

```
calender:
-----
sudo apt update
sudo apt install ncal
cal or ncal - calender with highlighted date
ncal <month> <year> - ncal 01 2025 -> provides jan 2025 calender

man command :
-----

manul page (man)

$ man ls

this provides the manual for how to use `ls`

Basic Command syntax:
-----

<command> <args>
echo      hello

for shell we provide command followed by arguments

echo hello [echo is command and hello is argument]

There are two types of Arguments
  * Positional arguments
  * Named arguments

Positional Arguments :
-----
<command> <arg1> <arg2>
cp 1.txt 2.txt

here position becomes very critical

Named arguments:
-----
<command> --<arg> <argvalue>

ping -c 4 google.com

* to see hidden files
ls -a
```

Linux Directory Hierarchy

- / => root directory

- /bin => Binaries and other executables
- /etc => system configuration files
- /home => home directory
- /opt => optional or third party softwares
- /tmp => Temporary spaces
- /usr => User related programs
- /var => variable data, log
- experiment with mkdir and rmdir.
- ls . [dot represents current directory]
- ls .. [one step back from current]
- ls ../lib [shows inside files of lib]

```
ubuntu@ip-172-31-83-19:~$ less 1.txt
ubuntu@ip-172-31-83-19:~$
ubuntu@ip-172-31-83-19:~$ touch 2.txt
ubuntu@ip-172-31-83-19:~$ diff -u 1.txt 2.txt
--- 1.txt          2024-08-01 02:03:21.100972929 +0000
+++ 2.txt          2024-08-01 02:06:05.571480447 +0000
@@ -1,33 +0,0 @@
-sadfasfasdfs
-asf
-as
-fas
-fs
-vas
-v
-a
-dvadsvdagwd
-g
-ad
-v
-sd
-v
-ad
-va
-dv
-a
-dv
-ad
-vbad
-gqdgadgadgagdas
-sf
-as
-
-as
-dg
-sag
-as
-f
-s
-g
-as
ubuntu@ip-172-31-83-19:~$ cp 1.txt 2.txt
ubuntu@ip-172-31-83-19:~$ diff -u 1.txt 2.txt
ubuntu@ip-172-31-83-19:~$ |
```

- file command is used to know more about that file

```
ubuntu@ip-172-31-83-19:~$ file 1.txt
1.txt: ASCII text
ubuntu@ip-172-31-83-19:~$ |
```

- find command

```
ubuntu@ip-172-31-83-19:~$ find . -type f -name "*.txt"
./2.txt
./1.txt
ubuntu@ip-172-31-83-19:~$ find . -name "*.txt"
./2.txt
./1.txt
ubuntu@ip-172-31-83-19:~$ |
```

- head and tail commands
 - head command show fist n lines where n is an integer

- default tail command shows last 10 lines of file

```
ubuntu@ip-172-31-83-19:~$ head -5 1.txt
sasdfasdfs
asf
as
fas
fs
ubuntu@ip-172-31-83-19:~$ tail -5 1.txt
as
f
s
g
as
ubuntu@ip-172-31-83-19:~$ tail 1.txt
as

as
dg
sag
as
f
s
g
as
ubuntu@ip-172-31-83-19:~$
```

Environmental and shell variables

- Shell variables : Shell can temporarily store variables called as shell variables
- once you logout and login back it cant remember previous values

```
<var-name>=<value>
Topic= linux
to access variable use $
echo $Topic
```

```
ubuntu@ip-172-31-83-19:~$ echo $Topic
linux
ubuntu@ip-172-31-83-19:~$ Topic=ubuntu
ubuntu@ip-172-31-83-19:~$ echo $topic
ubuntu
ubuntu@ip-172-31-83-19:~$ exit
logout
Connection to 3.92.2.115 closed.
PS C:\Users\aravi> ssh ubuntu@3.92.2.115
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1009-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Aug  1 02:24:59 UTC 2024

System load:  0.0                       Processes:            115
Usage of /:   26.0% of 6.71GB           Users logged in:     1
Memory usage: 23%                       IPv4 address for enX0: 172
.31.83.19
Swap usage:   0%

Expanded Security Maintenance for Applications is not enable
d.

34 updates can be applied immediately.
18 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security update
s.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Aug  1 02:00:32 2024 from 103.197.112.85
ubuntu@ip-172-31-83-19:~$ echo $Topic
ubuntu@ip-172-31-83-19:~$ |
```

