

4. Basic of Linux

Contents

1. Intro to Linux
2. Basic CLI commands
3. Understanding files in Linux
4. Filters and redirection
5. Users and groups
6. Sudo
7. Software management
8. Services and processes
9. good to know commands
10. server management

4 areas we will work

1. Commands
2. Files
3. Softwares
4. Servers

Open Source

It is a software with source code that anyone can inspect, modify, enhance and redistribute.

Linux is an open source kernel.

Linux origin

1984 : The GNU Project and the Free Software Foundation

- Creates open source version of UNIX utilities
- Creates the General Public License (GPL)
 - Software license enforcing open source principles.

1991 : Linus Torvalds

- Creates open source, UNIX-like kernel, released under the GPL

- Pors some GNU utilities, solicits assistance online

Today:

- Linux Kernel + GNU utilities = complete, open source, UNIX-like operating system
 - Packaged for targeted audiences as distributions.
-

Linux Principles

- Everything is a file
- Small single purpose Programs
- Ability to chain programs together for complex operations
- Avoid Captive User Interface (No Next next next finish)
- Configuration data stored in text file

Why Linux?

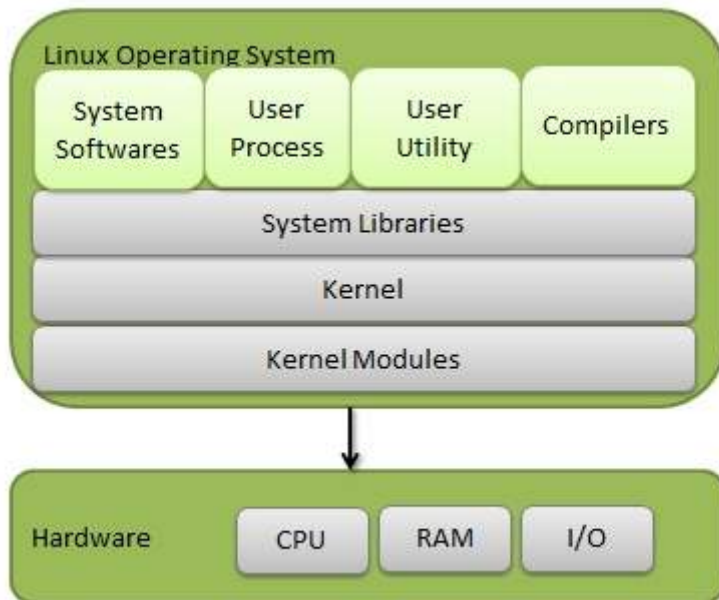
- Opensource
- Community Support (Get new updates)
- Support Wide Variety of hardware
- Customization
- Most servers runs on linux
- Automation

Components of Linux System

Kernel – Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.

System Library – System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features. These libraries implement most of the functionalities of the operating system and do not requires kernel module's code access rights.

System Utility – System Utility programs are responsible to do specialized, individual level tasks.

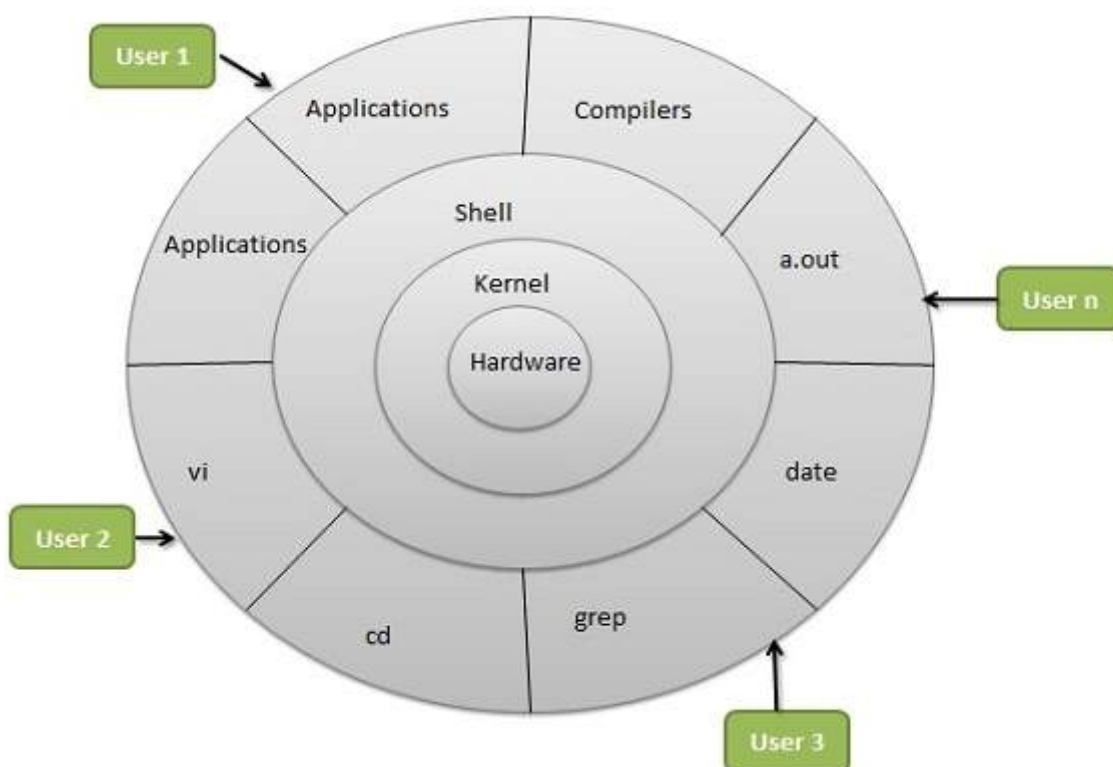


Kernel Mode vs User Mode

Kernel component code executes in a special privileged mode called **kernel mode** with full access to all resources of the computer. This code represents a single process, executes in single address space and do not require any context switch and hence is very efficient and fast. Kernel runs each processes and provides system services to processes, provides protected access to hardware to processes.

Support code which is not required to run in kernel mode is in System Library. User programs and other system programs works in **User Mode** which has no access to system hardware and kernel code. User programs/ utilities use System libraries to access Kernel functions to get system's low level tasks.

Architecture of Linux



Hardware - Computery sources : RAM CPU Networks

Kernel - Understand the hardware and pass signal to Shell. It is the core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.

Shell – An interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes kernel's functions.

Utilities – Utility programs that provide the user most of the functionalities of an operating systems.

Popular Linux distros

- Ubuntu Linux
- Mint Linux
- Arch Linux
- Fedora
- Debian
- OpenSuse

Popular Server Linux OS

- Red hat Enterprise Linux (Considered Most stable and secure Linux OS and it is not Open Source)
- Ubuntu Server
- Centos
- SUSE Enterprise Linux

Most used Linux distros currently in IT industry

- ROM based :- RHEL(Red Hat Enterprise Linux), Centos, Oracle Linux
- Debian based :- Ubuntu Server, Kali Linux

Some Important Directories

- Home Directories: `/root`, `/home/username`
- User Executable: `/bin`, `/usr/bin`, `/usr/local/bin`
- System Executables: `/sbin`, `/usr/sbin`, `/usr/local/sbin`
- Other Mountpoints: `/media`, `/mnt`
- Configuration: `/etc`
- Temporary Files: `/tmp`
- Kernels and Bootloader: `/boot`
- Server Data: `/var`, `/srv`
- System Information: `/proc`, `/sys`
- Shared Libraries: `/lib`, `/usr/lib`, `/usr/local/lib`

```
$ cat /etc/os-release - displays OS info
```

absolute path always begins with root directory for example

```
$ cd /bin
```

Relative Path begins with the pathname that is inside the directory you are in for example

```
$ cd dev/ - dev is a folder in your directory
```

To copy directory use

```
$ cp -r directory path
```

Syntax for commands in linux

command - `ls`

option - `-l`

argument - `path or anyother things`

command - `cp`

option - `-r`

argument - `path or anyother things`

VIM editor

To create a file

```
$ vim firstfile.txt
```

There are three types of VIM editor

1. Command mode (esc)
2. Insert mode (edit mode) (i and o)
3. extended command mode (:) - for saving files

Extended command mode

`Esc+:w` - To Save the changes

`Esc+:q` - To quit (Without saving)

`Esc+:wq` - To save and quit

`Esc+:w!` - To save forcefully

`Esc+wq!` - To save and quit forcefully

`Esc+:x` - To save and quit

`Esc+:X` - To give passw or d to the file and remove password

`Esc+:20 (n)` - To go to line no 20 or n

`Esc+: se nu` - To set the line numbers to the file !

