CSS 430 Operating Systems
Programming Assignment Report
Author: Jay Hennen
10/12/2013

-----> Part 1

processes.cpp is a simple program intended to emulate the way the linux shell executes processes.

1. The parent process forks a child(C) process, which forks a grandchild(GC) process, which forks a great grandchild(GGC) process.
2. The GGC process sets up it's cout to the pipe and then execlp into 'ps -A'.
3. The GC process puts the pipe as cin and a second pipe as cout, then execlp into 'grep argv[0]'.
4. Finally, the child puts the second pipe as cin and executes wc -l.
5. Once the child is done, then the parent exits and the program is over.

The included run script run.sh executes several tests of processes compared to the 'ps -A | grep REGEX | wc -l' benchmark.

-----> Part 2

Shell.java is a simple shell for ThreadOS. The algorithm for running a command string is as follows.

1. Read the command from keyboard
2. Split the command on ";" to get an array of sequential commands
3. For each sequential command, split on "&" to get an array of concurrent commands.
4. Execute each concurrent command simultaneously, adding their thread IDs to a set representing outstanding threads.
5. As each thread finishes, join() retrieves the thread IDs and checks/removes them from the set.
6. Once the set is empty, the concurrent commands are done, and the next command in sequence is executed.
7. Once all sequential commands are executed, print the shell prompt again in preparation for more commands from keyboard.

To test Shell.java, load it from the ThreadOS, then execute commands made up of PingPong calls, sequential, concurrent, and dummy command calls to make sure it can handle the various cases. Some examples are as follows

```
// Basic concurrent and sequential call
PingPong a 10 & PingPong b 10 ; PingPong c 10
// Concurrent dummy commands
PingPong a 10 & b & c & PingPong d 10 ; PingPong e 10
// Stacking delimiters
PingPong a 10 ;; PingPong b 10 ; ; & PingPong c 10 &&& PingPong d 10
```