Chapter 11

Purpose: Gain experience using shared memory.

Build a readers and writers program using share memory.

In your previous readers and writers problem, you didn't actually read and write a shared variable, you only printed the messages.

In this problem you don't have to print the messages, but you do have to do the reading and writing. The sleeps are designed to reveal if you haven't coordinated the use of the shared memory correctly.

You will need 3 bytes of shared memory.

The reader process does the following 5 times (loop):
Enter the critical section
read the first byte of shared memory
sleep 1 second
read the second byte of shared memory
sleep 1 second
read the third byte of shared memory
print the three bytes
sleep 2 second

The writer process does the following 5 times (loop):
Enter the critical section.
increment the first byte of shared memory
sleep 1 second.
increment the second byte of shared memory
sleep 1 second.
increment the third byte of shared memory
sleep 3 seconds.

Testing: start 3 readers and 2 writers.

Always start a writer first.

The writer that starts first (i.e., the one that creates the shared memory) should initialize the 3 bytes of shared memory to 1, 1 and 1.

Note 1: everytime a reader prints the three bytes, they should all be the same.

Note 2: cout tends to do the wrong thing here, it assumes your integers are characters because they are bytes. Printf with a cast will work.

When you are done. Remove the shared memory and the semaphores (by hand using the `ipcrm` command).

It is permissible (and smart) to start with a copy of your readers and writers programs from the previous assignment.

Demo: Your program. The instructor will also ask to look at the code.