Quiz #6, on Wednesday, November 28th, will consist of questions taken from or inspired Parts I and II of this homework.

Part I — Questions

1. Explain the difference between preemptive and nonpreemptive scheduling.

2. Discuss why a scheduler might decide to give to I/O-bound processes a higher priority than to CPU-bound processes.

3. Which of the following scheduling algorithms could result in starvation?

1. First-come, First-served
2. Shortest Job First
3. Round robin
4. Priority with aging
5. Shortest Remaining Time
6. Priority without aging

4. What is the difference between response time and turnaround time?

5. Why would a scheduler for an interactive system not necessarily be the best choice for a real-time system?

6. What is the convoy effect?

7. Give an example of internal priority, and an example of external priority for a priority-scheduling algorithm.

8. What is aging? Why is it useful?

9. What is processor affinity?

10. Why is isolation between processes such an important feature?

11. What is the difference between a physical address and a virtual address?

Part II – Problems

As usual, I'll assume that you will have successfully completed the following problems by Wednesday, November 28th, so don't wait and let me know if you had difficulties solving them.

Problem 1

Consider the following (idealized) workload:

Process	Burst Time	Priority	Arrival Time
P_1	50 ms	4	0 ms
P_2	20 ms	1	20 ms
P_3	100 ms	3	40 ms
P_4	40 ms	2	60 ms

As an example, if we were to apply the "First Come, First Served" scheduler, and ignoring the context-switch overhead, we would obtain the following (where 1 unit is 10 ms):

P_1	P_1	P_1	P_1	P_1	P_2	P_2	P_3	P_3	P_3	P_3	P_3	P_3	P_3	P_3	P_3	P_3	P_4	P_4	P_4	P_4

Answer the following, making the same assumption that we can ignore the context-switch overhead:

1. Ignoring the context-switch overhead, show the schedule using shortest remaining time, nonpremptive priority (a smaller priority number implies higher priority) and round robin with quantum 30 ms.
2. What is the average waiting time of the above scheduling policies?
3. What is the mean process turnaround time of the above scheduling policies?

Problem 2

Consider the following list of desirable features for a CPU scheduler:

Name	Description
Meeting deadlines	Avoid losing data
CPU utilization	Keep the CPU busy all the time
Balance	Keeping all parts of the system busy
Fairness	Giving each process a fair share of the CPU
Predictability	Avoid quality degradation
Responsive time	Respond to requests quickly
Throughput	Maximize jobs per hours
Policy enforcment	Seeing that stated policy is carried out
Proportionality	Meet users' expectations
Turnaround time	Minimize time between submission and termination

Some of them are critical for **batch systems**, **interactive systems**, **real-time systems**, or **all systems**, some are clearly dispensable for some systems. Identify the correspondences.

Problem 3

Consider the following (idealized) workload:

Process	Burst Time	Priority	Arrival Time
A	50 ms	4	0 ms
B	20 ms	2	10 ms
C	20 ms	1	20 ms
D	30 ms	3	40 ms
E	50 ms	1	60 ms

You will be asked to considerate several scheduling algorithms. If there is ambiguity (i.e., two processes are "equally qualified" to be the next scheduled by the algorithm), use "First Come, First Served" to decide.

1. Ignoring the context-switch overhead, show the schedule using non-preemptive shortest remaining time.
2. What is the average waiting time of the above scheduling policy? Show your calculation.
3. Ignoring the context-switch overhead, show the schedule using preemptive shortest remaining time.
4. What is the average turnaround time for the above scheduling policy?
5. Now, let us assume that context-switching takes 5 ms, and happen every time a process terminate or is removed from the CPU by the scheduler. Show the first 75 ms of the schedule for a premptive priority scheduler (a smaller priority number implies higher priority), taking context-switching into account.

Problem 4

Assume that you run a computer lab with a limited number of seats, and constantly more students than available seats. Every student will start working when given a seat, and will continue to work until completion of the work, or until (s)he is required by your regulation to free the seat (s)he was using.

For each of the following regulation, answer the following:

Q. i. Is the rule pre-emptive, i.e. does it require to ask students to free their seat even if they are still working?

Q. ii. Is this rule likely to create starvation, i.e., will some students be waiting forever to get access to a seat? If yes, describe a scenario where that would happen, if no, describe why.

Q. iii. Practically, what makes this regulation good or bad?

1. "Round Robin with a 10 min. quantum."
2. "CS and IT students have priority over students with other majors when a seat becomes available."
3. "Seats are provided on a first-come, first-served basis. After an hour of waiting, the student gets the seat of who has a seat for the longest period of time."