
CMPSCI 377: Operating Systems

Homework 1: Processes

Due: September 28, 2000, in class

1. (10 pts) What is multiprogramming and what are its advantages?
2. (5 pts) What is the purpose of system calls?
3. (5 pts) What is a context switch? Describe the actions taken by a kernel for a context switch?
4. (20 pts) Using the *fork()*, *waitpid()*, *exit()* and *kill()* system calls, write a program in which a parent creates a child and the child creates a grandchild. The grandchild prints "I am a grandchild" and sleeps for 60 seconds. The child then kills the grandchild and exits. The parent waits for the child to finish and prints "Child finished" and exits. You may use pseudo-code to write your program.

5. (10 pts) Consider an interprocess-communication that uses mailboxes:

```
mailbox::send(ToID, msg)          // send msg to Process Id, ToID
mailbox::receive(FromID, msg)    // receive msg from process id, FromID
```

Is it possible for a process to wait to get a message from any one of a number of processes? If your answer is yes, write a program that does so.

If we add a new mailbox primitive:

```
mailbox::empty(FromID) // returns false if no msg in mailbox,
                       // return true otherwise
```

Use it, to write a program that waits for a message from any one of a number of processes?