Linux Systems

Introduction to Linux:

Basics of Linux:

Contents:

1. Introduction to Linux
2. Basic CLI Commands
3. Understanding files in Linux
4. Filters & Redirection
5. Users & Groups
6. Sudo
7. Software Management
8. Services & Processes
9. Good to Know Commands for Devops
10. Server Management

There are Basically four areas to work on Linux

01. Commands – Syntax, helps & lots of commands
02. Files – Understanding, Editing, Filtering & Security
03. Software – Management
04. Server – Management

Open Source:

Open-Source Software is the Software with source code that anyone can inspect, modify and enhance.

Linux Origins

- **1984: The GNU Projects and the Free Software Foundations**
  - Creates open-source versions of Unix Utilities
  - Creates the General Public License
    a. Software License enforcing open-source principles
- **1991**: **Linus Torvalds**
  - Creates an open-source UNIX – Like Kernel, released under the GPL
  - Ports some GNU utilities, Solicits assistance online

Today:

- Linux Kernel + GNU Utilities = Complete, Open-Source Unix Like OS
- Packaged for target audiences for Distribution

Linux Principles

- Everything is a File (Including Hardware)
- Small Single Purpose Programs
- Ability of Change the Programs for Complex Operations
- Avoid Captive User Interface
- Configuration data stored in text file

Why Linux?

- It is an open source
- Community Support
- Support Wide Variety of Hardware
- Customization
- Most of Servers run on Linux
- Automation
- Security

Architecture of Linux

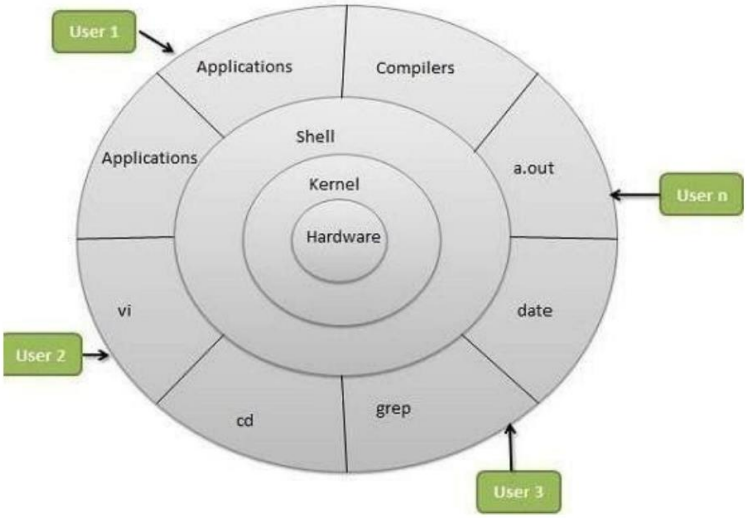


Fig. Linux Architecture

Popular Linux Distros

- **Popular Desktop Linux OS**
  - Ubuntu Linux
  - Linux Mint
  - Arch Linux
  - Fedora
  - Debian
  - OpenSUSE

- **Popular Server Linux OS**

    - Red Hat Enterprise Linux
    - Ubuntu Server
    - CentOS
    - SUSE Enterprise Linux

