Name: ______________________________

CS 4/53201 # Exam #1 **Operating Systems**

Friday 5 October 2001

---

1. **What is the difference between synchronous I/O and asynchronous I/O, and in what way is asynchronous I/O "better"? (10 points)**

   With synchronous I/O, the CPU initiates an I/O operation and then waits until the I/O operation completes, whereas with asynchronous I/O, the CPU continues executing instructions while the I/O operation proceeds concurrently.

   Asynchronous I/O is better because it allows the CPU to do useful work during the I/O operation, instead of just waiting.

2. **One of the states in the five-state process model is the "blocked" state (sometimes called the "waiting" state).**

   a. **A process or thread may go into the blocked state when it is waiting on an I/O operation to complete. Give an example that illustrates this situation. (5 points)**

      Waiting for the user to type a character on the keyboard, waiting for the user to move the mouse, waiting for a character to finishing being displayed on the screen, waiting for a disk write to finish, etc.

   b. **Besides waiting on an I/O operation to complete, why else might a process or thread be in the blocked state? (10 points)**

      Besides I/O, a process might block for several reasons. If it is communicating with another process using message-passing, a blocking send or recv operation may cause a process to block. If it is using semaphores or locks and condition variables, a process may also block under certain conditions when it performs a semaphore P/wait, a condition variable wait, or a lock acquire. (*Later on in the course, we will see a third reason why a process may block.*)

3. **Consider the various possible types of message-passing.**

   a. **Using direct communication, is it possible for a process to receive messages from more than one sender? Explain. (5 points)**

      Yes, most recv operations allow the use of a "wildcard" that allows a message to be received from any sender, rather than just one specific sender. *(The textbook refers to this as "asynchronous communication".)*

   b. **What is the major benefit that indirect communication provides over direct communication? (10 points)**

      Indirect communication uses mailboxes (or tags), so a receiving process can distinguished between different types of messages if each type is sent to a different mailbox.

Name: ______________________________

4. **Suppose three threads are all part of the same process.**

   a. **What resources do the three threads share? (8 points)**

      Program code, address space, global variables, heap, open files, I/O devices.

   b. **What resources are associated with each individual thread? (7 points)**

      Program counter, stack pointer, general-purpose registers.

5. **Semaphores can be used for both mutual exclusion and synchronization.**

   a. **In this context, what is meant by mutual exclusion? (8 points)**

      Semaphores can be used for mutual exclusion, meaning that when one thread is executing a particular piece of code (called the "critical section"), it receives exclusive access to that code — any other thread attempting to access that code must block until the first thread finishes.

   b. **In this context, what is meant by synchronization? (7 points)**

      Synchronization is used to coordinate two processes. For example, a producer can signal a consumer process when it has produced something, and a consumer process can wait for that production to occur.

6. **There are many of similarities between semaphores and condition variables.**

   a. **What are the major differences between semaphores used for synchronization and condition variables, with respect to their operation inside a critical section of code? (10 points)**

      There are two major differences. First, a condition variable wait operation can be used inside a locked region, while a semaphore P/wait operation can not be used inside a critical section. Second, semaphores have a value while condition variables do not (leading to the difference illustrated in parts (b) and (c) below).

   b. **When used for synchronization, suppose a semaphore V operation occurs before a semaphore P operation. What happens when the P operation executes, and why? (5 points)**

      Since the V operation will have already incremented the value of the semaphore in advance, when the P operation executes, it will decrement the value of the semaphore and continue. In effect, what it would have waited for otherwise has already occurred, so there is no need for it to wait.

   c. **Suppose a condition variable signal operation occurs before a condition variable wait operation. What happens when the wait operation executes, and why? (5 points)**

      It would wait for the next signal to occur. Since condition variables do not have a value, there is no "memory" of signal operations having occurred previously.

Name: ______________________________

7. What is the relationship between monitors, classes, and locks? (10 points)

Monitors are a programming language construct, essentially providing the functionality of a class with locks enforcing mutually exclusive access to all the member functions of the class. However, the functionality of the locks is provided by the implementation of the monitor, rather than explicitly being placed into the code by the programmer.