`Esc+:se nonu` - To Remove the set line numbers

Command mode

`G` - go to the last line

`gg` - go to line no.1

`yy` - copying (also called yunking) the line your cursor is on

`p` - to paste it below the line

`P` - to paste it above the line

`4yy` - copy 4 lines of code

`dd` - Delete the current line (also Cut)

`u` - undo

`5dd` - Delete 5 lines

`/` - search keyword (for eg : `/network`)

`ctrl + R` - to redo changes

`x` - delete key

Types of files in linux

Types of files in linux.

File Type	First Character in File Listing	Description
Regular file	<code>*</code>	Normal files such as text, data, or executable files
Directory	<code>d</code>	Files that are lists of other files
Link	<code>l</code>	A shortcut that points to the location of the actual file
Special file	<code>c</code>	Mechanism used for input and output, such as files in <code>/dev</code>
Socket	<code>s</code>	A special file that provides inter-process networking protected by the file system's access control
Pipe	<code>p</code>	A special file that allows processes to communicate with each other without using network socket semantics

Making a soft link

```
ln -s /opt/dev/ops/devops/test/commands.txt cmds
```

```
[vagrant@localhost ~]$ ln -s /opt/dev/ops/devops/test/commands.txt cmds
[vagrant@localhost ~]$ ls -l
total 4
lrwxrwxrwx. 1 vagrant vagrant 37 May 24 12:29 cmds -> /opt/dev/ops/devops/test/commands.txt
-rw-rw-r--. 1 vagrant vagrant 120 May 24 02:30 firstfile.txt
[vagrant@localhost ~]$ cat cmds
ls
pwd
whoami
cd
uptime
touch
mkdir
[vagrant@localhost ~]$ |
```

To remove the link

use `rm {linkname}` or `unlink {linkname}`

For viewing the file in timestamps(Latest in the first

```
ls -lt
```

In reverse (Latest one in the last)

```
ls -ltr
```

ls command options

Options	Description
-l	Long listing format of files and directories, one per line
-a	List all hidden files and directories started with '.'
-F	Add a '/' classification at the end of each Directory
-g	List all files and directories with the group name
-i	Print index number of each files and directories
-m	List all file and directories separated by comma ','
-n	List numeric UID and GID of Owner and Groups
-r	List all files and directories in reverse order
-R	Short list all directories
-t	Sorted by modified time, started with the newest file

Filter & IO redirections (InputOutput redirection)

Grep

`grep -i firewall text.txt` : -i will ignore the case sensitive, firewall is the word we are looking in text.txt file.

actually it is

`grep -i firewall < text.txt` : but the `<` is invisible and it comes by default. but what it means is that the `firewall` is an input for the `text.txt` file to look in it.

`grep -i firewall *` : all the files in the current working directory, ignoring the case sensitivity

`grep -iR firewall *` : will also check the input inside the directory, ignoring the case sensitivity

It is very handfull command because it will ease out to look for keywords in various files in a short time

```
root@centos7:~  
[root@centos7 ~]# grep -R SELINUX /etc/*  
grep: /etc/alternatives/mta-mailqman: No such file or directory  
grep: /etc/alternatives/mta-newaliasesman: No such file or directory  
grep: /etc/alternatives/mta-sendmailman: No such file or directory  
grep: /etc/alternatives/mta-aliasesman: No such file or directory  
/etc/selinux/config:# SELINUX= can take one of these three values:  
/etc/selinux/config:SELINUX=permissive  
/etc/selinux/config:# SELINUXTYPE= can take one of three values:  
/etc/selinux/config:SELINUXTYPE=targeted  
/etc/selinux/targeted/contexts/x_contexts:property _SELINUX_*          system_u:object_r:seclabel_xproperty_t:s0  
/etc/sysconfig/selinux:# SELINUX= can take one of these three values:  
/etc/sysconfig/selinux:SELINUX=permissive  
/etc/sysconfig/selinux:# SELINUXTYPE= can take one of three values:  
/etc/sysconfig/selinux:SELINUXTYPE=targeted
```

`grep -vi firewall text.txt` : `-v` is a reverse search meaning it will search everything except firewall with this, ignoring the case sensitivity (because there is `-i`)

`less text.txt` - less is a reader, it helps to read contents of the file, can also use `/` to search words.
press q to quit

`more text.txt` - is something like a reader which shows the content in a percentage.

`head text.txt` - first 10 lines

`head -n text.txt` - first n lines

same with tail

tail also has a function to show dynamic content which will show any changes on the file. `tail -f text.txt`

```
[root@centos7 log]# tail -f /var/log/messages  
May 24 14:17:48 localhost systemd-logind: Removed session 7.  
May 24 14:17:49 localhost kernel: 14:17:49.198746 timesync vgsvcTimeSyncWorker: Radical guest time change: 4 319 501 354 000ns (GuestNow=1 65  
401 869 198 727 000 ns GuestLast=1 653 397 549 697 373 000 ns fSetTimeLastLoop=true )  
May 24 14:18:19 localhost systemd-logind: New session 8 of user vagrant.  
May 24 14:18:19 localhost systemd: Started Session 8 of user vagrant.  
May 24 14:20:52 localhost chronyd[677]: Selected source 162.159.200.1  
May 24 14:23:03 localhost systemd-logind: New session 9 of user vagrant.  
May 24 14:23:03 localhost systemd: Started Session 9 of user vagrant.  
May 24 14:23:21 localhost systemd-logind: Removed session 9.  
May 24 14:23:48 localhost systemd-logind: New session 10 of user vagrant.  
May 24 14:23:48 localhost systemd: Started Session 10 of user vagrant.  
May 24 14:25:07 localhost systemd-logind: New session 11 of user vagrant.  
May 24 14:25:07 localhost systemd: Started Session 11 of user vagrant.
```

```
vagrant@centos7:~  
hello world@DESKTOP-1BL5LTS MINGW64 ~  
$ cd /c/vagrant-vms/centos7/  
hello world@DESKTOP-1BL5LTS MINGW64 /c/vagrant-vms/centos7  
$ vagrant ssh  
Last login: Tue May 24 14:23:48 2022 from 10.0.2.2  
[vagrant@centos7 ~]$
```

cut command

can be used in files to separate columns with the command

`cut -d: -f1 /etc/passwd` : where -d is a delimiter f1 is field 1 and /etc/passwd is the file


```
[root@centos7 log]# cut -d: -f1 /etc/passwd
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
operator
games
ftp
nobody
systemd-network
dbus
polkitd
sshd
postfix
chrony
vagrant
rpc
rpcuser
nfsnobody
tss
vboxadd
[root@centos7 log]#
```

this is good only if you have proper separator with something else we can use awk

awk is an intelligent filter tool which has lots of options

eg

```
awk -F':' '{print $1}' /etc/passwd : gets the first column of the file.
```

Moderna has create vaccine for coronavirus.
for coronavirus pfizer has also created vaccine.
coronavirus spreads through water droplets in air, coronavirus is a pandemic.

```
:%s/coronavirus/covid19|
```

`:%s/coronavirus/covid19` : but the problem is if you do this method it will not replace the same word if it is repeated in the same line.

```
:%s/coronavirus/covid19/g : which will replace globally
```

```
:%s/coronavirus//g : will replace coronavirus with nothing.
```

eg

```
sed 's/coronavirus/covid19/g' text.txt
```

(or whatever file) : but this will only print it just to see what you are changing for making actual change -i should be added which will be

```
sed -i 's/coronavirus/covid19/g' text.txt
```

input redirection : <

output redirection : >

`uptime > /tmp/sysinfo.txt` : for overwriting or creating the file.

