

# Lab Report

## 1. Screenshot of cd/ls practice.

```
jboicken@ns1:~$ cd
jboicken@ns1:~$ ls
jboicken@ns1:~$ pwd
/home/jboicken
jboicken@ns1:~$ cd /
jboicken@ns1:/$ ls var
backups cache crash lib local lock log mail opt run snap spool tmp
jboicken@ns1:/$
```

## 2. Screenshot of directory structure after mkdir/mv practice (use tree).

```
jboicken@ns1:~$ mkdir Documents Pictures Music bin
jboicken@ns1:~$ cd Pictures/
jboicken@ns1:~/Pictures$ mkdir catz report pets
jboicken@ns1:~/Pictures$ mv report/ ~/Documents/
jboicken@ns1:~/Pictures$ mv catz kittens
jboicken@ns1:~/Pictures$ tree ~
/home/jboicken
├── bin
├── Documents
│   └── report
├── Music
├── Pictures
│   ├── kittens
│   └── pets
└──
```

7 directories, 0 files  
jboicken@ns1:~/Pictures\$ \_

## 3. Difference between > and >>.

> overrides the right hand file, whereas >> appends to the right hand file.

Both will create a new empty file if the right hand file doesn't exist.

```
jboicken@ns1:~$ tail src clone append
==> src <==
This file is called src

==> clone <==
This file is called clone

==> append <==
This file is called append
jboicken@ns1:~$ cat src > clone
jboicken@ns1:~$ cat src >> append
jboicken@ns1:~$ tail src clone append
==> src <==
This file is called src

==> clone <==
This file is called src

==> append <==
This file is called append
This file is called src
jboicken@ns1:~$
```

#### 4. How do you search in less?

It uses at least similar vi(m) commands, so I type '/pattern' for top down searching or '?pattern' for bottom up searching, which starts at the top or bottom of what is shown on my terminal. Pattern is the text that I am searching for. Upon pressing enter, I am put into a search mode and can go to the next match with 'n' and the previous with 'N' meaning shift + 'n'.

#### 5. What do which, tail, head and grep do?

which shows you the absolute / full path of the file executed when a command is given. If the name passed is not a command, it will say it wasn't found. As well, if the command is part of the shell being used, it should say that it is built into the shell. (Depends on implementation)

tail shows you the final lines of a file, and head shows you the first lines of a file. Both default to 10 lines.

grep searches files for the existence of a pattern. The pattern can simply be the exact string to find or a string containing regular expressions that are parsed and searched for. The program will print out each line that matches the pattern. (Arguments can change the behavior in other ways as well.)

#### 6. Screenshot of last 10 lines of /var/log/auth.log using one of the commands learned in this lab.

```
jboicken@ns1:~$ cd /var/log
jboicken@ns1:/var/log$ tail -10 auth.log
Sep  2 17:51:17 ns1 sudo: pam_unix(sudo:session): session closed for user root
Sep  2 18:01:52 ns1 sudo: jboicken : TTY=tty1 ; PWD=/home/jboicken ; USER=root ; COMMAND=/usr/bin/cat /etc/shadow
Sep  2 18:01:52 ns1 sudo: pam_unix(sudo:session): session opened for user root by jboicken(uid=0)
Sep  2 18:01:52 ns1 sudo: pam_unix(sudo:session): session closed for user root
Sep  2 18:02:21 ns1 sudo: jboicken : TTY=tty1 ; PWD=/home/jboicken ; USER=root ; COMMAND=/usr/bin/cat /etc/passwd
Sep  2 18:02:21 ns1 sudo: pam_unix(sudo:session): session opened for user root by jboicken(uid=0)
Sep  2 18:02:21 ns1 sudo: pam_unix(sudo:session): session closed for user root
Sep  2 18:03:26 ns1 sudo: jboicken : TTY=tty1 ; PWD=/home/jboicken ; USER=root ; COMMAND=/usr/bin/passwd -S
Sep  2 18:03:26 ns1 sudo: pam_unix(sudo:session): session opened for user root by jboicken(uid=0)
Sep  2 18:03:26 ns1 sudo: pam_unix(sudo:session): session closed for user root
jboicken@ns1:/var/log$
```

#### 7. How do you view every process that's running with ps?

ps -e or ps -A or ps aux

ps is a command to get the status of processes, ps means process status. By default, it will return some bits of information about each process, which I think are run time, the terminal interface running the process, name of process, and PID. To show everything, the 3 commands above can be run. The man page says -A and -e are identical, so I'm guessing one is for Unix standard and the other is POSIX compliance. ps aux is the BSD formatting.

## 8. Screenshot of output from `top`, name of process with PID 1.

Surprisingly, the command is *systemd* and not *init*. Either way, I'm pretty sure that PID 1 is for running an "initial script" to start up and manage system services & daemons at boot.

```
top - 18:36:47 up 1:19, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 182 total, 1 running, 181 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.0 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 981.2 total, 159.0 free, 163.5 used, 658.7 buff/cache
MiB Swap: 1962.0 total, 1959.7 free, 2.3 used, 664.0 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	103056	12424	8160	S	0.0	1.2	0:07.08	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-kblockd
9	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
10	root	20	0	0	0	0	S	0.0	0.0	0:00.21	ksoftirqd/0
11	root	20	0	0	0	0	I	0.0	0.0	0:00.28	rcu_sched
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
13	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
16	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_kthre
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
21	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writeback
22	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kcompactd0
23	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
24	root	39	19	0	0	0	S	0.0	0.0	0:00.00	khugepaged
70	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kintegrityd
71	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kblockd
72	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	blkcg_punt_bio
73	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	tpm_dev_wq
74	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	ata_sff
75	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	md
76	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	edac-poller
77	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	devfreq_wq
78	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdogd

## 9. Method of running `ping` in background without output printed to the screen.

`ping XXX.XXX.XXX.254 -i 5 > ping.log &` (Saving a log from ping by redirecting output)

Or

`ping -q XXX.XXX.XXX.254 -i 5 &` (Telling ping to be quiet and not print anything)