Midterm Examination

COIS 3320H – Fundamentals of Operating Systems

Version 3

Due By: March 5th – 12:00PM (Noon) EST

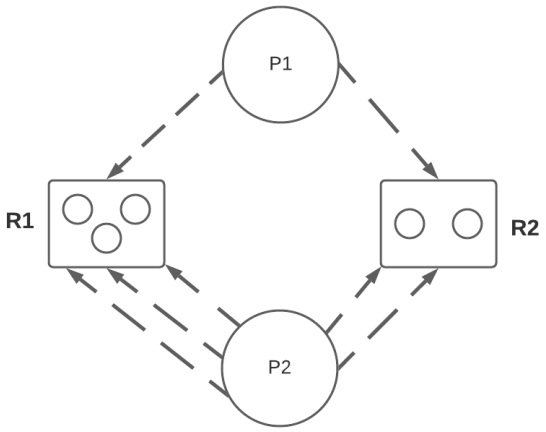
Instructions:

1- Please submit your answers as a single PDF titled firstname_lastname_midterm on Blackboard.
   a. Make sure you include your name and student number at the very top of your submitted PDF.
   b. The answers to the questions must be labelled and in order (e.g., answer Q1 first followed by Q2).
   c. You must show all your steps and all the formulas (note the purpose of your formula when working). Showing the final answer with no steps will earn you zero points!
   d. For the C programming question do not submit the .c file, simply program in your favorite text editor, test your code on Loki (screenshot the test) and copy and paste the code into a word processor along with the screenshot and save it as a PDF.
2- Your answers must be typed not handwritten and scanned. Handwritten midterms will not be accepted.
3- The deadline for the midterm is 12:00PM (noon) EST.
   a. Every minute you are late past the deadline constitutes a 2-point penalty.
4- Make sure you monitor the COIS 3320 MS Teams general chat.
   a. All your questions must be asked on MS Teams, please do not email me for your own sake!
   b. Any extensions will be announced on the MS Teams general chat.
5- This exam is open book. You can access all your notes and anything on the internet. However, you are not allowed to communicate with your classmates for help on the exam, and you are not allowed to communicate with any third party.

1- Resource Claim Graph

Given the following resource claim graph:

a- Show a sequence of operations leading from the given state to a deadlocked state. (1 point)
b- Show how the Banker's Algorithm would have prevented the deadlock. (2 points)



2- Banker's Algorithm

Given the following available resources, current allocation, and maximum demand:

a- Verify whether or not the available array was calculated correctly. (0.5 point)
b- Calculate the need matrix. (0.5 point)
c- Show that the current state is safe (show a safe sequence of processes). (2 points)
    a. Remember to show your steps.
d- Say a request (10,7,9,5) arrives from process 5, should this request be granted? Why? (1 point)
    a. Remember to show your steps.

Available Resources			
A	B	C	D
4	3	5	3

Current Allocation				
Process/Resource	A	B	C	D
P0	4	0	3	1
P1	0	2	1	5
P2	4	1	0	3
P3	1	0	0	4
P4	1	1	0	0
P5	1	0	1	1
Maximum Demand				
Process/Resource	A	B	C	D
P0	6	5	4	3
P1	2	2	3	8
P2	7	5	4	4
P3	3	4	2	8
P4	4	2	2	0
P5	4	4	2	3

3- Imagine a system that has n processes and a resource with m identical units. The maximum claim of any process is k.
   a. Why is the system deadlock free for all possible states of $m > n(k-1)$? (1 point)

4- Imagine a system that has 4 processes and a resource with 4 identical units, only two processes can claim resource units at the same time, and the maximum claim each process can make is 2 units.
   a. Is this system a deadlock free system? Why or why not? (1 point)

5- Given the six-state process activity model discussed in class (New, Ready, Running, Blocked, Suspended, Terminated), which of the following series of transitions are possible (assume you entered into a starting state from a valid series of transitions). For each series of transitions that is not possible explain why. (2.5 points)
   a. New→Ready→Running→Ready→Suspended→Ready→Terminated
   b. Ready→Running→Suspended→Blocked→Ready→Running
   c. Running→Blocked→Suspended→Ready→Running→Terminated
   d. Blocked→Ready→Running→Blocked→Suspended→Running→Terminated
   e. Suspended→Blocked→Ready→Suspended→Terminated

6- Consider the following set of processes, along with their burst times and arrival times:

Process	Total CPU Time	Arrival Time
P1	1	2
P2	3	1
P3	4	0
P4	3	10
P5	5	9

a. Complete the following Gantt chart illustrating the execution of these processes using FCFS, SRT, and RR (q=3). All the processes for each scheduling algorithm should be on the same chart. (3 points)

b. What is the Average Turnaround Time (ATT) for each of the scheduling algorithms in (a)? (3 points)

7- Write a program that accepts integers as command line arguments. The number of command line arguments accepted must be even. Say you input 6 command line arguments, the first three must go into array1, and the next three have to go into array2. Next, calculate a1[1]^a2[1], a1[2]^a2[2], … a1[n]^a2[n] and display the results. (5 points)