

*This homework is due at the beginning of class on September 24, 2014 and is worth 1.5% of your grade.*

Name: \_\_\_\_\_

CCIS Username: \_\_\_\_\_

<b>Problem</b>	<b>Possible</b>	<b>Score</b>
1	10	
2	30	
3	20	
4	25	
5	15	
Total	100	

**1.** What are the primary differences between a program, an executable, and a process? (10 pts)

**2a.** Describe the actions taken by the kernel to context-switch between processes. (10 pts)

**2b.** What are two events that can cause a running process to be context-switched out? (5 pts)

2c. Why might one want context switches to happen very frequently?

(5 pts)

2d. Why might one want context switches to happen infrequently?

(5 pts)

2e. How is multiprogramming able to achieve much more efficient use of the computer's resources (as opposed to running only a single process)?

(5 pts)

- Using the system calls `fork()`, `waitpid()`, and `kill()`, write a program in pseudo-C that creates two children, waits for the first child to complete, and then kills the second before exiting. (20 pts)

4a. Name three ways in which a process can be terminated.

(5 pts)

4b. What operations does the kernel perform once a process is terminated?

(10 pts)

4c. Suppose a computer does not have a hardware timer, and the running process enters a `while(1) {}` loop. Is there any way for the kernel to remove this process from the CPU? If so, what is it? If not, why not?

(10 pts)

5. Suppose we are implementing a kernel for a hardware architecture that does not have support for the mode bit (i.e., there is no distinction between user mode and kernel mode). Which of the following resources can we protect from illegal access? For each one, you must state why or why not.

5a. Memory (i.e., can we prevent a process from accessing another processes' memory?) (5 pts)

5b. CPU (i.e., can we prevent a process from refusing to give up the CPU?) (5 pts)

5c. Devices (i.e., can we prevent a process talking directly to devices?) (5 pts)