Youngmin Lee
CSS 430 – Operating Systems
Professor Deal
June 30 2017

Homework 1 Report

How to execute processes.cpp:

1) Compile processes.cpp "g++ -o processes processes.cpp".
2) Execute processes.cpp "./processes.cpp [arg]" where [arg] is some string you want to search the Process List for.
3) The output will be the number of lines matching [arg] in ps –A.

Discussion of algorithm:

1. Set up two pipes for process communication
2. Fork for ps and replace the child process's image with ps and its appropriate arguments
3. Wait until ps has exited gracefully after placing its output into the first pipe
4. Fork for grep, passing in arg[1] as an argument
5. Wait until grep has exited
6. Fork for wc, but don't redirect stdout so wc's stdout is sent to the terminal

How to execute Shell.java

1) Build Shell.class by running 'javac Shell.java'
2) Move Shell.class to the same ThreadOS folder as all the other commands
3) Launch Boot with 'java Boot'
4) Inside ThreadOS, launch the Shell class with 'l Shell'
5) At this point you should be at the Shell prompt and can enter any amount of commands delimited by & and ; for nonblocking and blocking execution, respectively

Discussion of algorithm:

1. Take the input string ("something & something2 & something3 ; something4") from SysLib.cin
2. If the input string is "exit" or "q", return back to ThreadOS
3. Split it at either the & or ; delimiters (using lookback of one character). This results in an array containing the elements "something &", "something2 &", "something 3;", and "something4".
4. Loop through the array of elements containing one command each
5. For each command in the array:
    a. Check whether each command ends in either & or ; (or nothing)
    b. Execute the command and decide whether to block until join has returned or continue immediately without waiting

Output Testing:

```
youngmin@fenixfox ~ % g++ test.cpp -o processes
[1]

youngmin@fenixfox ~ % ps -A | grep tty | wc -l
[0]

2

youngmin@fenixfox ~ % ./processes tty
[0]

2

youngmin@fenixfox ~ % ps -A | grep Sys | wc -l
[0]

0

youngmin@fenixfox ~ % ./processes Sys
[0]

0

youngmin@fenixfox ~ % ps -A | grep user | wc -l
[0]

0

youngmin@fenixfox ~ % ./processes user
[0]

0
```

```
$ java Boot

threadOS ver 1.0:

Type ? for help

threadOS: a new thread (thread=Thread[Thread-3,2,main] tid=0 pid=-1)

-->l Shell

l Shell

threadOS: a new thread (thread=Thread[Thread-5,2,main] tid=1 pid=0)

Shell[1]% PingPong abc 10 & PingPong xyz 10 ; PingPong 123 10 &

threadOS: a new thread (thread=Thread[Thread-7,2,main] tid=2 pid=1)

threadOS: a new thread (thread=Thread[Thread-9,2,main] tid=3 pid=1)

abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc xyz abc
xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc abc
xyz abc xyz xyz abc xyz abc xyz abc xyz abc abc xyz xyz

xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz
xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz
xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz
xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz

threadOS: a new thread (thread=Thread[Thread-11,2,main] tid=4 pid=1)

Shell[2]% 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123

q

-->

-->q

q
```