CS 374 Lab 3: Process Control in Linux

* Boot a computer in vLinux, or log into the nrs-labs.humboldt.edu server. **On this handout, the mark beginning this line just represents a Linux prompt. Please don’t be confused by it!** This lab gives you experience with common Linux commands related to process control.

Each person in your group should clearly print their name here to so I know who worked in your group; I only need one hand-in from each group:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Open up a vLinux connection.

Using the **man**ual pages, describe what each of the following commands do; if you have seen the command already in the previous item(s), then explain what the options do:

* 1. ps \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  2. ps –au Note: there are two versions of u here, one accepting a list, one without!

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. kill \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  2. kill -9 [PID] Read this man page carefully, you’ll have to type another command… \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  3. find

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. bg

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Try out some of the commands and their options to see what they do before going on (you can’t try out bg without applying it to another command).

1. Next you will run the find program with redirection. You can use the ‘>’ character to redirect the output of a command into a file (rather than printing it to the screen). Try this now with the following commands:

* find . -name be -print
* find . -name be -print > foo.bar
* ls

You should see a file called foo.bar in the directory you are in. Use the less command to view its contents. Then use the rm command to delete it.

1. Next you’ll learn about running a program in the background using & and ending it using kill. Compile the following program in Linux, making the executable program called “sleeper”:

#include <stdlib.h>

#include <stdio.h>

#include <unistd.h>

int main( ) {

sleep(100); // sleeps this many seconds

printf("My process ID is %d and my parent process ID is %d.\n", getpid(), getppid() );

exit(0);

}

Run the following commands *quickly*, one right after the other. To ensure you can run kill, I recommend that you type the four commands carefully into a **text file**, then cut and paste them to the open Linux window (you will need to do this again below, so this saves time). The ‘2>’ is supposed to allow secondary redirection, but on some systems that is done with ‘>&’ so try both to see which works *before* doing the command sequence!

* sleeper > pathnames 2> /dev/null &
* sleeper > foo.paths 2> /dev/null &
* sleeper > foobar.paths 2> /dev/null &
* ps -u

Note: the & at the end of a command works the same as bg on the front of a command, *both are ways of running processes in the background*.

When you run this block of commands, the results of ps -u should let you should see new processes active—namely, at least one or two of the processes you just started! For all processes you can see, jot down below the following:

command unique process id (pid) time started

What does the **parent process ID** printed by each instance of sleeper correspond to?

Once you have done this, you need to kill off the running sleeper processes. To do this, use the process ID number you wrote down above. For each process ID for a find command, do this:

* kill -9 <pid>

1. Reenter the multiple command sequence above using paste. Then type the following quickly:

* fg

The last program put to sleep will be brought up, so you may see it running—that is, you won’t get your prompt back. Type:

* <ctrl-c> /% Note: this is one character, hold down the ctrl key, type c
* ps -u

What happened? Is the process dead or alive?

Restart the block of processes. Now type:

* fg
* ctrl-z
* ps -u

What happened? What is the ***p****rocess* ***s****tatus* of the process you just handled?

**Summary:** you should know what the utilities ps, kill, bg, fg, and find are used for. You should be able to use the ‘&’ to make processes run in the background, and redirections < and > to provide input and output to programs.

Hand in once copy of the lab sheet for your group so I can check you off, and you’re good to go!