CSCI 3753: Operating Systems

**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Fall 2016

Midterm Exam

**10/18/2016**

**Answer all questions in the space provided**

**Multiple Choice Questions:** Choose one option that answers the question best.

**[30 Points]**

1. Resource manager view of operating systems is

* A. OS provides mechanisms to deal with the complexity of hardware
* B. OS provides support for developing applications for a computing system
* C. OS allows sharing and effective utilization of computing system resources
* D. OS provides equivalent of a virtual machine that is easier to use
* E. OS provides support for security and protection

1. Difference between multiprogramming and multitasking is
   * A. In multiprogramming, a long running CPU-bound process may significantly delay the execution of other processes, while in multitasking, this cannot happen.
   * B. In multitasking, a process can be preempted only when it blocks for I/O while in multiprogramming, a process can be preempted any time.
   * C. Context switch times are much larger in multiprogramming than in multitasking.
   * D. There is no difference between multiprogramming and multitasking if none of the processes block during execution.
   * E. A and D only.
2. Which of the following is FALSE about trap table?

* A. A trap table is used to implement systems calls.
* B. A trap table resides in the OS kernel.
* C. A user can modify trap table entries by writing an LKM.
* D. Entries in a trap table can be accessed only in supervisor mode.
* E. A user process can access trap table entries using system calls.

1. The processor mode bit in modern computing systems
   * A. allows OS to implement interprocess communication using shared memory.
   * B. allows OS to implement multi-threading inside the kernel.
   * C. allows OS to protect the kernel from interference from the user program.
   * D. ensures that user programs cannot execute kernel functions.
   * E. None of the above.
2. The /proc directory

* A. allows applications and users to peer into the kernel's view of the system.
* B. Contains a hierarchy of special files which represent the current state of the kernel.
* C. provides a mechanism for kernel and kernel modules to send info to processes in user space.
* D. contains a special type of files called a virtual files.
* E. All of the above.

1. Which of the following is not included in process control block?

* A. program counter value
* B. list of open files
* C. priority
* D. cached data
* E. stack

1. Advantages of threads over processes include

* A. lower context switch time
* B. no possibility of race conditions
* C. sharing of heap and stack
* D. smaller code size
* E. None of the above

1. Which of the following is FALSE about IPC via pipes?

* A. Basic primitives are send( ) and receive( ).
* B. Communication can be blocking or non-blocking.
* C. Pipes can be anonymous or named.
* D. Pipes can be used for only one-way communication.
* E. IPC via pipes is slower than IPC using shared memory.

1. IPC using shared memory

* A. requires processes to agree on a key name in advance
* B. uses send and receive primitives
* C. requires one process to create a shared memory segment using shmget( ), while the other process to attach to the existing shared memory using shmat( )
* D. A and C but not B
* E. None of the above

1. Which of the following is FALSE about Pthread mutex?

* A. It cannot be used for process synchronization
* B. It can be used for implementing mutual exclusion
* C. It is as expressive as a binary semaphore
* D. Deadlocks can occur if it is not properly used
* E. It can be used to solve the producer consumer problem

**Short Answer Questions [30 Points]:**

1. The following code is executed by 10 different processes that all share the integer variable *counter* whose initial value is 5. Explain if there is a possibility of race condition in this code.

*counter*++;

1. Using a simple system call as an example (e.g. getpid, or uptime), describe what is generally involved in providing the result, from the point of calling the function in the C library to the point where that function returns.
2. Explain two ways that I/O can be overlapped with CPU execution and how they are each an improvement over not overlapping I/O with the CPU.
3. Provide a step-by-step description for adding a new device in Linux operating system without requiring recompiling the kernel.
4. Both CPU and DMA controller move data to/from main memory. Describe three ways they can share access to memory.
5. In round robin scheduling, discuss the tradeoffs in choosing a large time slice vs a small time slice.

## Problems

1. **[8 Points]** Suppose a computer center has two printers, A and B, that are similar but not identical. There are three kinds of processes, K1, K2 and K3 that use these printers with the following rules: K1 processes can only use printer A, K2 processes can only use printer B, and K3 processes can use either A or B.

A process calls *request* function to request a printer. This function returns the identity of a free printer. After using that printer, the process returns it by calling the *release* function. Implement *request* and *release* functions for the three kinds of processes. Use semaphores for synchronization. Your solution should be fair and deadlock free assuming that a process using a printer eventually releases it.

1. **[12 Points]** You are the organizer of a gaming exhibit where you want attendees to play your startup’s new game demo. You model the attendees as threads, called players, and your job is to synchronize access to a single copy of the game, as follows:

* When a player arrives, he or she waits in a waiting area.
* When there are four or more players waiting to play, you allow exactly four of them to exit the waiting area to start playing. The four players leave the waiting area and approach the game console.
* When a player reaches the game console, the player waits until all four players are at the console, at which point all four players begin playing.
* Players may finish playing at any time. However, you cannot allow any new player to approach the game console until all four players have left.

Your task is to write a function *play*( ) that a player calls when he/she wants to play. Use monitors for synchronization.

1. **[20 Points]** Four processes, P0, P1, P2 and P3 are created in a system at times 0, 10, 15 and 20 respectively. Their CPU time requirements are 40, 100, 20 and 30 time units respectively. Assume that the context switching time is negligible.
2. Show the Gantt chart for the execution of these four processes, if the FCFS scheduling algorithm is used. Calculate the average turnaround time.
3. Show the Gantt chart for the execution of these four processes, if the (non-preemptive) shortest job next scheduling algorithm is used. Calculate the average wait time.
4. Show the Gantt chart for the execution of these four processes, if the (non-preemptive) earliest deadline first scheduling algorithm is used. The completion deadline for these processes are 100, 310, 40 and 60 respectively. Do all processes meet their deadlines?
5. Show the Gantt chart for the execution of these four processes, if the pure round-robin scheduling algorithm is used with time slice of 30 time units. Calculate the average response time.