
CMPSCI 377: Operating Systems

Homework 1: Processes and Threads

Due: October 7, 2008

1. (10 pts) What are differences between a program, an executable, and a process?
2. (10 pts) Give three ways of switching from user mode to kernel mode.
3. (10 pts) What happens on a context switch? Should context switches happen frequently or infrequently? Explain your answer.
4. (10 pts) In Round-Robin scheduling, what happens when the quantum is very very small? What happens when it is really big?
5. (10pts) What are the differences between user-level threads and kernel threads? Under what circumstances is one type better than the other?
6. (20 pts) Using the *fork()*, *waitpid()*, and *kill()* system calls, write a program in which a parent creates two children. The parent then waits for the first child to complete, and kills the second when the first completes. After that, the parent exits.
7. (20 pts) Consider the following piece of code:

```
main(int argc, char ** argv)
{
    int child = fork();
    int c = 5;

    if(child == 0)
    {
        c += 5;
    }
    else
    {
        child = fork();
        c += 10;
        if(child)
            c += 5;
    }
}
```

How many different copies of the variable *c* are there? What are their values?

8. (10 pts) A hardware designer argues that there are enough transistors on the chip to provide 1024 integer registers and 512 floating point registers. What is the effect of having such a large number of registers on the operating system from a CPU scheduling perspective?