Name:

Operating Systems
Fall 2011
Test 2
October 27, 2011

Closed books, notes, cell phones, PDAs, iPods, laptops, etc. No headphones, please. No calculator needed.

You have 75 minutes to solve 7 problems. You get 10 points for writing your name on the top of this page. As with any exam, you should read through the questions first and start with those that you are most comfortable with. If you believe that you cannot answer a question without making some assumptions, state those assumptions in your answer.

Partial credit will be offered only for meaningful progress towards solving the problems.

Please read and sign below if you agree with the following statement:

In recognition of and in the spirit of the academic code of honor, I certify that I will neither give nor receive unpermitted aid on this exam.

Signature:______________________________

0	/10
1	/10
2	/15
3	/15
4	/10
5	/10
6	/15
7	/15

Total	/100

1. (10 points) **Short Answer Questions**:
   a. A __________ is a programming language construct that encapsulates variables, access procedures, and initialization code within an abstract data type.

   b. In the case of competing processes three control problems must be faced: mutual exclusion, deadlock, and _________.

   c. A monitor supports synchronization by the use of __________ that are contained within the monitor and accessible only within the monitor.

   d. The _________ is a directed graph that depicts a state of the system of resources and processes, with each process and each resource represented by a node.

   e. _________ can be defined as the permanent blocking of a set of processes that either compete for system resources or communicate with each other.

   f. The _________ scheduler determines which programs are admitted to the system for processing.

   g. A risk with _________ scheduling algorithm is the possibility of starvation for longer processes, as long as there is a steady supply of shorter processes.

   h. __________ scheduling is part of the swapping function.

   i. A scheduling mechanism that requires no prior knowledge of process length, yet can nevertheless favor shorter jobs, is known as the __________ scheduling mechanism.

   j. __________ perform static analysis of feasible schedules of dispatching with the result of the analysis being a schedule that determines, at run time, when a task must begin execution.

2. (15 points) **Short attention span:** Answer the following questions (5 points each):
    a. Suppose a scheduling algorithm favors those processes that have used little processor time in the recent past.
        1. Explain why this algorithm favors I/O-bound processes.
        2. Explain why this algorithm does not permanently deny processor time to CPU-bound processes.
    b. Most round-robin schedulers use a fixed size quantum. Give an argument in favor of a small quantum. Now give an argument in favor of a large quantum. Compare and contrast the types of systems and jobs to which the arguments apply. Are there any for which both are reasonable?
    c. What information about a task might be useful in real-time scheduling?

3. (15 points) Consider a system with a total of 150 units of memory, allocated to three processes as shown:

Process	Max	Holds
1	70	45
2	60	40
3	60	15

Apply the Banker's algorithm to determine whether it would be safe to grant the request from each of the two scenarios below. If yes, indicate the sequence of requests that could be guaranteed possible. If no, show the unsafe state.

   a. A fourth process arrives, with a maximum memory in need of 60 and an initial need of 25 units. (The initial need will be granted right away).
   b. A fourth process arrives, with a maximum memory in need of 60 and an initial need of 35 units. (The initial need will be granted right away).

4. (10 points) The Dining Philosophers have worked up a solution to avoid deadlock, with a little help from a consultant with a recent doctorate in algorithms. Before eating, each philosopher will flip a coin to decide whether to pick up the left fork or the right fork first. If the second fork is taken, the philosopher will put the first fork down, then flip the coin again. Is this solution deadlock-free? Justify your answer by checking the four conditions for deadlock.

5. (10 points) Deadlock: In the code below, three processes are competing for six resources labeled A to F.
    a. Using a resource allocation graph show the possibility of a deadlock in this implementation.
    b. Modify the order of some of the get requests to prevent deadlock. (You cannot move requests across processes, only change the order inside each process). Show that your solution is deadlock free.

void P0()
{
while (true) {
  get(A);
  get(B);
  get(C);
  // critical region:
  // use A, B, C
  release(A);
  release(B);
  release(C);
  }
}
	void P1()
	{
	while (true) {
	  get(D);
	  get(E);
	  get(B);
	  // critical region:
	  // use D, E, B
	  release(D);
	  release(E);
	  release(B);
	  }
	}
		void P2()
		{
		while (true) {
		  get(C);
		  get(F);
		  get(D);
		  // critical region:
		  // use C, F, D
		  release(C);
		  release(F);
		  release(D);
		  }
		}

6. (15 points) Suppose the following jobs are to be executed in a uniprocessor system.

Job	Arrival Time	CPU Burst Time
P_1	0	9
P_2	2	2
P_3	4	1
P_4	6	12
P_5	8	2

Assume the overhead of context switching is one time unit. Neglect the time spent by the scheduler if the scheduling decision does not lead to a context switch. Assume an interrupt is generated every time a process finishes and every time a process arrives. For each of the following scheduling methods, give:

(i) a Gantt chart to illustrate the execution sequence,
(ii) the average job turnaround time, and
(iii) the processor efficiency (defined as the ratio between the time spent doing useful work for executing all processes and the total time spent until all processes finish, thus including overhead).

a. Round Robin with a time quantum of 3.

b. Multilevel Feedback Queue with queues numbered 1-10 (where the lower the number, the higher the priority, with queue 1 the highest priority and queue 10 the lowest), quantum = 2i, where i is the queue level number and processes are initially placed in the first queue (i.e., level 1). In this scheduling policy, each process executes at a particular level for one quantum and then moves down a level; processes never move up a level.

7. (15 points) Consider a set of three periodic tasks with the execution profiles below. Develop scheduling diagrams for (1) fixed priority (preemptive) scheduling and (2) earliest deadline scheduling using completion deadline.

Process	Arrival Time	Execution Time	Ending Deadline
A(1)	0	10	20
A(2)	20	10	40
…	…	…	…
B(1)	0	15	50
B(2)	50	15	100
…	…	…	…
C(1)	0	10	50
C(2)	50	10	100
…	…	…	…

Each square in the table below represents five time units. Use a letter in the square to refer to a process.

Questions:

1) Fill the table with the information on arrival time, execution time, and deadlines.

	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
A																				
B																				
C																				

2) Consider the fixed priority scheduling. Consider the priority as follows: A, B, C. What is the scheduling of the tasks for this system? Fill the table below and mark the missed deadlines.

	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
FPS																				

3) Consider the earliest deadline scheduling using completion deadline. What is the scheduling of the tasks for this system? Fill the table below and mark the missed deadlines:

	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
EDS																				