Advanced Computer Network Assignment

Topic: - Take screenshots of basic Linux commands II

Submitted By:
Anilect Jose
Roll No:17
RMCA A

Submitted To:
Rini Kurian

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**

```
kaj@kaj-VirtualBox:~/Documents$ echo My name is Anilect>>name.txt
kaj@kaj-VirtualBox:~/Documents$ ls
abcc.sh  abc.sh  files  name.txt
kaj@kaj-VirtualBox:~/Documents$
```

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**

```
kaj@kaj-VirtualBox:~/Documents$ head -n 5 name.txt
1. 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
kaj@kaj-VirtualBox:~/Documents$ head -n 10 name.txt
1. 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
```

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 5 filename.txt**

```
kaj@kaj-VirtualBox:~/Documents$ 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
kaj@kaj-VirtualBox:~/Documents$
```

4. Read

Read the contents of a line into a variable.

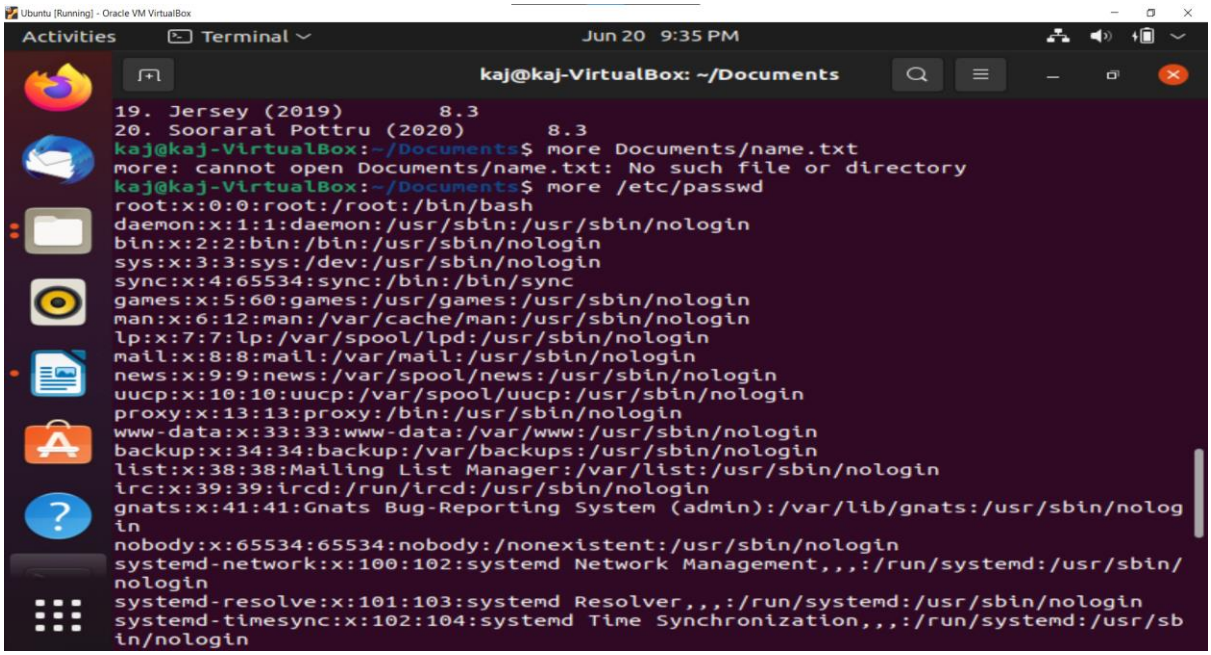
- The **read** command can be used with and without arguments.
- **Read** command is used to read [options] [name...]
    - $read
    - $read var1 var2 var3
    - $echo"[$var1] [$var2] [$var3] “

```
kaj@kaj-VirtualBox:~/Documents$ read name;
Anilect
kaj@kaj-VirtualBox:~/Documents$ read name2;
Jose
kaj@kaj-VirtualBox:~/Documents$ echo $name $name2;
Anilect Jose
kaj@kaj-VirtualBox:~/Documents$
```

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.

