Linux Create user

[root@server ~]#

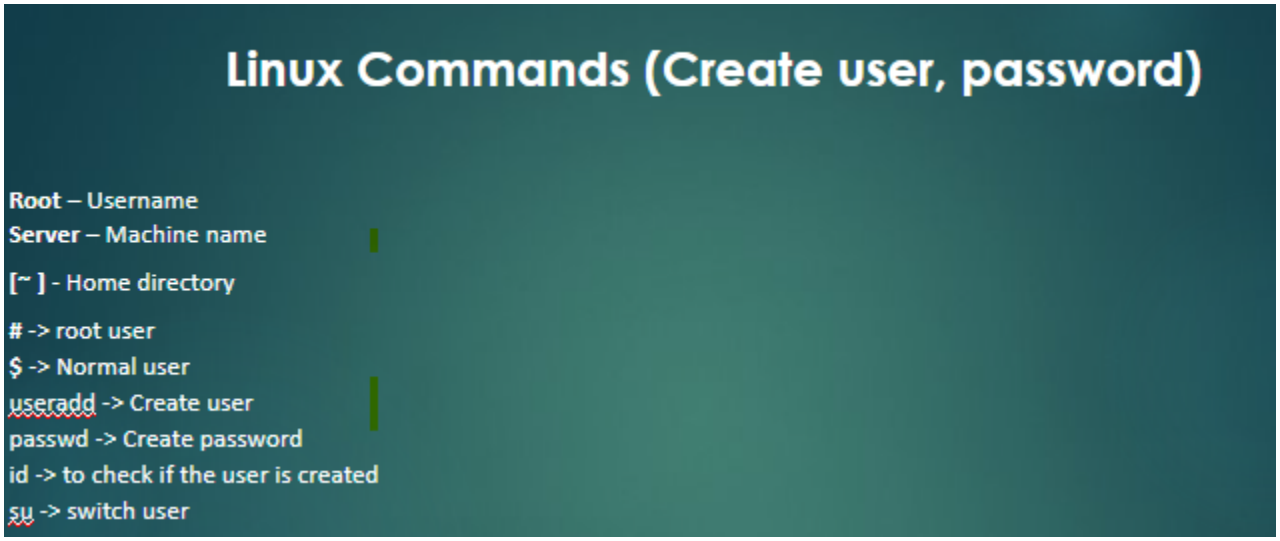
[username@hostname [home_directory]]

Shell:-

It Acts as an interpreter /mediator between user and Kernel.

Whoami:-

[root@server ~]# whoami ## displays the logged-in user name
root



Useradd:-

1 .[root@server ~]# useradd user1
[root@server ~]# id user1
uid=2007(user1) gid=2007(user1) groups=2007(user1)

2 .[root@server ~]# useradd user2
[root@server ~]# id user2
uid=2008(user2) gid=2008(user2) groups=New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@server ~]#

Ni

Switching Between Users:-

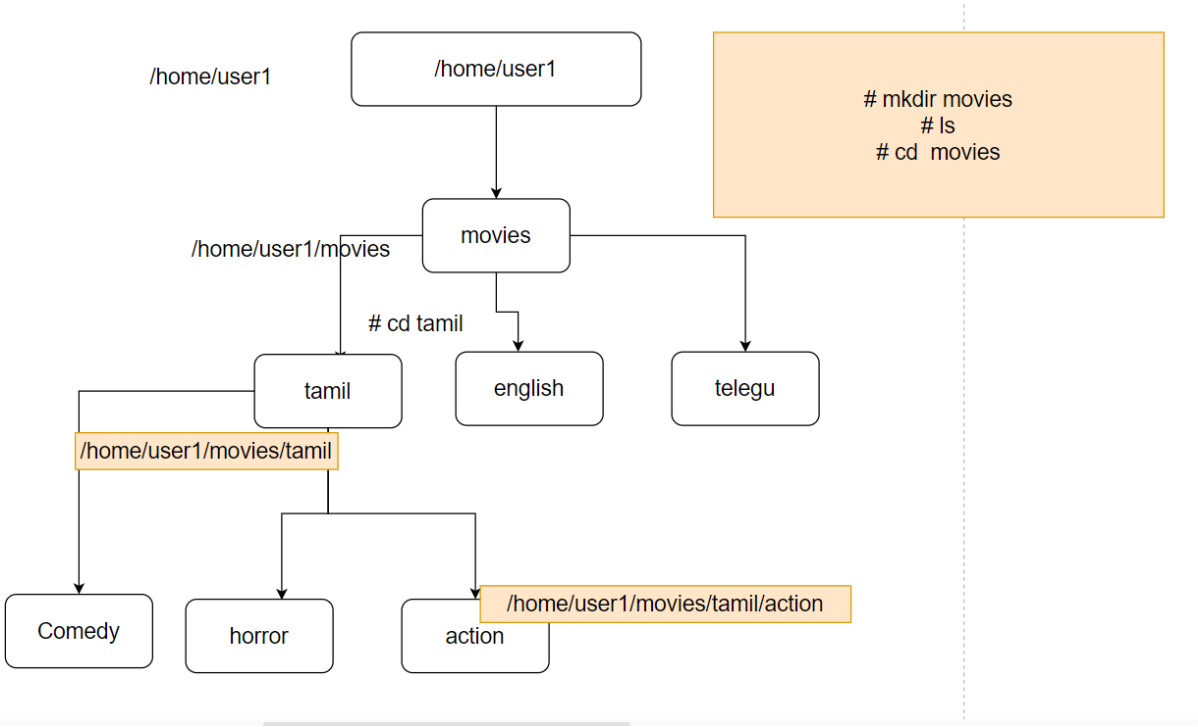
```
# from root switching to user1

[root@server ~]# su - user1
[user1@server ~]$ whoami
user1
[user1@server ~]$

2008(user2)
[root@server ~]#


[root@server ~]# passwd user1
Changing password for user user1.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.


[root@server ~]# passwd user2
Changing password for user user2.

# from user1 switch to user2
[user1@server ~]$ su - user2
Password:
[user2@server ~]$ whoami
user2
[user2@server ~]$
```

From Normal User to Switch the root User:-

```
[user2@server ~]$ su - root
Password:
[root@server ~]# whoami
root
[root@server ~]#
```

Browsing a Filesystem:

Ni

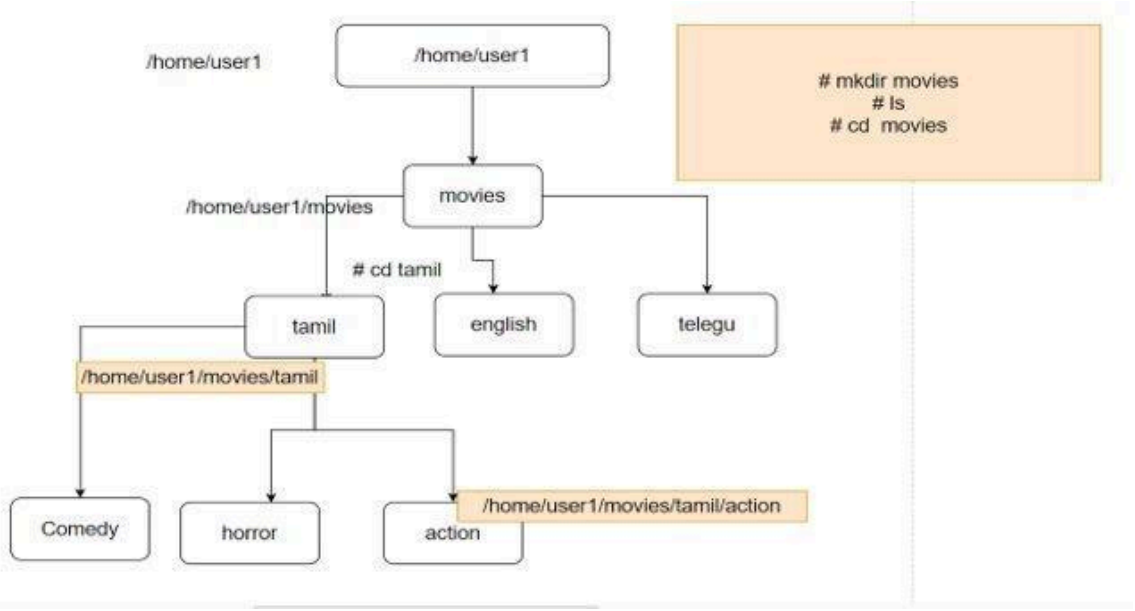


[root@server ~]#
username@hostname [home_directory] #

-> root user
$ -> Normal user

Create Dir:

Ni



mkdir <directory_name>

[user1@server ~]$ mkdir movies
[user1@server ~]$ ls
movies
[user1@server ~]$

$ ls => lists the files/directories on your pwd
$ cd <directory_name > ### change directory
$ pwd => prints working directory
$ cd .. => goes one step back from your pwd
$ touch <file_name> => creates an empty file
$

Create dir
[user3@server ~]$ mkdir Programming

Create Sub-dir
[user3@server Programming]$ mkdir c++

Create multiple Sub-dir in single command
[user3@server Programming]$ mkdir c java

Lists the files/directories on your pwd
[user3@server Programming]$ ls
c c++ java

[user3@server Programming]$ ls -l
total 0

Ni

```
drwxr-xr-x 2 user3 user3 6 Dec 17 14:05 c
drwxr-xr-x 2 user3 user3 6 Dec 17 14:04 c++
drwxr-xr-x 2 user3 user3 6 Dec 17 14:05 java

[user3@server Programming]$ cd c

Goes one step back from your pwd
[user3@server c]$ cd ..
[user3@server Programming]$ cd c
Goes two step back from your pwd
[user3@server c]$ cd ../..
[user3@server ~]$


creates an empty file
[user3@server c]$ touch file
[user3@server c]$ ls
file
```

Create a file with the content

```
[user2@localhost ~]$ cat > file2
Hello
This is Linux class
```

To view the file

```
[user2@localhost ~]$ cat file2
Hello
This is Linux class
```

1) cat

=======

cat(concatenate) command allows us to create single file, view contents of file, concatenate files and redirect output in terminal or files.

Example of creating file or files using cat command

#cat > filename

To create hidden file

#cat > .file

Ni

2) touch
=======
The touch command is the easiest way to create new, empty files that is 0kb file.
Example of creating file or files using touch command.

To create an empty file

#touch file1

To create multiple empty files

#touch file1 file2 file3 file4

#touch file{1..4}

To create hidden empty file

#touch .hiddenfile

[root@raghav ~]# cat samplefile
Welcome to the class
[root@raghav ~]# cat >>samplefile
thanks for joining
^C
[root@raghav ~]# cat samplefile
Welcome to the class
thanks for joining
[root@raghav ~]# vi samplefile
[root@raghav ~]# seq 1 100 > samplefile
[root@raghav ~]# cat samplefile
[root@raghav ~]# cat samplefile
Hello
[root@raghav ~]# echo linux class > samplefile
[root@raghav ~]# cat samplefile
linux class
[root@raghav ~]# echo linux class >> samplefile
[root@raghav ~]# cat samplefile
linux class
linux class
[root@raghav ~]# cal >> samplefile
[root@raghav ~]# cat samplefile
linux class
linux class

Ni

```
    February 2023
Su Mo Tu We Th Fr Sa
         1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

[root@raghav ~]# cp samplefile samplefile2
[root@raghav ~]# cat samplefile2
               Local time: Wed 2023-02-22 03:11:12 EST
           Universal time: Wed 2023-02-22 08:11:12 UTC
                 RTC time: Wed 2023-02-22 08:11:12
                Time zone: America/New_York (EST, -0500)

[root@raghav ~]# cp samplefile samplefile1
cp: overwrite 'samplefile1'? y
[root@raghav ~]# cat samplefile1
linux class
[root@raghav ~]# mkdir sampledir
[root@raghav ~]# cp samplefile sampledir/testsample
[root@raghav ~]# cd sampledir
[root@raghav sampledir]# ls -l
total 4
-rw-r--r-- 1 root root 12 Feb 22 03:20 testsample
[root@raghav sampledir]# cat testsample
linux class
[root@raghav sampledir]# cd
[root@raghav ~]# cp samplefile sampledir/
[root@raghav ~]# cd sampledir
```

Copy:-

Ni

```
1040  cp u4 u3
1041  ls -l
1042  cp u4 /vm/u3
1043  cp u4 vm/u3
1044  ls -l
1045  cd vm
1046  ls -l
1047  cd
1048  cp -r vm vm1
```

B

Move:-

```
1056  mv f2 vm1
```

Delete:-

```
rmdir file1
rm -r file1
```

Ni

Linux File and Directory Permissions

Types of permissions

1.Basic file/folder permissions

2.Special file/folder permissions

3.Access control list (ACL)

To check the default permission of a file/folder

ls -l

ll

FILE

- Read :- Permission to read the file content
- Write :- permission to modify the content
- Execute :- permission to run the script

Directory/Folder

- Read :- permission to list the contents (ls)
- Write :- permission to create or delete file/folder
- Execute :- permission to change directory

Ni

Fields in ls -l command output

- <file type> <file permission> <link count> <owner> <group> <file size> <last modify date and time> <file name

File type	User	Group	Global
d Directory	rwx	r-x	r-x
- Regular file	rw-	r--	r--
l Symbolic Link	rwx	rwx	rwx

In Linux, There are three general classes of access (access levels):

- User (u) The user who owns the file.
- Group (g) Users belonging to the file's defined ownership group.
- Others (o) Everyone else.