Most used Linux distros currently in IT Industry

RPM Based**: RHEL, Oracle Linux & Centos**

Debian Based**: Ubuntu Server, Kali Server**

Some Important Directories

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

Command and File Systems

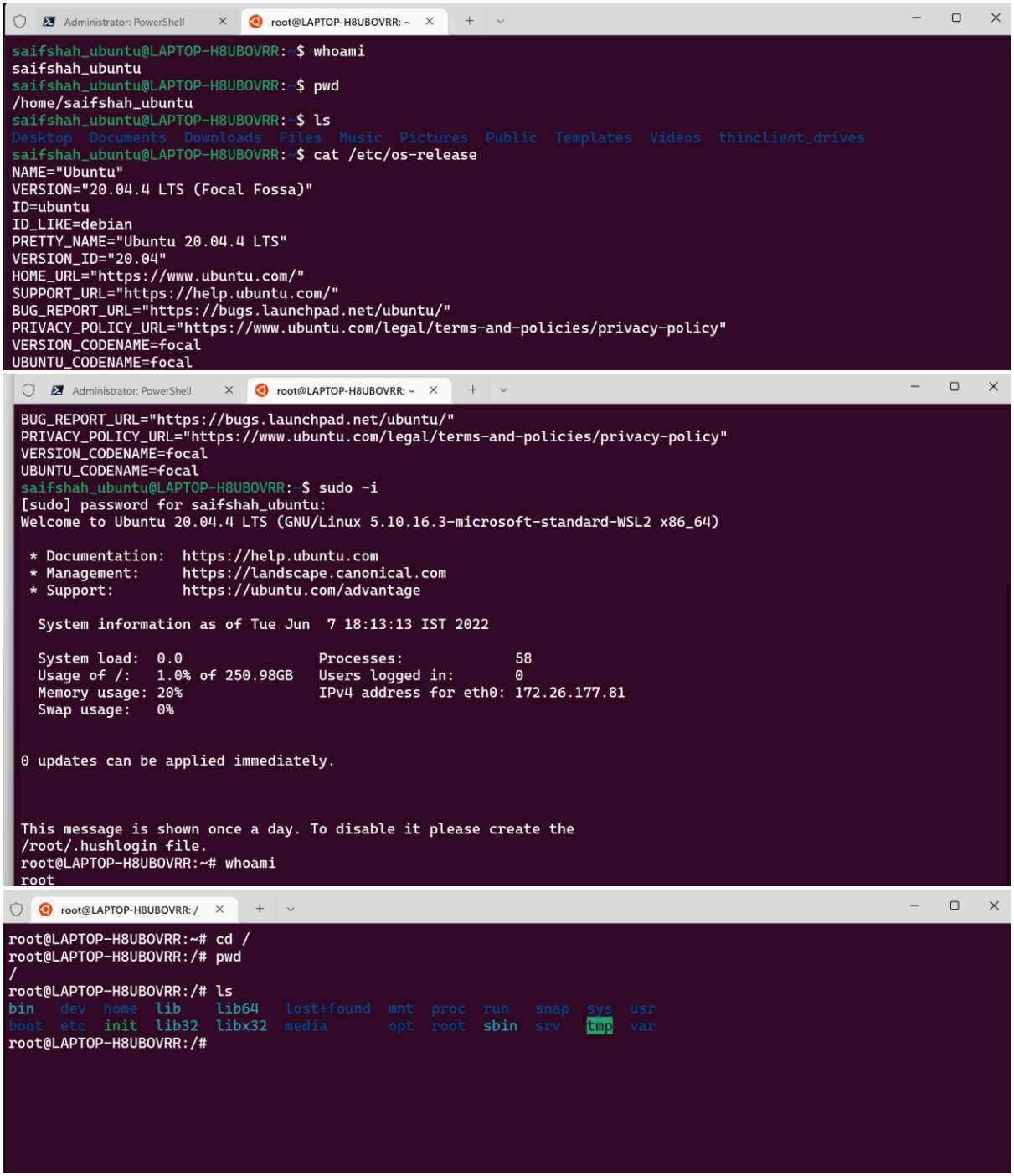


Fig. Basics Commands