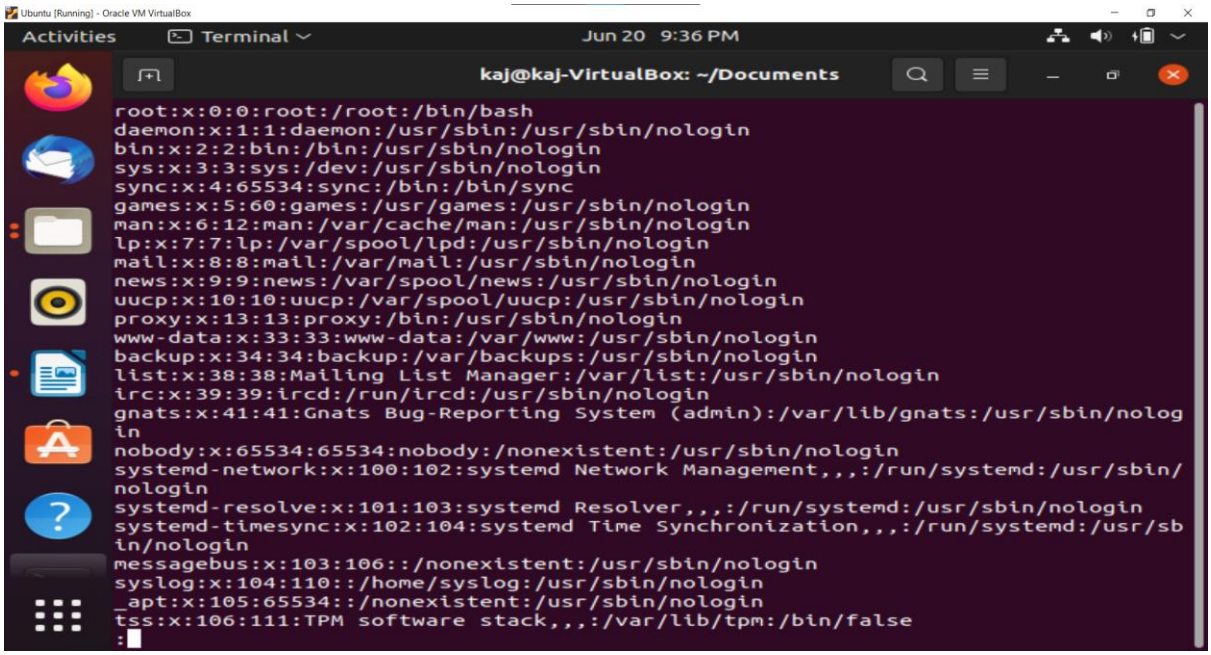
- Enter key: To scroll down page line by line.
- Space bar: To go to next page
- b key: To go to the backward page
- / key: Lets you search the string
- Syntax: more <file name>
- **more /etc/passwd**



6. less

The 'less' command is same as 'more' command but include some more features. It automatically adjusts with the width and height of the terminal window, while 'more' command cuts the content as the width of the terminal window get shorter

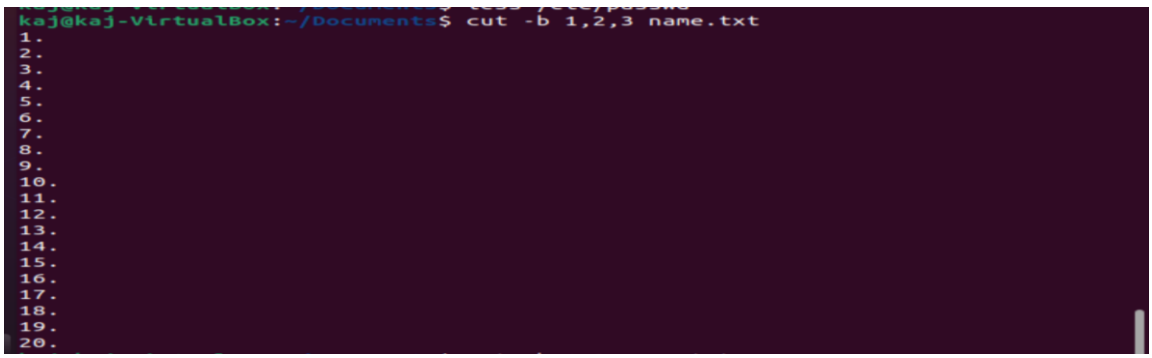
- **less <file name>**
- **less etc/passwd**



