1) Define multiprogramming and explain how it is able to increase CPU utilization.

2) Define SPOOLing and explain how the OS uses SPOOLing and multiprogramming to increase both CPU utilization and I/O device utilization.

3) Assume, in a multiprogrammed batched system, that the OS only switches to another process when the running process blocks for I/O. What additional state transition is necessary in a time sharing system and why is it necessary ?

4) Why are system calls necessary and describe the general categories into which all system calls can be placed.

5) Describe the capabilities of the virtual machine provided by the IBM VM/370 OS and the virtual 8086 machine provided by the Windows OS on Intel machines.

6) Why is it necessary to provide a hardware mechanism for disabling interrupts on the 8259A and the CPU separately ? (Think about this in terms of ISRs).

7) Describe the differences between a system call for input that is not available and the timer ISR handling by the OS for a process whose time slice has expired.

8) Describe the problems associated with threads with respect to OS functions and implementation. What are the tradeoffs in managing threads in user space verses kernel space ?

9) Describe the conditions for race conditions and describe how a shared memory variable value may become corrupt under such conditions. Describe how this can happen on a load/store architecture where memory variables must be moved into registers before being operated on. Use assemble language instructions in your explanation and describe the sequence of context switches that result in corruption.

10) Describe why it is not possible to change the while statement in Peterson's solution to

```
while (turn != process && interested[other] == TRUE)
```