`uptime >> /tmp/sysinfo.txt` : for appending

`free -m` : shows you the RAM info

`df -h` : shows you your harddisk info

`echo "Good morning"` : print command

we can use echo to put things in file by redirection. for eg

`echo "hello" >> /tmp/sysinfo.txt`

basically it is the same kind of concept

now for something useful we can do like this as well

`echo "#####" >> /tmp/sysinfo.txt`

`date >> /tmp/sysinfo.txt`

`echo "#####" >> /tmp/sysinfo.txt`

`uptime >> /tmp/sysinfo.txt`

`echo "#####" >> /tmp/sysinfo.txt`

`free -m >> /tmp/sysinfo.txt`

`echo "#####" >> /tmp/sysinfo.txt`

`df -h >> /tmp/sysinfo.txt`

`echo "#####" >> /tmp/sysinfo.txt`

```
[root@centos7 ~]# echo "#####" >> /tmp/sysinfo.txt
[root@centos7 ~]# date >> /tmp/sysinfo.txt
[root@centos7 ~]# echo "#####" >> /tmp/sysinfo.txt
[root@centos7 ~]# uptime >> /tmp/sysinfo.txt
[root@centos7 ~]# echo "#####" >> /tmp/sysinfo.txt
[root@centos7 ~]# free -m >> /tmp/sysinfo.txt
[root@centos7 ~]# echo "#####" >> /tmp/sysinfo.txt
[root@centos7 ~]# df -h >> /tmp/sysinfo.txt
[root@centos7 ~]# echo "#####" >> /tmp/sysinfo.txt
[root@centos7 ~]#
[root@centos7 ~]# cat /tmp/sysinfo.txt
#####
Tue May 24 15:01:37 UTC 2022
#####
15:01:59 up 1:29, 4 users, load average: 0.00, 0.01, 0.02
#####
              total        used        free      shared  buff/cache   available
Mem:           486          108          232           4          146          361
Swap:          1023           0          1023
#####
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        233M   0 233M   0% /dev
tmpfs           244M   0 244M   0% /dev/shm
tmpfs           244M  4.6M 239M   2% /run
tmpfs           244M   0 244M   0% /sys/fs/cgroup
/dev/mapper/centos-root 50G  1.7G  49G   4% /
/dev/mapper/centos-home 28G   33M  28G   1% /home
/dev/sda1       1014M 167M  848M  17% /boot
vagrant         447G 214G 234G  48% /vagrant
tmpfs           49M   0  49M   0% /run/user/1000
#####
[root@centos7 ~]#
```

by using this much command you just created a log file so easily and it will be very hand when writing bash scripts

About /dev/null : it is something like a blackhole in the linux whatever you throw in it , its gone
eg

```
yum install vim -y > /dev/null
```

it can also be used in other ways such as:

```
cat /dev/null > /tmp/sysinfo.txt
```

all the contents of the sysinfo.txt is gone because we have given /dev/null as an input

we can also use >> to show errors fo eg

free -m >> /tmp/error.log in this it doesnt show the number it is actually y default this command:

```
free -m 1 >> /tmp/error.log
```

 : this will give you the **Standard output**

similarly

```
free -m 2>> /tmp/error.log
```

 : this will give you the **Standard error**

But

```
free -m &>> /tmp/error.log
```

 : using this command will throw both **standard output** as well as **standard error**

```
wc -l /etc/passwd
```

 : counts line on the file

Piping

what is pipe?

pipe is '|' this thing.

for example if you want to count the files in the directory we can use

```
ls | wc -l
```

in this 'ls' will generate output that output goes as an input to the command after the pipe. It is quite a handy thing. we can also use it other ways like

```
ls | grep host
```

 : this will show all the files that has the name host in it.

we can also use it like this

```
tail -20 /var/log/messages | grep -i vagrant
```

 : this will show the word vagrant in the last 20 lines of the file messages. -i is for case sensitivity as mentioned above as well.

```
free -m | grep -i mem
```

 ; only show ram size

find command

find /etc -name host

locate command

we need to install locate first

```
$ yum install mlocate -y
```

also locate is not a real time search so we need to run the command updatedb before running it as it may show some old deleted files as well.

```
$ updatedb
```

```
$ locate host
```

Users and Groups

Some Important Points related to Users:

- Users and groups are used to control access to files and resources
- Users login to the system by supplying their username and password
- Every file on the system is owned by a user and associated with a group
- Every process has an owner and group affiliation, and can only access the resources its owner or group can access.
- Every user of the system is assigned a unique user ID number (the UID)
- Users name and UID are stored in **/etc/passwd**
- User's password is stored in **/etc/shadow** in encrypted form.
- Users are assigned a **home directory** and a program that is run when they login (**Usually a shell**)
- Users cannot read, write or execute each other's files without permission.

Types of user

TYPE	EXAMPLE	USER ID (ID)	GROUP ID (GID)	HOME DIR	SHELL
ROOT	root	0	0	/root	/bin/bash
REGULAR	imran, vagrant	1000 to 60000	1000 to 60000	/home/username	/bin/bash
SERVICE	ftp, ssh, apache	1 to 999	1 to 999	/var/ftp etc	/sbin/nologin

proper explanation of /etc/passwd

Total 7 columns in this file where

```
root:x:0:0:root:/root:/bin/bash
```

in which

first - username

second - link to a shadow file (encrypted password file)

third - userid(UID)

forth - groupid

fifth - comment

sixth - home directory

seventh - login shell

in /etc/group : stores all the group information

vagrant:x:1000:vagrant

first - group name

second - link to a shadow file (encrypted password file)

third - group id (the numeric group id)

last - group members

adding and modifying users

```
useradd jenkins
```

```
groupadd devops
```

```
usermod -G devops jenkins
```

 : -G is for group, devops is the groupname, jenkins is the username

```
last
```

 : will show who logged in before

```
who
```

 : will show current logged in user

```
lsuf -u vagrant
```

 : will list all the opened file by this(vagrant) user. (maybe you need to install it :

```
yum
```

```
install lsuf -y
```

)

deleting user

```
userdel -r aws
```

 : will remove everything for this user. without -r user's home directory and mailing pool will remain.

```
groupdel devops
```