7. cut

The cut command is used for cutting out the sections from each line of files and writing the 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**



8. paste

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

- **paste [OPTION] ... [FILES] …**
- **$paste state.txt capital.txt**

```
kaj@kaj-VirtualBox:~/Documents/files$ cat >newfile.txt
kaj@kaj-VirtualBox:~/Documents/files$ ls
file1.txt  file2.txt  file3.txt  newfile.txt
kaj@kaj-VirtualBox:~/Documents/files$ paste file2.txt newfile.txt
Also I'm from AJCE
```

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**

```
kaj@kaj-VirtualBox:~/Documents/files$ uname
Linux
kaj@kaj-VirtualBox:~/Documents/files$ uname -r
5.11.0-18-generic
kaj@kaj-VirtualBox:~/Documents/files$
```

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.

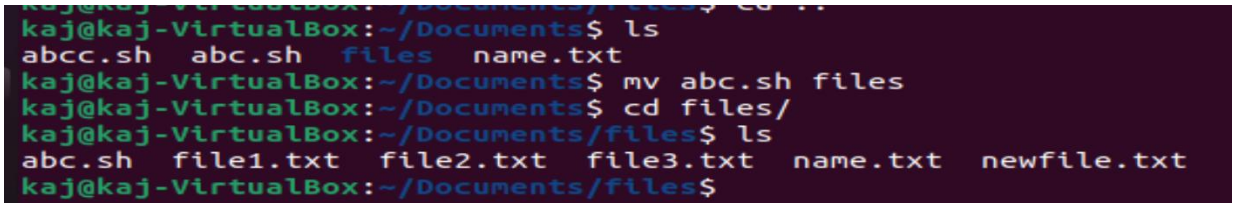
- **cp -i** will ask for user's consent in case of a potential file overwrite
- **cp -p** will preserve source files' mode, ownership and timestamp
- **cp -r** will copy directories recursively
- **cp -u** copies files only if the destination file is not existing or the source file is newer
- than the destination file

```
kaj@kaj-VirtualBox:~/Documents$ cp name.txt files
kaj@kaj-VirtualBox:~/Documents$ ls
abcc.sh  abc.sh  files  name.txt
kaj@kaj-VirtualBox:~/Documents$ cd files
kaj@kaj-VirtualBox:~/Documents/files$ ls
file1.txt  file2.txt  file3.txt  name.txt  newfile.txt
kaj@kaj-VirtualBox:~/Documents/files$
```

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



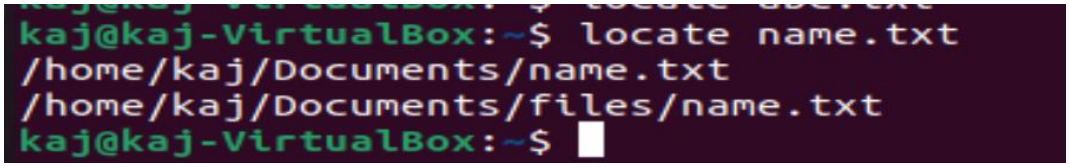
12. locate

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.



13. find

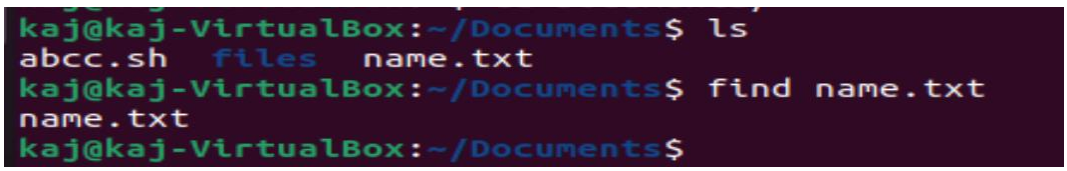
Similar to the locate command, using find also searches for files and directories.