- In Linux, There are three general classes of access (access levels):
- User (u) The user who owns the file.
- Group (g) Users belonging to the file's defined ownership group.
- Others (o) Everyone else.

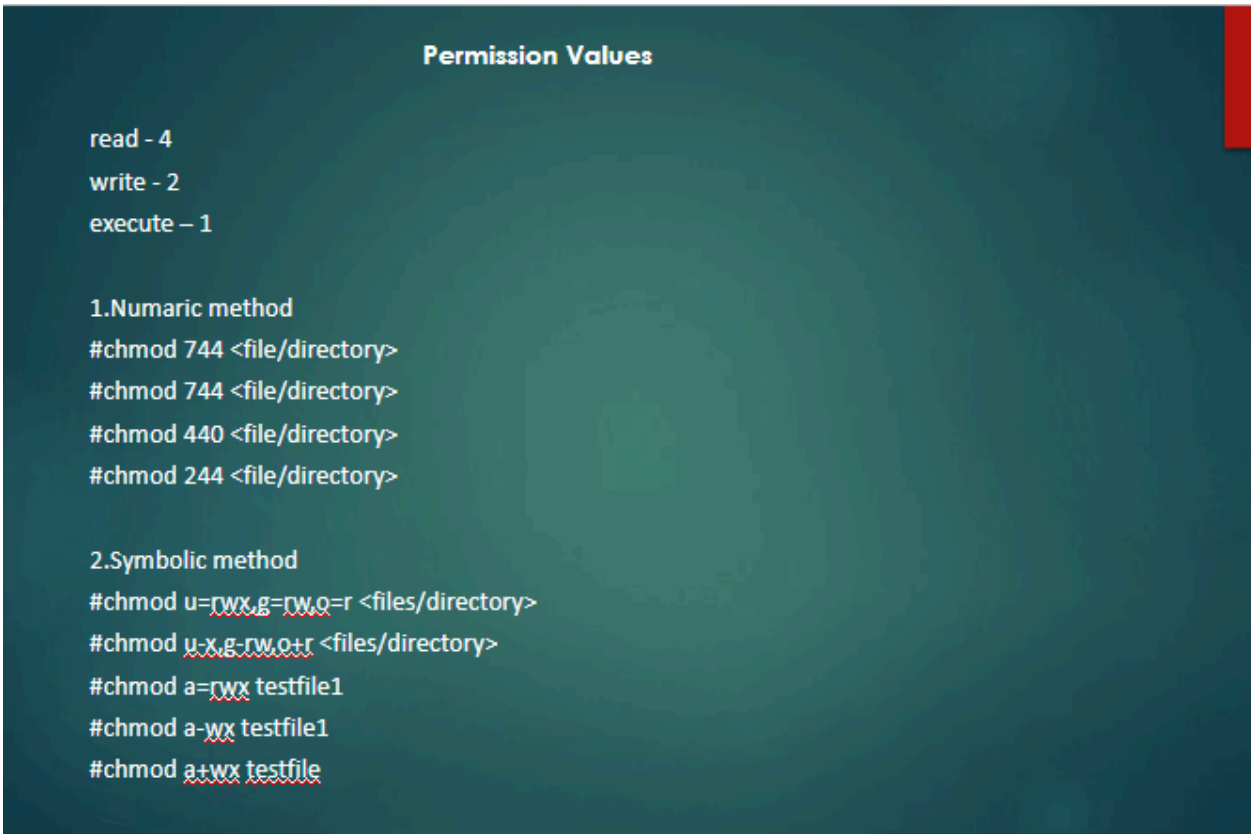
To change the default permission of a file/directory

- # chmod (permission value) <file/directory >

We can use two method to set permissions to files/directories

- 1.octal or numaric -->421
- 2.symbolic --> rwx

Ni




```
root@raghav ~]# mkdir /perm2
[root@raghav ~]# ls -ld /
afs/           db2part/       lib/           per1/          srv/
.autorelabel   dev/           lib64/         perm2/         stickybit/
bin/           dhparams.pem   media/         proc/          sys/
boot/          dir1/          mnt/           root/          test/
class/         etc/           newdir/        run/           tmp/
classdemo/     fileperm1/     nwfile1/       sbin/          usr/
comman-dir/    gid/           opt/           softlink       var/
db1part/       home/          partition/     softlink.file
[root@raghav ~]# ls -ld /per
per1/  perm2/
[root@raghav ~]# ls -ld /perm2
drwxr-xr-x 2 root root 6 Feb 27 03:38 /perm2
[root@raghav ~]# ls /perm2
[root@raghav ~]# ls -ld /perm2
drwxr-xr-x 2 root root 6 Feb 27 03:38 /perm2
[root@raghav ~]# tail /etc/group
u3:x:2016:
group6:x:2017:
Tariner:x:2018:u1
Trainer:x:2019:u1
```

Ni

```
u4:x:2020:
apache:x:48:
httpd:x:2021:
manoj:x:2022:
test:x:2023:
Vinoth:x:2024:

[root@raghav ~]# su - u1
[u1@raghav ~]$ cd /perm2
[u1@raghav perm2]$ touch perm
touch: cannot touch 'perm': Permission denied
[u1@raghav perm2]$ cd
[u1@raghav ~]$ logout

[root@raghav ~]# ls -ld /perm2
drwxr-xr-x 2 root root 6 Feb 27 03:38 /perm2

[root@raghav ~]# chmod 757 /per
per1/  perm2/
[root@raghav ~]# chmod 757 /perm2
[root@raghav ~]# ls -ld /perm2
drwxr-xrwx 2 root root 6 Feb 27 03:38 /perm2
[root@raghav ~]# su - u1
[u1@raghav ~]$ cd /perm2
[u1@raghav perm2]$ cat > perm
hi
^C
[u1@raghav perm2]$ logout
[root@raghav ~]# cd /per
per1/  perm2/
[root@raghav ~]# cd /perm2
[root@raghav perm2]# cat > sample
Permission class
^C
[root@raghav perm2]# ls -l
total 8
-rw-r--r-- 1 u1   Trainer  3 Feb 27 03:46 perm
-rw-r--r-- 1 root root    17 Feb 27 03:47 sample


[root@raghav perm2]# su - u1
[u1@raghav ~]$ cd /perm2
[u1@raghav perm2]$ ls -l
total 8


Ni
```

```
-rw-r--r-- 1 u1   Trainer  3 Feb 27 03:46 perm
-rw-r--r-- 1 root root    17 Feb 27 03:47 sample
[u1@raghav perm2]$ cat >> sample
-bash: sample: Permission denied
[u1@raghav perm2]$ logout
[root@raghav perm2]# ls -l
total 8
-rw-r--r-- 1 u1   Trainer  3 Feb 27 03:46 perm
-rw-r--r-- 1 root root    17 Feb 27 03:47 sample
[root@raghav perm2]# su - u1
[u1@raghav ~]$ cd /perm2
[u1@raghav perm2]$ cat sample
Permission class
[u1@raghav perm2]$ logout
[root@raghav perm2]# ls -l sample
-rw-r--r-- 1 root root 17 Feb 27 03:47 sample
[root@raghav perm2]# chmod 641 samole
chmod: cannot access 'samole': No such file or directory
[root@raghav perm2]# chmod 641 sample
[root@raghav perm2]# ls -l sample
-rw-r----x 1 root root 17 Feb 27 03:47 sample
[root@raghav perm2]# su - u1
[u1@raghav ~]$ cd /perm2
[u1@raghav perm2]$ cat sample
cat: sample: Permission denied
[u1@raghav perm2]$ logout

[root@raghav ~]# chown u1 /perm2
[root@raghav ~]# ls -ld /perm2
drwxr-xr-x 2 u1 root 32 Feb 27 03:47 /perm2
[root@raghav ~]# chmod 555 /perm2
[root@raghav ~]# ls -ld /perm2
dr-xr-xr-x 2 u1 root 32 Feb 27 03:47 /perm2

[root@raghav ~]# su - u1
[u1@raghav ~]$ cd /perm2
[u1@raghav perm2]$ touch sam1
touch: cannot touch 'sam1': Permission denied
[u1@raghav perm2]$ logout

[root@raghav ~]# groupadd perm
[root@raghav ~]# cat /etc/group

[root@raghav ~]# usermod -aG perm u2


Ni
```

```
[root@raghav ~]# cat /etc/group

[root@raghav ~]# ls -ld /perm2
dr-xr-xr-x 2 u1 root 32 Feb 27 03:47 /perm2
[root@raghav ~]# cho
choom  chown
[root@raghav ~]# chown :perm /perm2
[root@raghav ~]# ls -ld /perm2
dr-xr-xr-x 2 u1 perm 32 Feb 27 03:47 /perm2
[root@raghav ~]# chmod 775 /perm2
[root@raghav ~]# ls -ld /perm2
drwxrwxr-x 2 u1 perm 32 Feb 27 03:47 /perm2
[root@raghav ~]# su - u2
[u2@raghav ~]$ cd /perm2
[u2@raghav perm2]$ touch f1
[u2@raghav perm2]$ logout
[root@raghav ~]# cd /perm2
[root@raghav perm2]# ls -l
total 8
-rw-r--r-- 1 u2   u2       0 Feb 28 03:24 f1
-rw-r--r-- 1 u1   Trainer  3 Feb 27 03:46 perm
-rw-r----x 1 root root    17 Feb 27 03:47 sample
[root@raghav perm2]# chmod o+x,g+wx,u+rw /perm2
[root@raghav perm2]# ls -ld /perm2
drwxrwxr-x 2 u1 perm 42 Feb 28 03:24 /perm2
[root@raghav perm2]# chmod o+x,g+wx,u+rw sample
[root@raghav perm2]# ls -l
total 8
-rw-r--r-- 1 u2   u2       0 Feb 28 03:24 f1
-rw-r--r-- 1 u1   Trainer  3 Feb 27 03:46 perm
-rw-rwx--x 1 root root    17 Feb 27 03:47 sample
[root@raghav perm2]# chmod o+rwx,g+rwx,u+rwx sample
[root@raghav perm2]# ls -l
total 8
-rw-r--r-- 1 u2   u2       0 Feb 28 03:24 f1
-rw-r--r-- 1 u1   Trainer  3 Feb 27 03:46 perm
-rwxrwxrwx 1 root root    17 Feb 27 03:47 sample
[root@raghav perm2]# chmod o-x,g-x,u-x sample
[root@raghav perm2]# ls -l
total 8
-rw-r--r-- 1 u2   u2       0 Feb 28 03:24 f1
-rw-r--r-- 1 u1   Trainer  3 Feb 27 03:46 perm
-rw-rw-rw- 1 root root    17 Feb 27 03:47 sample
```

Ni

Vi editor

```
[user@host ~]$ vim filename
```

To save and quit from command mode

==========

- shift + zz
- Insert Mode :- content write mode which is needed to modify the file
- Shift form of command mode to Insert mode.
- i --> insert the text
- I --> insert the text in beginning of line
- a --> adds the text after the current cursor position
- A --> adds the text at the end of a line
- o --> inserts the text one line below current cursor position
- O --> inserts the text one line above current cursor position

Ni

- • Command Mode - press Esc (default mode)
- • Insert Mode - press i,I,A,a,o,O
- • Execution Mode (or) Escape Mode - : (shift + ;)
- • Command Mode :- default mode. In this mode we can copy, paste and delete.
- yy --> to copy a line in command mode
- 3yy --> to copy three lines in commandmode
- p --> to paste copied line one time and paste the below the cursor
- P - paste the above the cursor
- 4p --> to copy copied line 4 times
- dd --> to delete a line (or)cut
- 4dd --> to delete 4 lines
- shift + g --> moves the cursor to end of the file.
- gg --> moves the cursor to the beginning of file
- /word --> to search any key word in file
- ?word
- n --> next search
- N --> previous search
- u --> undo
- ctrl + r --> redo

To move cursor in command mode

==========

- j k l h
- h - move to left
- j - move to down
- k - move to top
- l - move to right

Ex Mode :- extended command mode

- :q --> without saving quite from file
- :q! --> forcefully quite
- :w --> save the changes
- :w! --> forcefully save changes
- :wq --> save and quite
- :wq! --> forcefully save and quite (it will modify the time stamp)
- :x --> save and quite
- :X --> to encrypt the file contents
- :25 --> moves cursor to 25th line
- :set nu --> to set line numbers to line in the file
- :set nonu --> remove line numbers
- :5d --> delete 5th line
- :$d --> delete last line
- :3,6d --> delete 3rd line to 6th line
- :$ --> place the cursor in the late line

:%s/old/new/g ---> TO search a word and replace it with new word

:s/old/new/g line number -----> will replace specific word in that specific line number

- /<string>
- ?<string>Finding text

Ni

USERS AND GROUP ADMINISTRATION IN LINUX

- User :- A person who utilizes the Operating System services is a user.
- Group :- A group is collection of users.
-

Linux Users

1. Super user or root user(0) :-The user with all permissions, access and privileges.
2. Static system user(1-200) :- The user created by Operating System automatically. Like daemon
3. System user(201-999) :- The user created when we install any package like DNS, HTTP (Apache)and chronyd because these services and daemons runs as a system user.
4. Normal user(1000-60000) :- Manually created users are normal user. Not having much privileges, access and permissions.

There are main user and group administration files:

- /etc/passwd
- /etc/group
- /etc/shadow
- /etc/gshadow
- /etc/login.defs ----> file provides default configuration information for several user account parameters

Fields in Below Files:

- /etc/passwd - Username: User's password mask : UID : GID : User Comment : home directory of user: Default log-in shell
- /etc/shadow - Username: Encrypted password:Last password change:Minimum :Maximum :Warning days:Inactive:Expire:reserverd
- /etc/group - Groupname:password:GID:Members of group
-
- /etc/gshadow - Groupname:group password:group id admin:Members of group

To remove pass$word of a user

- #passwd -d username

Ni

To delete an user account

- #userdel username

To delete an user's account with its home directory and mail directory

- #userdel --remove username
- or
- #userdel -r username

To edit user's default parameters (/etc/default/useradd)

- #vim /etc/default/useradd

To create group

- #groupadd groupname
-

To see the information of groups

- # cat /etc/group
-

Add user into a group (secondary group)

- #gpasswd -a user group
- or
- #usermod -a -G group user

Remove user from a group

- #gpasswd -d user group

Change users primary group

- usermod -g groupname/id username

To remove group

- #groupdel groupname

Check the password status;

Ni

- #passwd -S <username>

To list the shells

- # cat /etc/shells

To change ownership of file/folder

- #chown username file/folder

To change group ownership of file/folder

- #chgrp groupname file/folder

To change ownership and group ownership of file/folder

- #chown username:groupname file/folder

/etc/shadow -Username:Encrypted password:Last password change:Minimum :Maximum :Warning days:Inactive:Expire:reserverd

· Username of the account this password belongs to. The encrypted password of the user.

· The format of encrypted passwords is discussed later in this section. The day on which the password was last changed.

· This is set in days since 1970-01-01, and is calculated in the UTC time zone. The minimum number of days that have to elapse since the last password change before the user can change it again.

· The maximum number of days that can pass without a password change before the password expires.

· An empty field means it does not expire based on time since the last change.

Ni

- Warning period. The user will be warned about an expiring password when they login for this number of days before the deadline.

- Inactivity period. Once the password has expired, it will still be accepted for login for this many days. After this period has elapsed, the account will be locked.

- The day on which the account expires.

- This is set in days since 1970-01-01, and is calculated in the UTC time zone. An empty field means it does not expire on a particular date.

- The last field is usually empty and is reserved for future use

Managing User Password:-

[user01@host ~]$ chage -m 0 -M 90 -W 7 -I 14 user03

The preceding chage command uses the -m, -M, -W, and -I options to set the minimum age, maximum age, warning period, and inactivity period of the user's password, respectively. The chage -d 0 user03 command forces the user03 user to update its password on the next login. The chage -l user03 command displays the password aging details of user03. The chage -E 2019-08-05 user03 command causes the user03 user's account to expire on 2019-08-05 (in YYYY-MM-DD format).

chage -M 90 operator1

chage -l operator1

Ni

chage -d 0 operator1

Chage –E yyyy/mm/dd or yyyy-mm-dd – Account Expire

Change users primary group

usermod -g groupname/id username

To remove group

#groupdel groupname

check the password status

#passwd -S <username>

To list the shells

cat /etc/shells

To change ownership of file/folder

#chown username file/folder

To change group ownership of file/folder

#chgrp groupname file/folder

To change ownership and group ownership of file/folder

#chown username:groupname file/folder

Ni

Special Permissions

setuid permission

```
[root@localhost ~]# chmod 775 /gid
[root@localhost ~]# chgrp Trainer /gid
[root@localhost ~]# ls -ld /gid
drwxrwxr-x. 2 root Trainer 6 Jan 19 02:37 /gid
[root@localhost ~]# chmod 2775 /gid
[root@localhost ~]# ls -ld /gid
drwxrwsr-x. 2 root Trainer 19 Jan 19 02:42 /gid
[root@localhost ~]# chgrp root /gid
[root@localhost ~]# chmod 777 /gid
[root@localhost ~]# chmod 1777 /gid
```

The setuid permission on an executable file means that commands run as the user owning the file, not as the user that ran the command. One example is the passwd command:

In a long listing, you can identify the setgid permissions by a lowercase s where you would normally expect the x (group execute permissions) to be. If the group does not have execute permissions, this is replaced by an uppercase S.

```
[user@host ~]$ ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 35504 Jul 16  2010 /usr/bin/passwd
```

Setgid permission

The special permission setgid on a directory means that files created in the directory inherit their group ownership from the directory, rather than inheriting it from the creating user. This is commonly used on group collaborative directories to automatically change a file from the default private group to the shared group, or if files in a directory should be always owned by a specific group

```
[user@host ~]$ ls -ld /run/log/journal
drwxr-sr-x. 3 root systemd-journal 60 May 18 09:15 /run/log/journal
```

Sticky bit:

Ni

The sticky bit for a directory sets a special restriction on deletion of files. Only the owner of the file (and root) can delete files within the directory. An example is /tmp:

In a long listing, you can identify the sticky permissions by a lowercase t where you would normally expect the x (other execute permissions) to be. If other does not have execute permissions, this is replaced by an uppercase T.

```
[user@host ~]$ ls -ld /tmp
drwxrwxrwt. 39 root root 4096 Feb  8 20:52 /tmp
```

Setting Special Permissions

• Symbolically: setuid = u+s; setgid = g+s; sticky = o+t

• Numerically (fourth preceding digit): setuid = 4; setgid = 2; sticky = 1

Managing Links Between Files

Hard Links and Soft Links It is possible to create multiple names that point to the same file. There are two ways to do this: by creating a hard link to the file, or by creating a soft link (sometimes called a symbolic link) to the file. Each has its advantages and disadvantages.

Soft Links :

The ln -s command creates a soft link, which is also called a "symbolic link." A soft link is not a regular file, but a special type of file that points to an existing file or directory. Soft links have some advantages over hard links:

• They can link two files on different file systems.

• They can point to a directory or special file, not just a regular file.

Ni

In the Softlink, the link can create both files and directory. Once the original file or directory is deleted the created Link file will be inactive.

Hard Link:

Every file starts with a single hard link, from its initial name to the data on the file system. When you create a new hard link to a file, you create another name that points to that same data. The new hard link acts exactly like the original file name. Once created, you cannot tell the difference between the new hard link and the original name of the file.

You can find out if a file has multiple hard links with the ls -l command. One of the things it reports is each file's link count, the number of hard links the file has.

In the Hardlink, the link can create only files, if the original file is deleted, the link file remains active

Softlink command:

ln –s <source file> <destination file>

Hardlink command:

ln <source file> <destination file>

Softlink:

```
51835145 drwxr-xr-x. 2 hari  linux     6 Dec 30 00:46 sdir1
18343975 lrwxrwxrwx. 1 root  root      5 Jan 20 02:19 sdir1.h -> sdir1
```

Hardlink:

Ni

```
18360686 -rw-r--r--. 2 root  root      5 Jan 14 03:29 softfile1
18360686 -rw-r--r--. 2 root  root      5 Jan 14 03:29 softfile1_h
```

Monitoring and Managing Linux Processes

Definition of a Process

A process is a running instance of a launched, executable program. A process consists of:

An address space of allocated memory

Security properties including ownership credentials and privileges • One or more execution threads of program code

Process state The environment of a process includes:

Local and global variables

A current scheduling context

Allocated system resources, such as file descriptors and network ports

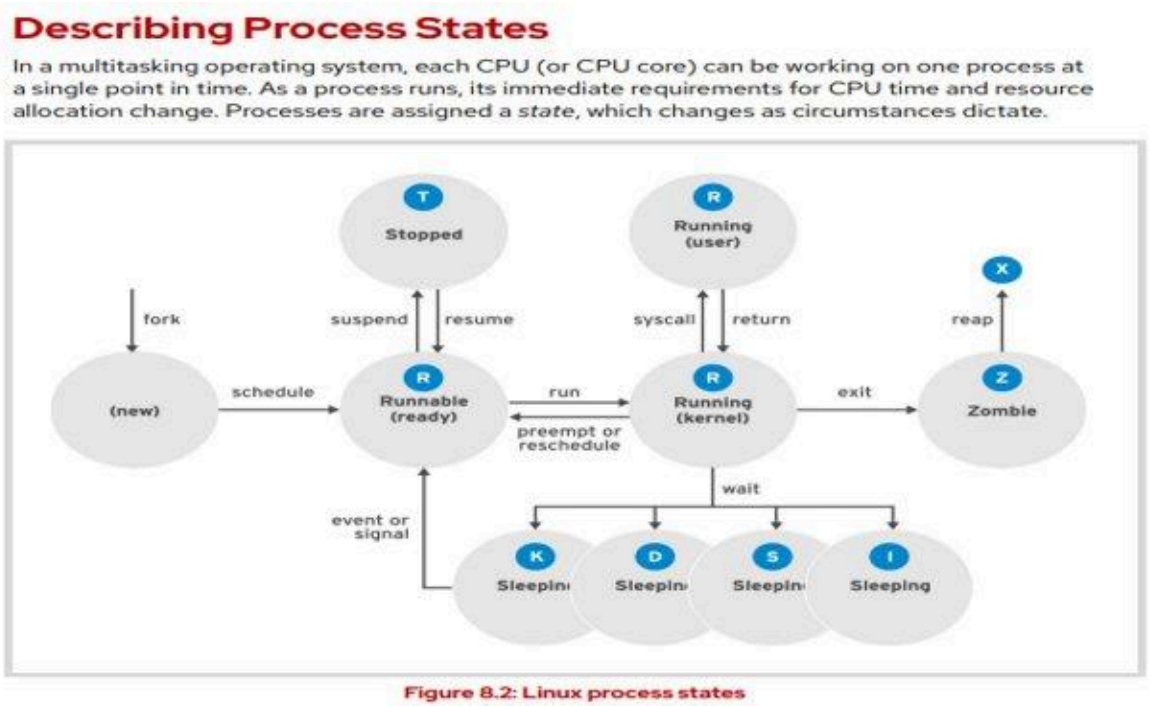


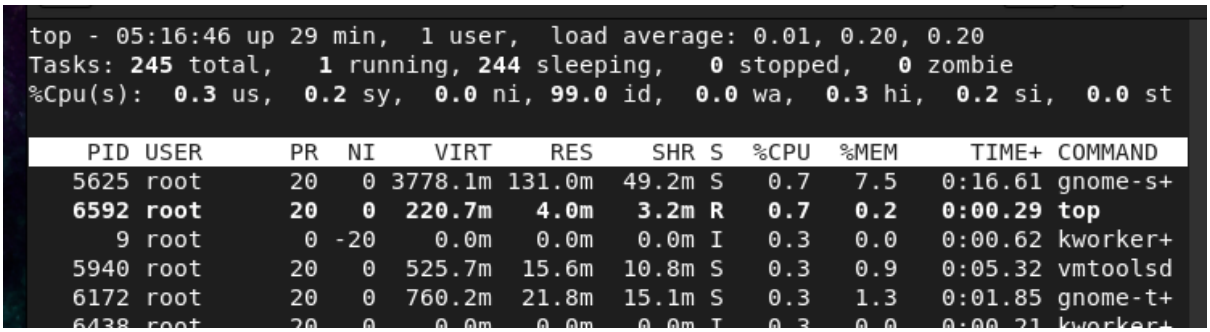
Figure 8.2: Linux process states

Ni

Name	Flag	Kernel-defined state name and description
Running	**R**	TASK_RUNNING: The process is either executing on a CPU or waiting to run. Process can be executing user routines or kernel routines (system calls), or be queued and ready when in the *Running* (or *Runnable*) state.
Sleeping	**S**	TASK_INTERRUPTIBLE: The process is waiting for some condition: a hardware request, system resource access, or signal. When an event or signal satisfies the condition, the process returns to *Running*.
	D	TASK_UNINTERRUPTIBLE: This process is also *Sleeping*, but unlike **S** state, does not respond to signals. Used only when process interruption may cause an unpredictable device state.
	K	TASK_KILLABLE: Identical to the uninterruptible **D** state, but modified to allow a waiting task to respond to the signal that it should be killed (exit completely). Utilities frequently display *Killable* processes as **D** state.

Name	Flag	Kernel-defined state name and description
Running	**R**	TASK_RUNNING: The process is either executing on a CPU or waiting to run. Process can be executing user routines or kernel routines (system calls), or be queued and ready when in the *Running* (or *Runnable*) state.
Sleeping	**S**	TASK_INTERRUPTIBLE: The process is waiting for some condition: a hardware request, system resource access, or signal. When an event or signal satisfies the condition, the process returns to *Running*.
	D	TASK_UNINTERRUPTIBLE: This process is also *Sleeping*, but unlike **S** state, does not respond to signals. Used only when process interruption may cause an unpredictable device state.
	K	TASK_KILLABLE: Identical to the uninterruptible **D** state, but modified to allow a waiting task to respond to the signal that it should be killed (exit completely). Utilities frequently display *Killable* processes as **D** state.