 : delete group

File permissions

every files has its ownership

-rwxr-xr-x 1 root root

in this

rwx - for user

r-x - for group

r-x - for other

root root : this file is owned by root user and root group

-rw-----. 1 root root 2232 Dec 6 08:32 anaconda-ks.cfg

'-' : file type

'rw-': User

'---': Group

'---': Others

practical

```

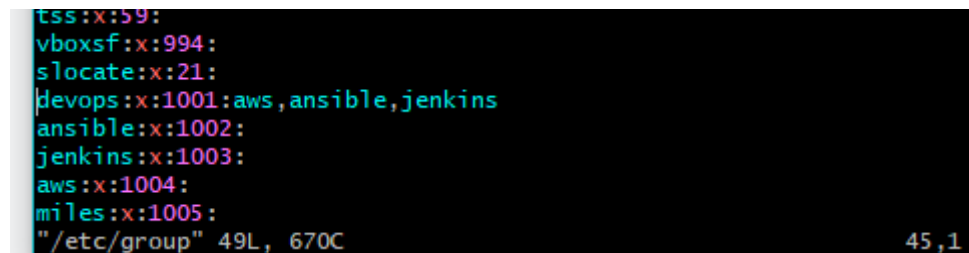
mkdir /opt/devopsdir
ls -l /opt/

drwxr-xr-x 3 root root 17 Dec 6 08:00 dev
drwxr-xr-x 2 root root 6 Dec 6 13:35 devopsdir
drwxr-xr-x 8 root root 136 Nov 6 08:00 Vboxguestaddition

groupadd devops
useradd ansible
useradd jenkins
useradd aws
useradd miles

vim /etc/groups

```



```

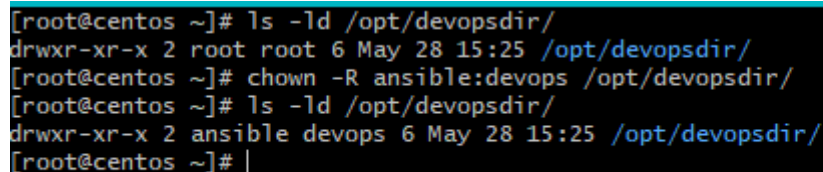
tss:x:59:
vboxsf:x:994:
slocate:x:21:
devops:x:1001:aws,ansible,jenkins
ansible:x:1002:
jenkins:x:1003:
aws:x:1004:
miles:x:1005:
"/etc/group" 49L, 670C 45,1

```

```

id ansible
ls -ld /opt/devopsdir
chown -R ansible:devops /opt/devopsdir

```



```

[root@centos ~]# ls -ld /opt/devopsdir/
drwxr-xr-x 2 root root 6 May 28 15:25 /opt/devopsdir/
[root@centos ~]# chown -R ansible:devops /opt/devopsdir/
[root@centos ~]# ls -ld /opt/devopsdir/
drwxr-xr-x 2 ansible devops 6 May 28 15:25 /opt/devopsdir/
[root@centos ~]# |

```

-ld will show the properties of only the directory

Use -R only if you want to give every subdirectory permissions

to change the execute permission for others use:

```
chmod o-x /opt/devopsdir
```

o : is for others

-x : is for removing execute permission

NOTE: removing execute permission will also remove to cd to that directory

```
chmod g+w /opt/devopsdir
```

this will add write permission to the groups

like this anyone in the devops group can have permission to the directory but not other users like miles

Changing File Ownership

- Only root can change a file's owner
- Only root or the owner can change a file's group
- Ownership is changed with **chown**:
 - **chown [-R] user_name file|directory ...**
- Group-Ownership is changed with **chgrp**:
 - **chgrp [-R] group_name file|directory ...**

Two methods for this

- Symbolic method

Changing Permissions - Symbolic Method

- To change access modes:
 - **chmod [-OPTION] ... mode[,mode] file|directory ...**
 - *mode* includes:
 - **u,g** or **o** for user, group and other
 - **+** - or **=** for grant, deny or set
 - **r, w** or **x** for read, write and execute
 - Options include:
 - **-R** Recursive
 - **-v** Verbose
 - **--reference** Reference another file for its mode
 - Examples:
 - **chmod ugo+r file**: Grant read access to all for *file*
 - **chmod o-wx dir**: Deny write and execute to others for *dir*
-

- Numeric method

Changing Permissions - Numeric Method

- Uses a three-digit mode number
 - first digit specifies owner's permissions
 - second digit specifies group permissions
 - third digit represents others' permissions
- Permissions are calculated by adding:
 - 4 (for read)
 - 2 (for write)
 - 1 (for execute)
- Example:
 - **chmod 640 myfile**

SUDO

Gives power to a normal user to execute commands which is owned by root user.

if a user have sudoers privilege it can become root user any time.

```
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
ansible ALL=(ALL)        NOPASSWD: ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOC
ATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)        ALL

## Same thing without a password
# %wheel    ALL=(ALL)        NOPASSWD: ALL
```

added user ansible to sudoers file with NOPASSWD required for any kind of sudo commands

sometimes if you type something wrong in sudoers

```
[root@centos ~]# visudo
>>> /etc/sudoers: syntax error near line 1 <<<
What now? e
[root@centos ~]#
```

press e to edit it and correct it on the stated line

Another solution instead of editing

```
[root@centos ~]# cd /etc/sudoers.d/
[root@centos sudoers.d]# ls
vagrant
[root@centos sudoers.d]# cat vagrant
vagrant    ALL=(ALL)        NOPASSWD: ALL
[root@centos sudoers.d]# cp vagrant devops
[root@centos sudoers.d]# vim devops
[root@centos sudoers.d]# cat devops
%devops    ALL=(ALL)        NOPASSWD: ALL
[root@centos sudoers.d]# cat *
%devops    ALL=(ALL)        NOPASSWD: ALL
vagrant    ALL=(ALL)        NOPASSWD: ALL
[root@centos sudoers.d]#
```

this is a safer option instead of editing the sudoers file

Software/Package Management

Some handy commands::

Commands	Description	
wget link	to download file from link	
curl link	access file from link	
curl link -o outputfile	access file and store o/p to a file	
REDHAT RPM commands		
rpm -ivh {rpm-file}	Install the package	rpm -ivh mozilla-mail-1.7.5-17.i586.rpm rpm -ivh --test mozilla-mail-1.7.5-17.i586.rpm
rpm -Uvh {rpm-file}	Upgrade package	rpm -Uvh mozilla-mail-1.7.6-12.i586.rpm rpm -Uvh --test mozilla-mail-1.7.6-12.i586.rpm

rpm -ev {package}	Erase/remove/ an installed package	rpm -ev mozilla-mail
rpm -ev --nodeps {package}	Erase/remove/ an installed package without checking for	rpm -ev --nodeps mozilla-mail

	dependencies	
rpm -qa	Display list all installed packages	rpm -qa rpm -qa less
rpm -qi {package}	Display installed information along with package version and short description	rpm -qi mozilla-mail
rpm -qf {/path/to/file}	Find out what package a file belongs to i.e. find what package owns the file	rpm -qf /etc/passwd rpm -qf /bin/bash
rpm -qc {package-name}	Display list of configuration file(s) for a package	rpm -qc httpd
rpm -qcf {/path/to/file}	Display list of configuration files for a command	rpm -qcf /usr/X11R6/bin/xeyes
rpm -qa --last	Display list of all recently installed RPMs	rpm -qa --last rpm -qa --last less

So from this we installed tree

```
[root@centos ~]# tree /var/log/
/var/log/
├── anaconda
│   ├── anaconda.log
│   ├── ifcfg.log
│   ├── journal.log
│   ├── ks-script-CIP9d6.log
│   ├── ks-script-MyPjfZ.log
│   ├── ks-script-z__Dbe.log
│   ├── packaging.log
│   ├── program.log
│   ├── storage.log
│   └── syslog
├── audit
│   └── audit.log
├── boot.log
├── boot.log-20220524
├── boot.log-20220525
├── boot.log-20220528
├── btmp
├── btmp-20220524
├── chrony
├── cron
├── cron-20220524
├── dmesg
├── dmesg.old
├── grubby
├── grubby_prune_debug
├── lastlog
├── maillog
├── maillog-20220524
├── messages
├── messages-20220524
├── rhsm
├── samba
│   └── old
├── secure
├── secure-20220524
├── spooler
├── spooler-20220524
├── tallylog
├── tuned
│   └── tuned.log
├── vboxadd-install.log
├── vboxadd-setup.log
├── vboxadd-setup.log.1
├── vboxadd-setup.log.2
└── vboxadd-setup.log.3
```

it is actually a very handy command since it will display everything in a tree like structure just like in the picture above.