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

As an example, **find / 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 notes txt
- To look for directories use, type d name notes txt



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".**

```
kaj@kaj-VirtualBox:~/Documents$ cd files
kaj@kaj-VirtualBox:~/Documents/files$ grep Anilect file3.txt
I'm Anilect Jose
kaj@kaj-VirtualBox:~/Documents/files$
```

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**

```
kaj@kaj-VirtualBox:~/Documents/files$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
tmpfs              99556     1412     98144   2% /run
/dev/sda3        9735476  7798920   1422304  85% /
tmpfs             497772        0    497772   0% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
tmpfs               4096        0      4096   0% /sys/fs/cgroup
/dev/sda2         524252     5340    518912   2% /boot/efi
tmpfs              99552      136     99416   1% /run/user/1000
kaj@kaj-VirtualBox:~/Documents/files$
```

16. ds

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

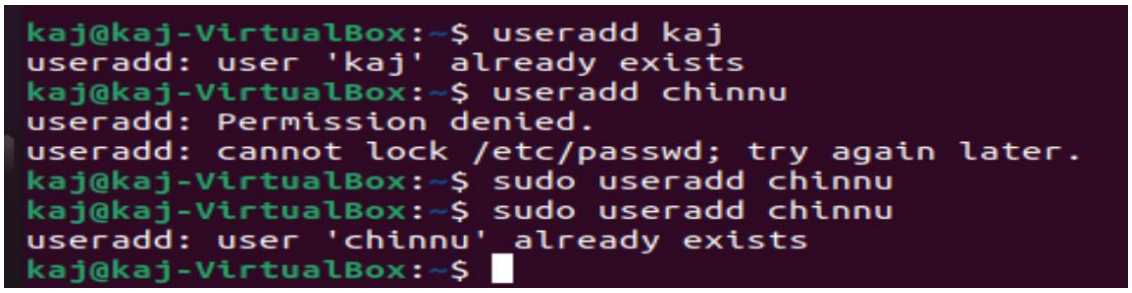
```
kaj@kaj-VirtualBox:~$ du
4       ./.ssh
4       ./Desktop
4       ./Videos
4       ./Downloads
24      ./Documents/files
36      ./Documents
12      ./.gnupg
4       ./Public
8       ./.config/goa-1.0
32      ./.config/evolution/sources
36      ./.config/evolution
4       ./.config/gnome-session/saved-session
8       ./.config/gnome-session
16      ./.config/ibus/bus
20      ./.config/ibus
8       ./.config/gedit
12      ./.config/dconf
4       ./.config/enchant
84      ./.config/pulse
8       ./.config/gtk-3.0
4       ./.config/nautilus
4       ./.config/libreoffice/4/user/autocorr
4       ./.config/libreoffice/4/user/extensions/tmp/extensions
4       ./.config/libreoffice/4/user/extensions/tmp/registry/com.sun.star.comp.
deployment.executable.PackageRegistryBackend
```

17.useradd

This is available only to system admins.

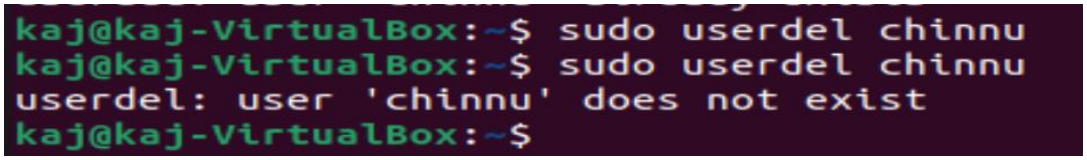
Since Linux is a multi-user system, this means more than one person can interact with the same system at the same time.

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**



18. userdel

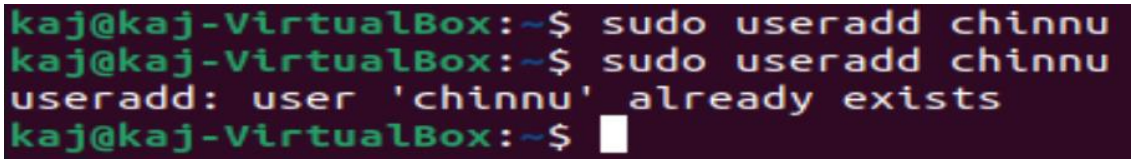
R emove a user is very similar to adding a new user To delete the users account type, **userdel UserName**



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 maria



20. passwd

Changes passwords for user accounts

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

passwd[option] [username]

passwd

passwd user 1