CS 3100: Operating Systems and Concurrency
Spring 2024
Take-home Final Exam

Total Marks: 53

Carefully Read the following Instructions at least two times:

- Write down your Name and A-number on top of the answer script.
- Clearly mark your answers, where **Ans-i** will be considered as the answer of **Q-i** (i = 1, 2, …, 10). An answer that is not clearly marked cannot be graded.
- Your answer should be typed in font size: 11.
- The answer script needs to be submitted on Canvas as a single pdf file. Do Not email your submission to Instructor/TA – it is not acceptable.
- Deadline for submission: **4:50 PM on Thursday, April 25**. If your submission is turned in after that time, even just 1 second after, will be considered late. **No late submission will be accepted for the Final Exam.** Do not email us with a request for late submission, it is not acceptable.
- No make-up exam will be offered.
- Individual submission is required from each student. Every student MUST work on their own submission.
- It is your responsibility to make sure your answer sheet is properly submitted via Canvas, before the deadline. **Double check** that you have submitted the correct answer sheet. No answer sheet will be accepted after the submission deadline.
- **Any sort of plagiarism/cheating will result in severe penalties**.

	Questions	Marks
Q-1.	Explain the following statement: “Semaphore can provide a more sophisticated way of process synchronization, as compared to Mutex Lock.”	4
Q-2.	Considering the security guarantee and performance overhead, state one prevention technique that you would choose (among the techniques discussed in lecture) for gaining protection against both direct and indirect buffer overflow attacks. Explain the rationale behind your choice.	1.5 + 4 = 5.5
Q-3.	Consider that there are three memory frames; according to LRU page replacement algorithm, how many page faults would occur for this reference string: 5, 1, 4, 3, 0, 2, 1, 3, 0, 5, 4, 0, 1, 0, 5, 2, 1, 2, 3, 1, 0, 5, 3? **Note**: You must show the steps how you calculated the page faults.	4
Q-4.	For passing parameters from a user program to the operating system during context switching, which method would you prefer most, and why?	1 + 2.5 = 3.5
Q-5.	Describe how the ‘mailbox’ is used in indirect communication between processes.	3
Q-6.	Explain the deadlock situation in the resource-allocation graph shown in Figure 1. Here, T1, T2, and T3 are threads – presented by circles; R1, R2, and R3 are resource types – presented by rectangles. Each resource type has multiple instances, where each instance is presented by a dot within the rectangle.	5

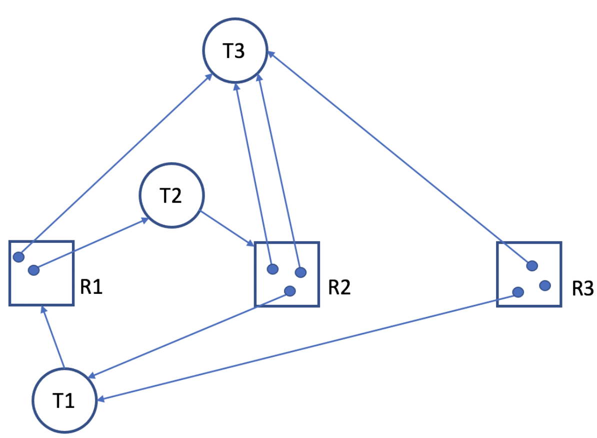

Figure 1: Resource-allocation Graph

Q-7. Which method would you prefer most for dynamic storage allocation when a system has to satisfy a request of memory size n from a list of available holes? Why would you prefer this method? — 0.5 + 2.5 = 3

Q-8. Consider five processes with given burst time: P1 (burst time: 24 units of time), P2 (burst time: 3 units of time), P3 (burst time: 10 units of time), P4 (burst time: 5 units of time), and P5 (burst time: 3 units of time). Here, Time Quantum is 4 units of time. Initially, processes are in the ready queue in this order: P1 (first process in the ready queue), P2, P3, P4, P5 (last process in the ready queue). The time for context switch is 1 unit of time. Answer the following questions based on Round-Robin scheduling algorithm: — 4 + 3 + 3 + 3 = 13

a. Draw a Gantt chart to demonstrate the order in which processes will be selected for scheduling.
b. What will be the waiting time for each of five processes?
c. What will be the turnaround time for each of five processes?
d. What will be the average turnaround time, if we add the time for context switch to the turnaround time?

Q-9. In which cases, does a process switch from its waiting state to the ready state and is put back in the ready queue? — 3

Q-10. Consider five threads: T0, T1, T2, T3, T4; and three resource types: A, B, C. Here, A has 12 instances, B has 8 instances, and C has 9 instances. The current state of the system is presented on Table 1, where 'Allocation' represents the number of instances for each resource type currently allocated to each thread, and 'Max' represents the maximum demand of each thread. Answer the following questions based on Banker's Algorithm: — 0.5 + 5.5 = 6

a. Is the system in a safe state now?
b. If the system is in a safe state, what is the corresponding safe sequence? It would suffice to present one safe sequence in your answer. **Note**: You must show the steps how you found the safe sequence.

Threads	Allocation A, B, C	Max A, B, C
T0	3, 1, 1	10, 1, 2
T1	2, 1, 1	3, 4, 4
T2	1, 2, 1	6, 4, 2

T3	1, 0, 2	4, 2, 4
T4	1, 1, 1	3, 4, 3

Table 1

Q-11. In the context of memory allocation, which method would you use to tackle the external fragmentation issue? Why can/can't you use the same method to address internal fragmentation? 1 + 2 = 3