Dan Herve
CSCD 240
Steiner

Lab 3

1) Clearly explain the difference between locate, which, whereis, grep, and find.

find: finds one or more files by their approximate filenames.

locate: finds files by name.

grep: finds text within a file.

whereis: the whereis command is used to locate the binary, the source code and the online manual page for any specified program.

which: used to identify the location of executables.

2) In class we discussed the umask command. Explain how to set the umask value so group members have write permissions. Capture the output of setting the command a long list after creating a new file to show the umask command was properly set.

```
dherve@cslinux:~$ umask g+w
...
dherve@cslinux:~/cscd240u14/lab3$ touch test.txt
dherve@cslinux:~/cscd240u14/lab3$ ls -l
total 0
-rw-rw-r-- 1 dherve IT-GenericLinuxGroup 0 Jul  3 20:18 test.txt
```

3) In class we discussed chmod and the different ways to use the command including chmod g+x or chmod 756. Use the chmod command with +t on a normal file contained in your directory. Now try to remove that file. Explain the +t of chmod.

The +t adds a sticky bit to the file so that only the owner and root can edit the file. Otherwise anyone with write permissions would be able to. I was allowed to remove it since I created the file.

```
dherve@cslinux:~/cscd240u14/lab3$ chmod +t test.txt
dherve@cslinux:~/cscd240u14/lab3$ rm test.txt
dherve@cslinux:~/cscd240u14/lab3$ ls
test2.txt
```

```
dherve@cslinux:~/cscd240u14/lab3$ touch test.txt
dherve@cslinux:~/cscd240u14/lab3$ ls -l
total 0
-rw-r--r-- 1 dherve IT-GenericLinuxGroup 0 Jul  3 20:20 test2.txt
-rw-r--r-- 1 dherve IT-GenericLinuxGroup 0 Jul  3 20:25 test.txt
dherve@cslinux:~/cscd240u14/lab3$ chmod +t test.txt
dherve@cslinux:~/cscd240u14/lab3$ ls -l
total 0
-rw-r--r-- 1 dherve IT-GenericLinuxGroup 0 Jul  3 20:20 test2.txt
-rw-r--r-T 1 dherve IT-GenericLinuxGroup 0 Jul  3 20:25 test.txt
```

4) Issue the find command looking for the file named ldd starting at the root directory.

    a. Assuming you are not logged in as root, you should get a list of errors as well as where the file was found. Capture the output and include it in your submission – you do not need to include all the permission errors just a few to get the idea but do include where the file was found.

```
find: `./etc/ssl/private': Permission denied
find: `./tmp/tmux-900688935': Permission denied
find: `./root': Permission denied
./usr/bin/ldd
find: `./sys/kernel/debug': Permission denied
find: `./lost+found': Permission denied
```

    b. Repeat the command (again not as root) – illustrating a method of eliminating the error messages and printing only what was found.

```
dherve@cslinux:/$ find -name ldd 2>/dev/null
./usr/bin/ldd
```

5) In class we talked about the '–name' option for the find command.

    a. Explain how to use the size option.

```
dherve@cslinux:~$ find -size +10M
./.cache/mozilla/firefox/rhs5lt9h.default/Cache/_CACHE_003_
```

    M can be replaced with the units desired. For instance, c for bytes, G for gigs, k for kilobytes, etc.