now we will install httpd which is a web server software.

```
root@centos:~
[root@centos ~]# curl
curl: try 'curl --help' or 'curl --manual' for more information
[root@centos ~]#
[root@centos ~]#
[root@centos ~]# curl https://rpmfind.net/linux/centos/7.9.2009/os/x86_64/Packages/httpd-2.4.6-95.el7.centos.x86_64.rpm -o httpd-2.4.6-95.el7.c
entos.x86_64.rpm
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 2779k  100 2779k    0     0  366k      0  0:00:07  0:00:07 --:--:-- 538k
[root@centos ~]# ls
anaconda-ks.cfg  httpd-2.4.6-95.el7.centos.x86_64.rpm  original-ks.cfg  samplefile.txt  tree-1.6.0-10.el7.x86_64.rpm
```

But while installing it with rpm it shows an error cause it needs the following dependencies.

Ofcourse you can manually search the dependencies and install it all
but what if you require 100s of dependencies?

```
[root@centos ~]# rpm -ivh httpd-2.4.6-95.el7.centos.x86_64.rpm
error: Failed dependencies:
    /etc/mime.types is needed by httpd-2.4.6-95.el7.centos.x86_64
    httpd-tools = 2.4.6-95.el7.centos is needed by httpd-2.4.6-95.el7.centos.x86_64
    libapr-1.so.0()(64bit) is needed by httpd-2.4.6-95.el7.centos.x86_64
    libaprutil-1.so.0()(64bit) is needed by httpd-2.4.6-95.el7.centos.x86_64
[root@centos ~]#
```

So for this we use yum, it will automate the process of package installation.

Link to YUM cheatsheet

https://access.redhat.com/sites/default/files/attachments/rh_yum_cheatsheet_1214_jcs_print-1.pdf

```
cd /etc/yum.repos.d
```

```
[root@centos ~]# cd /etc/yum.repos.d/
[root@centos yum.repos.d]# ls
CentOS-Base.repo  CentOS-Debuginfo.repo  CentOS-Media.repo  CentOS-Vault.repo  epel.repo
CentOS-CR.repo   CentOS-fasttrack.repo  CentOS-Sources.repo  CentOS-x86_64-kernel.repo  epel-testing.repo
[root@centos yum.repos.d]#
```

in this directory there are files which points to repository on the internet. There are various URLs and repository information. which will help to install softwares with all the dependencies it needs.

We can search by using the command

```
yum search httpd
```

so to install

```
yum install httpd -y
```

Package	Arch	Version	Repository	Size
Installing:				
httpd	x86_64	2.4.6-97.el7.centos.5	updates	2.7 M
Installing for dependencies:				
apr	x86_64	1.4.8-7.el7	base	104 k
apr-util	x86_64	1.5.2-6.el7	base	92 k
httpd-tools	x86_64	2.4.6-97.el7.centos.5	updates	94 k
mailcap	noarch	2.1.41-2.el7	base	31 k
Transaction Summary				
Install 1 Package (+4 Dependent packages)				
Total download size: 3.0 M				
Installed size: 10 M				
Downloading packages:				
(1/5): apr-1.4.8-7.el7.x86_64.rpm			104 kB	00:00:00
(2/5): mailcap-2.1.41-2.el7.noarch.rpm			31 kB	00:00:00
(3/5): apr-util-1.5.2-6.el7.x86_64.rpm			92 kB	00:00:00
(4/5): httpd-2.4.6-97.el7.centos.5.x86_64.rpm			2.7 MB	00:00:00
(5/5): httpd-tools-2.4.6-97.el7.centos.5.x86_64.rpm			94 kB	00:00:00
Total			3.1 MB/s	3.0 MB 00:00:00
Running transaction check				
Running transaction test				
Transaction test succeeded				
Running transaction				
Warning: RPMDB altered outside of yum.				
Installing :	apr-1.4.8-7.el7.x86_64			1/5
Installing :	apr-util-1.5.2-6.el7.x86_64			2/5
Installing :	httpd-tools-2.4.6-97.el7.centos.5.x86_64			3/5
Installing :	mailcap-2.1.41-2.el7.noarch			4/5
Installing :	httpd-2.4.6-97.el7.centos.5.x86_64			5/5
Verifying :	apr-1.4.8-7.el7.x86_64			1/5
Verifying :	mailcap-2.1.41-2.el7.noarch			2/5
Verifying :	httpd-tools-2.4.6-97.el7.centos.5.x86_64			3/5
Verifying :	apr-util-1.5.2-6.el7.x86_64			4/5
Verifying :	httpd-2.4.6-97.el7.centos.5.x86_64			5/5
Installed:				
httpd.x86_64 0:2.4.6-97.el7.centos.5				
Dependency Installed:				
apr.x86_64	0:1.4.8-7.el7	apr-util.x86_64	0:1.5.2-6.el7	httpd-tools.x86_64 0:2.4.6-97.el7.centos.5 mailcap.noarch 0:2.1.41-2.el7
Complete!				
[root@centos ~]#				

to remove

`yum remove httpd`

so if we need to install jenkins what do we do>?

```
[root@centos ~]# yum install jenkins
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: centos.excellmedia.net
 * epel: download.nus.edu.sg
 * extras: centos.excellmedia.net
 * updates: centos.excellmedia.net
No package jenkins available.
Error: Nothing to do
[root@centos ~]#
```

It shows an error

WHY?

Because it doesnot have anything named jenkins in its repository.

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
sudo yum upgrade
# Add required dependencies for the jenkins package
sudo yum install java-11-openjdk
sudo yum install jenkins
sudo systemctl daemon-reload
```

we follow the instruction from the official jenkins site.

Services

we installed package httpd, its a service

`systemctl status httpd`

its stopped (inactive)

`systemctl start httpd`

it starts

`systemctl restart httpd`

`systemctl reload httpd` (will reload the configuration without restarting the service)

you can use `systemctl` command for a service

this service is started but what if you reboot the machine

this service will not start again

for this you need to enable the service for eg

systemctl enable httpd

you need to start for the first time

to check if it is enabled or not just do

systemctl is-enable httpd

for checking start

systemctl is-active httpd

we are able to access the machine with the service sshd

to check

systemctl status sshd

the way systemctl works is through its configuration file so for httpd there should be configuration file. it comes when you install the service

cat /etc/systemd/system/multi-user.target.wants/httpd.service

```
[root@centos ~]# cat /etc/systemd/system/multi-user.target.wants/httpd.service
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target
Documentation=man:httpd(8)
Documentation=man:apachectl(8)

[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/httpd
ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful
ExecStop=/bin/kill -WINCH ${MAINPID}
# We want systemd to give httpd some time to finish gracefully, but still want
# it to kill httpd after TimeoutStopSec if something went wrong during the
# graceful stop. Normally, Systemd sends SIGTERM signal right after the
# ExecStop, which would kill httpd. We are sending useless SIGCONT here to give
# httpd time to finish.
KillSignal=SIGCONT
PrivateTmp=true

[Install]
WantedBy=multi-user.target
[root@centos ~]#
```

Processes

the command **top** will show all the dynamic processes