User Executables: /bin, /usr/bin, /usr/local/bin

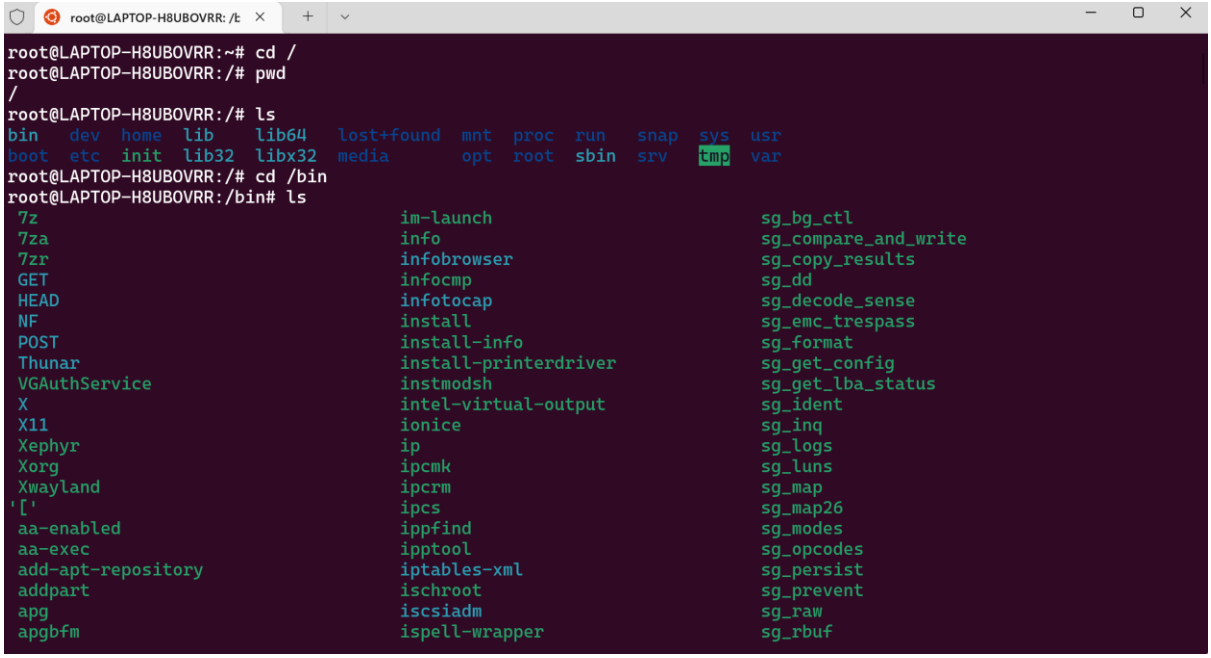


Fig. User Executables

System Executables: /sbin, /usr/sbin, /usr/local/sbin

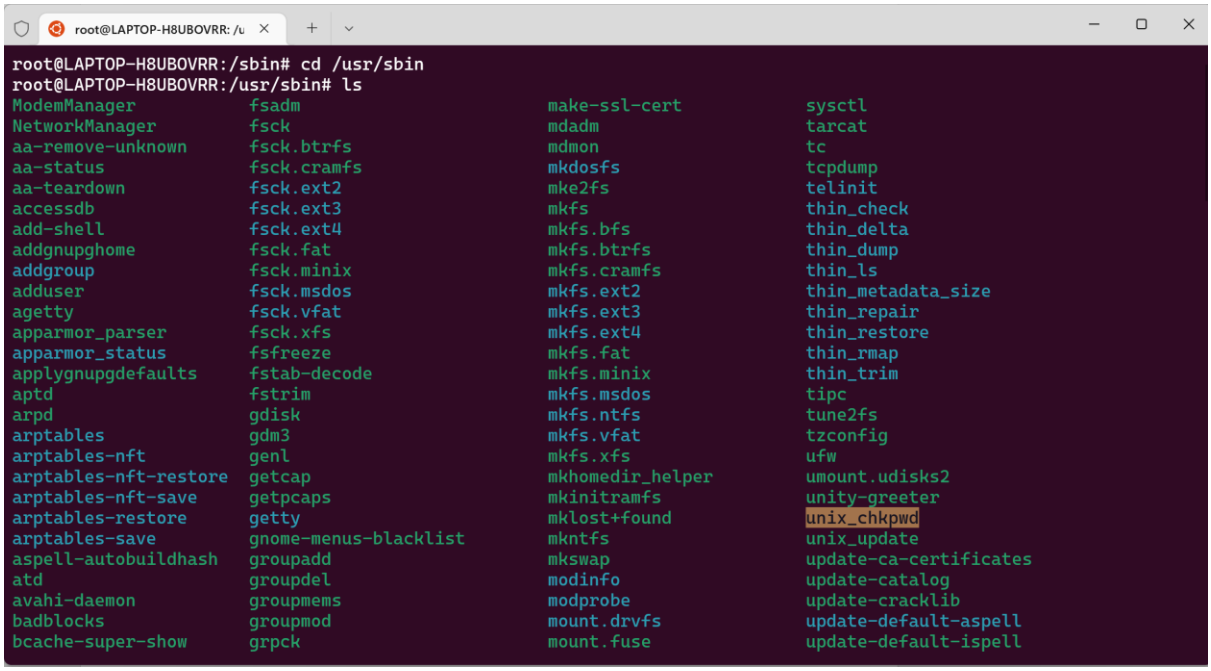


Fig. System Executables

Configuration: /etc

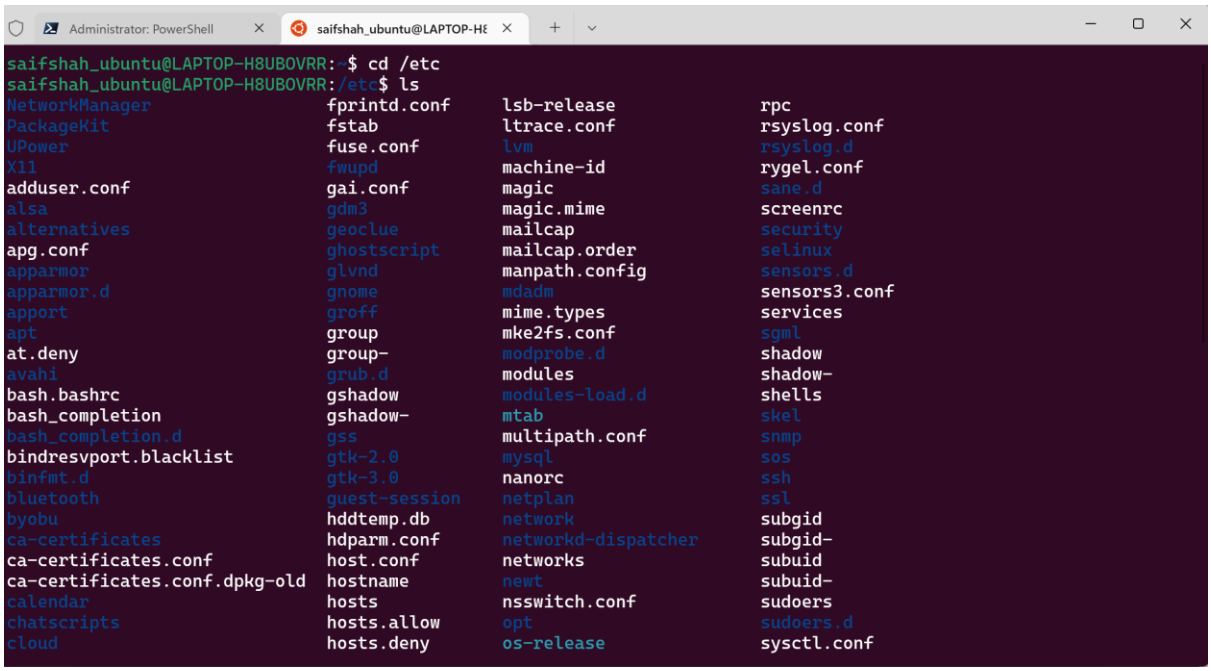


Fig. Configuration

saifshah_ubuntu@LAPTOP-H8UBOVRR:/etc$ cat /etc/hostname

LAPTOP-H8UBOVRR

saifshah_ubuntu@LAPTOP-H8UBOVRR:/etc$ cd /tmp/

saifshah_ubuntu@LAPTOP-H8UBOVRR:/tmp$ ls

Temp-3e9ab343-1b3f-424b-b596-b9d0836a4786 ssh-MLfaB9lwy4kU ssh-Z7gY3c6CDnF7 ssh-r7C8ZJ5nt4HI tmpoinmphoxpulse-PKdhtXMmr18n ssh-Sr0LS7ijgWCQ ssh-fOExaGFZ42WE ssh-sW3dYgBkFVID

