4. Filter & IO redirectioncommand.

Grep

grep command is used to find texts from any text input.

Passwd file: stores information about all the users in the

system

```
imran@DevOps:~/linux-practices$ cat /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
```

➔ Finding line which contains word as “root” from /etc/passwd file.

```
imran@DevOps:~/linux-practices$ grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
imran@DevOps:~/linux-practices$
```

➔ Linux is case sensetive, Root is diffrent that root. Ignoring case in grep with -i option.

```
imran@DevOps:~/linux-practices$ grep Root /etc/passwd
imran@DevOps:~/linux-practices$ grep -i Root /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

➔ To display things except the given word use -v option

Filter Commands

- **less: Displays file content page wise or line wise. Ex: less /etc/passwd**

Note: -press **Enter** key to scroll down line by line (or)
Use **d** to go to next page
Use **b** to go to previous page
Use **/** to search for a word in the file
Use **v** to go vi mode where you can edit the file and once you save it you will back to less command

- **more**

more is exactly same like **less**
Ex: #more /etc/passwd
Note: -press **Enter** key to scroll down line by line (or)
Use **d** to go to next page
Use **/** to search for a word in the file
Use **v** to go vi mode where you can edit the file and once you save it you will back to more command

- **head**

It is used to display the top **10 lines** of the file.
Ex:# head /etc/passwd

```
[root@ktlinux ~]# head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
```

- **tail**

It is used to display the **last 10** lines of the file
#tail /etc/passwd

```
[root@ktlinux ~]# tail /etc/passwd
apache:x:48:48:Apache:/var/www:/sbin/nologin
nslcd:x:65:55:LDAP Client User:/:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
pulse:x:496:494:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
visitor:x:500:500:visitor:/home/visitor:/bin/bash
ktuser:x:501:501::/home/ktuser:/bin/bash
```

- **cut**

 # cut -d -f filename (where d stands for delimiter ex. : , " " etc and f stands for field)

```
[root@ktlinux ~]# cut -d: -f1 /etc/passwd
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
uucp
```

To delimit spaces and print the field
#cut –d " " –f1 filename

- **sed**

sed stands for **stream editor**, which is used to search a word in the file and replace it with the word required to be in the output

Note: it will only modify the output, but there will be no change in the original file.

#sed 's/searchfor/replacewith/g' filename

```
[root@ktlinux ~]# cat ktfile
Welcome to Kernel Tech
[root@ktlinux ~]# sed 's/Tech/Technologies/g' ktfile
Welcome to Kernel Technologies
[root@ktlinux ~]# cat ktfile
Welcome to Kernel Tech
```

I/O redirection

Redirection is a process where we can copy the output of any command(s), file(s) into a new file. There are two ways of redirecting the output into a file.
Using **>** or **>>** **filename** after the command, and

➔ Create a file named devopstools with below content.

