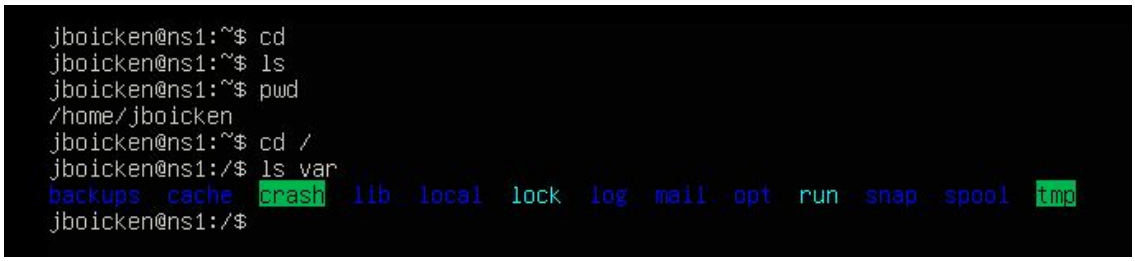
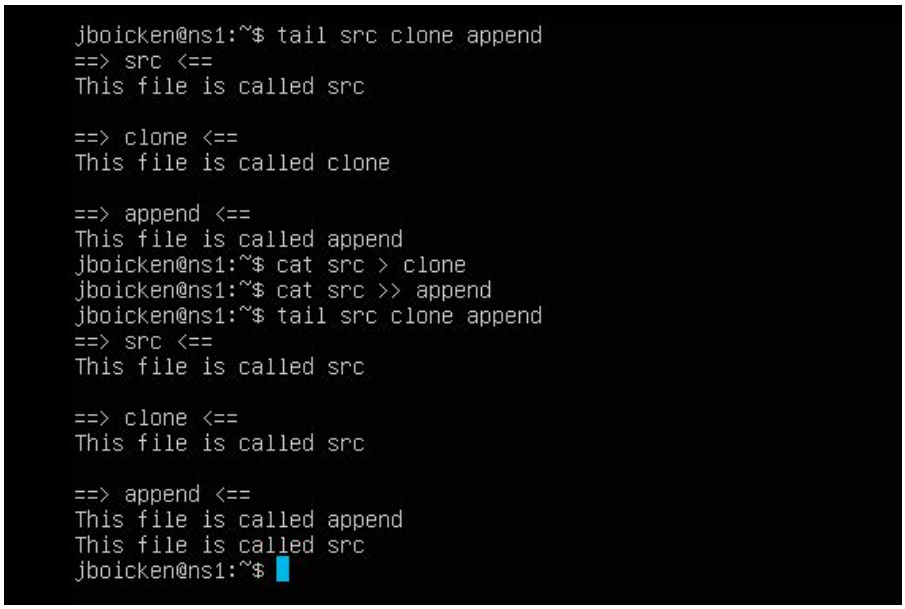
Lab Report

1. **Screenshot of `cd`/`ls` practice.**



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

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.



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 final. 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.)

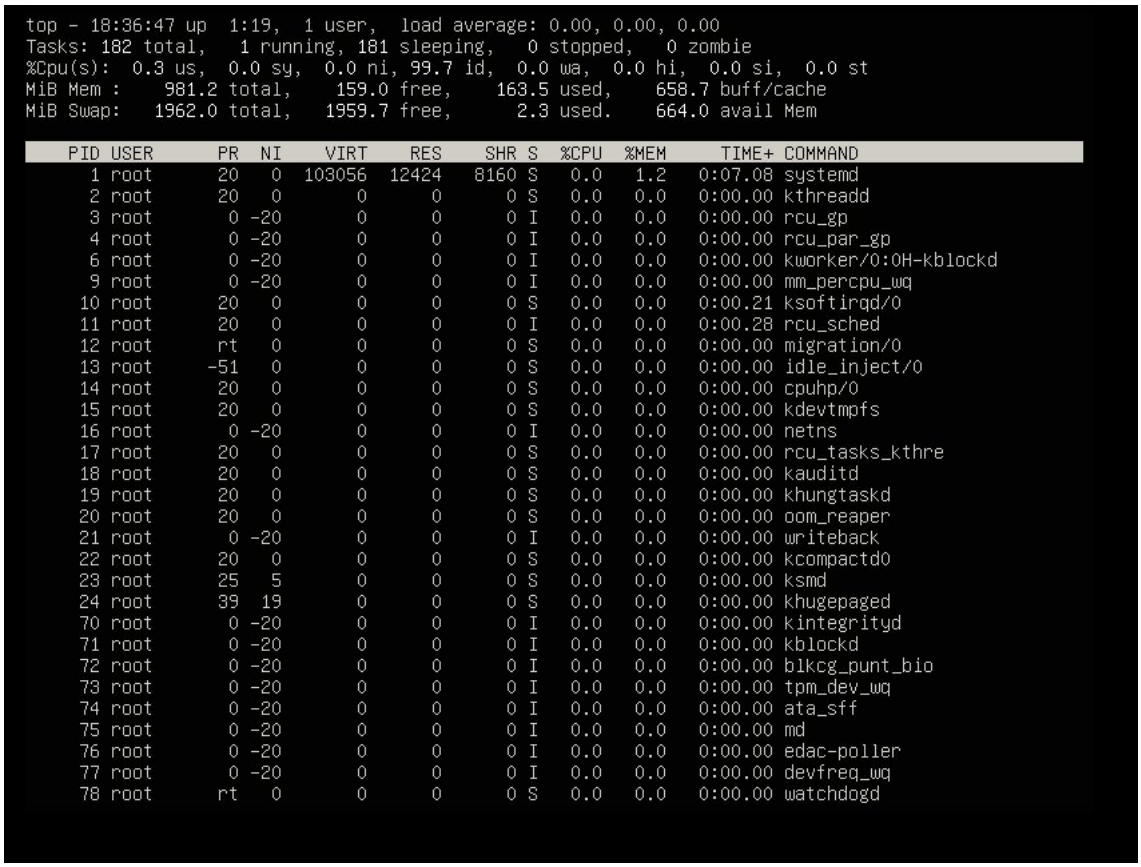
6. **Screenshot of last 10 lines of /var/log/auth.log using one of the commands learned in this lab.**

```
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/ca
t /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/ca
t /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/pa
sswd -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.



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)