CSE 410 - Computer Systems
Homework 2

Assigned: Wednesday, November 14, 2001

Due: Wednesday, November 21, 2001
            At the start of class

Your name: _____

1. Briefly define the following terms and highlight the key differences.

a.   program

b.   process

c.   thread

2. Provide short answers to the following questions.

a.   What is an advantage of kernel threads?

b.   What is a disadvantage of kernel threads?

c.   What is an advantage of user threads?

d.   What is a disadvantage of user threads?

3. Of the many variables or values that change when a process switches from one thread to another, name two of them and describe what they represent.

4. Of the many variables or values that change when the operating system switches from one process to another, name two of them and describe what they represent. Do not repeat your answer from 3.

5. Consider the process tree shown in the lecture of November 14 on slide 19. This slide shows the output from the tlist program, running under Win2K on my laptop.

    a.   What is the name of the tlist program file?

    b.   What is the name of the program file that is running in the parent process of tlist?

    c.   What is the name of the program file that is running in the grandparent process? (ie, the parent of the parent of tlist).

    d.   Do you think that the parent of tlist used a blocking or non-blocking call to create the tlist process? Describe your reasoning.

    e.   Do you think that the grandparent of tlist used a blocking or non-blocking call to create the parent of tlist? Describe your reasoning.

6. The slide that follows the tlist slide is titled "Fork Example." Draw a process tree showing the situation in this example at the moment when the "ls" command is executing. Label the nodes of the tree with the process ID numbers and the names of the program files. You don't have to include the command interpreter (bash) or any parent processes that are not shown in the example.

7. Consider the slide labeled "One Thread Three Threads" on page 10 of the Threads lecture dated November 16. Given what you know about 32-bit virtual address spaces, describe how you might allocate stack space to each of the various threads. Does the fact that the stacks and the heap all grow various amounts during execution cause a problem? Why or why not?