**MdShahid Emdad**

**11-23-2021**

**Fall 2021**

**CSC332,Sec M**

**Quiz 2**

Q1. (50 Points)

Consider the interrupt mechanism discussed in class with the following modification. In the interrupt vector, each PSW value has the mode bit=”user”.

A special machine instruction is provided, whose meaning is to change the mode bit in the PSW register to “supervisor”. The first instruction of each service routine will be this special instruction.

Assume that the logical and physical addresses are same and memory protection is not being done (so before fetching an instruction in the service routine, the cpu will not complain that the address is outside user’s memory area.)

There are no other changes to the interrupt mechanism

that we discussed in class.

Will this scheme work?

Explain in LESS THAN 50 WORDS.

**SOLUTION:**

There are 2 different scenarios that needs to be take into consideration:

1. If the special instruction is privileged it will not work because the mode bit is set to"user.
2. But, if it is not privileged, it will work because user can run non privileged instructions

Q2. (50 Points)

Consider process management with RR as discussed in Ch 4.

Suppose we did the same shell command that created the 4 processes for ./cpu

Make these assumptions:

1. There is only one cpu.
2. There are only 5 processes in the system—A, B, C, D, and the shell.
3. Initially all the 4 user processes and the shell are in ready list in the order A, B, C,D, shell.
4. **All processes are entirely in physical memory.**
5. The only interrupts in the system are the Hardware Timer and interrupts caused by execution of these processes.
6. **The cpu scheduler:**

**The cpu scheduler strictly follows the RR algorithm, with time slice = 0.3 seconds.**

When a ready proc gets cpu, it gets a fresh time slice of of above duration. This slice has nothing to do with how much time it has already executed before becoming ready.

1. When a proc terminates (normal termination or runtime abortion), an interrupt is generated and a service routine is called. The service routine will do some work such as updating state of the process etc. and the cpu scheduler gives cpu to another process. Any unused time of the process that just terminated, is gone and is irrelevant.
2. **The spin(1) in the code will first record its initial starting time. Then it keeps checking the current time. If (current time – initial time) >= 1, then it returns.**
3. Any computation other than spin(1) in the system will take nearly zero time, ex. the initial if statement, checking the loop condition, executing a service routine, etc. But spin(1) will run in real time, not in zero time.
4. **When a process executes the printf statement in cpu.c, it does NOT get blocked; its output is stored in memory and the process continues its execution.** The OS sends various outputs to the printer, in FIFO order, from time to time when the printer becomes idle. One print operation of a text line takes 1 second. The output device prints things in strictly First-come-first-served order.

What is the sequence of the first three letters printed?

Write down **all such possible sequences** of the first 3 letters.

Explain your answer in less than 100 words.

**SOLUTION:**

The sequence is ABC. At time 0 process A starts execution since ready list is managed in FIFO order and A is the first one in it. Due to the time slide, 0.3, A goes to back of ready list and B starts execution. At time 0.6 B goes to back of ready list and C starts. Continuing like this, processes get 0.3 seconds each, spin (1) returns only if (current time – initial time) >= 1. Hence, A will send its printout first, then B, then C as followed.