```
top - 14:46:59 up 16 min, 1 user, load average: 0.00, 0.01, 0.03
Tasks: 117 total, 1 running, 116 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 498484 total, 320820 free, 82516 used, 95148 buff/cache
KiB Swap: 1048572 total, 1048572 free, 0 used. 398260 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	191172	4124	2640	S	0.0	0.8	0:01.70	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
4	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:+
6	root	20	0	0	0	0	S	0.0	0.0	0:00.09	ksoftirqd/0
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0.0	0.0	0:00.34	rcu_sched
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	lru-add-dr+
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	watchdog/0
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/1
13	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	migration/1
14	root	20	0	0	0	0	S	0.0	0.0	0:00.33	ksoftirqd/1
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/1:+
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
19	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
21	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	writeback

first column is PID (Process ID)
second column User
9th column is the status

Q to quit from the **top** command

ps aux is also a similar command to **top** command but it quits after showing all the processes.

```
[root@centos ~]# ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.1	0.8	191172	4124	?	Ss	14:30	0:01	/usr/lib/systemd/systemd --switched-root --system --deserialize 22
root	2	0.0	0.0	0	0	?	S	14:30	0:00	[kthreadd]
root	4	0.0	0.0	0	0	?	S<	14:30	0:00	[kworker/0:0H]
root	6	0.0	0.0	0	0	?	S	14:30	0:00	[ksoftirqd/0]
root	7	0.0	0.0	0	0	?	S	14:30	0:00	[migration/0]
root	8	0.0	0.0	0	0	?	S	14:30	0:00	[rcu_bh]
root	9	0.0	0.0	0	0	?	S	14:30	0:00	[rcu_sched]
root	10	0.0	0.0	0	0	?	S<	14:30	0:00	[lru-add-drain]
root	11	0.0	0.0	0	0	?	S	14:30	0:00	[watchdog/0]
root	12	0.0	0.0	0	0	?	S	14:30	0:00	[watchdog/1]
root	13	0.0	0.0	0	0	?	S	14:30	0:00	[migration/1]
root	14	0.0	0.0	0	0	?	S	14:30	0:00	[ksoftirqd/1]
root	16	0.0	0.0	0	0	?	S<	14:30	0:00	[kworker/1:0H]
root	18	0.0	0.0	0	0	?	S	14:30	0:00	[kdevtmpfs]
root	19	0.0	0.0	0	0	?	S<	14:30	0:00	[netns]
root	20	0.0	0.0	0	0	?	S	14:30	0:00	[khungtaskd]
root	21	0.0	0.0	0	0	?	S<	14:30	0:00	[writeback]
root	22	0.0	0.0	0	0	?	S<	14:30	0:00	[kintegrityd]
root	23	0.0	0.0	0	0	?	S<	14:30	0:00	[bioaset]
root	24	0.0	0.0	0	0	?	S<	14:30	0:00	[bioaset]
root	25	0.0	0.0	0	0	?	S<	14:30	0:00	[bioaset]
root	26	0.0	0.0	0	0	?	S<	14:30	0:00	[kblockd]
root	27	0.0	0.0	0	0	?	S<	14:30	0:00	[md]
root	28	0.0	0.0	0	0	?	S<	14:30	0:00	[edac-poller]
root	29	0.0	0.0	0	0	?	S<	14:30	0:00	[watchdogd]
root	35	0.0	0.0	0	0	?	S	14:30	0:00	[kswapd0]
root	36	0.0	0.0	0	0	?	SN	14:30	0:00	[ksmd]
root	37	0.0	0.0	0	0	?	S<	14:30	0:00	[crypto]
root	45	0.0	0.0	0	0	?	S<	14:30	0:00	[kthrotld]
root	46	0.0	0.0	0	0	?	S	14:30	0:00	[kworker/u4:1]
root	47	0.0	0.0	0	0	?	S<	14:30	0:00	[kmpath_rdacd]
root	48	0.0	0.0	0	0	?	S<	14:30	0:00	[kaluad]
root	49	0.0	0.0	0	0	?	S	14:30	0:00	[kworker/0:1]
root	50	0.0	0.0	0	0	?	S<	14:30	0:00	[kpsmouse]
root	51	0.0	0.0	0	0	?	S	14:30	0:00	[kworker/0:2]
root	52	0.0	0.0	0	0	?	S<	14:30	0:00	[ipv6_addrconf]
root	65	0.0	0.0	0	0	?	S<	14:30	0:00	[deferwq]
root	101	0.0	0.0	0	0	?	S	14:30	0:00	[kauditd]
root	275	0.0	0.0	0	0	?	S<	14:30	0:00	[ata_sff]
root	289	0.0	0.0	0	0	?	S	14:30	0:00	[scsi_eh_0]
root	290	0.0	0.0	0	0	?	S<	14:30	0:00	[scsi_tmf_0]
root	291	0.0	0.0	0	0	?	S	14:30	0:00	[scsi_eh_1]
root	292	0.0	0.0	0	0	?	S<	14:30	0:00	[scsi_tmf_1]
root	294	0.0	0.0	0	0	?	S	14:30	0:00	[kworker/u4:3]
root	359	0.0	0.0	0	0	?	S<	14:30	0:00	[kdmflush]
root	360	0.0	0.0	0	0	?	S<	14:30	0:00	[bioaset]
root	369	0.0	0.0	0	0	?	S<	14:30	0:00	[kdmflush]

The first process in linux is systemd in older version its also called init. This process will start so many

other process and also handle child processes

The process which can be seen on the square bracket is the Kernel threads

Other normal processes looks like this::

```
root      654  0.0  0.0      0 0 ?        Ss   14:30  0:00 [xprtlib]
polkitd   654  0.0  1.8 716704 9160 ?        Ss   14:30  0:00 /usr/lib/polkit-1/polkitd --no-debug
dbus      656  0.0  0.4  58216 2476 ?        Ss   14:30  0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --system
rpc       657  0.0  0.2  69256 1008 ?        Ss   14:30  0:00 /sbin/rpcbind -w
chrony    660  0.0  0.3 117808 1736 ?        S    14:30  0:00 /usr/sbin/chronyd
root      666  0.0  0.2 195208 1256 ?        Ss   14:30  0:00 /usr/sbin/gssproxy -D
root      673  0.0  1.7 550252 8772 ?        Ss   14:30  0:00 /usr/sbin/NetworkManager --no-daemon
root      675  0.0  0.3  26384 1744 ?        Ss   14:30  0:00 /usr/lib/systemd/systemd-logind
root      677  0.0  0.2  21540 1240 ?        Ss   14:30  0:00 /usr/sbin/irqbalance --foreground
root      711  0.0  1.1 102988 5536 ?        S    14:30  0:00 /sbin/dhclient -d -q -sf /usr/libexec/nm-dhcp-helper -pf /var/run/dhclient-enp
root      999  0.0  0.7 212304 3516 ?        Ss   14:30  0:00 /usr/sbin/rsyslogd -n
root     1001  0.0  3.5 574284 17448 ?        Ss   14:30  0:00 /usr/bin/python2 -Es /usr/sbin/tuned -l -P
root     1005  0.0  0.8 112984 4336 ?        Ss   14:30  0:00 /usr/sbin/sshd -D
root     1012  0.0  0.3 126388 1568 ?        Ss   14:30  0:00 /usr/sbin/crond -n
root     1028  0.0  0.1 110208   864 tty1     Ss+  14:30  0:00 /sbin/agetty --noclear tty1 linux
root     1216  0.0  0.4  89788 2116 ?        Ss   14:31  0:00 /usr/libexec/postfix/master -w
postfix   1232  0.0  0.8  89892 4092 ?        S    14:31  0:00 pickup -l -t unix -u
postfix   1233  0.0  0.8  89960 4112 ?        S    14:31  0:00 qmgr -l -t unix -u
root     1869  0.0  1.1 154776 5500 ?        Ss   14:32  0:00 sshd: vagrant [priv]
vagrant   1871  0.0  0.5 155084 2608 ?        S    14:32  0:00 sshd: vagrant@pts/0
vagrant   1872  0.0  0.4 115548 2048 pts/0    Ss   14:32  0:00 -bash
root     1933  0.0  0.9 241412 4660 pts/0    S    14:33  0:00 sudo -i
root     1934  0.0  0.4 115548 2056 pts/0    S    14:33  0:00 -bash
root     1958  0.0  1.0 230440 5204 ?        Ss   14:33  0:00 /usr/sbin/httpd -DFOREGROUND
apache    1959  0.0  0.6 230440 2992 ?        S    14:33  0:00 /usr/sbin/httpd -DFOREGROUND
apache    1960  0.0  0.6 230440 2992 ?        S    14:33  0:00 /usr/sbin/httpd -DFOREGROUND
apache    1961  0.0  0.6 230440 2992 ?        S    14:33  0:00 /usr/sbin/httpd -DFOREGROUND
apache    1962  0.0  0.6 230440 2992 ?        S    14:33  0:00 /usr/sbin/httpd -DFOREGROUND
apache    1963  0.0  0.5 230440 2984 ?        S    14:33  0:00 /usr/sbin/httpd -DFOREGROUND
root     1984  0.0  0.0      0 0 ?        S    14:35  0:00 [kworker/1:0]
root     1994  0.0  0.3 155452 1876 pts/0    R+   14:51  0:00 ps aux
[root@centos ~]#
```

we can see multiple httpd services

we can also use the command **ps -ef** in which there are shown UID PID PPID C STIME TTY TIME CMD

something like this