b. Issue and capture the results of the find command in your home directory that display all files that are greater than 1K. Do not search for more than 3 subfolders. Do not display error messages.

```
dherve@cslinux:~$ find -maxdepth 3 -size +1k -type f 2>/dev/null
./.bash_history
./Documents/herved_lab1.pdf
./Documents/herved_lab1.zip
./Documents/240 - Lab 1.odt
./Documents/herve_lab2.odt
./Documents/herved_lab1.pdf.odt
./.local/share/recently-used.xbel
./.jedit/activity.log
./.pulse/21f7ebd9adda17b1ea2c8b45529f6c94-device-volumes.tdb
./.pulse/21f7ebd9adda17b1ea2c8b45529f6c94-stream-volumes.tdb
./Downloads/viQuickRef.pdf
./Downloads/doc.pdf
./Downloads/cscd240_u14_syllabus.pdf
./Downloads/toc (1).pdf
./Downloads/doc (2).pdf
./Downloads/toc.pdf
./Downloads/doc (1).pdf
./Downloads/cscd240_u14_lab1.pdf
./.ICEauthority
./.bashrc
./cscd240u14/lab1/my.copy.bashrc
./.cache/tracker/meta.db-wal
./.cache/tracker/meta.db-shm
./.cache/tracker/meta.db
./.cache/tracker/ontologies.gvdb
./.cache/event-sound-cache.tdb.21f7ebd9adda17b1ea2c8b45529f6c94.x86_64-pc-linux-g
nu
./.zcompdump
./.pki/nssdb/key4.db
./.pki/nssdb/cert9.db
./.gnome2/accels/nautilus
./.config/dconf/user
./.config/gedit/accels
./.config/chromium/Safe Browsing Bloom Prefix Set
./.config/chromium/Safe Browsing Extension Blacklist
./.config/chromium/Safe Browsing Cookies
./.config/chromium/Safe Browsing Bloom
./.config/chromium/Safe Browsing Download
```

./.config/chromium/Safe Browsing Csd Whitelist
./.config/chromium/Safe Browsing Cookies-journal
./.config/chromium/Local State
./.config/chromium/Certificate Revocation Lists
./.config/chromium/Safe Browsing Download Whitelist
./.thumbnails/normal/cb7c058fcf61d0230740234182fd4d2c.png
./.thumbnails/normal/cbd30cc1923c6afdaffb3fc0a3504736.png
./.thumbnails/normal/db0dc63939a10a3920b4499ab82fd6e9.png
./.xsession-errors

6) Use a text editor on the remote machine to create a file named frost.poem that contains the following text:

The Road Not Taken by Robert Frost
Two roads diverged in a yellow wood,
And sorry I could not travel both
and be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;
Then took the other, as just as fair,
And having perhaps the better claim
Because it was grassy and wanted wear,
Though as for that tHe passing there
Had worn them really about the same,

a. Use the grep command, capture both the command and the output, to finds all lines, including the line number, that end with a comma.

dherve@cslinux:~/cscd240u14/lab3$ grep -n ,$ frost.poem
2:Two roads diverged in a yellow wood,
8:Then took the other, as just as fair,
10:Because it was grassy and wanted wear,
12:Had worn them really about the same,

b. Use the grep command, capture both the command and the output, to finds all lines, including the line number, containing the word as.

dherve@cslinux:~/cscd240u14/lab3$ grep -n ' as ' frost.poem
5:And looked down one as far as I could
8:Then took the other, as just as fair,
11:Though as for that tHe passing there

c. Use the grep command, capture both the command and the output, to finds all lines, including the line number that starts with the word and (case DOES NOT matter).

dherve@cslinux:~/cscd240u14/lab3$ grep -ni and frost.poem
3:And sorry I could not travel both
4:and be one traveler, long I stood
5:And looked down one as far as I could
9:And having perhaps the better claim
10:Because it was grassy and wanted wear,

d. Use the grep command, capture both the command and the output, to finds all lines, including the line number that starts with the word and (case DOES matter).

dherve@cslinux:~/cscd240u14/lab3$ grep -n and frost.poem
4:and be one traveler, long I stood
10:Because it was grassy and wanted wear,

7) Capture, creating a directory named lab3.

dherve@cslinux:~/cscd240u14$ mkdir lab3

a. Capture placing a copy of frost.poem in the directory lab3. There should be one copy of frost.poem in your home directory and one in lab3.

dherve@cslinux:~/cscd240u14$ cp frost.poem ./lab3

b. Within your home directory, capture the grep command and its output that will recursively find all instances of the word I (case DOES matter) in all files that end with .poem.

dherve@cslinux:~$ grep -r I *.poem
And sorry I could not travel both
and be one traveler, long I stood
And looked down one as far as I could

c. Capture translating all Th in the file frost.peom to TH and all th to TH but no cases of tH should be translated. Save this output on the command line with redirection to frosttr.poem.

dherve@cslinux:~$ sed -i 's/Th/TH/' frost.poem
THe Road Not Taken by Robert Frost
Two roads diverged in a yellow wood,
And sorry I could not travel both

and be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;

