**wc:**

wc stands for word count.

- Used for counting purpose.
- It is used to find out number of lines, word count, byte and characters count in the files specified in the file arguments.
- #wc state.txt 6 8 54 state.txt
- #wc state.txt capital.txt
- wc -l state.txt
- wc -w state.txt capital.txt
- wc -c state.txt
- wc -m state.txt

```
┌──(raman㉿kali)-[~]
└─$ cat myfile.txt
hello everyone
hope u all safe
stay home stay safe

┌──(raman㉿kali)-[~]
└─$ cat myfile2.txt
hai hello

┌──(raman㉿kali)-[~]
└─$ wc myfile.txt
 3 10 52 myfile.txt

┌──(raman㉿kali)-[~]
└─$ wc myfile2.txt
 1  2 10 myfile2.txt

┌──(raman㉿kali)-[~]
└─$ wc -l myfile.txt
3 myfile.txt

┌──(raman㉿kali)-[~]
└─$ wc -w myfile.txt myfile2.txt
10 myfile.txt
 2 myfile2.txt
12 total
```

**Tar:**

• The Linux ‘tar’stands for tape archive, is used to create Archive and extract the Archive files • Linux tar command to create compressed or uncompressed Archive files • Options:

- -c : Creates Archive
- -x : Extract the archive
- -f : creates archive with given filename
- -t: displays or lists files in archived file
- -u: archives and adds to an existing archive file
- -v: Displays Verbose Information
- -A : Concatenates the archive files
- -z : zip, tells tar command that creates tar file using gzip

- -j : filter archive tar file using tbzip
- -W : Verify a archive file
- -r : update or add file or directory in already existed .tar file

#tar cf archive.tar state.txt capital.txt //create archive file

#ls archive.tar

#tar tf /archive.tar // list contents of tar archive file

- Extract an archive created with tar

#mkdir backup

#cd backup

#tar xf/home/meera/Documents/Meera_Linux/archive.tar

- **Compression Types**

gzip(z),bzip2(j), xz(J) #tar czf /abc.tar.gz /etc

#tar cjf /abcd.tar.bz2 /etc

**#tar cJf /abcde.tar.xz /etc**

- **Extract an archive**

#mkdir backup1

#cd backup1

#tar xzf /abc.tar.gz

#mkdir backup2

#cd backup2

#tar xjf /abcd.tar.bz2

#mkdir backup3

#cd backup3

#tar xJf /abcde.tar.xz

- **Bzip2**


- **gzip**

```
┌──(raman㉿kali)-[~]
└─$ ls
Desktop    f1.txt  f3.txt.gz  Music        Pictures   Videos
Documents  f2.txt  f3.txt.xz  myfile2.txt  Public
Downloads  f3.txt  f4.txt     myfile.txt   Templates

┌──(raman㉿kali)-[~]
└─$ gzip f1.txt f2.txt

┌──(raman㉿kali)-[~]
└─$ ls
Desktop    f1.txt.gz  f3.txt.gz  Music        Pictures   Videos
Documents  f2.txt.gz  f3.txt.xz  myfile2.txt  Public
Downloads  f3.txt     f4.txt     myfile.txt   Templates

┌──(raman㉿kali)-[~]
└─$ gzip -c f1.txt > f2.txt
gzip: f1.txt: No such file or directory

┌──(raman㉿kali)-[~]
└─$ gzip -c f1.txt.gz                                                    1
L�af1.txt.gz������iKd`N3�+�(a`f�,\

┌──(raman㉿kali)-[~]
└─$ gzip -d f2.txt.gz
gzip: f2.txt already exists; do you wish to overwrite (y or n)? y
```

- **xz**

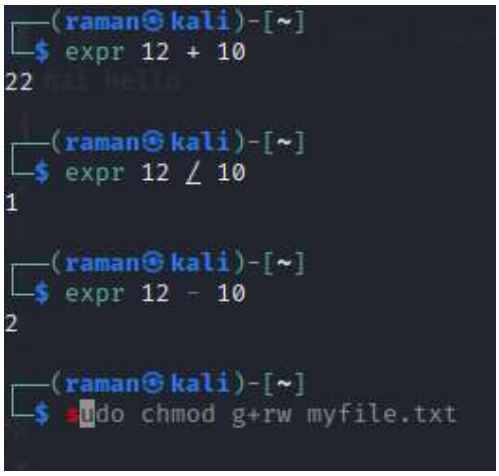
```
┌──(raman㉿kali)-[~]
└─$ ls
Desktop    f1.txt.gz  f3.txt.gz  Music        Pictures   Videos
Documents  f2.txt     f3.txt.xz  myfile2.txt  Public
Downloads  f3.txt     f4.txt     myfile.txt   Templates

┌──(raman㉿kali)-[~]
└─$ xz f2.txt

┌──(raman㉿kali)-[~]
└─$ xz -k f3.txt
xz: f3.txt.xz: File exists

┌──(raman㉿kali)-[~]
└─$ xz -k f4.txt                                                         1

┌──(raman㉿kali)-[~]
└─$ xz -c f3.txt > f3.txt.gz

┌──(raman㉿kali)-[~]
└─$ ls
Desktop    f1.txt.gz  f3.txt.gz  f4.txt.xz    myfile.txt  Templates
Documents  f2.txt.xz  f3.txt.xz  Music        Pictures    Videos
Downloads  f3.txt     f4.txt     myfile2.txt  Public

┌──(raman㉿kali)-[~]
```

