

Friday 4 October 2002

1. An important innovation in bringing computing to the masses was interactive timesharing.

a. Define timesharing. (5 points)

In a time-shared system, multiple users connect to one computer, but the OS switches rapidly between their various processes, giving each user rapid response time and the illusion of a dedicated computer.

b. How does interactive timesharing compare to using a personal computer? (10 points)

On a timeshared computer, there are often multiple users running processes at the same time, whereas on a personal computer, there is usually only a single user running processes. However, on both a timeshared computer and a personal computer, the OS may be switching between multiple processes to provide rapid response time.

2. A CPU can communicate with I/O devices in a variety of ways, one of which is software-polling synchronous I/O.

a. Briefly define software-polling synchronous I/O. (5 points)

The OS starts an I/O operation, then continuously checks (“polls”) that I/O device until the I/O operation finishes. During this time the CPU busy-waits, rather than doing useful work.

b. Why is this technique not ideal? (5 points)

It wastes CPU time looping and checking, instead of doing useful work executing another process while the I/O proceeds.

3. In the five-state process model, two of the states are “ready” and “blocked”. How do these two states differ, in terms of what it means to be in that state, how that state is entered, how that how that state is left, etc.? (20 points)

A process in the “ready” state is waiting to use the CPU, and if placed on the CPU will start executing; a process in the “blocked” state is waiting on some event (e.g., an I/O operation to complete or a semaphore signal to be received), and would not be able to the CPU if it were placed on the CPU.

A process enters the “ready” state when created, when it unblocks, or when it is pre-empted from the CPU (e.g., by a timeout); a process enters the “blocked” state when it is running, but must now wait on some event (see list above).

Name: _____

A process leaves the “ready” state by going onto the CPU (i.e., into the “running” state); a process leaves the “blocked” state when the event it is waiting on occurs.

4. Message-passing between processes can be classified as direct or indirect.

a. Briefly define indirect communication. (5 points)

Indirect communication is a form of message-passing where the sending process communicates to the receiving process through a mailbox (usually owned by the receiving process), instead of contacting that receiving process directly.

b. What advantage does indirect communication provide over direct communication? (5 points)

Indirection communication allows a receiving process to separate different types of communication — one mailbox for one type of communication, another for another type of communication.

5. Why might threads be particularly useful in writing a server, for example a server that provides access to items from a database? (15 points)

One thread can receive user requests and act as a dispatcher, passing each request to a thread that will service that request. By using threads instead of processes, the threads can share common data, such as the database, perform fast context switches, execute another thread if one blocks, etc.

6. Semaphores can be implemented using a variety of techniques, including busy-waiting, disabling interrupts, and test-and-set instructions. What are the disadvantages of each of these three techniques? (20 points)

Busy-waiting: wastes CPU time doing nothing useful, doesn't support a queue of blocked threads, and doesn't provide a means to protect the code that implements wait / signal

Disabling interrupts: also wastes CPU time busy-waiting and doesn't support a queue of blocked threads, doesn't work on multiprocessors, and can interfere with timer

Test & set instructions: also wastes CPU time busy-waiting and doesn't support a queue of blocked threads

7. Explain the similarities and differences between a monitor and a class (in an object-oriented language). (10 points)

Like a class, a monitor hides private data and allows it to be accessed only through the monitor's access functions. However, a monitor goes beyond classes by allowing only one access function to be called at a time, ensuring mutually exclusive access to its private data (an effect that can be achieved by acquiring a lock at the beginning of every access function and releasing it at the end).

Name: _____