THen took the other, as just as fair,
And having perhaps the better claim
Because it was grassy and wanted wear,
THough as for that tHe passing there
Had worn them really about the same,

8) Using a text editor create a file named myScript that contains the following:
    #!/bin/bash
    string="Hello World"
    echo $string

    a. Try to execute the script with ./myScript and capture the output.

    dherve@cslinux:~$ echo `./myScript`
    -bash: ./myScript: Permission denied

    b. Execute and capture the command that will change the permissions on myScript to be
    user executable without changing any other permissions.

    dherve@cslinux:~$ chmod u+x myScript

    c. Execute the script with ./myScript and capture the output.

    dherve@cslinux:~$ ./myScript
    ./myScript: line 2: $'World\342\200\235': command not found

9) Using a text editor create a file named secondScript that contains the following:
    #!/bin/ksh
    string="Hello World"
    print $string

    a. Try to execute the script with ./secondScript and capture the output.

    dherve@cslinux:~$ echo `./secondScript`
    -bash: ./secondScript: Permission denied

    b. Execute and capture the command that will change the permissions on secondScript
    to be user executable without changing any other permissions.

dherve@cslinux:~$ chmod u+x secondScript

c. Execute the script with ./secondScript and capture the output.

dherve@cslinux:~$ ./secondScript
./secondScript[2]: World": not found [No such file or directory]

d. What does the #! mean?

run from this shell

e. In problem 7 what shell did the code execute in?

.bash

f. In problem 8 what shell did the code execute in?

.ksh

10) Copy secondScript into a file named thirdScript, ensure you preserve the timestamp of secondScript. Capture the output of stat on both files to prove you preserved the timestamp.

dherve@cslinux:~$ stat secondScript
  File: `secondScript'
  Size: 50   Blocks: 8   IO Block: 32768 regular file
Device: 15h/21d Inode: 53960717   Links: 1
Access: (0744/-rwxr--r--) Uid: (900724753/ dherve)   Gid:
(800000001/IT-GenericLinuxGroup)
Access: 2014-07-03 21:35:42.000000000 -0700
Modify: 2014-07-03 21:49:21.000000000 -0700
Change: 2014-07-03 21:49:21.000000000 -0700
 Birth: -

dherve@cslinux:~$ stat thirdScript
  File: `thirdScript'
  Size: 50   Blocks: 8   IO Block: 32768 regular file
Device: 15h/21d Inode: 53960719   Links: 1
Access: (0744/-rwxr--r--) Uid: (900724753/ dherve)   Gid:
(800000001/IT-GenericLinuxGroup)
Access: 2014-07-03 21:35:42.000000000 -0700
Modify: 2014-07-03 21:49:21.000000000 -0700
Change: 2014-07-03 22:03:47.000000000 -0700

Birth: -

11) Capture the diff command, ignoring case and white space, on frost.poem and frosttr.poem.

dherve@cslinux:~$ diff ./frost.poem ./frosttr.poem
1c1
< The Road Not Taken by Robert Frost

> THe Road Not Taken by Robert Frost
8c8
< Then took the other, as just as fair,

> THen took the other, as just as fair,
11c11
< Though as for that tHe passing there

> THough as for that tHe passing there

12) Define what a process is and what a job is, clearly explain how jobs differ from processes.

A process is a program being run by a computer. A job is a collection of related processes.

