Networking & System Administration Lab

Assignment-2

Basic Linux Commands

Submitted to:
Meera Miss

Submitted by :
Vimal Thomson
Roll No.: 40
RMCA-B

Basic Linux Commands

1. **echo**

echo command is used to move some data into a file.

If you want to add the text, “Hello, my name is John” into a file called name.txt,
you would type echo Hello, my name is John >> name.txt

```
amalthomson@amalthomson:~/Desktop$ echo Hello, my name is Vimal >> name.txt
amalthomson@amalthomson:~/Desktop$
```

2. **head**

The head command is used to view the first lines of any text file.

By default, it will show the first ten lines, but you can change this number to your
liking.

If you only want to show the first five lines, type head -n 5 filename.txt

```
amalthomson@amalthomson:~/Desktop$ head -n 5 name.txt
Hello, my name is Vimal1. Pather Panchali (1955)        8.5
2. Nayakan (1987)       8.5
3. Pariyerum Perumal (2018)     8.5
4. Anbe Sivam (2003)    8.5
5. Hanky Panky (1979)   8.5
amalthomson@amalthomson:~/Desktop$
```

3. **tail**

This one has a similar function to the head command, but instead of showing the
first lines, the tail command will display the last ten lines of a text file.

tail -n filename.txt

```
amalthomson@amalthomson:~/Desktop$ tail -n 5 name.txt
16. Visaaranai (2015)   8.4
17. 3 Idiots (2009)     8.3
18. Like Stars on Earth (2007)  8.3
19. Jersey (2019)       8.3
20. Soorarai Pottru (2020)      8.3
amalthomson@amalthomson:~/Desktop$
```

4. **read**

read the contents of a line into a variable.
The read command can be used with and without arguments
read command is used to read [options] [name...]
$read
$read var1 var2 var3
$echo "[$var1] [$var2] [$var3]“

```
amalthomson@amalthomson:~/Desktop$ read name
Vimal Thomson
amalthomson@amalthomson:~/Desktop$ echo [$name]
[Vimal Thomson]
amalthomson@amalthomson:~/Desktop$
```

5. **more**

• Like cat command, more command displays the content of a file. Only difference is that, in case of larger files, 'cat' command output will scroll off your screen while 'more' command displays output one screenful at a time.

Enter key: To scroll down page line by line.

Space bar: To go to next page.

b key: To go to the backward page.

/ key: Lets you search the string.

Syntax: more <file name> more /etc/passwd

```
amalthomson@amalthomson:~/Desktop$ more /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologi
n
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/n
ologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbi
--More--(40%)
```

6. **less**

The 'less' command is same as 'more' command but include some more features.

It automatically adjust with the width and height of the teminal window, while 'more'
command cuts the content as the width of the terminal window get shorter.

less <file name>

$less /etc/passwd

```
amalthomson@amalthomson:~/Desktop$ less /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologi
n
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/n
ologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbi
/etc/passwd
```

7. **cut**

The cut command is used for cutting out the sections from each line of files and writingthe result to standard output. It can be used to cut parts of a line by byte position,
character and field

cut OPTION... [FILE]...

$cut -b 1,2,3 state.txt

```
amalthomson@amalthomson:~/Desktop$ cut -b 1,2,3 name.txt
Hel
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
amalthomson@amalthomson:~/Desktop$
```

8. **paste**

It is used to join files horizontally (parallel merging) by outputting linesconsisting of lines from each file specified, separated by tab as delimiter, to thestandard output.

paste [OPTION]... [FILES]...

$ paste state.txt capital.txt

```
amalthomson@amalthomson:~/Desktop$ paste name1.txt name.txt
hai!!!!!!!!!!!!!!!!!!!!!!!!    Hello, my name is Vimal1. Pather Panchali (1955)
8.5
        2. Nayakan (1987)       8.5
        3. Pariyerum Perumal (2018)     8.5
        4. Anbe Sivam (2003)    8.5
        5. Hanky Panky (1979)   8.5
        6. C/o Kancharapalem (2018)     8.5
        7. The World of Apu (1959)      8.5
        8. Kireedam (1989)      8.4
        9. Manichitrathazhu (1993)      8.4
        10. Natsamrat (2016)    8.4
        11. 96 (2018)   8.4
        12. Thevar Magan (1992) 8.4
        13. Black Friday (2004) 8.4
        14. Kumbalangi Nights (2019)    8.4
        15. Drishyam 2 (2021)   8.4
        16. Visaaranai (2015)   8.4
        17. 3 Idiots (2009)     8.3
        18. Like Stars on Earth (2007)  8.3
        19. Jersey (2019)       8.3
        20. Soorarai Pottru (2020)      8.3
amalthomson@amalthomson:~/Desktop$
```

9. **uname**

The uname command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on.

$uname

$uname -r

```
amalthomson@amalthomson:~/Desktop$ uname
Linux
amalthomson@amalthomson:~/Desktop$ uname -r
5.8.0-55-generic
amalthomson@amalthomson:~/Desktop$
```

10. **cp**

cp command is used to copy files from the current directory to a different directory. For instance, the command cp scenery.jpg /home/username/Pictures would create a copy of scenery.jpg (from your current directory) into the Pictures directory.

cp -i will ask for user's consent in case of a potential file overwrite.

cp -p will preserve source files' mode, ownership and timestamp.

cp -r will copy directories recursively.

cp -u copies files only if the destination file is not existing or the source file is newer than the destination file.

```
amalthomson@amalthomson:~/Desktop$ sudo cp name1.txt /Downloads
[sudo] password for amalthomson:
amalthomson@amalthomson:~/Desktop$
```

11. **mv**

The primary use of the mv command is to move files, it can also be used to rename files. The arguments in mv are similar to the cp command. You need to type mv, the file's name, and the destination's directory.

mv file.txt /home/username/Documents

To rename files, the Linux is mv oldname.ext newname.ext

```
amalthomson@amalthomson:~/Desktop$ sudo mv name1.txt /Downloads
amalthomson@amalthomson:~/Desktop$
```

12. **locate**

To locate a file, just like the search command in Windows.

