This practice midterm is DEFINITELY MUCH TOO LONG. The real midterm will be shorter.

Homework problems are also possible.

PLEASE WRITE YOUR NAME AND ANSWERS ON ALL 8 QUESTION SHEETS. You may use the backs of the question sheets to continue your answers. You may also use the blank sheet after question 7 to further continue your answers. Scrap paper is available. GOOD LUCK!!

1A. Draw the process state diagram from the notes. This diagram contains nodes (i.e. circles) labeled with the possible states for a process. It also contains arcs (i.e. arrows) showing the various state transitions possible. Label the nodes and arcs (for example one node should be labeled running and one arc should be labeled schedule.

1B. Shortest Job First and Preemptive Shortest First have a very favorable property. What is it? Why are they impractical for general purpose operating systems? Even if they were practical, they also have an unfavorable property that would make them unsuitable purpose systems. What is it? What would you add to these scheduling algorithms to alleviate this unfavorable property.

1C. Some arcs change the NUMBER of processes in the system. Which arcs are these and what system calls do they correspond to.

2A. Define, but do **NOT** solve the Readers Writers problem.

2B. Some systems simply treat the readers writers problems as critical section problems and hence the implementation simply use P and V. What requirement of the Readers Writers problem does this implementation not satisfy?

3. Consider the following set of processes) run using RR scheduling with q=3. All times are in milliseconds and context switching takes zero time. The CPU time (column **TWO**) is the total time required for the process. The creation time (column **THREE**) is the time when the process is created. So P1 is created when the problem begins and P2 is created 4 milliseconds later. If ties occur use the lab 2 tie-breaking rule. After P0 has run for 5ms it blocks for 9ms and never blocks again. After P1 has run for 4ms it blocks for 6ms and never blocks again. P2 never blocks. At what time did does each process finish.

Process	CPU Time	Creation Time	Blocks after	Blocks for
P0	10	0	5	9
P1	11	4	4	6
P2	9	4	never	blocks

4A. Draw a resource allocation graph (also called a reusable resource graph) that represents a deadlock state. Your graph must contain at least two resources and at least two tasks. Each resource must contain 3 units.

4B. Of course when execution started there were no arcs in the resource allocation graph. Give a scenario starting from this initial condition of no arcs and ending in the graph you gave for 4A. That is, tell what requests and releases occur and in what order. For this part you should assume a naive (i.e., optimistic) resource manager that grants every request as soon as it can.

4C. Recall that the Banker’s algorithm for resource allocation never enters a deadlocked state like the one in part A. Consider the scenario you gave for part B and tell at what point the Banker’s algorithm will depart from the scenario. That is, indicate the first request that the Banker’s algorithm will refuse to grant that the optimistic manager did grant. Don’t forget that whenever you deal with Banker’s algorithm or (un)safe states, the claims of each process are important.

5A. Draw a reusable resource graph that represents an **UN**safe state that is **NOT** deadlocked. (Hopefully you remembered that the claims of the processes are important).

5B. Process A is already written and requests the printer, the plotter, the tape drive, and the robotic arm, in that order. You need to develop processes B and C. Both need the printer and the arm, B needs the tape drive, and C needs the plotter. What order should the requests be made? Why.

6. Consider a system containing a total of 12 units of resource R and 24 units of resource S managed by the banker's algorithm. There are three processes P1, P2, and P3. P1's claim is 0 units of R and 12 units of S, written (0,12). P2's claim is (8,15). P3's claim is (8,20). Currently P1 has 4 units of S, P2 has 8 units of R, P3 has 8 units of S, and there are no outstanding requests.

6A. What is the largest number of units of S that P1 can request at this point that the banker will grant?

6B. If P2 instead of P1 makes the request, what is the largest number of units of S that the banker will grant?

6C. If P3 instead of P1 or P2 makes the request, what is the largest number of units of S that the banker will grant?

7. Fill in the blanks with the appropriate technical term or phrase.

(1) When a linker converts a relative address to the corresponding absolute address, we say it is ________________________ the relative address.

(2) A program in execution is called ________________________.

(3) The processor scheduling algorithm resulting in the smallest average waiting time is ________________________.

(4) A directed graph with processes represented as circles, resources represented as squares, and requests and allocations represented as arcs is called ________________________.

(5) The Banker's algorithm ensures that the system is always in ________________________.