13) In the lab3 directory create the C file named lab3.c with the following code.
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
printf("Hello World\n");
return 0;
}// end main
```

dherve@cslinux:~/cscd240u14/lab3$ ls
frost.poem lab3.c

14) Give the grep command that will start in your home directory and show the file names and line numbers containing the term "stdio" in all .c files in the home directory and all directories below the home.

dherve@cslinux:~/cscd240u14/lab3$ grep -rnH "stdio" *.c
lab3.c:1:#include <stdio.h>

15) Consider the following command ls – al | more.

a. How many processes are created with that command?

2 processes are created

b. What exactly does "|" do in this command?

It sends the output of ls -al to more, but the output is not compatible so it does nothing

16) Using the cut command and the ls –l command, issue the command that will remove the year from the long listing.

dherve@cslinux:~/cscd240u14/lab3$ ls -l | cut -d" " -f1-5,10-
total 8
-rw-r--r-- 1 dherve IT-GenericLinuxGroup 403 frost.poem
-rw-r--r-- 1 dherve IT-GenericLinuxGroup 104 lab3.c

17) Using the man page for ps

a. Issue and capture the ps command with the appropriate options to allow listing of all processes in the system.

dherve@cslinux:~/cscd240u14/lab3$ ps -A
  PID TTY   TIME CMD
        1 ?   00:00:02 init
        2 ?   00:00:00 kthreadd
        3 ?   00:00:15 ksoftirqd/0
        5 ?   00:00:00 kworker/u:0
        6 ?   00:00:01 migration/0
        7 ?   00:00:04 watchdog/0
        8 ?   00:00:01 migration/1
   10 ?   00:00:11 ksoftirqd/1
   12 ?   00:00:04 watchdog/1
   13 ?   00:00:00 cpuset
   14 ?   00:00:00 khelper
   15 ?   00:00:00 kdevtmpfs
   16 ?   00:00:00 netns
   17 ?   00:00:18 kworker/u:1
   18 ?   00:00:37 sync_supers
   19 ?   00:00:00 bdi-default
   20 ?   00:00:00 kintegrityd
   21 ?   00:00:00 kblockd
   22 ?   00:00:00 ata_sff
   23 ?   00:00:00 khubd

```
 24 ?           00:00:00 md
 26 ?           00:00:00 khungtaskd
 27 ?           00:00:02 kswapd0
 28 ?           00:00:00 ksmd
 29 ?           00:00:00 khugepaged
 30 ?           00:00:00 fsnotify_mark
 31 ?           00:00:00 ecryptfs-kthrea
 32 ?           00:00:00 crypto
 40 ?           00:00:00 kthrotld
 41 ?           00:00:00 scsi_eh_0
 42 ?           00:00:00 scsi_eh_1
 64 ?           00:00:00 devfreq_wq
156 ?           00:00:00 mpt_poll_0
159 ?           00:00:00 mpt/0
206 ?           00:00:00 scsi_eh_2
220 ?           00:00:00 jbd2/sda1-8
221 ?           00:00:00 ext4-dio-unwrit
306 ?           00:00:00 upstart-udev-br
308 ?           00:00:00 udevd
457 ?           00:00:00 kpsmoused
481 ?           00:00:00 ttm_swap
494 ?           00:00:00 upstart-socket-
513 ?           00:00:01 rpcbind
680 ?           00:00:00 reiserfs
723 ?           00:00:00 sshd
724 ?           00:00:00 rpciod
745 ?           00:00:50 rsyslogd
746 ?           00:00:00 rpc.statd
747 ?           00:00:00 nfsiod
754 ?           00:00:00 rpc.idmapd
759 ?           00:00:00 dbus-daemon
775 ?           00:00:14 avahi-daemon
777 ?           00:00:00 avahi-daemon
831 tty4       00:00:00 getty
837 tty5       00:00:00 getty
854 tty2       00:00:00 getty
855 tty3       00:00:00 getty
857 tty6       00:00:00 getty
862 ?           00:00:00 acpid
866 ?           00:00:01 cron
867 ?           00:08:19 automount
872 ?           00:00:00 atd
912 ?           00:01:46 irqbalance
```

```
 914 ?         00:00:26 whoopsie
 920 ?         00:03:21 .vasd
 937 ?         00:06:01 .vasd
 938 ?         00:00:08 flush-8:0
1092 tty1      00:00:00 getty
1108 ?         00:10:42 .vasd
1109 ?         00:14:07 .vasd
1110 ?         00:00:10 .vasd
2120 ?         00:00:00 sshd
2165 ?         00:00:00 sshd
2167 ?         00:00:00 sftp-server
3790 ?         00:00:00 kworker/0:0
7718 pts/1     00:00:00 a.out
8010 ?         00:00:00 sshd
8063 ?         00:00:00 sshd
8064 pts/4     00:00:00 bash
8270 ?         00:00:00 sshd
8343 ?         00:00:01 sshd
8344 pts/0     00:00:00 bash
8495 ?         00:00:00 sshd
8539 ?         00:00:00 sshd
8540 pts/5     00:00:00 bash
9070 ?         00:00:00 sshd
9122 ?         00:00:00 sshd
9124 pts/6     00:00:00 bash
9195 ?         00:00:22 kworker/0:1
9244 ?         00:00:00 kworker/1:1
10182 ?        00:00:00 kworker/1:2
10307 ?        00:00:00 flush-0:21
10567 ?        00:00:00 kworker/1:0
10726 pts/5    00:00:00 a.out
10728 pts/5    00:00:00 a.out
10729 pts/5    00:00:00 a.out
11429 pts/2    00:00:00 ps
12283 ?        00:00:16 cifsd
15841 ?        00:00:00 xfs_mru_cache
15842 ?        00:00:00 xfslogd
15843 ?        00:00:00 xfsdatad
15844 ?        00:00:00 xfsconvertd
15847 ?        00:00:00 jfsIO
15848 ?        00:00:00 jfsCommit
15849 ?        00:00:00 jfsCommit
15850 ?        00:00:00 jfsSync
```