- to make it remember we write values in some files like /etc/environment, later even if you login back and use them

```
ubuntu@ip-172-31-83-19:~$ Topic=linux
ubuntu@ip-172-31-83-19:~$ echo $Topic
linux
ubuntu@ip-172-31-83-19:~$ sudo vi /etc/environment
ubuntu@ip-172-31-83-19:~$ source /etc/environment
ubuntu@ip-172-31-83-19:~$ exit
logout
Connection to 3.92.2.115 closed.
PS C:\Users\aravi> ssh ubuntu@3.92.2.115
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1009-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Aug  1 02:35:55 UTC 2024

System load:  0.0                       Processes:            113
Usage of /:   26.0% of 6.71GB           Users logged in:     1
Memory usage: 23%                       IPv4 address for enX0: 172
.31.83.19
Swap usage:   0%

Expanded Security Maintenance for Applications is not enable
d.

34 updates can be applied immediately.
18 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security update
s.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Aug  1 02:33:42 2024 from 103.197.112.85
ubuntu@ip-172-31-83-19:~$ echo $Topic
linux
ubuntu@ip-172-31-83-19:~$
```

CLASSESS

Class 1:

Directoroy Navigation

- `.` represents a present folder
- `..` represents parent folder

- ~ Home folder
- linux contains folder and files
- when you get linux fully then it feels Everything in linux is a file.

Class 2:

Linux Directories

- Purpose of following directories
 - /etc
 - /bin
 - /home
 - /var
 - /tmp
- /etc :

Contains configuration files[password configuration,service files, user management, network configuration,package management(apt, dnf)]

- what is configuration ?
- configuration is change in settings of any program, for example : you can change chrome lightning mode from dark to colour or colour to dark this makes a chages in configuration files.
- In linux configuration files are present in /etc folder.
- simply it is settings for linux
- /bin :

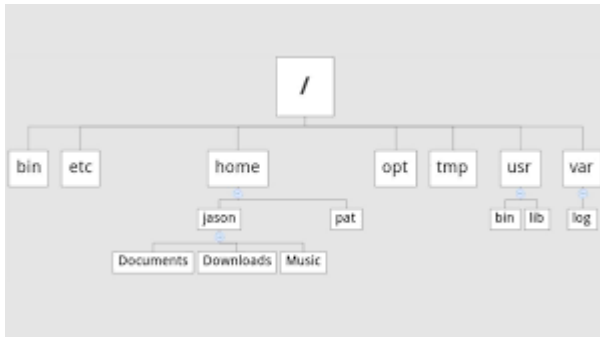
Contains binary files

- binary files are files which are not human readable, they are machine readable that is OS. a file consists of instructions which an operating system can understand and whenever you executues it cpu and memory is allocating and program starts running
- executables
 - Binaris
 - all linux commands will exist here
 - when you do cat for binary files it displays in non-understandable languages
 - Scripts
 - when you cat for these files you can see the scripts
 - !#/bin/bash
- /home :

Home directory is used to store multiple users data, when multiple users use the machine they should have data under their names since all thse are under home directory , in windows it is c:/users folder.

- /var : Contains Variable data like logs which get changes all the time.

- tmp/ : Temporary files are stored here. Temporary files are files which are created for a short period of time and then deleted. For example,...when you download a file from the internet, it is first saved as a temporary file in the /tmp directory, and then moved to its final destination once the download is complete.
- tree structure of linux folders



Navigation and file system

- Absolute paths
- Relative paths
- Commands :
 - pwd - present directory
 - ls - list out files
 - cd - get into
- go to cd /var/log and do ls, now how to find whether the listed names are files or folders

```

ubuntu@ip-172-31-9-1:/var/log$ ls -al
total 588
drwxrwxr-x  11 root    syslog      4096 Aug 11 07:43 .
drwxr-xr-x  13 root    root         4096 Aug 11 07:43 ..
lrwxrwxrwx   1 root    root           39 Jul  1 16:00 README -> ../..
/usr/share/doc/systemd/README.logs
-rw-r--r--   1 root    root         444 Jul  1 16:03 alternatives.log
drwx-----   3 root    root         4096 Aug 11 07:43 amazon
-rw-r-----   1 root    adm           0 Aug 11 07:43 apport.log
drwxr-xr-x   2 root    root         4096 Jul  1 16:06 apt
-rw-r-----   1 syslog  adm        41556 Aug 11 09:18 auth.log
-rw-rw----   1 root    utmp       31488 Aug 11 08:38 btmp
drwxr-x---   2 _chrony  _chrony    4096 Aug 11 07:43 chrony
-rw-r-----   1 root    adm        4286 Aug 11 07:43 cloud-init-outp