TOP command:



ps command

Ni

```
[root@192 ~]# ps
    PID TTY          TIME CMD
   6199 pts/0    00:00:00 bash
   6612 pts/0    00:00:00 ps
```

ps -aux

```
[root@192 ~]# ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.1  0.5 106300  9332 ?        Ss   04:47   0:03 /usr/lib/syst
root           2  0.0  0.0      0     0 ?        S    04:47   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   04:47   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   04:47   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   04:47   0:00 [netns]
```

ps -ef

```
[root@192 ~]# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 04:47 ?        00:00:03 /usr/lib/systemd/systemd rhg
root           2       0  0 04:47 ?        00:00:00 [kthreadd]
root           3       2  0 04:47 ?        00:00:00 [rcu_gp]
root           4       2  0 04:47 ?        00:00:00 [rcu_par_gp]
root           5       2  0 04:47 ?        00:00:00 [netns]
root           7       2  0 04:47 ?        00:00:00 [kworker/0:0H-events_highpri
```

Listing Processes

The ps command is used for listing current processes. It can provide detailed process information, including:

Perhaps the most common set of options, aux, displays all processes including processes without a controlling terminal. A long listing (options lax) provides more technical detail, but may display faster by avoiding user name lookups. The similar UNIX syntax uses the options -ef to display all processes.

Ni

```
[user@host ~]$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.1  0.1  51648  7504 ?        Ss   17:45   0:03 /usr/lib/systemd/
syst
root           2  0.0  0.0      0     0 ?        S    17:45   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    17:45   0:00 [ksoftirqd/0]
root           5  0.0  0.0      0     0 ?        S<   17:45   0:00 [kworker/0:0H]
root           7  0.0  0.0      0     0 ?        S    17:45   0:00 [migration/0]
...output omitted...
[user@host ~]$ ps lax
F   UID   PID  PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY      TIME COMMAND
4     0     1     0  20   0  51648  7504 ep_pol Ss   ?        0:03 /usr/lib/
systemd/
1     0     2     0  20   0      0     0 kthrea S    ?        0:00 [kthreadd]
1     0     3     2  20   0      0     0 smpboo S    ?        0:00 [ksoftirqd/0]
1     0     5     2   0 -20      0     0 worker S<   ?        0:00 [kworker/0:0H]
1     0     7     2 -100  -      0     0 smpboo S    ?        0:00 [migration/0]
...output omitted...
[user@host ~]$ ps -ef
UID          PID  PPID  C STIME TTY          TIME CMD
root           1     0  0 17:45 ?        00:00:03 /usr/lib/systemd/systemd --
switched-ro
root           2     0  0 17:45 ?        00:00:00 [kthreadd]
root           3     2  0 17:45 ?        00:00:00 [ksoftirqd/0]
root           5     2  0 17:45 ?        00:00:00 [kworker/0:0H]
root           7     2  0 17:45 ?        00:00:00 [migration/0]
...output omitted...
```

Running Jobs in Background

```
[root@192 ~]# sleep 500 &
[1] 6680
```

You can display the list of jobs that Bash is tracking for a particular session with the jobs command.

```
[root@192 ~]# jobs
[1]+  Running                 sleep 500 &
```

A background job can be brought to the foreground by using the **fg** command with its job ID (*%job number*).

```
[user@host ~]$ fg %1
sleep 10000
```

In the preceding example, the **sleep** command is now running in the foreground on the controlling terminal. The shell itself is again asleep, waiting for this child process to exit.

To send a foreground process to the background, first press the keyboard generated *suspend* request (**Ctrl+z**) in the terminal.

```
sleep 10000
^Z
[1]+  Stopped                 sleep 10000
[user@host ~]$
```

The job is immediately placed in the background and is suspended.

Ni

The ps j command displays information relating to jobs. The PID is the unique process ID of the process. THe PPID is the PID of the parent process of this process, the process that started (forked) it. The PGID is the PID of the process group leader, normally the first process in the job's pipeline. The SID is the PID of the session leader, which (for a job) is normally the interactive shell that is running on its controlling terminal. Since the example sleep command is currently suspended, its process state is T.

```
[root@192 ~]# ps j
  PPID     PID    PGID     SID TTY        TPGID STAT   UID   TIME COMMAND
  5472    5528    5528    5528 tty2         5528 Ssl+     0   0:00 /usr/libexec/
  5528    5535    5528    5528 tty2         5528 Sl+      0   0:00 /usr/libexec/
  6172    6199    6199    6199 pts/0        6733 Ss       0   0:00 bash
  6199    6680    6680    6199 pts/0        6733 T        0   0:00 sleep 500
  6199    6733    6733    6199 pts/0        6733 R+       0   0:00 ps j
```

To start the suspended process running in the background, use the bg command with the same job ID.

```
[root@192 ~]# bg %1
[1]+ sleep 500 &
[root@192 ~]# jobs
[1]+  Running                 sleep 500 &
```

Killing process

Ni

Fundamental Process Management Signals

Signal number	Short name	Definition	Purpose
1	HUP	Hangup	Used to report termination of the controlling process of a terminal. Also used to request process reinitialization (configuration reload) without termination.
2	INT	Keyboard interrupt	Causes program termination. Can be blocked or handled. Sent by pressing INTR key sequence (**Ctrl+c**).
3	QUIT	Keyboard quit	Similar to SIGINT; adds a process dump at termination. Sent by pressing QUIT key sequence (**Ctrl+**).
9	KILL	Kill, unblockable	Causes abrupt program termination. Cannot be blocked, ignored, or handled; always fatal.
15 *default*	TERM	Terminate	Causes program termination. Unlike SIGKILL, can be blocked, ignored, or handled. The "polite" way to ask a program to terminate; allows self-cleanup.
18	CONT	Continue	Sent to a process to resume, if stopped. Cannot be blocked. Even if handled, always resumes the process.
19	STOP	Stop, unblockable	Suspends the process. Cannot be blocked or handled.
20	TSTP	Keyboard stop	Unlike SIGSTOP, can be blocked, ignored, or handled. Sent by pressing SUSP key sequence (**Ctrl+z**).

Commands for Sending Signals by Explicit Request

You signal the current foreground process by pressing a keyboard control sequence to suspend (**Ctrl+z**), kill (**Ctrl+c**), or core dump (**Ctrl+**) the process. However, you will use signal-sending commands to send signals to a background process or to processes in a different session.

Signals can be specified as options either by name (for example, **-HUP** or **-SIGHUP**) or by number (the related **-1**). Users may kill their own processes, but root privilege is required to kill processes owned by others.

The **kill** command sends a signal to a process by PID number. Despite its name, the kill command can be used to send any signal, not just those for terminating programs. You can use the **kill -l** command to list the names and numbers of all available signals.

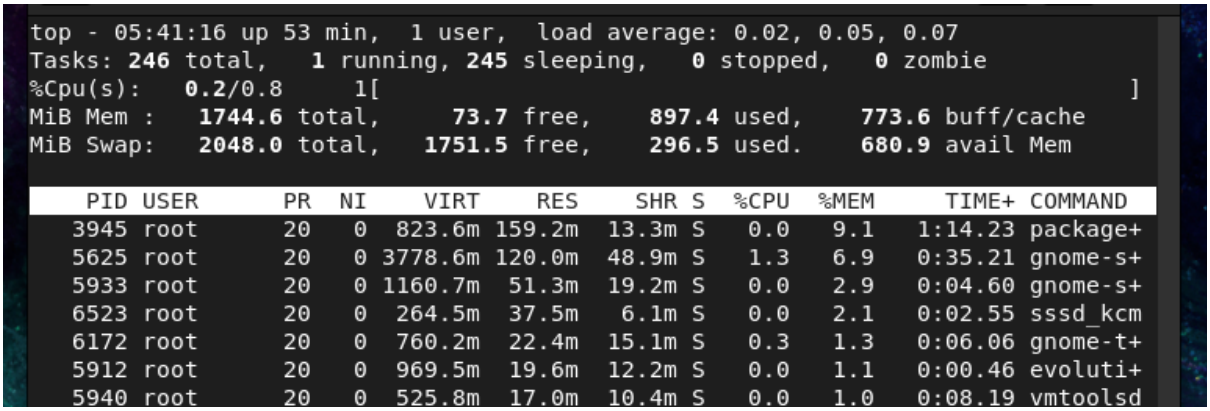
```
[user@host ~]$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
...output omitted...
[user@host ~]$ ps aux | grep job
5194  0.0  0.1 222448  2980 pts/1    S    16:39   0:00 /bin/bash /home/user/bin/
control job1
5199  0.0  0.1 222448  3132 pts/1    S    16:39   0:00 /bin/bash /home/user/bin/
control job2
5205  0.0  0.1 222448  3124 pts/1    S    16:39   0:00 /bin/bash /home/user/bin/
control job3
5430  0.0  0.0 221860  1096 pts/1    S+   16:41   0:00 grep --color=auto job
[user@host ~]$ kill 5194
[user@host ~]$ ps aux | grep job
user   5199  0.0  0.1 222448  3132 pts/1    S    16:39   0:00 /bin/bash /home/
user/bin/control job2
user   5205  0.0  0.1 222448  3124 pts/1    S    16:39   0:00 /bin/bash /home/
user/bin/control job3
user   5783  0.0  0.0 221860   964 pts/1    S+   16:43   0:00 grep --color=auto
 job
[1]   Terminated              control job1
[user@host ~]$ kill -9 5199
```

4. In the right terminal shell, observe the top display. Toggle between load, threads and

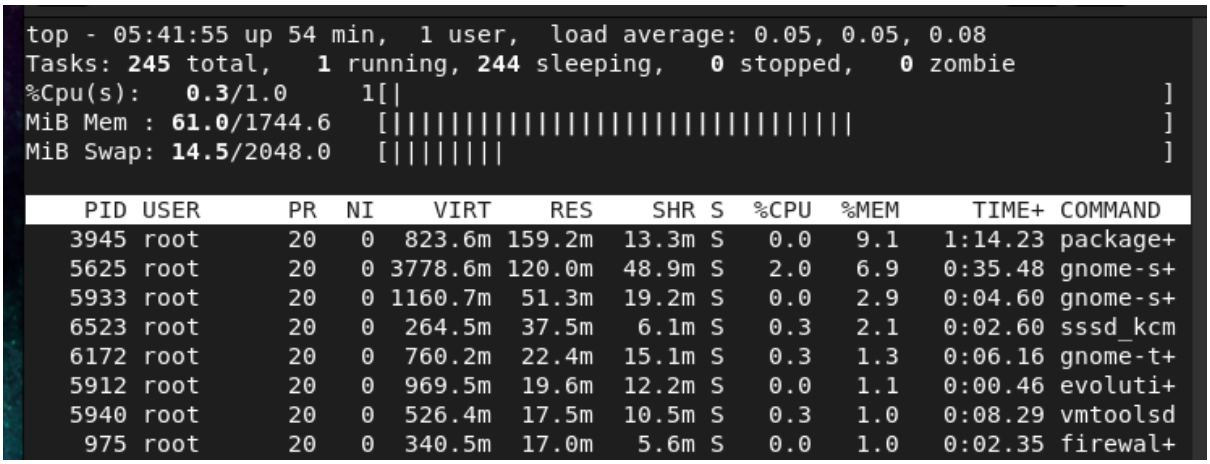
Ni

memory. Note the process ID (PID) for process101. View the CPU percentage. It should hover around 10% to 15%. Ensure that top is showing CPU usage once you have viewed load, threads, and memory.

Press Shift+m.

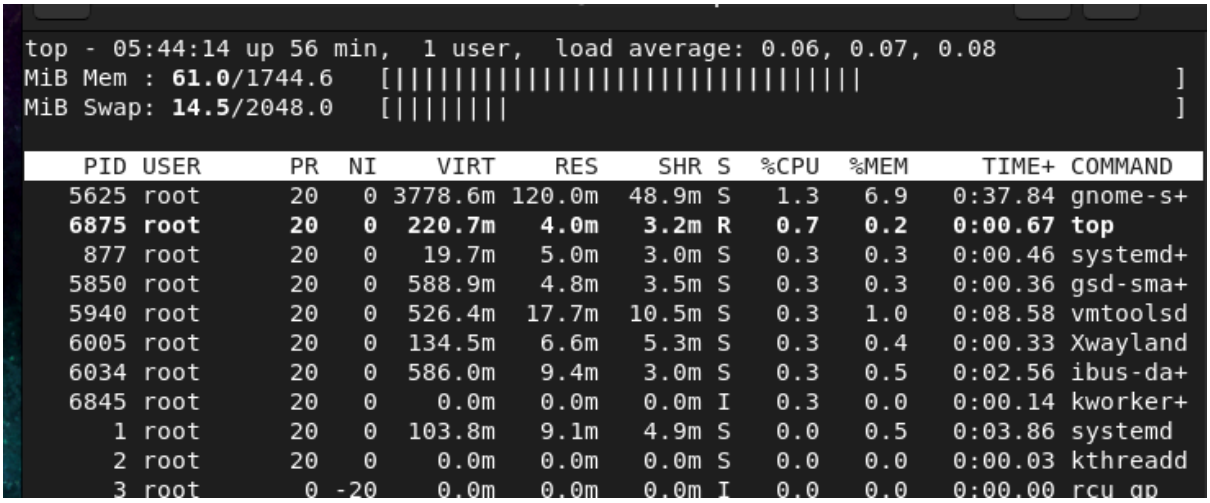


Press m - Shows Memory

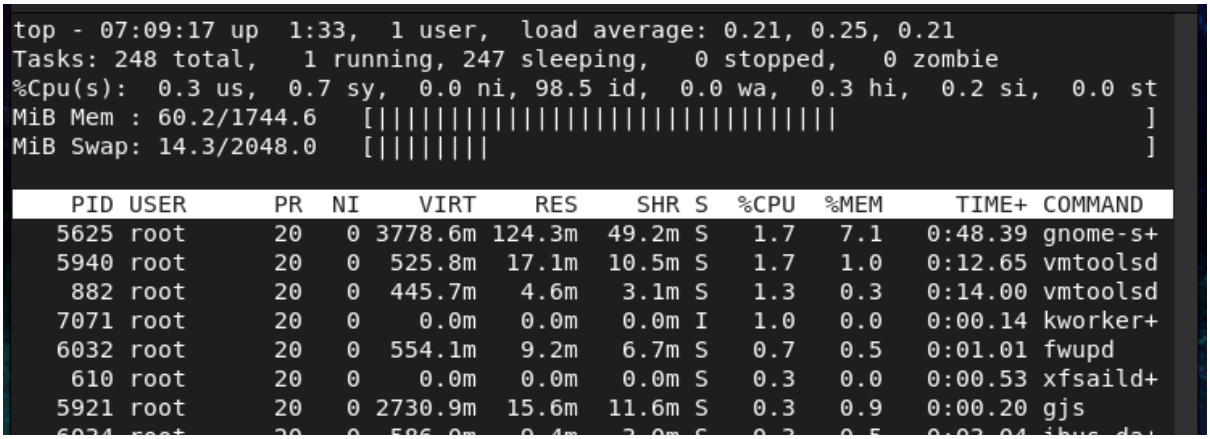


Press t

Ni



Press Shift+b switch the use of bold off.