ssh-LTAzDJ4CwiiM ssh-WULG7NyGwSQe ssh-iShxOmGJWvkV tmpaddon

System Information: /proc, /sys

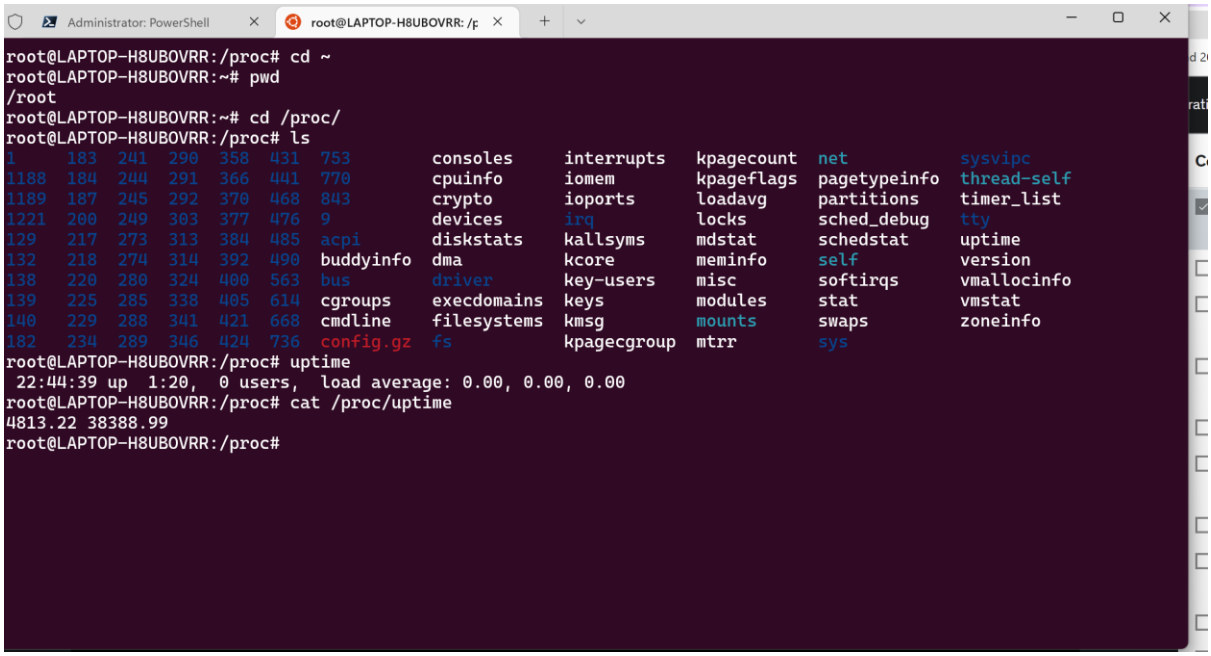