- **expr**

The expr command evaluates a given expression and displays its corresponding output. It is used for:

- Basic operations like addition, subtraction, multiplication, division, and modulus on integers.
- Evaluating regular expressions, string operations like substring, length of strings etc.
- Performing operations on variables inside a shell script

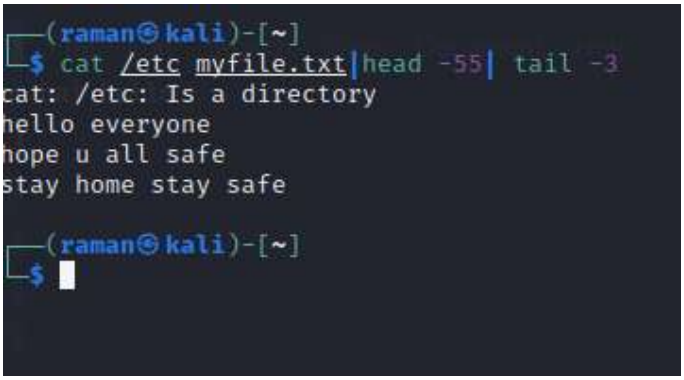
#expr 10 + 2




- **Redirections & Piping :**
  - A pipe is a form of redirection to send the output of one command/program/process to another command/program/process for further processing.
  - Pipe is used to combine two or more commands, the output of one command acts as input to another command, and this command's output may act as input to the next command and so on.

#ls -l | wc - l

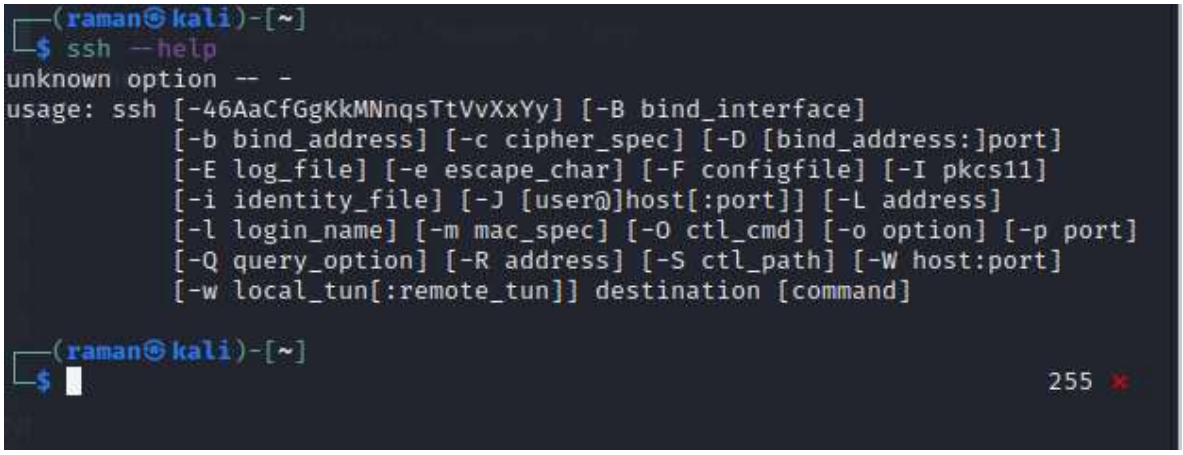
#cat /etc.passwd.txt | head -7 | tail -5




- **ssh**
  - *ssh*stands for **"Secure Shell"**.
  - It is a protocol used to securely connect to a remote server/system.
  - ssh is secure in the sense that it transfers the data in encrypted form between the host and the client.
  - It transfers inputs from the client to the host and relays back the output. ssh runs at TCP/IP port 22.

