Operating Systems I

CS 474

Homework #2

Due date: 11/8/2020 at 11:59 pm
Instructions:
1) To complete assignment two, you need to read Chapters 5, 6, and 7 of the textbook.
2) HW must be submitted on typed pdf or word document.
All questions need to be worked on and answered individually. No collaboration of any form is permitted and will amount to plagiarism if performed.

Questions from Chapter 5

5.3 What is the meaning of the term busy waiting? What other kinds of waiting are there in an operating system? Can busy waiting be avoided altogether? Explain your answer. (6 pts)

5.10 Explain why implementing synchronization primitives by disabling interrupts is not appropriate in a single-processor system if the synchronization primitives are to be used in user-level programs. (5 pts)

5.14 Describe how the compare and swap() instruction can be used to provide mutual exclusion that satisfies the bounded-waiting requirement. (15 pts)

5.17 Assume that a system has multiple processing cores. For each of the following scenarios, describe which is a better locking mechanism—a spinlock or a mutex lock where waiting processes sleep while waiting for the lock to become available: (6 pts)

- The lock is to be held for a short duration.
- The lock is to be held for a long duration.
- A thread may be put to sleep while holding the lock.

Questions from Chapter 6

6.6 Suppose that a scheduling algorithm (at the level of short-term CPU scheduling) favors those processes that have used the least processor time in the recent past. Why will this algorithm favor I/O-bound programs and yet not permanently starve CPU-bound programs? (6 pts)

6.11 Discuss how the following pairs of scheduling criteria conflict in certain settings. (9 pts)

- CPU utilization and response time
- Average turnaround time and maximum waiting time
- I/O device utilization and CPU utilization

6.16 Consider the following set of processes, with the length of the CPU burst given in milliseconds: (12 pts)

Thread	Burst	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0.

1) Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).
2) What is the turnaround time of each process for each of the scheduling algorithms?
3) What is the waiting time of each process for each of these scheduling algorithms?
4) Which of the algorithms results in the minimum average waiting time (over all processes)?

6.17 The following processes are being scheduled using a preemptive, round-robin scheduling algorithm. Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an idle task (which consumes no CPU resources and is identified as Pidle). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue. (12 pts)

Thread	Priority	Burst	Arrival
P_1	40	20	0
P_2	30	25	25
P_3	30	25	30
P_4	35	15	60
P_5	5	10	100
P_6	10	10	105

1) Show the scheduling order of the processes using a Gantt chart.
2) What is the turnaround time for each process?
3) What is the waiting time for each process?
4) What is the CPU utilization rate?

Questions from Chapter 7

7.10 Is it possible to have a deadlock involving only one single-threaded process? Explain your answer. (5 pts)

7.16 In a real computer system, neither the resources available nor the demands of processes for resources are consistent over long periods (months). Resources break or are replaced, new processes come and go, and new resources are bought and added to the system. If deadlock is controlled by the banker's algorithm, which of the following changes can be made safely (without introducing the possibility of deadlock), and under what circumstances? (12 pts)

1) Increase Available (new resources added).
2) Decrease Available (resource permanently removed from system).
3) Increase Max for one process (the process needs or wants more resources than allowed).
4) Decrease Max for one process (the process decides it does not need that many resources).
5) Increase the number of processes.
6) Decrease the number of processes.

7.23 Consider the following snapshot of a system: (12 pts)

	Allocation	*Max*	*Available*
	A B C D	*A B C D*	*A B C D*
P_0	2 0 0 1	4 2 1 2	3 3 2 1
P_1	3 1 2 1	5 2 5 2	
P_2	2 1 0 3	2 3 1 6	
P_3	1 3 1 2	1 4 2 4	
P_4	1 4 3 2	3 6 6 5	

Answer the following questions using the banker's algorithm:

1) Illustrate that the system is in a safe state by demonstrating an order in which the processes may complete.
2) If a request from process P1 arrives for (1, 1, 0, 0), can the request be granted immediately?
3) If a request from process P4 arrives for (0, 0, 2, 0), can the request be granted immediately?