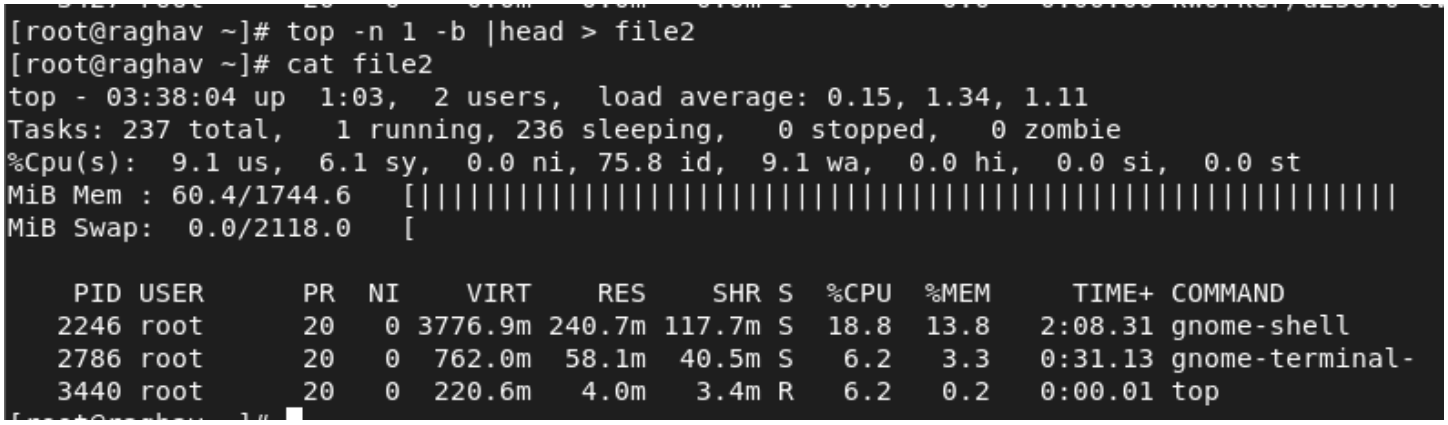
You can sort which column you

ps -aux –sort =pid

need to.

Take the copy for the TOP current status.

Ni



Services:

https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-centos-7#step-2-checking-your-web-server **`a**

Ni

```
[root@localhost ~]# systemctl
  UNIT
  proc-sys-fs-binfmt_misc.automount
  sys-devices-pci0000:00-0000:00:07.1-ata2-host1-target1:0:0-1:0:0:0-block-sr0
  sys-devices-pci0000:00-0000:00:10.0-host2-target2:0:0-2:0:0:0-block-sda-sda1
  sys-devices-pci0000:00-0000:00:10.0-host2-target2:0:0-2:0:0:0-block-sda-sda2
  sys-devices-pci0000:00-0000:00:10.0-host2-target2:0:0-2:0:0:0-block-sda.devi
  sys-devices-pci0000:00-0000:00:10.0-host2-target2:0:1-2:0:1:0-block-sdb.devi
  sys-devices-pci0000:00-0000:00:10.0-host2-target2:0:2-2:0:2:0-block-sdc.devi
  sys-devices-pci0000:00-0000:00:11.0-0000:02:01.0-sound-card0-controlC0.device
  sys-devices-pci0000:00-0000:00:15.0-0000:03:00.0-net-ens160.device
  sys-devices-pci0000:00-0000:00:16.0-0000:0b:00.0-net-ens192.device
  sys-devices-platform-serial8250-tty-ttyS1.device
  sys-devices-platform-serial8250-tty-ttyS2.device
  sys-devices-platform-serial8250-tty-ttyS3.device
  sys-devices-pnp0-00:05-tty-ttyS0.device
  sys-devices-virtual-block-dm\x2d0.device
  sys-devices-virtual-block-dm\x2d1.device
  sys-devices-virtual-block-dm\x2d2.device
  sys-devices-virtual-misc-rfkill.device
  sys-module-configfs.device
  sys-module-fuse.device
  sys-subsystem-net-devices-ens160.device
  sys-subsystem-net-devices-ens192.device
  -.mount
  boot.mount
  dev-hugepages.mount
  dev-mqueue.mount
  root-vm.mount
  run-media-root-CentOS\x2dStream\x2d9\x2dBaseOS\x2dx86_64.mount
  run-user-0-gvfs.mount
  run-user-0.mount
  run-vmblock\x2dfuse.mount
```

Systemctl list-units –type=service

```
[root@localhost ~]# systemctl list-units --type=service
  UNIT                              LOAD   ACTIVE SUB     DESCRIPTION          >
  accounts-daemon.service           loaded active running Accounts Service
  alsa-state.service                loaded active running Manage Sound Card St>
  atd.service                       loaded active running Deferred execution s>
  auditd.service                    loaded active running Security Auditing Se>
  avahi-daemon.service              loaded active running Avahi mDNS/DNS-SD St>
  chronyd.service                   loaded active running NTP client/server
  colord.service                    loaded active running Manage, Install and >
```

Systemctl list-unit-files –type=service

Ni

```
[root@localhost ~]# systemctl list-unit-files --type=service
UNIT FILE                                  STATE           VENDOR PRESET
accounts-daemon.service                    enabled         enabled
alsa-restore.service                       static          -
alsa-state.service                         static          -
arp-ethers.service                         disabled        disabled
atd.service                                enabled         enabled
auditd.service                             enabled         enabled
autovt@.service                            alias           -
avahi-daemon.service                       enabled         enabled
blk-availability.service                   disabled        disabled
bluetooth.service                          enabled         enabled
bolt.service                               static          -
brltty.service                             disabled        disabled
canberra-system-bootup.service             disabled        disabled
```

Service States in the Output of `systemctl`

Keyword	Description
loaded	Unit configuration file has been processed.
active (running)	Running with one or more continuing processes.
active (exited)	Successfully completed a one-time configuration.
active (waiting)	Running but waiting for an event.
inactive	Not running.
enabled	Is started at boot time.
disabled	Is not set to be started at boot time.
static	Cannot be enabled, but may be started by an enabled unit automatically.

xz

Service status check

Ni

```
[root@localhost ~]# systemctl status sshd.service
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor pres>
     Active: active (running) since Wed 2023-01-25 00:40:07 EST; 8min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 1089 (sshd)
      Tasks: 1 (limit: 10778)
     Memory: 2.9M
        CPU: 70ms
     CGroup: /system.slice/sshd.service
             └─1089 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 25 00:40:06 localhost systemd[1]: Starting OpenSSH server daemon...
Jan 25 00:40:07 localhost sshd[1089]: Server listening on 0.0.0.0 port 22.
Jan 25 00:40:07 localhost sshd[1089]: Server listening on :: port 22.
Jan 25 00:40:07 localhost systemd[1]: Started OpenSSH server daemon.
```

Service start. Stop, restart command

```
[root@localhost ~]# systemctl stop sshd.service
[root@localhost ~]# systemctl start sshd.service
[root@localhost ~]# systemctl restart sshd.service
```

Check the enabled and active command

```
[root@localhost ~]# systemctl is-enabled sshd.service
enabled
[root@localhost ~]# systemctl is-active sshd.service
active
```

Service Disable and Enable command

Configuring and Securing SSH

What is OpenSSH?

OpenSSH implements the Secure Shell or SSH protocol in the Red Hat Enterprise Linux systems. The SSH protocol enables systems to communicate in an encrypted and secure fashion over an insecure network.

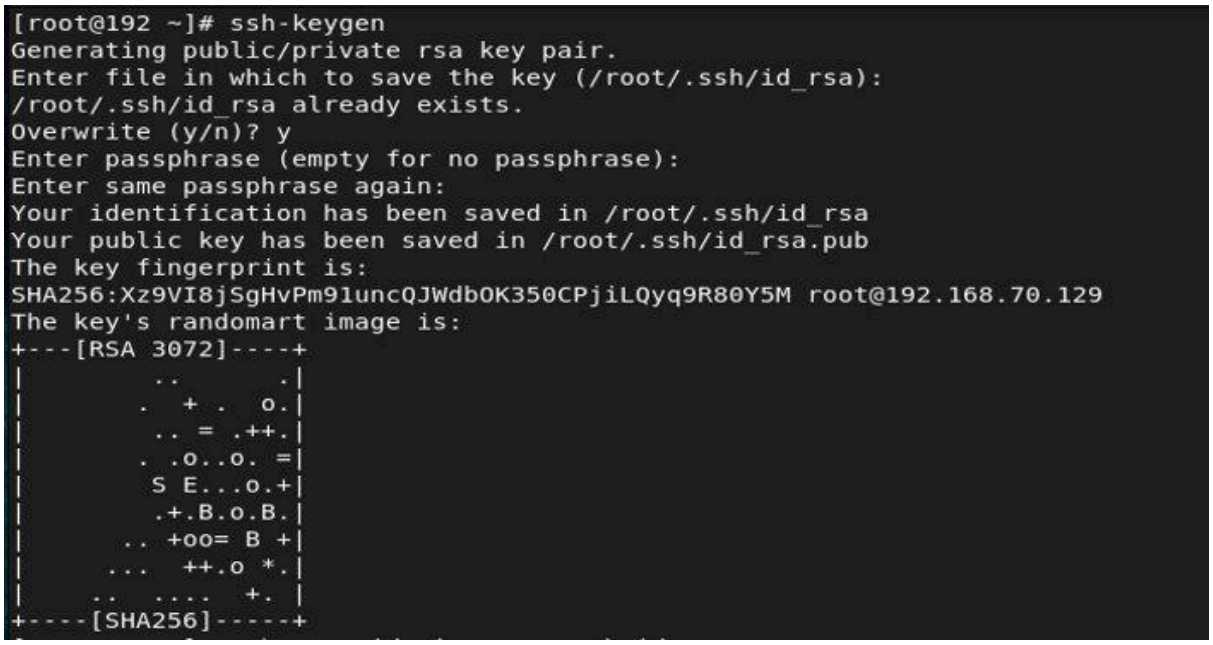
Ni

You can use the ssh command to create a secure connection to a remote system, authenticate as a specific user, and get an interactive shell session on the remote system as that user. You may also use the ssh command to run an individual command on the remote system without running an interactive shell.