#ssh user_name@host(IP/Domain_name)

#ssh –X root@server1.example.com

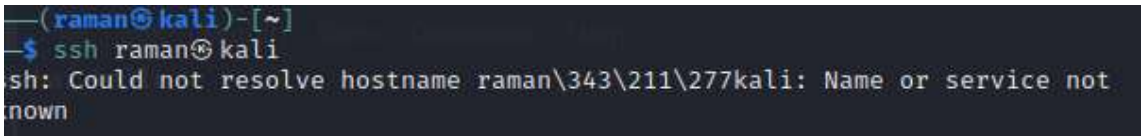



- **scp**
  - SCP (secure copy) is a command-line utility that allows you to securely
  - copy files and directories between two locations.
  - With scp, you can copy a file or directory:
  - From your local system to a remote system.
  - From a remote system to your local system.
  - Between two remote systems from your local system.
  - Remote file system locations are specified in format
  - [user@]host:/path Syntax:

scp [OPTION] [user@]SRC_HOST:]file1 [user@]DEST_HOST:]file2

$scp/etc/yum.config/etc/hosts ServerX:/home/student

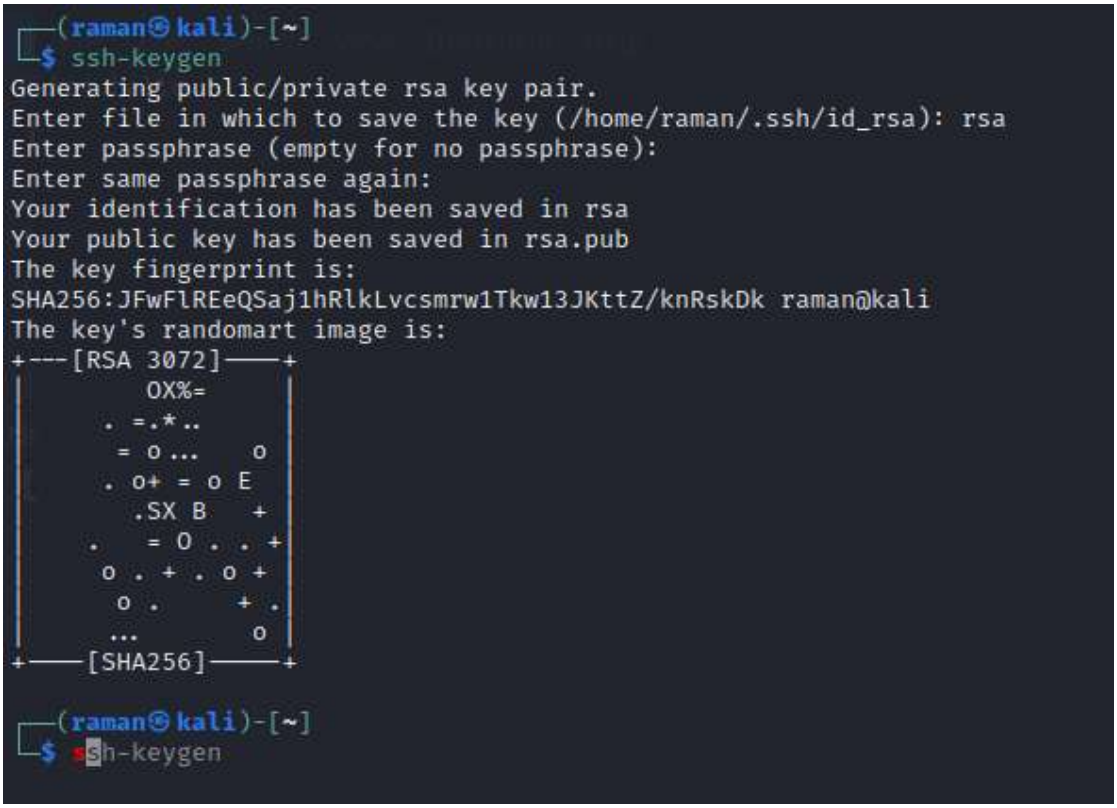
$scp ServerX:/etc/hostname /home/student




- **ssh-keygen**

ssh-keygen command to generate a public/private authentication key pair. Authentication keys allow a user to connect to a remote system without supplying a password. Keys must be generated for each user separately. If you generate key pairs as the root user, only the root can use the keys.

$ssh-keygen -t rsa




- **ssh-copy-id**
  - The ssh-copy-id command allows you to install an SSH key on a remote server's authorized keys.
  - This command facilitates SSH key login, which removes the need for a password for each login, thus ensuring a password-less, automatic login process.
  - $ssh-copy-id username@remote_host