Fig. System Information

Shared Libraries: /lib, /usr/lib, /usr/local/lib

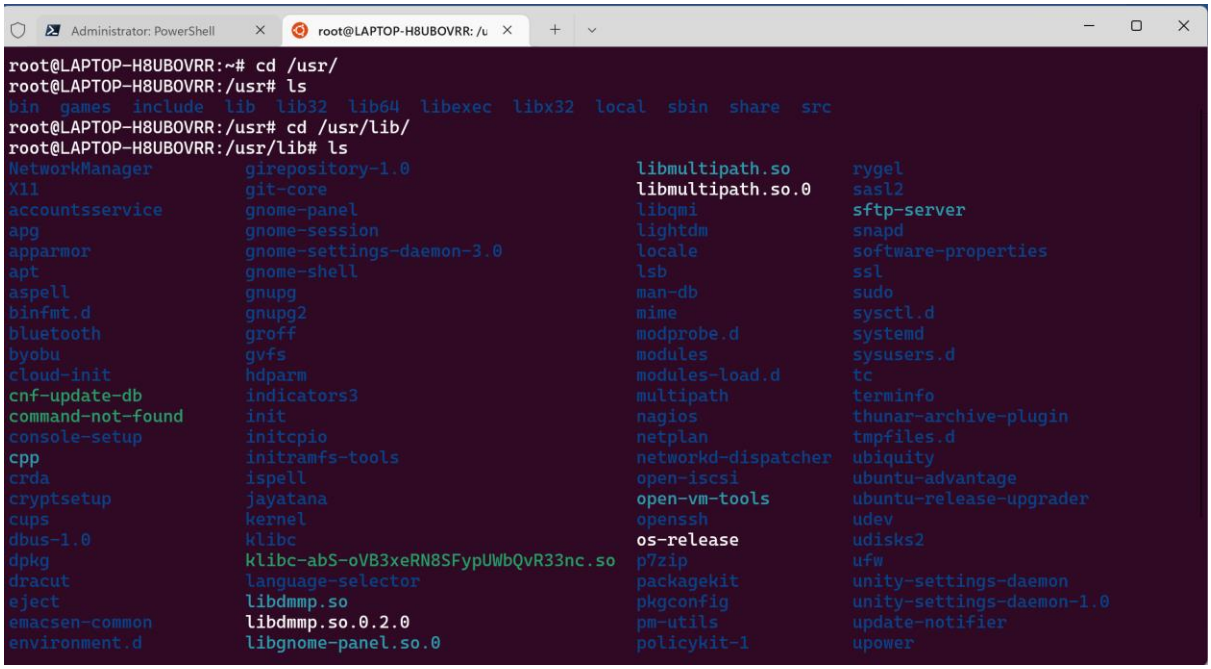


Fig. Shared Libraries

Basic Commands:

- **Know where you are? Present Working Directory**

Fig. present working directory

- **Creating a directory Folders in Home Directory**

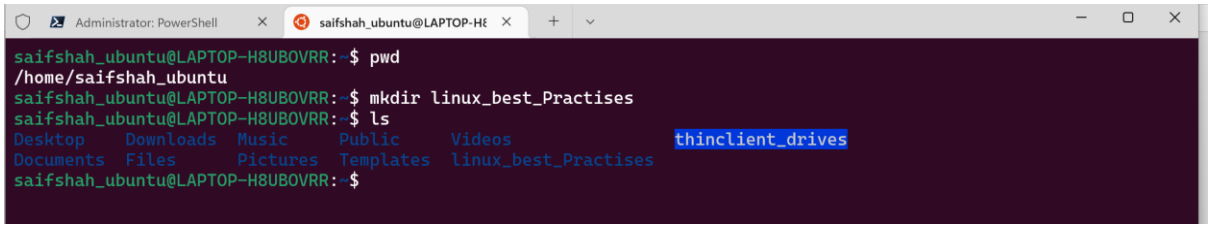


Fig. directory creation

- **Changing your Current directory to Linux Best Practises**

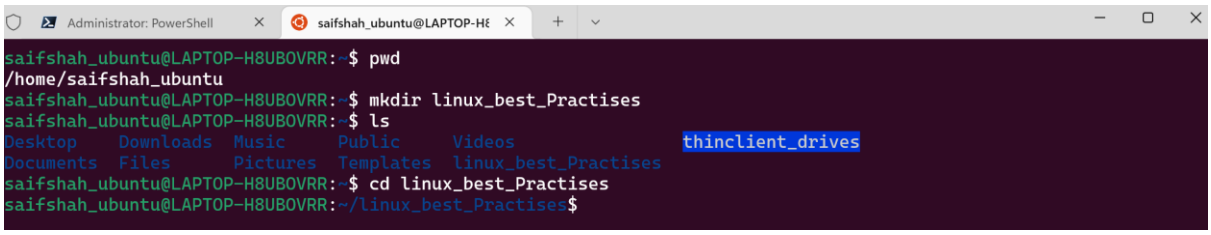


Fig. current directory to linux best practises

- **Create some more directories and list them with “ls” command**

Fig. Creating directory and listing with commands

- **Create some empty files with “touch” command and list them**

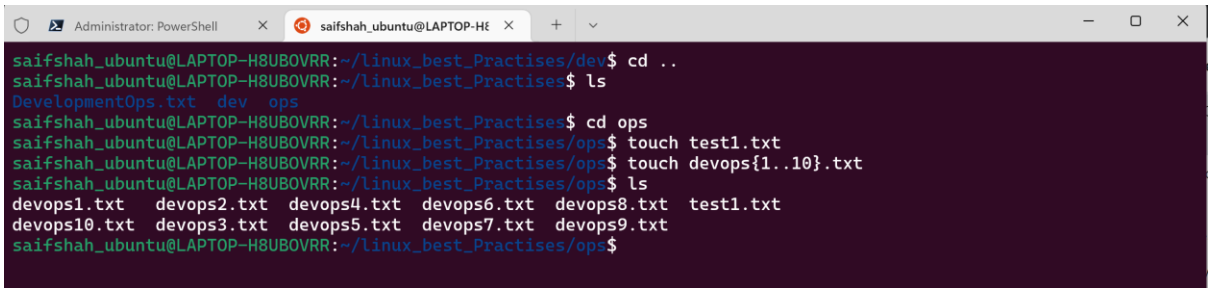


Fig. Creating some empty files with “touch” command

- **Reconfirm your location in your system**

Fig. reconfirm your location in your system

Absolute path and Relative path

What is a path?

A path is a unique location to a file or a folder in a file system of an OS. A path to a file is a combination of / and alpha-numeric characters.

What is an absolute path?

An absolute path is defined as the specifying the location of a file or directory from the root directory(/). In other words we can say absolute path is a complete path from start of actual filesystem from / directory. Some examples of absolute path

Some examples of absolute path:

/var/ftp/pub

/etc/samba.smb.conf

/boot/grub/grub.conf

If you see all these paths started from / directory which is a root directory for every Linux/Unix machines

What is the relative path?

Relative path is defined as path related to the present working directory(pwd). Suppose I am located in /home/saif_shah and I want to change directory to /home/saif_shah /linux-practices. I can use relative path concept to change directory to linux-practices and also ops directory.

```
saifshah_ubuntu@LAPTOP-H8UBOVRR:~$ pwd
/home/saifshah_ubuntu
saifshah_ubuntu@LAPTOP-H8UBOVRR:~$ ls
Desktop    Downloads  Music     Public     Videos                thinclient_drives
Documents  Files      Pictures  Templates  linux_best_Practises
saifshah_ubuntu@LAPTOP-H8UBOVRR:~$ cd linux_best_Practises/
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises$ ls
DevelopmentOps.txt  dev  ops
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises$ pwd
/home/saifshah_ubuntu/linux_best_Practises
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises$ cd ops/
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/ops$ ls
devops1.txt   devops2.txt  devops4.txt  devops6.txt  devops8.txt  test1.txt
devops10.txt  devops3.txt  devops5.txt  devops7.txt  devops9.txt
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/ops$ pwd
/home/saifshah_ubuntu/linux_best_Practises/ops
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/ops$
```