```
22458 ?      00:00:00 winbindd
22459 ?      00:00:01 winbindd
22507 ?      00:00:02 smbd
22536 ?      00:00:00 winbindd
22537 ?      00:00:00 winbindd
22542 ?      00:00:00 winbindd
22544 ?      00:00:28 nmbd
22676 ?      00:00:00 smbd
23360 ?      00:00:00 udevd
23655 ?      00:00:00 sshd
23710 ?      00:00:03 sshd
23711 pts/3  00:00:01 bash
26445 ?      00:00:00 sshd
26495 ?      00:00:00 sshd
26496 pts/1  00:00:00 bash
28466 ?      00:00:00 sshd
28516 ?      00:00:09 sshd
28517 pts/2  00:00:01 bash
30665 pts/1  00:00:00 nano
```

b. Using the output from part A, what was the first process started and by whom was it started?

The first process was started by root and it was /sbin/init

c. What was the first non-root process that was started?

rsyslogd -c5

d. What was the last process started and by whom?

stodd2 started myScript last

18) Using a text editor create the following C programmed named almostEndless.c
```
#include <stdio.h>
int main()
{
int x = 0;
while(x < 20)
{
printf("..");
fflush(stdout);
sleep(2);
```

x++;
}// end while
return 0;
}// end main

a. Compile your program with gcc almostEndless.c
b. Start your program with ./a.out
c. With your program running, describe the commands you would use (without using ctrl-c) to terminate that program, from the same terminal window in which it was started

I would use ctrl-z to put the program in the background; I would then use ps to find the process id. Finally, I would use kill -9 plus the pid to kill it.

d. Execute and capture the commands, using process notation, to terminate a.out

dherve@cslinux:~/cscd240u14/lab3$ ./a.out almostEndless.c
....^Z
[1]+ Stopped   ./a.out almostEndless.c
dherve@cslinux:~/cscd240u14/lab3$ ps
  PID TTY   TIME CMD
12524 pts/2   00:00:00 a.out
12567 pts/2   00:00:00 ps
28517 pts/2   00:00:01 bash
dherve@cslinux:~/cscd240u14/lab3$ kill -9 12524
[1]+ Killed   ./a.out almostEndless.c

e. Restart your program with ./a.out
f. Execute and capture the commands, using job notation, to terminate a.out

19) In a single terminal window capture the command to start a.out 3 times each running as background jobs:

a. What are the job numbers of the above?

1
2
3

b. What are the process ID numbers of the above?

13062
13074
13081

c. Capture the command and output to bring the second a.out to the foreground.

dherve@cslinux:~/cscd240u14/lab3$ fg %2

d. Capture the command(s) to send a.out back to the to the background.

dherve@cslinux:~/cscd240u14/lab3$ bg %2

e. Capture the command(s) to kill the third a.out using its job number.

dherve@cslinux:~/cscd240u14/lab3$ kill -9 %3

[3]+ Stopped   ./a.out almostEndless.c

f. Capture the command(s) to kill the first a.out using its process number.

dherve@cslinux:~/cscd240u14/lab3$ kill -9 13062

g. Can CTRL C be used to kill any job? Why or why not? Clearly explain why or why not.

No, ctrl-c can only kill jobs in the foreground. Not all programs can run in the foreground