What's more, using the -i argument along with this command will make it case-insensitive, so you can search for a file even if you don't remember its exact name.

To search for a file that contains two or more words, use an asterisk (*).

For example, locate -i school*note command will search for any file that contains the word "school" and "note", whether it is uppercase or lowercase.

```
amalthomson@amalthomson:~/Desktop$ locate name.txt
/home/amalthomson/Desktop/name.txt
/usr/share/doc/syslinux-common/asciidoc/com-name.txt
amalthomson@amalthomson:~/Desktop$
```

13. **find**

Similar to the locate command, using find also searches for files and directories.

The difference is, you use the find command to locate files within a given directory.

As an example, find /home/ -name notes.txt command will search for a file called notes.txt within the home directory and its subdirectories.

Other variations when using the find are:

To find files in the current directory use, find .-name.txt

To look for directories use, / -type d -name. Txt

```
amalthomson@amalthomson:~/Desktop$ find name.txt
name.txt
amalthomson@amalthomson:~/Desktop$
```

14. **grep**

Another basic Linux command that is undoubtedly helpful for everyday use is grep. It lets you search through all the text in a given file. To illustrate, grep blue notepad.txt will search for the word blue in the notepad file.

Lines that contain the searched word will be displayed fully. Usually output of a previous command is piped into the grep command. For example ls -l | grep "kernel"

```
amalthomson@amalthomson:~/Desktop$ grep Vimal name.txt
Hello, my name is Vimal1. Pather Panchali (1955)        8.5
amalthomson@amalthomson:~/Desktop$
```

15. **df**

Use df command to get a report on the system's disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type df -m.

```
amalthomson@amalthomson:~/Desktop$ df -m
Filesystem     1M-blocks  Used Available Use% Mounted on
udev                3912     0      3912   0% /dev
tmpfs                789     2       787   1% /run
/dev/sda5         467922 34591    409494   8% /
tmpfs               3941     0      3941   0% /dev/shm
tmpfs                  5     1         5   1% /run/lock
tmpfs               3941     0      3941   0% /sys/fs/cgroup
/dev/loop1            65    65         0 100% /snap/caprine/48
/dev/loop7            62    62         0 100% /snap/core20/1026
/dev/loop3            56    56         0 100% /snap/core18/2066
/dev/loop2             2     2         0 100% /snap/fast/4
/dev/loop8           160   160         0 100% /snap/signal-desktop/354
/dev/loop9            62    62         0 100% /snap/core20/975
/dev/loop10            9     9         0 100% /snap/pdfmixtool/862
/dev/loop11           66    66         0 100% /snap/gtk-common-themes/1515
/dev/loop13           65    65         0 100% /snap/gtk-common-themes/1514
/dev/loop0           100   100         0 100% /snap/core/11167
/dev/loop12           56    56         0 100% /snap/core18/1997
/dev/loop6           713   713         0 100% /snap/intellij-idea-community/302
/dev/loop14          167   167         0 100% /snap/signal-desktop/358
/dev/loop15           51    51         0 100% /snap/snap-store/542
/dev/loop4           219   219         0 100% /snap/gnome-3-34-1804/66
/dev/loop19           56    56         0 100% /snap/teams-for-linux/156
```

16. **du**

If you want to check how much space a file or a directory takes, the du (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format.

If you want to see it in bytes, kilobytes, and megabytes, add the -h argument to the command line.

$du -h

```
amalthomson@amalthomson:~/Desktop$ du -h
24K     ./.idea
12K     ./src
12K     ./out/production/java-progs
16K     ./out/production
20K     ./out
132K    .
amalthomson@amalthomson:~/Desktop$
```

17. **useradd**

This is available only to system admins

Since Linux is a multi-user system, this means more than one person can interact with the same system at the same time.

useradd is used to create a new user, while passwd is adding a password to that user's account. To add a new person named John type, useradd John and then to add his password type, passwd 123456789

```
amalthomson@amalthomson:~/Desktop$ sudo useradd VimalT
useradd: user 'VimalT' already exists
amalthomson@amalthomson:~/Desktop$
```

18. **userdel**

Remove a user is very similar to adding a new user. To delete the users account type, userdel UserName

```
amalthomson@amalthomson:~/Desktop$ sudo userdel VimalT
amalthomson@amalthomson:~/Desktop$ sudo userdel VimalT
userdel: user 'VimalT' does not exist
amalthomson@amalthomson:~/Desktop$
```

19. **sudo**

Short for "SuperUser Do", this command enables you to perform tasks that require administrative or root permissions. You must have sufficient permissions to use this command.

sudo useradd Vimal

```
amalthomson@amalthomson:~/Desktop$ sudo useradd VimalT
amalthomson@amalthomson:~/Desktop$ sudo useradd VimalT
useradd: user 'VimalT' already exists
```

20. passwd

Changes passwords for user accounts.

A normal user may only change the password for their own account, while thesuperuser may change the password for any account.

passwd[option] [username]

passwd

passwd user1

```
amalthomson@amalthomson:~/Desktop$ passwd amalthomson
Changing password for amalthomson.
Current password:
New password:
Retype new password:
Password unchanged
```