Secure Shell Examples The following ssh command would log you in on the remote server remotehost using the same user name as the current local user. In this example, the remote system prompts you to authenticate with that user's password

```
[user01@host ~]$ ssh remotehost
user01@remotehost's password: redhat
...output omitted...
[user01@remotehost ~]$
```

SSH Keygen



Sharing Public key

Ni

```
[root@192 ~]# ssh-copy-id -i /root/.ssh/id_rsa.pub vm1@192.168.70.128
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa
.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
vm1@192.168.70.128's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'vm1@192.168.70.128'"
and check to make sure that only the key(s) you wanted were added.
```

Shared Public kay location in Remote user

```
[vm2@192 ~]$ ls -la
total 16
drwx------.  7 vm2  vm2  169 Jan 25 00:49 .
drwxr-xr-x. 12 root root 150 Jan  3 14:24 ..
-rw-r--r--.  1 vm2  vm2    0 Jan 24 21:41 1
-rw-------.  1 vm2  vm2  174 Jan 24 22:52 .bash_history
-rw-r--r--.  1 vm2  vm2   18 Nov  5  2021 .bash_logout
-rw-r--r--.  1 vm2  vm2  141 Nov  5  2021 .bash_profile
-rw-r--r--.  1 vm2  vm2  492 Nov  5  2021 .bashrc
drwx------.  2 vm2  vm2    6 Jan 24 20:42 .cache
drwxr-xr-x.  2 vm2  vm2   55 Jan 25 00:47 file
-rw-r--r--.  1 vm2  vm2    0 Jan 24 21:43 file2
drwxr-xr-x.  4 vm2  vm2   39 Dec 17 12:05 .mozilla
drwxr-xr-x.  3 vm2  vm2   31 Jan 25 00:49 r1
drwx------.  2 vm2  vm2   29 Jan 24 20:46 .ssh
```

```
[vm2@192 ~]$ cd .ssh
[vm2@192 .ssh]$ ls -l
total 4
-rw-------. 1 vm2 vm2 573 Jan 24 20:46 authorized_keys
```

Ni

SCP (Secure copy)

Copy file from local to remote user

```
[root@192 ~]# scp myfile vm2@192.168.70.128:/home/vm2/file
myfile                                        100%   12     0.1KB/s   00:00
```

Copy directory from local to remote user (use –r for dir)

```
[root@192 ~]# scp -r r1 vm2@192.168.70.128:/home/vm2/
demo1                                         100%    0     0.0KB/s   00:00
```

SCP (Secure copy)

- **Copy file from remote user to local machine**

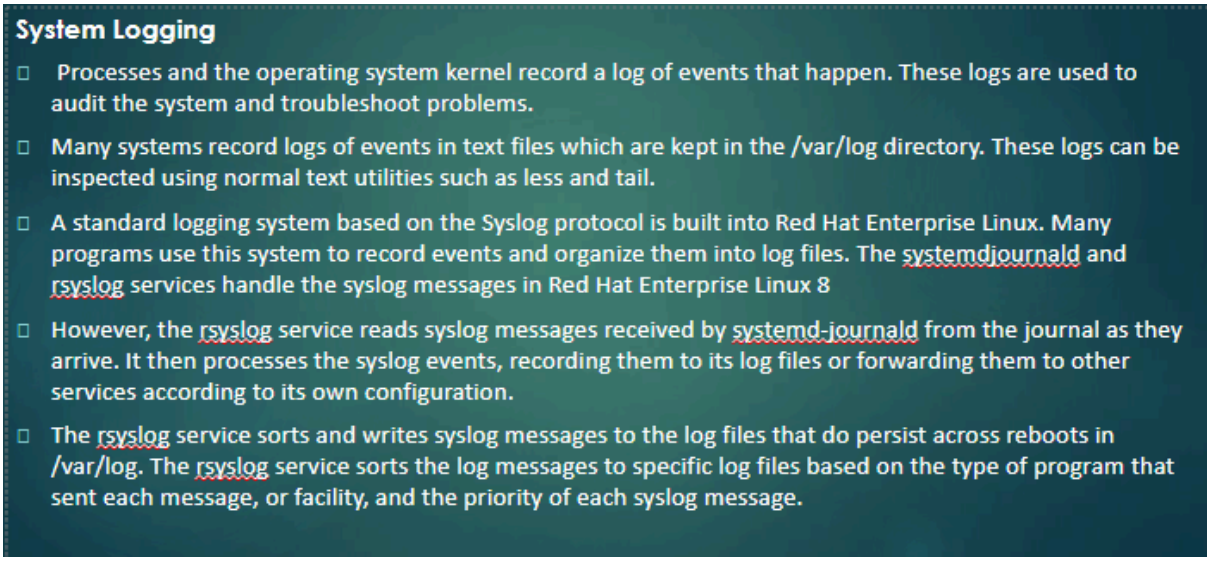
```
[root@192 ~]# scp vm2@192.168.70.128:/home/vm2/file/text1 /root/dir3
[root@192 ~]# cd dir3
-rw-r--r--  1 root  root   0 Jan 24 14:17 text1
```

- **Copy directory from remote user to local machine (use –r for dir)**

```
[root@192 ~]# scp -r vm2@192.168.70.128:/home/vm2/file /root
myfile                                        100%   12     0.7KB/s   00:00
```

Ni

Analyzing and Storing Logs



Rsyslog Configuration for both Server and client

```
Step 1: Install rsyslog

# yum install rsyslog

Step 2: To configure rsyslog using TCP:

1. Configure the remote server to accept remote log messages using TCP.

Uncomment the following lines in the MODULES section of /etc/rsyslog.conf:

#ModLoad imtcp.so
#InputTCPServerRun 514

Restart rsyslog.

[root@server ~]# service rsyslog restart

2. Configure the rsyslog server to send rsyslog events to another server using TCP.

Add the following line to the RULES section of /etc/rsyslog.conf:

# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional

#*.* @remote-host:514

*.* @@10.10.10.1:514

You can also specify the severity to send, for example info messages:

*.info @@10.10.10.1:514

Restart rsyslog.
```

Ni

```
[root@system ~]# service rsyslog restart

Step 3: Configure the remote server to accept remote log messages using UDP.

1. Configure the server to accept remote log messages using UDP.

Uncomment the following lines in the MODULES section of /etc/rsyslog.conf:

# Provides UDP syslog reception
$ModLoad imudp.so
$UDPServerRun 514

Restart rsyslog.

[root@server ~]# service rsyslog restart

2. Configure the rsyslog server to send rsyslog events to another server using UDP.

Add the following line to the RULES section of /etc/rsyslog.conf:
```

```
#*.* @remote-host:514
```

```
[root@192 ~]# rpm -q rsyslog
rsyslog-8.2102.0-105.el9.x86_64
[root@192 ~]# systemctl restart rsyslog.service
[root@192 ~]# vi /etc/rsyslog.config
[root@192 ~]# vi /etc/rsyslog.conf
[root@192 ~]# systemctl restart rsyslog.service
[root@192 ~]# fire
firefox                 firewalld
firewall-cmd            firewall-offline-cmd
[root@192 ~]# firewall-
firewall-cmd            firewall-offline-cmd
[root@192 ~]# firewall-cmd --permanent --add-port=514/tcp
success
[root@192 ~]# firewall-cmd --reload
success
[root@192 ~]# systemctl restart rsyslog.service
```

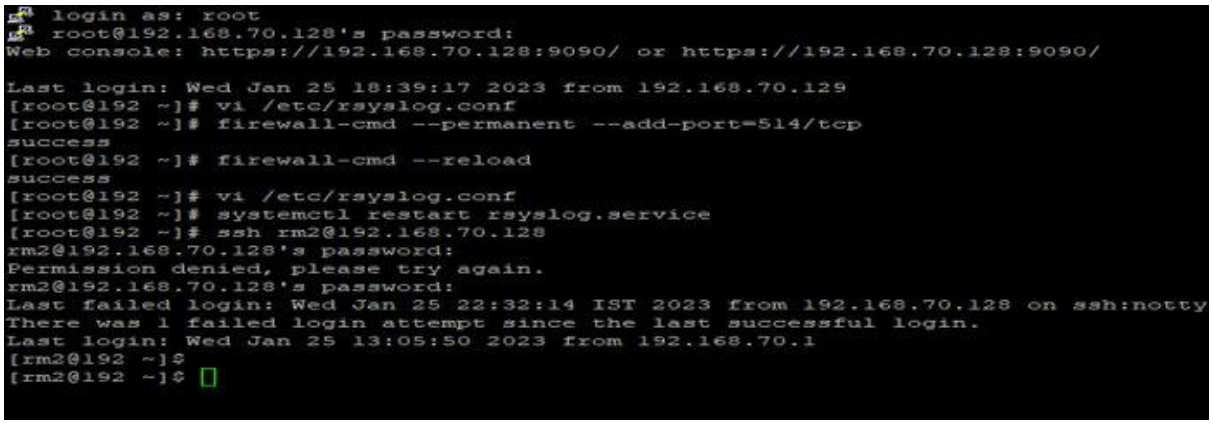
Rsyslog config file

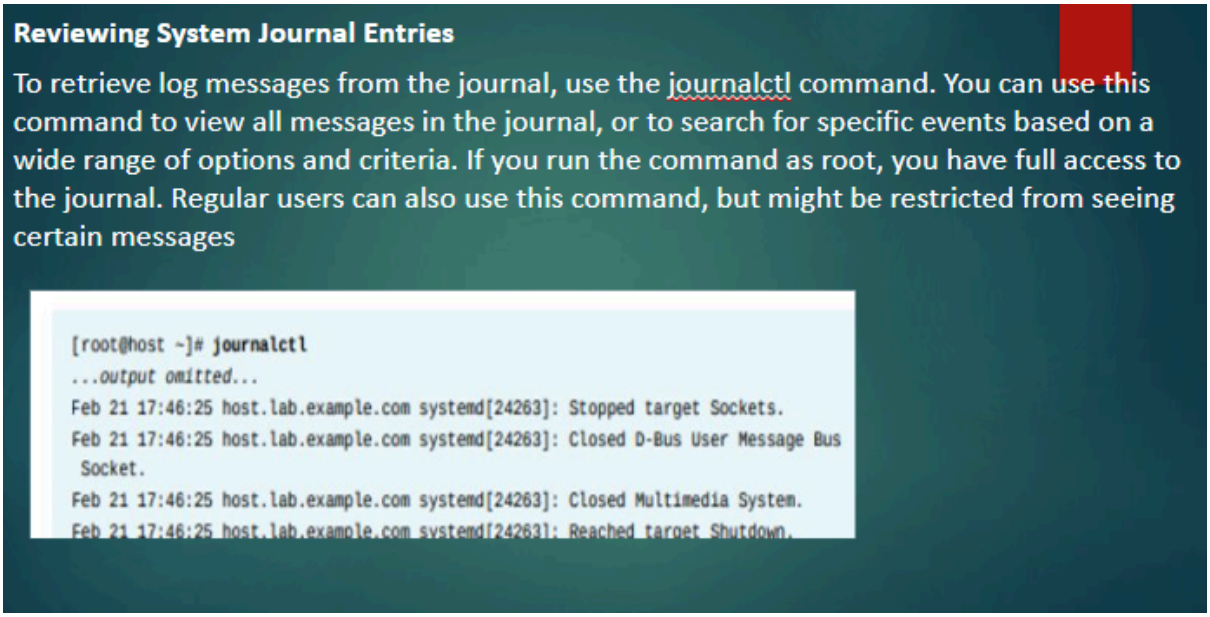
Vi /etc/rsyslog.config - uncomment → Provides TCP syslog reception→ line - module and line - input

```
# Provides UDP syslog reception
# for parameters see http://www.rsyslog.com/doc/imudp.html
#module(load="imudp") # needs to be done just once
#input(type="imudp" port="514")

# Provides TCP syslog reception
# for parameters see http://www.rsyslog.com/doc/imtcp.html
#module(load="imtcp") # needs to be done just once
#input(type="imtcp" port="514")
```

Ni

Client side Configuration





Ni

To view live logs : tail -f

```
[root@192 ~]# tail -f /var/log/secure
Jan 26 14:46:44 localhost gdm-launch-environment][1212]: pam_unix(gdm-launch-
ironment:session): session opened for user gdm(uid=42) by (uid=0)
Jan 26 14:46:48 localhost polkitd[882]: Registered Authentication Agent for u
-session:c1 (system bus name :1.25 [/usr/bin/gnome-shell], object path /org/f
desktop/PolicyKit1/AuthenticationAgent, locale en_US.UTF-8)
Jan 26 14:47:11 localhost gdm-password][2166]: gkr-pam: unable to locate daem
control file
Jan 26 14:47:11 localhost gdm-password][2166]: gkr-pam: stashed password to t
later in open session
Jan 26 14:47:11 localhost systemd[2175]: pam_unix(systemd-user:session): sess
 opened for user root(uid=0) by (uid=0)
Jan 26 14:47:11 localhost gdm-password][2166]: pam_unix(gdm-password:session)
ession opened for user root(uid=0) by (uid=0)
```

Journalctl -f command display all current logs

```
[root@192 ~]# journalctl -f
Jan 26 14:47:46 192.168.70.129 systemd[1]: systemd-hostnamed.service: Deactivat
d successfully.
Jan 26 14:47:46 192.168.70.129 systemd[1]: systemd-localed.service: Deactivated
successfully.
Jan 26 14:47:49 192.168.70.129 geoclue[1846]: Service not used for 60 seconds.
hutting down..
Jan 26 14:47:49 192.168.70.129 systemd[1]: geoclue.service: Deactivated success
ully.
Jan 26 14:47:52 192.168.70.129 realmd[2086]: quitting realmd service after time
ut
Jan 26 14:47:52 192.168.70.129 realmd[2086]: stopping service
Jan 26 14:47:52 192.168.70.129 systemd[1]: realmd.service: Deactivated successf
lly.
Jan 26 14:47:55 192.168.70.129 chronyd[907]: Selected source 162.159.200.123 (2
centos.pool.ntp.org)
Jan 26 14:48:14 192.168.70.129 systemd[2175]: Starting Virtual filesystem metad
```

Systemd Journald config file to change the storage

Ni

The **Storage** parameter in the **/etc/systemd/journald.conf** file defines whether to store system journals in a volatile manner or persistently across reboot. Set this parameter to **persistent**, **volatile**, **auto**, or **none** as follows:

- **persistent**: stores journals in the **/var/log/journal** directory which persists across reboots.

  If the **/var/log/journal** directory does not exist, the **systemd-journald** service creates it.

- **volatile**: stores journals in the volatile **/run/log/journal** directory.

  As the **/run** file system is temporary and exists only in the runtime memory, data stored in it, including system journals, do not persist across a reboot.

- **auto**: if the **/var/log/journal** directory exists, then **systemd-journald** uses persistent storage, otherwise it uses volatile storage.

  This is the default action if the **Storage** parameter is not set.

