Full Name: ____________________________

CS 450 Fall 2009
Midterm Exam

October 14, 2009

Instructions:

- This is a closed-book, closed-notes exam.
- Calculators are **not** permitted.
- Write your full name on the front, and make sure your exam is not missing any pages.
- The problems are of varying difficulty: dispatch the easy ones first and tackle the harder ones later.
- Good luck!

Problem 1	(/30)	:
Problem 2	(/16)	:
Problem 3	(/16)	:
TOTAL	(/62)	:

Problem 1 (30 points)

Circle the single best answer to each of the following multiple choice problems.

1. A valid argument *against* a monolithic kernel design is that, in such an architecture:
   a. message passing between kernel modules is inefficient
   b. an increased number of context switches is required
   c. a bug in one kernel module (e.g., a device driver) may crash the entire kernel
   d. system calls require hardware intervention
2. Which of the following potential priority scheduling issues is targeted by the Highest Penalty-Ratio Next (HPRN) scheduling algorithm?
   a. priority inversion
   b. starvation
   c. deadlock
   d. greediness
3. Which of the following techniques can be used to reduce the likelihood of deadlock due to the mutually exclusive use of certain resources?
   a. all-or-nothing resource utilization
   b. resource preemption
   c. spooling
   d. round-robin allocation
4. Which of the following scheduling algorithms would be a good fit for a general purpose OS running variable length processes with frequent and short CPU and I/O bursts?
   a. selfish RR
   b. regular RR
   c. SJF
   d. FCFS
5. A *preemptive* scheduler is best characterized by its ability to carry out this state transition:
   a. ready → running
   b. running → blocked
   c. blocked → running
   d. running → ready

6. Which of the following scheduling algorithms likely requires the use of a predictive indicator such as an exponential moving average (EMA)?

   a. FCFS

   b. selfish RR

   c. regular RR (with small Q)

   d. SJF

7. Which of the following events would most likely cause a multilevel feedback-queue scheduler to move a given process from a FCFS queue to a RR queue with quantum Q?

   a. The process completes a CPU burst with duration ≤ Q

   b. The process returns from an I/O burst with duration ≥ Q

   c. The RR queue is empty

   d. The process initiates an I/O burst

8. Which of the following best describes the "bounded waiting" constraint on a solution to a synchronization problem?

   a. A thread waiting to enter its critical section will *eventually* be allowed to enter

   b. The first thread to request entry into a critical section will be allowed to enter immediately

   c. After a given thread starts waiting to enter its critical section, there is a limit on how many other threads can enter that critical section

   d. After a given thread starts waiting to enter its critical section, no thread can execute that critical section and "lap" the first thread to reenter the critical section

9. Which of the following semaphore declarations is best used to guarantee mutual exclusion to a critical section?

   a. `Semaphore(-1)`

   b. `Semaphore(0)`

   c. `Semaphore(1)`

   d. `Semaphore(N)  #  N ≥ 2`

10. Which of the following synchronization patterns is best used to implement categorical mutual exclusion?

    a. the light-switch

    b. the turnstile

    c. the footman

    d. rendezvous

11. What assumption is made by the modified Banker's algorithm presented in class for the purposes of deadlock *detection*?

   a. processes will not request more than a pre-declared max of any resource type

   b. processes can complete once granted all resources it is currently requesting

   c. processes will request resources in a predefined order

   d. processes will request resources currently held by other processes

12. The "wait-for" resource-allocation graph was introduced primarily to:

   a. reduce the number of nodes in the resource allocation graph

   b. allow for deadlock detection via cycles in the resource allocation graph

   c. eliminate "false positives" while detecting deadlock

   d. prove that deadlock detection is *impossible*

13. An important implication of the halting problem is that:

   a. deadlock prevention is provably impossible

   b. deadlock prevention is provably possible, given enough time

   c. deadlock detection requires $O(N^2)$ time

   d. all operating systems courses are duds

14. Which of the following responsibilities falls upon the kernel (as opposed to the hardware memory management unit)?

   a. virtual to physical page number translation

   b. TLB lookups

   c. page eviction / replacement

   d. segment base + offset computation

15. Which of the following is a strong argument *against* using only segmentation for memory virtualization?

   a. poor memory utilization

   b. inefficient/complex mapping

   c. lack of logical address space partioning

   d. difficulty of cache integration

Problem 2 (16 points):

All parts of this problem are based on the following table of four processes and their associated arrival and CPU burst times. You may assume context switch times are nonexistent.

Process	Arrival Time	Burst Time
P_1	0	6
P_2	1	3
P_3	2	6
P_4	3	3

1. Using a pre-emptive SJF scheduling algorithm, mark off the following grid to chart the execution of the four processes -- enter a 'X' in a column if the corresponding process is running, enter 'W' if it is waiting, and leave it blank if it has not yet arrived or completed.

P_1	X																			
P_2																				
P_3																				
P_4																				

2. What are the total wait and turnaround times for each of the processes from part (1)?

P_1 wait: P_2 wait: P_3 wait: P_4 wait:

P_1 tnrd: P_2 tnrd: P_3 tnrd: P_4 tnrd:

3. Using a round-robin scheduling algorithm with time quantum Q = 2, mark the following grid to chart the execution of the four processes. Use the same conventions you did for part (1).

P_1	X																			
P_2																				
P_3																				
P_4																				

4. What are the total wait and turnaround times for each of the processes from part (3)?

P_1 wait: P_2 wait: P_3 wait: P_4 wait:

P_1 tnrd: P_2 tnrd: P_3 tnrd: P_4 tnrd:

Problem 3 (16 points):

At the VanderCook College of Music, there is a practice studio that contains three pianos. Up to three student pianists are welcome to enter the studio to use the instruments simultaneously. When they are done practicing, they leave. However, when a virtuoso arrives, she waits until there are no others in the studio, at which point she enters the room and is permitted to practice alone until she is ready to leave.

You are to model this problem using semaphores and count variables for synchronization. In additional to meeting the criteria above, your solution should avoid starvation of any of the pianists (e.g., a virtuoso will not be starved of the room if there is a steady stream of students arriving).

Your solution will consist of two threads (one for the student, one for the virtuoso). The following are recommended semaphore and count variable declarations. Feel free to declare your own.

```
studentCount = 0
mutex = Semaphore(1)
turnstile = Semaphore(1)
studioEmpty = Semaphore(1)
pianoMultiplex = Semaphore(3)
```

Student thread:	Virtuoso thread: