CS 162   Prof. Alan J. Smith

Problem Set 1

Please remember that your answers to these questions should be prepared on the computer system and handed in via the usual process.

1. Operating systems can be considered as *coordinators* and as *extended machines*. Please define and explain each.

2. Define a process and explain process state.

3. The following algorithm, developed by Dekker, is the first known correct software solution to the critical section problem for two processes. The two processes, P0 and P1, share the following variables:

```
var flag: array [0..1] of boolean; (*initially false*)
    turn: 0..1;
```

The following program is for process Pi (i=0 or 1), with Pj (j=0 or 1) being the other process:

```
repeat
  flag[i]:=true;
  while flag[j]
    do if turn = j
      then begin
        flag[i]:=false;
        while turn=j do noop;
        flag[i]:=true;
      end;
  ...
  critical section
  ...
  turn:=j;
  flag[i]:=false;
```

```
        ...
      remainder section
        ...
      until false;
```

Prove that the algorithm satisfies all three requirements (mutual exclusion, progress, bounded waiting) for the critical section problem.

4. Explain why interrupts are necessary for multiprogramming to be very useful.

5. Consider a set of sequential processes that cannot share any variables except synchronization variables, such as semaphores. Can these processes communicate with each other? Please assume that the scheduling algorithm is such that each process is guaranteed to be able to run for at least .1 seconds out of every second, and that processes can also read the real time clock.

6. Redo the readers and writers problem from class to give priority to readers instead of writers.

7. Consider a system consisting of 4 resources of the same type, being shared by 3 processes, each of which needs at most 2 resources. Can a deadlock occur in this system? If so, show how. If not, explain why not. You can assume that the processes are totally independent except for their need to use the 4 resources in the shared pool.