- **none**: do not use any storage. All logs are dropped but log forwarding will still work as expected.

```
[root@192 ~]# vi /etc/systemd/journald.conf
```

```
[Journal]
#Storage=auto
```

Journalctl commands

```
[root@host ~]# journalctl -p err
-- Logs begin at Wed 2019-02-20 16:01:17 +07, end at Thu 2019-02-21 18:01:01 +07.
 --
```

```
[root@host ~]# journalctl --since today
-- Logs begin at Wed 2019-02-20 16:01:17 +07, end at Thu 2019-02-21 18:31:14 +07.
 --
```

Ni

```
[root@host ~]# journalctl --since "2019-02-10 20:30:00" \
--until "2019-02-13 12:00:00"
...output omitted...
```

You can also specify all entries since a time relative to the present. For example, to specify all entries in the last hour, you can use the following command:

```
[root@host ~]# journalctl --since "-1 hour"
...output omitted...
```

```
[root@host ~]# journalctl --since today
-- Logs begin at Wed 2019-02-20 16:01:17 +07, end at Thu 2019-02-21 18:31:14 +07
 --
```

```
[root@host ~]# journalctl --since "-1 hour"
...output omitted...
```

Archiving and Transfe

Archive file in different location

```
[root@192 sample]# tar -cvf /root/archive.tar *.txt
file1.txt
file2.txt
file3.txt
file4.txt
file5.txt
[root@192 sample]# cd
[root@192 ~]# ls -l
total 72
-rw-r--r--. 1 root  root       21 Dec 29 03:08 192.168.70.128
-rw-------. 1 root  root      819 Dec 17 07:56 anaconda-ks.cfg
-rw-r--r--  1 root  root    10240 Jan 26 13:15 archive.tar
```

Ni

Extract Archive file

```
-rw-r--r-- 1 root root 10240 Jan 26 12:59 archive.tar
[root@192 sample]# tar -xvf archive.tar *.txt
file1.txt
file2.txt
file3.txt
file4.txt
file5.txt
file1.txt
[root@192 sample]# ls -l
total 16
-rw-r--r-- 1 root root 10240 Jan 26 12:59 archive.tar
-rw-r--r-- 1 root root     6 Jan 26 12:58 file1.txt
-rw-r--r-- 1 root root     0 Jan 26 08:39 file2.txt
-rw-r--r-- 1 root root     0 Jan 26 08:39 file3.txt
-rw-r--r-- 1 root root     0 Jan 26 08:39 file4.txt
-rw-r--r-- 1 root root     0 Jan 26 08:39 file5.txt
```

Creating a file for size

```
[root@192 ~]# fallocate -l +1G f4
[root@192 ~]# ls -sh f4
1.0G f4
```

Ni

Compression command: gunzip, bunzip, zip

```
[root@192 sample]# tar -zcvf gunzip.tar.gz file1.txt file2.txt
file1.txt
file2.txt
[root@192 sample]# tar -jcvf gunzip.tar.bz2 file1.txt file2.txt
file1.txt
file2.txt
```

```
[root@192 sample]# zip sample.zip file1.txt file2.txt
  adding: file1.txt (deflated 98%)
  adding: file2.txt (deflated 98%)
```

4

Check the size before and after compression of the files

Before Compression

```
[root@192 sample]# du -sh file1.txt file2.txt
12K     file1.txt
20K     file2.txt
```

After compression for Archive, gunzip, bunzip, zip

```
[root@192 sample]# du -sh archive.tar
32K     archive.tar
[root@192 sample]# du -sh gunzip.tar.bz2
4.0K    gunzip.tar.bz2
[root@192 sample]# du -sh gunzip.tar.gz
4.0K    gunzip.tar.gz
```

```
[root@192 sample]# du -sh sample.zip
4.0K    sample.zip
```

Ni

List the compressed file

```
[root@192 sample]# ls -l
total 76
-rw-r--r-- 1 root root 30720 Jan 26 13:38 archive.tar
-rw-r--r-- 1 root root 10138 Jan 26 13:30 file1.txt
-rw-r--r-- 1 root root 16400 Jan 26 13:30 file2.txt
-rw-r--r-- 1 root root     0 Jan 26 08:39 file3.txt
-rw-r--r-- 1 root root     0 Jan 26 08:39 file4.txt
-rw-r--r-- 1 root root     0 Jan 26 08:39 file5.txt
-rw-r--r-- 1 root root   610 Jan 26 13:33 gunzip.tar.bz2
-rw-r--r-- 1 root root   502 Jan 26 13:33 gunzip.tar.gz
drwxr-xr-x 2 root root     6 Jan 26 13:09 s1
-rw-r--r-- 1 root root   858 Jan 26 13:40 sample.zip
```

Extract file command: gunzip, bunzip, zip

```
[root@192 sample]# tar -zxvf gunzip.tar.gz file1.txt file2.txt
file1.txt
file2.txt
```

```
[root@192 sample]# tar -jxvf gunzip.tar.bz2 file1.txt file2.txt
file1.txt
file2.txt
```

```
[root@192 sample]# unzip sample.zip file1.txt file2.txt
Archive:  sample.zip
  inflating: file1.txt
  inflating: file2.txt
```

Ni

TCP/IP Network Model

The *TCP/IP network model* is a simplified, four-layered set of abstractions that describes how different protocols interoperate in order for computers to send traffic from one machine to another over the Internet. It is specified by RFC 1122, *Requirements for Internet Hosts -- Communication Layers*. The four layers are:

- **Application**

  Each application has specifications for communication so that clients and servers may communicate across platforms. Common protocols include SSH (remote login), HTTPS (secure web), NFS or CIFS (file sharing), and SMTP (electronic mail delivery).

- **Transport**

  Transport protocols are TCP and UDP. TCP is a reliable connection-oriented communication, while UDP is a connectionless *datagram* protocol. Application protocols use TCP or UDP ports. A list of well-known and registered ports can be found in the **`/etc/services`** file.

  When a packet is sent on the network, the combination of the service port and IP address forms a *socket*. Each packet has a source socket and a destination socket. This information can be used when monitoring and filtering.

- **Internet**

  The Internet, or network layer, carries data from the source host to the destination host. The IPv4 and IPv6 protocols are Internet layer protocols. Each host has an IP address and a prefix used to determine network addresses. Routers are used to connect networks.

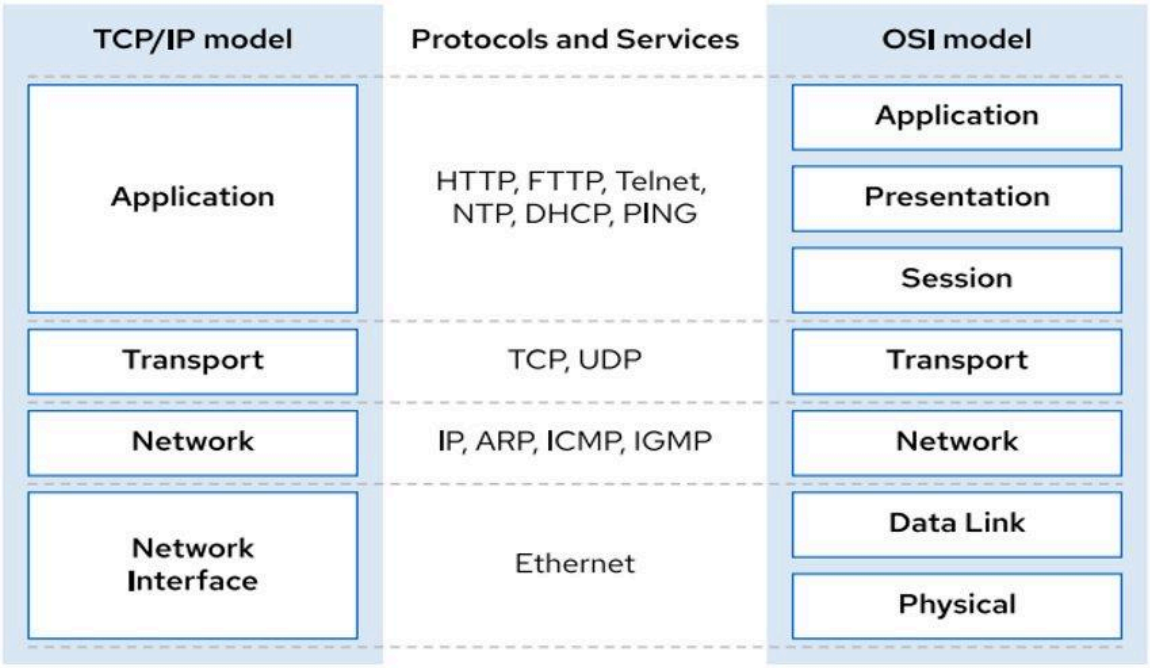


Figure 12.1: Comparison of the TCP/IP and OSI network models

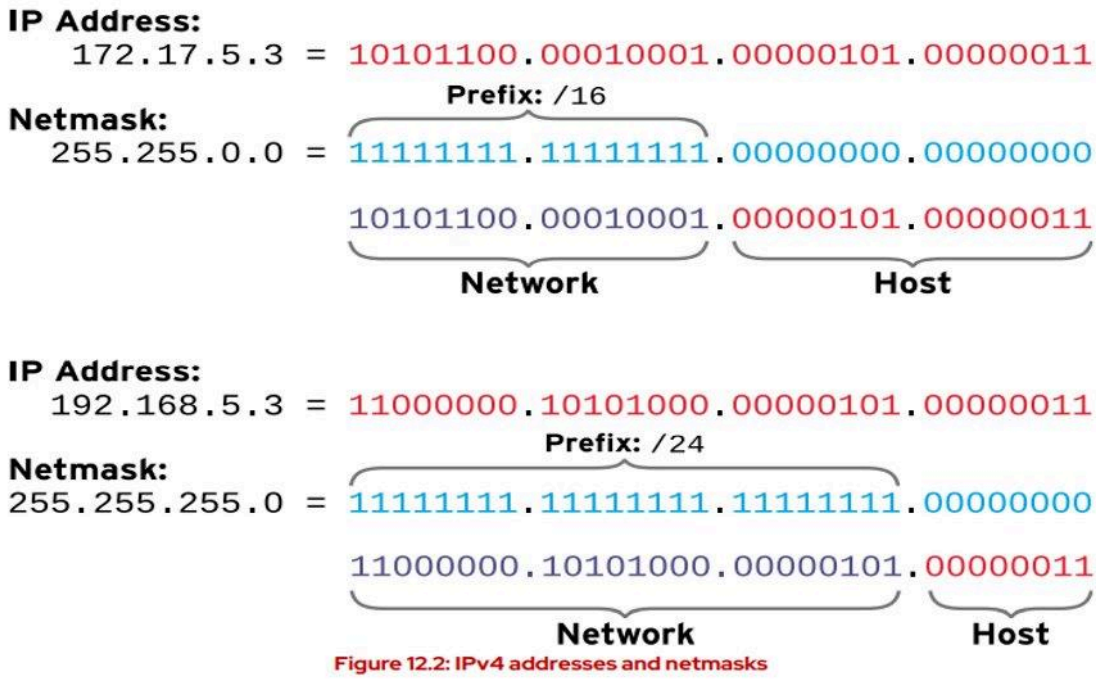


Figure 12.2: IPv4 addresses and netmasks

```
[root@192 ~]# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.70.129  netmask 255.255.255.0  broadcast 192.168.70.255
        inet6 fe80::20c:29ff:feb7:c55  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:b7:0c:55  txqueuelen 1000  (Ethernet)
        RX packets 36976  bytes 48659871 (46.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14684  bytes 862624 (842.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 00:0c:29:b7:0c:5f  txqueuelen 1000  (Ethernet)
        RX packets 2353  bytes 160102 (156.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 137  bytes 9619 (9.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 137  bytes 9619 (9.3 KiB)
```

Ni

```
[root@localhost ~]# ifup ens224
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
[root@localhost ~]# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.70.129  netmask 255.255.255.0  broadcast 192.168.70.255
        inet6 fe80::20c:29ff:fe18:6860  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:18:68:60  txqueuelen 1000  (Ethernet)
        RX packets 19391  bytes 26439749 (25.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7683  bytes 431305 (421.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.70.131  netmask 255.255.255.0  broadcast 192.168.70.255
        ether 00:0c:29:18:68:74  txqueuelen 1000  (Ethernet)
        RX packets 107  bytes 7814 (7.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 46  bytes 4308 (4.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
[root@192 ~]# nmcli
ens160: connected to ens160
        "VMware VMXNET3"
        ethernet (vmxnet3), 00:0C:29:B7:0C:55, hw, mtu 1500
        ip4 default
        inet4 192.168.70.129/24
        route4 default via 192.168.70.2 metric 100
        route4 192.168.70.0/24 metric 100
        inet6 fe80::20c:29ff:feb7:c55/64
        route6 fe80::/64 metric 1024

ens192: disconnected
        "VMware VMXNET3"
        ethernet (vmxnet3), 00:0C:29:B7:0C:5F, hw, mtu 1500

lo: unmanaged
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536

DNS configuration:
        servers: 192.168.70.2
        domains: localdomain
        interface: ens160

Use "nmcli device show" to get complete information about known devices and
"nmcli connection show" to get an overview on active connection profiles.
```

```
[root@192 ~]# nmcli device status
DEVICE  TYPE      STATE         CONNECTION
ens160  ethernet  connected     ens160
ens192  ethernet  disconnected  --
lo      loopback  unmanaged     --
```

Ni

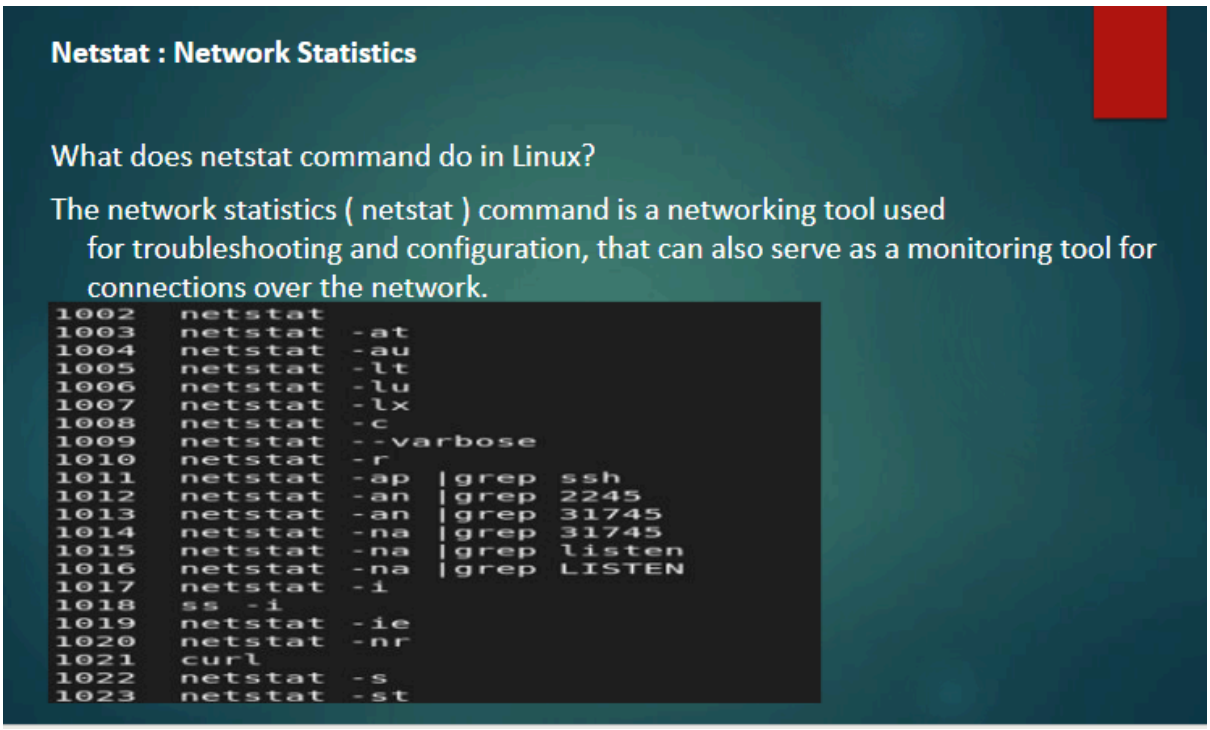
```
[root@192 ~]# nmcli device  show ens160
GENERAL.DEVICE:                         ens160
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         00:0C:29:B7:0C:55
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     ens160
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnection/1
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         192.168.70.129/24
IP4.GATEWAY:                            192.168.70.2
IP4.ROUTE[1]:                           dst = 0.0.0.0/0, nh = 192.168.70.2, mt = 100
IP4.ROUTE[2]:                           dst = 192.168.70.0/24, nh = 0.0.0.0, mt = 100
IP4.DNS[1]:                             192.168.70.2
IP4.DOMAIN[1]:                          localdomain
IP6.ADDRESS[1]:                         fe80::20c:29ff:feb7:c55/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 1024
```

To assign ip-address to network adapter Manually:

nmcli connection add type ethernet con-name ens192 ifname ens192 ipv4.gateway 192.168.233.2 ipv4.addresses 192.168.233.50/24 autoconnect yes ipv4.method manual

To assign ip-address to network adapter Automatically:

nmcli connection add type ethernet con-name ens224 ifname ens224 autoconnect yes ipv4.method auto



Ni

Troubleshooting ports and services

TCP services use sockets as end points for communication and are made up of an IP address, protocol, and port number. Services typically listen on standard ports while clients use a random available port. Well-known names for standard ports are listed in the **/etc/services** file.

The **ss** command is used to display socket statistics. The **ss** command is meant to replace the older tool **netstat**, part of the *net-tools* package, which may be more familiar to some system administrators but which is not always installed.

```
[user@host ~]$ ss -ta
State      Recv-Q Send-Q      Local Address:Port          Peer Address:Port
LISTEN     0      128                 *:sunrpc                   *:*
LISTEN     0      128                 *:ssh                      *:*
LISTEN     0      100         127.0.0.1:smtp                     *:*
LISTEN     0      128                 *:36889                    *:*
ESTAB      0      0       172.25.250.10:ssh          172.25.254.254:59392
LISTEN     0      128              :::sunrpc                  :::*
LISTEN     0      128              :::ssh                     :::*
LISTEN     0      100             ::1:smtp                    :::*
LISTEN     0      128              :::34946                   :::*
```

1. The port used for SSH is listening on all IPv4 addresses. The "*" is used to represent "all" when referencing IPv4 addresses or ports.
2. The port used for SMTP is listening on the **127.0.0.1** IPv4 loopback interface.
3. The established SSH connection is on the 172.25.250.10 interface and originates from a system with an address of **172.25.254.254**.
4. The port used for SSH is listening on all IPv6 addresses. The "::" syntax is used to represent all IPv6 interfaces.
5. The port used for SMTP is listening on the ::1 IPv6 loopback interface.

Option	Description
-n	Show numbers instead of names for interfaces and ports.
-t	Show TCP sockets.
-u	Show UDP sockets.
-l	Show only listening sockets.
-a	Show all (listening and established) sockets.

Option	Description
-p	Show the process using the sockets.
-A inet	Display active connections (but not listening sockets) for the **inet** address family. That is, ignore local UNIX domain sockets.

Ni

Configuring Host Names and Name Resolution

```
[root@192 ~]# hostname
192.168.70.129
[root@192 ~]# hostname
hostname     hostnamectl
[root@192 ~]# hostnamectl set-hostname raghav.com
[root@192 ~]# hostname
raghav.com
[root@192 ~]# hostnamectl
 Static hostname: raghav.com
       Icon name: computer-vm
         Chassis: vm
      Machine ID: 2a6565f7736140cb9576ed323c0073ed
         Boot ID: 79a31abcffc44c8eacb7fe574d4fa1fc
  Virtualization: vmware
Operating System: CentOS Stream 9
     CPE OS Name: cpe:/o:centos:centos:9
          Kernel: Linux 5.14.0-115.el9.x86_64
```

Installing and Updating Software Packages

Software packages and RPM

The RPM Package Manager, originally developed by Red Hat, provides a standard way to package software for distribution. Managing software in the form of RPM packages is much simpler than working with software that has simply been extracted into a file system from an archive.

It lets administrators track which files were installed by the software package and which ones need to be removed if it is uninstalled, and check to ensure that supporting packages are present when it is installed. Information about installed packages is stored in a local RPM database on each system.

All software provided by Red Hat for Red Hat Enterprise Linux is provided as an RPM package.

Ni

RPM package files names consist of four elements (plus the **.rpm** suffix): **name-version-release.architecture**:

Figure 14.3: RPM file name elements

- NAME is one or more words describing the contents (coreutils).
- VERSION is the version number of the original software (8.30).
- RELEASE is the release number of the package based on that version, and is set by the packager, who might not be the original software developer (4.el8).
- ARCH is the processor architecture the package was compiled to run on. **noarch** indicates that this package's contents are not architecture-specific (as opposed to **x86_64** for 64-bit, **aarch64** for 64-bit ARM, and so on).

Ni

rpm Commands

The general form of a query is:

- **`rpm -q [select-options] [query-options]`**

RPM queries: General information about installed packages

- **`rpm -qa`**: List all installed packages
- **`rpm -qf FILENAME`**: Find out what package provides FILENAME

```
[user@host ~]$ rpm -qf /etc/yum.repos.d
redhat-release-8.0-0.39.el8.x86_64
```

RPM queries: Information about specific packages

- **`rpm -q`**: List what version of the package is currently installed

```
[user@host ~]$ rpm -q yum
yum-4.0.9.2-4.el8.noarch
```

- **`rpm -qi`**: Get detailed information about the package
- **`rpm -ql`**: List the files installed by the package

Ni

```
[root@example ~]# rpm -q yum
yum-4.12.0-2.el9.noarch
```

```
[root@example ~]# rpm -qi yum
Name        : yum
Version     : 4.12.0
Release     : 2.el9
Architecture: noarch
Install Date: Sat 17 Dec 2022 07:48:02 AM EST
Group       : Unspecified
Size        : 77593
License     : GPLv2+
Signature   : RSA/SHA256, Thu 26 May 2022 04:40:28 AM EDT, Key ID 05
5d
Source RPM  : dnf-4.12.0-2.el9.src.rpm
Build Date  : Mon 23 May 2022 09:59:30 AM EDT
Build Host  : x86-02.stream.rdu2.redhat.com
Packager    : builder@centos.org
Vendor      : CentOS
URL         : https://github.com/rpm-software-management/dnf
Summary     : Package manager
Description :
Utility that allows users to manage packages on their systems.
It supports RPMs, modules and comps groups & environments.
```

```
[root@example ~]# rpm -qa|wc -l
1174
```

```
[root@example ~]# rpm -ql yum
/etc/dnf/protected.d/yum.conf
/etc/yum.conf
/etc/yum/pluginconf.d
/etc/yum/protected.d
/etc/yum/vars
/usr/bin/yum
/usr/share/man/man1/yum-aliases.1.gz
/usr/share/man/man5/yum.conf.5.gz
/usr/share/man/man8/yum-shell.8.gz
/usr/share/man/man8/yum.8.gz
```

Ni

- **`rpm -qc`**: List just the configuration files installed by the package

```
[user@host ~]$ rpm -qc openssh-clients
/etc/ssh/ssh_config
/etc/ssh/ssh_config.d/05-redhat.conf
```

- **`rpm -qd`**: List just the documentation files installed by the package

```
[user@host ~]$ rpm -qd openssh-clients
/usr/share/man/man1/scp.1.gz
/usr/share/man/man1/sftp.1.gz
/usr/share/man/man1/ssh-add.1.gz
/usr/share/man/man1/ssh-agent.1.gz
/usr/share/man/man1/ssh-copy-id.1.gz
/usr/share/man/man1/ssh-keyscan.1.gz
/usr/share/man/man1/ssh.1.gz
/usr/share/man/man5/ssh_config.5.gz
/usr/share/man/man8/ssh-pkcs11-helper.8.gz
```

Installing RPM Packages

The **`rpm`** command can also be used to install an RPM package that you have downloaded to your local directory.

```
[root@host ~]# rpm -ivh wonderwidgets-1.0-4.x86_64.rpm
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:wonderwidgets-1.0-4              ################################# [100%]
[root@host ~]#
```

However, the next section of this chapter will discuss a more powerful tool for managing RPM installation and updates from the command line, **`yum`**.

Ni

```
[root@example Packages]# rpm -ivh vsftpd-3.0.3-49.el9.x86_64.rpm
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:vsftpd-3.0.3-49.el9              ################################# [100%]
```

```
[root@example Packages]# rpm -ivh vsftpd-3.0.3-49.el9.x86_64.rpm --force
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:vsftpd-3.0.3-49.el9              ################################# [100%]
```

```
[root@example Packages]# rpm -qa |grep -i ftp
vsftpd-3.0.3-49.el9.x86_64
```

Package Update

```
[root@example Packages]# rpm -Uvh firefox-91.10.0-1.el9.x86_64.rpm
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
       package firefox-91.10.0-1.el9.x86_64 is already installed
```

Nodeps command

```
[root@example Packages]# rpm -ivh yajl-2.1.0-20.el9.i686.rpm --nodeps
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:yajl-2.1.0-20.el9                ################################# [100%]
```

Uninstall commands

```
[root@example Packages]# rpm -qa |grep -i ftp
vsftpd-3.0.3-49.el9.x86_64
[root@example Packages]# rpm -ev vsftpd
Preparing packages...
vsftpd-3.0.3-49.el9.x86_64
[root@example Packages]# rpm -qa |grep -i ftp
```

Ni

Task:	Command:
List installed and available packages by name	`yum list [NAME-PATTERN]`
List installed and available groups	`yum group list`
Search for a package by keyword	`yum search KEYWORD`
Show details of a package	`yum info PACKAGENAME`
Install a package	`yum install PACKAGENAME`
Install a package group	`yum group install GROUPNAME`
Update all packages	`yum update`
Remove a package	`yum remove PACKAGENAME`
Display transaction history	`yum history`

Yum rep configuration

```
1019  rpm -qa|grep firewall
1020  systemctl status firewalld.service
1021  systemctl disable firewalld.service
1022  systemctl status firewalld.service
1023  vi /etc/selinux/config
1024  mkdir /var/ftp/repo
1025  lsblk
1026  cd /run/media/root/CentOS-Stream-9-BaseOS-x86_64
1027  ll
1028  cp -av * /var/ftp/repo
1029  vi /etc/yum.repos.d/create.repos
1030  yum clean all
1031  cd /etc/yum.repos.d/
1032  ll
1033  rm -rf centos-addons.repo centos.repo
1034  ll
1035  cat create.repos
1036  yum list all
1037  c
1038  cd
1039  history
```

Ni

```
[baseos]
name=createbaseos
baseurl=file:///var/ftp/repos/BaseOS
enabled=1
gpgcheck=0



[appstream]
name=createappstream
baseurl=file:///var/ftp/repos/AppStream
enabled=1
gpgcheck=0
```

Ni

Ni