Index = 0 $\quad Finish_0 == false, \; Need_0 \le Work$

	A	B	C	D
Work	1	5	3	2

	0	1	2	3	4
Finish	T	F	F	F	F

Index = 1 $\quad$ Need > work so can't do it right now

Index = 2 $\quad Finish_2 == false, \; Need_2 \le Work$

	A	B	C	D
Work	2	8	8	6

	0	1	2	3	4
Finish	T	F	T	F	F

Index = 3 $Finish_3$ == false $Need_3 \le work$

	A	B	C	D
work	2	14	11	8

	0	1	2	3	4
Finish	T	F	T	T	F

Index = 4 $Finish_4$ == false $Need_4 \le work$

	A	B	C	D
work	2	14	12	12

	0	1	2	3	4
Finish	T	F	T	T	T

Index = 1 $Finish_1$ == false $Need_1 \le work$

	A	B	C	D
work	3	14	12	12

	0	1	2	3	4
Finish	T	T	T	T	T

The system is in a safe state since Finish[i] == true for all i.

7. *(10 points)* If a request from T4 for (2,1,1,4) arrives while the system is in this state, could it immediately be satisfied? Justify your answer.

   The request could not be immediately satisfied as we can see the currently available resources are (1, 5, 2, 0) so we do not have enough of A or D available to satisfy the request from T4.

8. *(15 points)* Change this system so that it is unsafe. Show it is unsafe by running through the safety algorithm.

8) If we changed the max of T_1 to (1, 99, 5, 0) then the system would be unsafe.
The need of T_1 would be (0, 99, 5, 0)

The safety algorithm would proceed as it did last time where T_0 ran, T_1 was skipped, T_2 ran, T_3 ran, and T_4 ran. However, T_1 would still not run after T_4. The state after T_4 runs is,

	A	B	C	D
Work	2	14	12	12

	0	1	2	3	4
Finish	T	F	T	T	T

We seed that $Need_1 > Work$ since (0, 99, 5, 0) means there is not enough B available. So we have Finish[1] != true so the system is not in a safe state