```

- ls -al : list out files and folders with their permissions, ownership and size
- anything which is starting with d is directory.
- get into home directory and find out hidden files
- anything which is starting with .txt is a hidden file.
- ls -a : list out all files and folders including hidden files

File Manipulation

- File Management
 - create :
 - touch => creates an empty file

- edit
 - text editor
 - vim (learnig site openvim.com)
 - nano
- delete
 - rm -r directoryname
 - rm -rf force deletion
 - rm -i file name -interactive deletion
 - rm *.txt -deletes all the files with extension .txt
- Folder Management
 - create
 - mkdir directoryname
 - mkdir f1 f2 d1 d2 -create multiple foders
 - edit
 - move
 - mv d1 d2 -folder d1 will move to folder d2.
 - delete
 - rmdir dirname -to delete an empty directory
 - rm -r directoryname
 - rm -rf force deletion
 - rm -i file name -interactive deletion
 - rm *.txt -deletes all the files with extension .txt

Users and Groups

Users

- Check Users in linux :
 - cat /etc/passwd
 - getent passwd
 - Check Single user:
 - sudo chage -l sai
 - Check whether account locked:(L defines locked)
 - sudo passwd -S sai
 - Check User exist or not
 - getent passwd | grep sai
 - list users above number 1000
 - getent passwd {1000..1010}
 - Check current user
 - **who** or **users**
- Create User :
 - sudo adduser aravindh
- Delete User :

- `sudo deluser aravindh`
- Change password :
 - `sudo passwd sai`
- Set account expiry date
 - `sudo chage -E 2025-05-30 sai`
- Set account expiry to never
 - `sudo chage -E -1 sai`
- Lock User
 - `sudo passwd -l sai`
- Unlock User
 - `sudo passwd -u sai`
- change Username :
 - `sudo usermod -l "saib" sai`
- Add User to a group :
 - `sudo usermod -aG group1 saib`
- Remove User form a Group
 - `sudo gpasswd -d saib group1`
- This command sets:
 - Minimum days between password changes to 7 days.
 - Maximum days before the password expires to 90 days.
 - Warning days before expiration to 14 days.
 - Account expiration date to December 31, 2024.
 - `sudo chage -m 7 -M 90 -W 14 -E 2024-12-31 sai`

Groups

- Check groups in linux :
 - `cat /etc/group`
 - `getent group`
 - `getent group | grep group1`
 - `getent group {1000..1010}`
- To check groups assigned to him
 - `groups`
- To check groups assigned for specific user
 - `groups saib`
- To checks users identity

- id sai
- To check id
 - id
- Create Newgroup :
 - sudo groupadd group1
- Delete Group :
 - sudo delgroup group1
- Display all users in specific group
 - getent group group1

VIM Editor :

- Basically VIM has two modes
 - insert mode
 - Normal mode
- insert mode allows you to write text same as text editor.
- Normal mode provides an efficient way to manipulate and navigate to text.
- At any time, you can see which mode you are in on the status bar which is located at the top of the editor.
- To change between modes, use **Esc** for normal mode and **i** for insert mode

Cursor movement :

- h - moves left similar to left navigation key **<**.
- l - moves right similar to right navigation key **>**.
- k - move top
- j - move down

Shell and Terminal

Terminal :

- Terminal is a software that allows you to type commands ex: git,bash which are installed on your system

Shell :

- the commands which you wrote in the terminal understands by Shell.
- the software which understands the commands is called shell. windows : * DOS * PowerShell Linux : * bash * flash ..etc
- how to list all the shells

```
cat /etc/shells
```

```

Ara@Linux-TestMachine:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/usr/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/usr/bin/dash
/usr/bin/screen
/usr/bin/tmux
Ara@Linux-TestMachine:~$
```

- in Linux there are two types of users
 - System Users
 - this is created to run some applications/services in linux
 - Users
- How to tell whether a user is System user / User.
 - when you see all the users list, where the users doesn't have 'nologin' at last is a system user
 - generally users will be associated with shell and system user will not be associated with shells
- On all linux machines we have **root** User, root user will have full control over machine
- how to change to root user

```
sudo -i
```