```
UID      PID    PPID  C  STIME TTY          TIME CMD
root      1      0  0 14:30 ?          00:00:01 /usr/lib/systemd/systemd --switched-root --system --deserialize 22
root      2      0  0 14:30 ?          00:00:00 [kthreadd]
root      4      2  0 14:30 ?          00:00:00 [kworker/0:0H]
root      6      2  0 14:30 ?          00:00:00 [ksoftirqd/0]
root      7      2  0 14:30 ?          00:00:00 [migration/0]
root      8      2  0 14:30 ?          00:00:00 [rcu_bh]
root      9      2  0 14:30 ?          00:00:00 [rcu_sched]
root     10      2  0 14:30 ?          00:00:00 [lru-add-drain]
root     11      2  0 14:30 ?          00:00:00 [watchdog/0]
root     12      2  0 14:30 ?          00:00:00 [watchdog/1]
root     13      2  0 14:30 ?          00:00:00 [migration/1]
root     14      2  0 14:30 ?          00:00:00 [ksoftirqd/1]
root     16      2  0 14:30 ?          00:00:00 [kworker/1:0H]
root     18      2  0 14:30 ?          00:00:00 [kdevtmpfs]
root     19      2  0 14:30 ?          00:00:00 [netns]
root     20      2  0 14:30 ?          00:00:00 [khungtaskd]
root     21      2  0 14:30 ?          00:00:00 [writeback]
root     22      2  0 14:30 ?          00:00:00 [kintegrityd]
root     23      2  0 14:30 ?          00:00:00 [bioaset]
root     24      2  0 14:30 ?          00:00:00 [bioaset]
root     25      2  0 14:30 ?          00:00:00 [bioaset]
root     26      2  0 14:30 ?          00:00:00 [kblockd]
root     27      2  0 14:30 ?          00:00:00 [md]
root     28      2  0 14:30 ?          00:00:00 [edac-poller]
root     29      2  0 14:30 ?          00:00:00 [watchdogd]
root     35      2  0 14:30 ?          00:00:00 [kswapd0]
root     36      2  0 14:30 ?          00:00:00 [ksmd]
root     37      2  0 14:30 ?          00:00:00 [crypto]
root     45      2  0 14:30 ?          00:00:00 [kthrotld]
root     46      2  0 14:30 ?          00:00:00 [kworker/u4:1]
root     47      2  0 14:30 ?          00:00:00 [kmpath_rdacd]
root     48      2  0 14:30 ?          00:00:00 [kaluad]
root     49      2  0 14:30 ?          00:00:00 [kworker/0:1]
root     50      2  0 14:30 ?          00:00:00 [kpsmouse]
root     51      2  0 14:30 ?          00:00:00 [kworker/0:2]
root     52      2  0 14:30 ?          00:00:00 [ipv6_addrconf]
root     65      2  0 14:30 ?          00:00:00 [deferwq]
root    101      2  0 14:30 ?          00:00:00 [kauditd]
root    275      2  0 14:30 ?          00:00:00 [ata_sff]
root    289      2  0 14:30 ?          00:00:00 [scsi_eh_0]
root    290      2  0 14:30 ?          00:00:00 [scsi_tm_f_0]
root    291      2  0 14:30 ?          00:00:00 [scsi_eh_1]
```

PPID stands for Parent process ID

```

root      1958      1  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1959    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1960    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1961    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1962    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1963    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
root      2005      1  0 15:01 ?        00:00:00 /usr/sbin/anacron -s
root      2009     673  0 15:10 ?        00:00:00 /sbin/dhclient -d -q -sf /usr/libexec/nm-dhcp-helper -pf /var/run/dhclient-enp0s3.pid -lf /var/
root      2088      2  0 15:17 ?        00:00:00 [kworker/1:0]
root      2089      2  0 15:22 ?        00:00:00 [kworker/1:1]
root      2090    1005  0 15:28 ?        00:00:00 sshd: vagrant [priv]
vagrant   2092    2090  0 15:28 ?        00:00:00 sshd: vagrant@pts/1
vagrant   2093    2092  0 15:28 pts/1    00:00:00 -bash
root      2115      2  0 15:28 ?        00:00:00 [kworker/1:2]
vagrant   2118    2093  0 15:29 pts/1    00:00:00 ps -ef

```

in this image we can see how parents process ID and PID starts the other one

so we can say that the root process has started all the other apache processes by looking at their PPID. this process is also called as **forking**

to kill or stop a process there is a command called **kill**

to show all the httpd we can use grep

```
ps -ef | grep httpd | grep -v 'grep'
```

the first pipe to show the httpd and the second pipe to not show the grep word.

```

[vagrant@centos ~]$ ps -ef | grep httpd
root      1958      1  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1959    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1960    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1961    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1962    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1963    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
vagrant   2165    2093  0 15:34 pts/1    00:00:00 grep --color=auto httpd
[vagrant@centos ~]$ ps -ef | grep httpd | grep -v 'grep'
root      1958      1  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1959    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1960    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1961    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1962    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    1963    1958  0 14:33 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
[vagrant@centos ~]$

```

we can kill the process 1958 by **kill 1958**

```

[root@centos ~]# kill 1958
[root@centos ~]# ps -ef | grep httpd | grep -v 'grep'
[root@centos ~]#

```

by this all other child process are closed at first then it will close itself

we also have a command to forcefully close the process which is by **kill -9 {process id}**

```
[root@centos ~]# ps -ef | grep httpd | grep -v 'grep'
root      2200      1  0 15:37 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2201    2200  0 15:37 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2202    2200  0 15:37 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2203    2200  0 15:37 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2204    2200  0 15:37 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2205    2200  0 15:37 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
[root@centos ~]# kill -9 2200
[root@centos ~]# ps -ef | grep httpd | grep -v 'grep'
apache    2201      1  0 15:37 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2202      1  0 15:37 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2203      1  0 15:37 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2204      1  0 15:37 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2205      1  0 15:37 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
[root@centos ~]# |
```

when closing forcefully the other child processes becomes orphan which means it doesn't close. This is because the parent process has no idea that it is being closed. And later we can see that the orphan child processes are adopted by the systemd process which is **1**. Most of the time orphan processes are killed automatically in advance systems.

In cases like this sometimes it is hard to kill all the orphan processes one by one so we can use multiple commands with pipe.

something like this:

```
ps -ef | grep httpd | grep -v 'grep' | awk '{print $2}'
```

by this command it will show only the 2nd column of the table.

and now we can use this second column as an input for the kill command by using the command **xargs kill -9**. Something like this.