```
imran@DevOps:~/linux-practices$ cat devopstools
chef tech
ansible tech
git tech
docker tech
aws tech
```

➔ Search for text “tech” replace it with “tools” and redirect output to a new file.

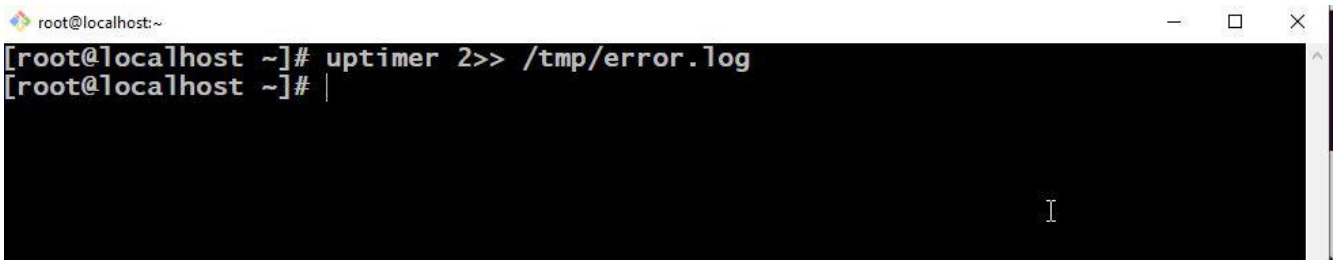
```
imran@DevOps:~/linux-practices$ sed 's/tech/tools/g' devopstools
chef tools
ansible tools
git tools
docker tools
aws tools
imran@DevOps:~/linux-practices$ sed 's/tech/tools/g' devopstools > newtools.txt
imran@DevOps:~/linux-practices$ cat newtools.txt
chef tools
ansible tools
git tools
docker tools
aws tools
```

Note: if the given name of the file is not available a new file will be created automatically. If the file already exists then it will overwrite contents of that file.

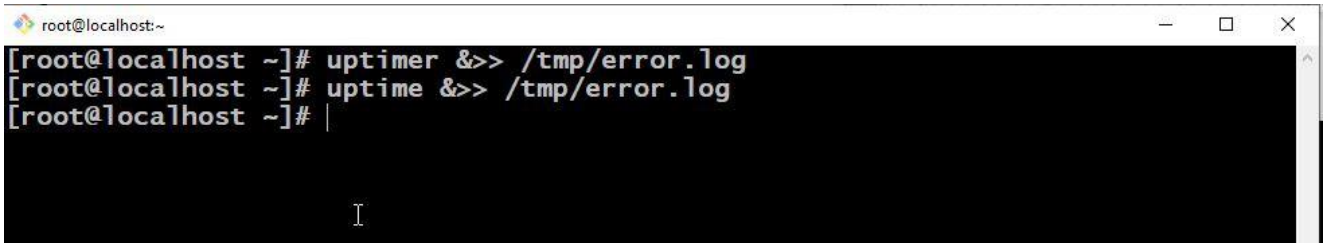
➔ Appending another output in same file with “>>” .

```
imran@DevOps:~/linux-practices$ tail /etc/passwd >> newtools.txt
imran@DevOps:~/linux-practices$ cat newtools.txt
chef tools
ansible tools
git tools
docker tools
aws tools
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
imran:x:1000:1000:Imran,,,:/home/imran:/bin/bash
jenkins:x:121:131:Jenkins,,,:/var/lib/jenkins:/bin/bash
guest-lx1wni:x:999:999:Guest:/tmp/guest-lx1wni:/bin/bash
guests:x:1001:1001:,,,:/home/guests:/bin/bash
nvidia-persistenced:x:122:132:NVIDIA Persistence Daemon,,,:/:/sbin/nologin
guest-yjz1gk:x:998:998:Guest:/tmp/guest-yjz1gk:/bin/bash
imran@DevOps:~/linux-practices$
```

➔ **Redirecting only error to a file “2>>”.**



➔ **Redirecting all the output to a file “&>>”.**



Piping

So far we've dealt with sending data to and from files. Now we'll take a look at a mechanism for sending data from one program to another. It's called piping and the operator we use is (|). What this operator does is feed the output from the program on the left as input to the program on the right.

```
imran@DevOps:~$ cd linux-practices/
imran@DevOps:~/linux-practices$ ls
chefdk_1.2.22-1_amd64.deb  file2          logdir        tree_1.7.0-3_amd64.deb
devopstools                firstfile.txt  newtools.txt  vpdir
imran@DevOps:~/linux-practices$ ls | head -3
chefdk_1.2.22-1_amd64.deb
devopstools
file2
imran@DevOps:~/linux-practices$ ls | grep logdir
logdir
imran@DevOps:~/linux-practices$ cat /etc/passwd | grep root
root:x:0:0:root:/root:/bin/bash
imran@DevOps:~/linux-practices$
```

Find

find command is used to find the files or directory's path, it is exactly like the find option in windows where you can search for a file.

```
imran@DevOps:~/linux-practices$ find /home/imran/ -name newtools.txt
/home/imran/linux-practices/newtools.txt
```

Options that can be used with find command:

Option	Usage
-name	For searching a file with its name
-inum	For searching a file with particular inode number
-type	For searching a particular type of file
-user	For files whose owner is a particular user
-group	For files belonging to particular group