- to know who is current user

```
whoami
```

- users can be associated with the groups
- Lets create a Group
 - developer
 - tester
 - devops
- every user has a unique UID and every group will have unique GID

```
Username:password:UID:GID:GECOS:home_directory:login_shell
```

- the users will be listed in above format
- here when you create a user UID will be automatically created, and GID(group id) if you have not assigned to any group it will assign it will create a new group with username and brings its id.
- if you assign any group, it shows respective group id.
- lets create a 3 Users
 - dev1
 - group : developer
 - shell : /bin/bash
 - home_directory : /home/dev1
 - test1
 - group : developer
 - shell : /bin/bash
 - home_directory : /home/dev1
 - devops1
 - group : developer
 - shell : /bin/bash
 - home_directory : /home/dev1

```
sudo adduser dev2 --shell /bin/bash --gid 1005
```

```
Ara@Linux-TestMachine:~$ sudo adduser dev2 --shell /bin/bash --gid 1005
info: Adding user `dev2' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new user `dev2' (1010) with group `developer (1005)' ...
info: Creating home directory `/home/dev2' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for dev2
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
info: Adding new user `dev2' to supplemental / extra groups `users' ...
info: Adding user `dev2' to group `users' ...
Ara@Linux-TestMachine:~$ cat /etc/passwd
```

```
sudo useradd -s /bin/bash -g 1006 -m test1
```

```
Ara@Linux-TestMachine:~$ sudo useradd -s /bin/bash -g 1006 -m test2
Ara@Linux-TestMachine:~$ getent passwd
```

- adding users to **sudoers** group
 - visudo
 - add to sudo group **usermod -aG sudo dev1**
- visudo :
 - open **sudo visuo**

```
# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
%developer ALL=(ALL:ALL) NOPASSWD:ALL
dev1    ALL=(ALL:ALL) NOPASSWD:ALL
# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

- if a line starts with **%** it is a group and if it starts normally it is a user.

- see above image

```
# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

%developer ALL=(ALL:ALL) NOPASSWD:ALL
dev1 ALL=(ALL:ALL) NOPASSWD:ALL
# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

- see above image the blue colored box content is written for not to ask password for group %developer and user dev1

to check all disks in linux machine

```
dev1@Linux-TestMachine:~$ sudo lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   30G  0 disk
├─sda1       8:1    0   29G  0 part /
├─sda14      8:14   0    4M  0 part
├─sda15      8:15   0  106M  0 part /boot/efi
└─sda16     259:0   0   913M  0 part /boot
sdb          8:16   0    4G  0 disk
└─sdb1       8:17   0    4G  0 part /mnt
dev1@Linux-TestMachine:~$ |
```


sudo visudo[second way of providing sudo permissions to user]

- if you create new user and add it in this file, he will get full permissions as **sudoers**

```
# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
raju ALL=(ALL:ALL) NOPASSWD:ALL
# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d
|
```

^{^G} Help ^{^O} Write Out ^{^W} Where Is ^{^K} Cut ^{^T} Execute
^{^X} Exit ^{^R} Read File ^{^N} Replace ^{^U} Paste ^{^J} Justify

- if you create a new user, new group and add user to new group. then if you add this new group to this file the group will get **sudoers** permissions. ex: * create a new user john
 - create a new group Tester
 - add john to Tester group
 - open **sudo visudo** and add group with all permissions
 - %Tester ALL(ALL:ALL) NOPASSWD:ALL**
 - save and exit.
 - now switch user **su john** using john

- now try to do `sudo apt update` it will work.

```
# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
raju ALL=(ALL:ALL) NOPASSWD:ALL
%Tester ALL=(ALL:ALL) NOPASSWD:ALL
# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d
|
```

^{^G} Help ^{^O} Write Out ^{^W} Where Is ^{^K} Cut ^{^T} Execute
^{^X} Exit ^{^R} Read File ^{^_\} Replace ^{^U} Paste ^{^J} Justify