```
ps -ef | grep httpd | grep -v 'grep' | awk '{print $2}' | xargs kill -9 '
```

```
[root@centos ~]# ps -ef | grep httpd | grep -v 'grep'
root      2233      1  1 15:46 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2234    2233  0 15:46 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2235    2233  0 15:46 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2236    2233  0 15:46 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2237    2233  0 15:46 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2238    2233  0 15:46 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
[root@centos ~]# kill -9 2233
[root@centos ~]# ps -ef | grep httpd | grep -v 'grep' | awk '{print $2}'
2234
2235
2236
2237
2238
[root@centos ~]# ps -ef | grep httpd | grep -v 'grep' | awk '{print $2}' | xargs kill -9
[root@centos ~]# ps -ef | grep httpd | grep -v 'grep' | awk '{print $2}'
[root@centos ~]# |
```

if you're using -9, always filter it.

Zombie processes are the processes which have finished their task and still take an entry in the process table. It will appear with the status of Z in the table.

Archiving

There are many times when we need to archive files and directory mostly in backup processes

we are in /var/log directory

and lets say we want to archive the jenkins direcotry

TAR

so for that we do

```
tar -czvf jenkins_06122022.tar.gz jenkins
```

c is to create

z is to compress

v is for verbose

f is for file

tar represent tar ball

gz represent gun zip

jenkins is the file that we are archiving

jenkins_06122022.tar.gz is the file output of the archive

```
[root@centos ~]# cd /var/log
[root@centos log]# ls jenkins/
[root@centos log]# clear
[root@centos log]# cd /var/log/
[root@centos log]# pwd
/var/log
[root@centos log]# tar -czvf jenkins_06122022.tar.gz jenkins
jenkins/
[root@centos log]# ls -ltr
-rw----- 1 root root 62162 Jun 2 16:01 messages
-rw-r--r-- 1 root root 117 Jun 2 16:03 jenkins_06122022.tar.gz
[root@centos log]# file jenkins_06122022.tar.gz
jenkins_06122022.tar.gz: gzip compressed data, from Unix, last modified: Thu Jun 2 16:03:52 2022
[root@centos log]# |
```

to unarchive

```
tar -xzvf jenkins_06122022.tar.gz
```

where

x is for extract

if want to extract somewhere else you can give -C. For example

```
tar -xzvf jenkins_06122022.tar.gz -C /opt/
```

tar is very legacy command and it is still used

There are many other options as well:: which can be found by
tar --help

ZIP and UNZIP

There is also much more simple command which is **zip** and **unzip**.
we install it by

```
yum install zip unzip -y
```

we zip the file by

```
zip -r jenkins_06122022.zip jenkins
```

zip - command

-r : for compressing or archiving a directory

jenkins_06122022.zip : the output file you will get

jenkins : the input directory or file

```
[root@centos log]# zip -r jenkins_06122022.zip jenkins
  adding: jenkins/ (stored 0%)
[root@centos log]# clear
[root@centos log]# ls -ltr jenkins*
-rw-r--r-- 1 root  root   117 Jun  2 16:03 jenkins_06122022.tar.gz
-rw-r--r-- 1 root  root   166 Jun  3 01:17 jenkins_06122022.zip

jenkins:
total 0
```

to unzip simply

```
unzip jenkins_06122022.zip
```

Ubuntu Commands

In ubuntu we use **adduser** to add user. if we use **useradd** command it will not make any kind of home directory

in ubuntu the default editor is nano editor

for changing it into vim editor we do

```
export EDITOR=vim
```

with this the default editor is **vim**

but it is temporary because it is setting variables temporary in your current shell. so if you log out and log in it will change again

Package manager for ubuntu

first we will download tree package. In centos or Lets say red hat it was rpm for debian based OS it is dpkg

so to install the tree package which we got from the link by running the command

```
wget http://archive.ubuntu.com/ubuntu/pool/universe/t/tree/tree_1.8.0-1_amd64.deb
```

we install it by doing

```
dpkg -i tree_1.8.0-1_amd64.deb
```

```
root@ubuntu-bionic:~# wget http://archive.ubuntu.com/ubuntu/pool/universe/t/tree/tree_1.8.0-1_amd64.deb
--2022-06-06 15:55:30-- http://archive.ubuntu.com/ubuntu/pool/universe/t/tree/tree_1.8.0-1_amd64.deb
Resolving archive.ubuntu.com (archive.ubuntu.com)... 91.189.91.38, 185.125.190.36, 185.125.190.39, ...
Connecting to archive.ubuntu.com (archive.ubuntu.com)|91.189.91.38|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 43044 (42K) [application/x-debian-package]
Saving to: 'tree_1.8.0-1_amd64.deb'

tree_1.8.0-1_amd64.deb      100%[=====>] 42.04K  --.-KB/s   in 0.009s

2022-06-06 15:55:30 (4.36 MB/s) - 'tree_1.8.0-1_amd64.deb' saved [43044/43044]

root@ubuntu-bionic:~# ls
tree_1.8.0-1_amd64.deb
root@ubuntu-bionic:~# dpkg -i tree_1.8.0-1_amd64.deb
Selecting previously unselected package tree.
(Reading database ... 60148 files and directories currently installed.)
Preparing to unpack tree_1.8.0-1_amd64.deb ...
Unpacking tree (1.8.0-1) ...
Setting up tree (1.8.0-1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

-i to install

-r to remove

-l to list every thing

as we have yum in red hat machine here we have **apt**

for installing we need to update apt first by running

```
apt update
```

```
apt search tree
```

```
apt install tree
```

there is also

```
apt-get install tree
```

-get is an older version of apt people dont often use it these days

for installing httpd here it is apache2

apt install apache2

in debian after installing any kind of service it will enable and run it as soon as it installs

```
root@ubuntu-bionic:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Mon 2022-06-06 16:02:44 UTC; 38s ago
     Main PID: 3021 (apache2)
        Tasks: 55 (limit: 1151)
       CGroup: /system.slice/apache2.service
               └─3021 /usr/sbin/apache2 -k start
                 └─3022 /usr/sbin/apache2 -k start
                   └─3023 /usr/sbin/apache2 -k start

Jun 06 16:02:43 ubuntu-bionic systemd[1]: Starting The Apache HTTP Server...
Jun 06 16:02:44 ubuntu-bionic apachectl[2995]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 1
Jun 06 16:02:44 ubuntu-bionic systemd[1]: Started The Apache HTTP Server.
root@ubuntu-bionic:~# systemctl is-enabled apache2
enabled
root@ubuntu-bionic:~# |
```

apt upgrade will upgrade all the packages where as apt update will only update the package list.
there is a difference

for removing

```
apt remove apache2
```

there is also another option for removing the package

because simply removing the package but not the configuration so for that we do

```
apt purge apache2
```

this will do a clean uninstall which will remove the package, its data and all its configuration

when installing web service it will also update the firewall rules

ufw is the ubuntu firewall

these are some differences