Name: ______________________________

CS 4/53201

Exam #1

Operating Systems

Friday 30 September 1998

---

1. (10 points) Explain what is meant by the term "multiprogramming".

Multiprogramming refers to the operating system keeping several processes in memory, and switching between them when one process terminates or waits, in an attempt to keep the CPU fully utilized. (Note that this is different from timesharing, which switches between processes more quickly so as to provide better response to interactive users.)

2. (10 points) In what way is memory-mapped I/O "better" than interrupt-based asynchronous I/O?

Memory-mapped I/O allows the I/O device to transfer an entire block of data to or from memory. It does not require intervention by the CPU during this process, and it only interrupts the CPU when the entire block has been transferred.

3. (15 points) Most modern CPUs and operating systems support two modes of operation: user mode and kernel mode.

a. What can a process / thread do in kernel mode that it can not do in user mode?

It can execute privileged instructions, which might be able to access I/O devices, control interrupts, manipulate memory hardware, etc.

b. Briefly explain how a process / thread can get into kernel mode.

It makes a system call to request a specific service from the operating system. As part of the system call, a trap instruction is executed, which atomically sets kernel mode and jumps to the handler for that specific trap.

4. (10 points) What is the difference between a "program" and a "process"?

A program is a passive entity — just object code on a disk. A process is a program in execution, and includes not only the program's object code but its entire execution state (data in memory, data in registers, PC and SP values, etc.). Multiple processes may be executing the same program.

5. (15 points) In the 5 state process model, processes in both the "ready" and "blocked" states are waiting on something before they can move into the "running" state.

Name: ______________________________

a. **What are the processes waiting on in the "ready" and "blocked" states?**

Processes in the ready state are waiting to be scheduled onto the CPU. Processes in the blocked state are waiting for something to occur (e.g., for I/O to finish).

b. **In the entire system, how many processes can be in each of these three states at any point in time?**

Only one process can be in the running state, but any number of processes can be in the ready and blocked states (although the OS may actually impose some practical limit).

6. **(15 points) Briefly explain the major differences between a client communicating with a server via message-passing and via remote procedure calls (RPCs).**

To communicate via message-passing, the client packs information in to messages and uses a Send to send it to the server. Then it uses a Receive to wait for a response.

To communicate via RPCs, the client performs what looks almost like a local procedure call; the operating system takes care of the details of the communication (packing and unpacking arguments, handing errors, etc.).

7. **(10 points) Why is a context switch between threads faster than a context switch between processes?**

To switch between threads, only the registers, PC, SP, and possibly some accounting information must be saved. To switch between processes, much more must be saved, including the memory image, records of OS resources in use, etc.

8. **(15 points) Explain how a server (e.g., a file server) can be structured to make effective use of threads.**

One thread accepts requests from clients, and it forks a new "worker" thread to handle each request. Furthermore, if one worker blocks, others can continue executing.