CSCI 40300/ECE 40800
Operating Systems
Fall 2020
Midterm Exam

This is a take home exam. You have 24 hours to complete it. You may complete the answers by hand, and then upload a scanned file (preferably a single pdf file) to Canvas. If you wish, you may also type the answers in word or some other word processor and upload the file to Canvas. Please make sure that everything is in a single file, otherwise it will be unmanageable. And *do not submit multiple files put together in a zip archive.*

Please follow the instructions in the questions. Also it is very important that you show your work at each step along the way to the problems. This will allow me to assign partial credit if you get parts of the question wrong.

Name: _____

Question	Points	Score
1	4	
2	4	
3	4	
4	4	
5	4	
6	4	
7	4	
8	15	
9	12	
10	20	
11	5	
12	10	
13	5	
14	5	
Total:	100	

Name: _____

1. (4 points) You're hired by AB Computers to improve the performance of their system. They point out that their applications only use 10 of the CPU's 32 registers; so to improve the performance of the applications, they suggest that you change the OS's context switch routine so it only saves the 10 registers used by the applications. Assume that you can correctly change the context switch routine. Is this a good or bad idea? Why?

2. (4 points) Name *two ways* in which the processor can transition from user mode to kernel mode. Can the user execute arbitrary code after transitioning?

3. (4 points) Name *two* ways in which context-switching can happen in a *non-preemptive* scheduler.

4. (4 points) Explain the difference between response time and turnaround time. These times are both used to measure the effectiveness of scheduling schemes.

5. (4 points) Explain the process of starvation and how aging can be used to prevent it.

6. (4 points) What is priority inversion? Explain how a priority scheduler could be modified to avoid priority inversion.

7. (4 points) Consider the following code segment:

```
pid = fork();
if (pid == 0) { /* child process */
    fork();
}
fork();
```

How many unique processes are created at the end of this code snippet?

Process Scheduling

8. On a system using round-robin scheduling, let s represent the time needed to perform a process switch, q represent the round-robin time quantum, and r represent the average time a process runs before blocking on I/O. For each part, give a formula for CPU efficiency. CPU efficiency is defined as the ratio of time CPU does useful computation to the total amount of time that includes overhead.

   (a) (3 points) $q = \infty$

   (b) (3 points) $q > r$

   (c) (3 points) $s < q < r$

   (d) (3 points) $s = q < r$

   (e) (3 points) q nearly 0.

9. (a) (4 points) Explain what a *multi-level feedback queue* scheduler is. Give all the components that need to be specified to make it work.

   (b) (4 points) It has been claimed by some that a multi-level feedback queue scheduler approximates shortest-remaining-time-first (SRTF). Give an explanation why this could be true.

   (c) (4 points) Suggest a method by which one can fool the multi-level feedback scheduler's heuristics into giving a long-running task more CPU cycles.

10. Your boss asks you to pick a scheduling algorithm to run the following processes:

    - P1, which requires 1 hour of CPU time to complete;
    - P2, which requires 2 hours of CPU time to complete;
    - P3, which requires 1 min of CPU time to complete;

    Initially, assume all processes are ready at the same time.

    (a) (5 points) Assume you use First Come First Serve (FCFS) to schedule these processes. Which is the schedule with the largest average completion time (turnaround time)? Which is the schedule with the smallest average completion time (turnaround time)? Compute the average completion time (turnaround time) in each case.

    (b) (5 points) Assume you use Round-Robin with a time quanta (slice) of 100 ms. What is the completion time of each job in this case? Assume that there is no context switch overhead (i.e., it is negligible).

(c) (5 points) Now assume that P1 arrives at $t = 0$, P2 arrives at time $t = 10$ min, and P3 at time $t = 20$ min. What will be the completion time of each process when using the Shortest Remaining Time First (SRTF)?

(d) Assume now that P3 is performing an I/O operation every 10ms and that the I/O operation takes 40 ms, but would still take 1 minute to run if it was the only process running. Assume that P1, P2, and P3 are submitted at the same time and you use Round- Robin scheduling (like in part (b)).

i. (3 points) How long does it take P3 to complete if the time slice is 100 ms?

ii. (2 points) What time slice length would minimize the completion time of P3?

11. (5 points) Suppose your notion of fairness was to give every thread in a multithreaded system an equal portion of the CPU. Could you do this with *strictly user-level threads* (every process has only one kernel thread but multiple user-level threads)? Explain your answer.

12. Consider a system running ten I/O-bound tasks and one CPU-bound task. Assume that the I/O-bound tasks issue an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context-switching overhead is 0.1 millisecond and that all processes are long-running tasks.

    (a) (5 points) What is the CPU utilization for a round-robin scheduler when the time quantum is 1 millisecond?

    (b) (5 points) What is the CPU utilization for a round-robin scheduler when the time quantum is 10 millisecond?

Name: _____

Synchronization

13. (5 points) In lecture, we discussed using atomic instructions (e.g., test-and-set and swap) to implement spinlocks. With spinlocks, threads spin in a loop (busy waiting) until the lock is freed. We motivated the use of blocking locks and semaphores as an improvement over spinlocks because having threads spin can be very inefficient in terms of processor utilization. However, spinlocks are not always less efficient than blocking locks. In two to three sentences, briefly describe a scenario where spinlocks would be more efficient than blocking locks.

14. (5 points) Consider the Nachos implementation of semaphores given by default in your projects. Is this implementation free from starvation? Briefly explain why or why not.