- this command shows all the devices connected to the linux machine `sudo ls /dev`

```
john@Linux-TestMachine:~$ sudo ls /dev
autofs          loop7          stderr         tty34          tty8           ttyprintk
block           mapper         stdin          tty35          tty9           udmabuf
bsg             mcelog         stdout         tty36          ttyS0          uinput
btrfs-control   mem            tpm0           tty37          ttyS1          urandom
cdrom           mqueue         tpmrm0         tty38          ttyS10         userfaultfd
char            net            tty            tty39          ttyS11         vcs
console         null           tty0           tty4           ttyS12         vcs1
core            nvme-fabrics   tty1           tty40          ttyS13         vcs2
cpu             nvram          tty10          tty41          ttyS14         vcs3
cpu_dma_latency port           tty11          tty42          ttyS15         vcs4
cuse            ppp            tty12          tty43          ttyS16         vcs5
disk            psaux          tty13          tty44          ttyS17         vcs6
dma_heap        ptmx           tty14          tty45          ttyS18         vcsa
dri             ptp0           tty15          tty46          ttyS19         vcsa1
ecryptfs        ptp_hyperv     tty16          tty47          ttyS2          vcsa2
fb0             pts            tty17          tty48          ttyS20         vcsa3
fd              random         tty18          tty49          ttyS21         vcsa4
full            rfkill         tty19          tty5           ttyS22         vcsa5
fuse            root           tty2           tty50          ttyS23         vcsa6
hpet            rtc            tty20          tty51          ttyS24         vcsu
hugepages       rtc0           tty21          tty52          ttyS25         vcsu1
hwrng           sda            tty22          tty53          ttyS26         vcsu2
initctl         sda1           tty23          tty54          ttyS27         vcsu3
input           sda14          tty24          tty55          ttyS28         vcsu4
kmsg            sda15          tty25          tty56          ttyS29         vcsu5
log             sda16          tty26          tty57          ttyS3          vcsu6
loop-control    sdb            tty27          tty58          ttyS30         vfio
loop0           sdb1           tty28          tty59          ttyS31         vga_arbiter
loop1           sg0            tty29          tty6           ttyS4          vhost-net
loop2           sg1            tty3           tty60          ttyS5          vhost-vsock
loop3           sg2            tty30          tty61          ttyS6          vmbus
loop4           shm            tty31          tty62          ttyS7          zero
loop5           snapshot       tty32          tty63          ttyS8          zfs
loop6           sr0            tty33          tty7           ttyS9
```

FILE PERMISSIONS

- `chmod` command is used to change file permissions
- `chown` command is used to change file ownership
- `chgrp` command is used to change file group ownership

chmod

Practical :


- Create a new file `touch 1.sh`

- do `ls -l 1.sh`

```
Ara@Linux-TestMachine: ~/file × + v
Ara@Linux-TestMachine:~/files$ touch 1.sh
Ara@Linux-TestMachine:~/files$ ls
1.sh
Ara@Linux-TestMachine:~/files$ ls -l 1.sh
-rw-rw-r-- 1 Ara Ara 0 Nov 29 09:32 1.sh
Ara@Linux-TestMachine:~/files$ |
```

- removing all permissions

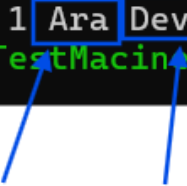
```
Ara@Linux-TestMachine:~/files$ chmod u-rwx 1.sh
Ara@Linux-TestMachine:~/files$ chmod g-rwx 1.sh
Ara@Linux-TestMachine:~/files$ chmod o-rwx 1.sh
Ara@Linux-TestMachine:~/files$ ls -l 1.sh
----- 1 Ara Ara 0 Nov 29 09:32 1.sh
Ara@Linux-TestMachine:~/files$ |
```



- here we have three different access **owner** **group** **other**.
- Owner = u
- Group = g
- Other = o
- the command comes as below
 - `chmod u+r 1.sh` - single permission at a time
 - `chmod u+rw 1.sh` - double permission at a time
 - `chmod u+rwx 1.sh` - triple permission at a time
 - `chmod g+r 1.sh`
 - `chmod g+rw 1.sh`
 - `chmod g+rwx 1.sh`
 - `chmod o+r 1.sh`
 - `chmod o+rw 1.sh`

- `chmod o+rw 1.sh`

```
Ara@Linux-TestMachine:~/files$ sudo chown :Developer 1.sh
Ara@Linux-TestMachine:~/files$ ls -l 1.sh
-rwxrwxrwx 1 Ara Developer 0 Nov 29 09:32 1.sh
Ara@Linux-TestMachine:~/files$ |
```



```
Ara@Linux-TestMachine:~/files$ sudo chown john:Devops 1.sh
Ara@Linux-TestMachine:~/files$ ls -l 1.sh
-rwxrwxrwx 1 john Devops 0 Nov 29 09:32 1.sh
Ara@Linux-TestMachine:~/files$ |
```

**