Fig. Relative Path Examples

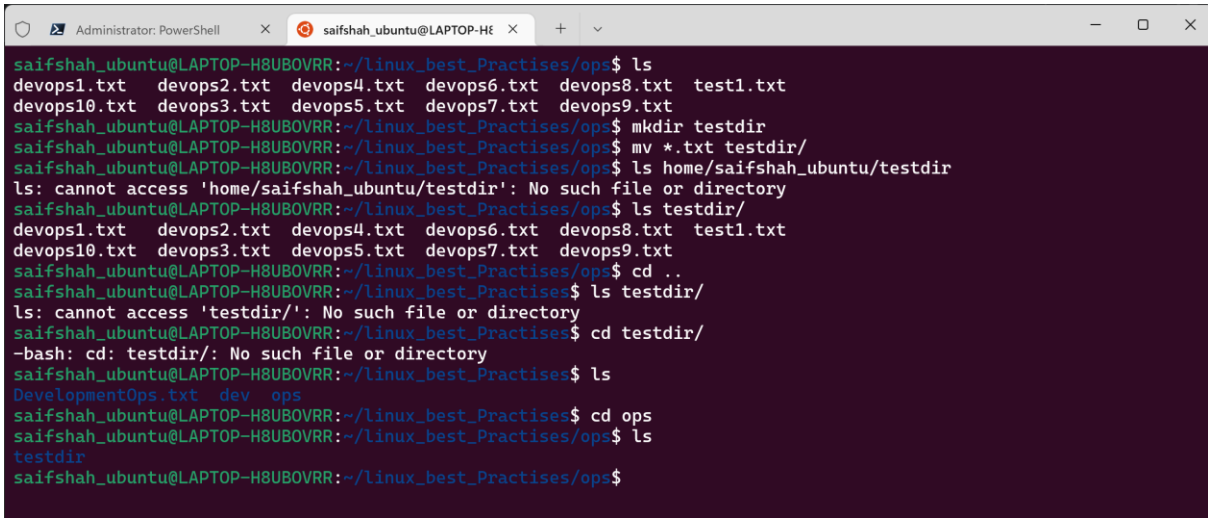


Fig. Moving files into another directory

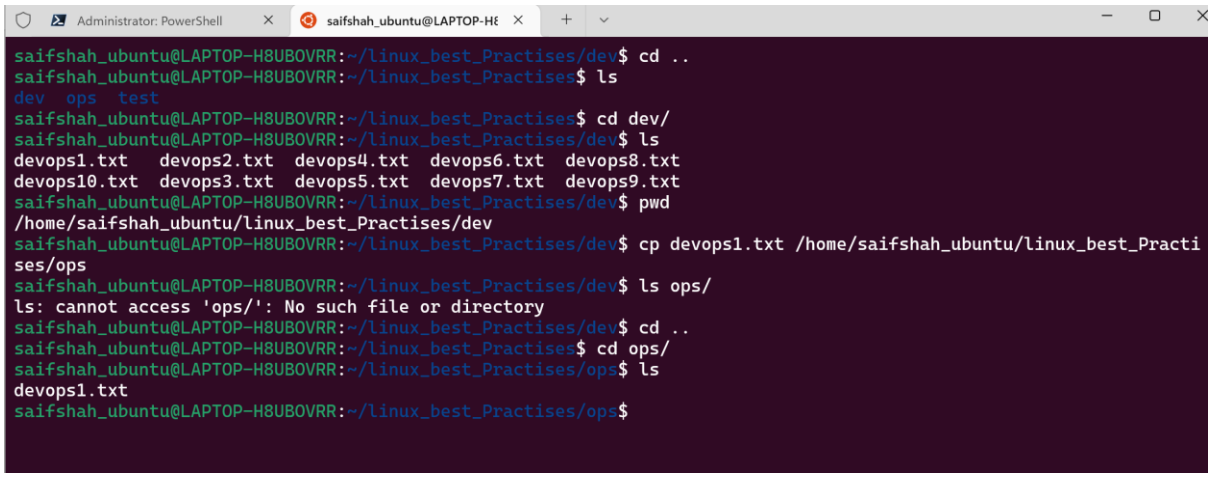


Fig. Copying the Files from one directory to another

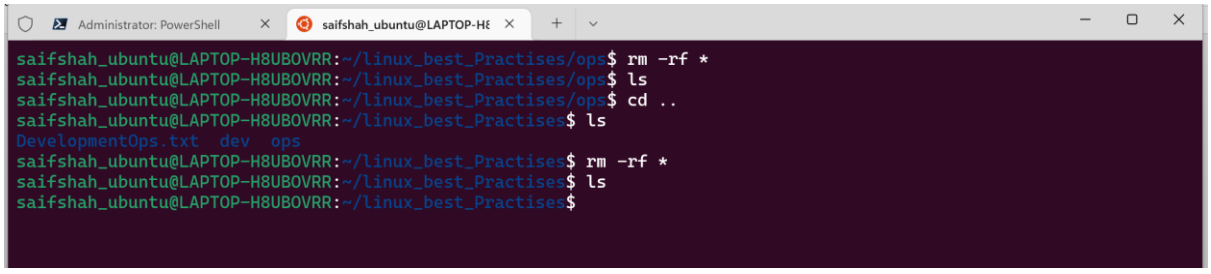


Fig. Removing Everything from Files

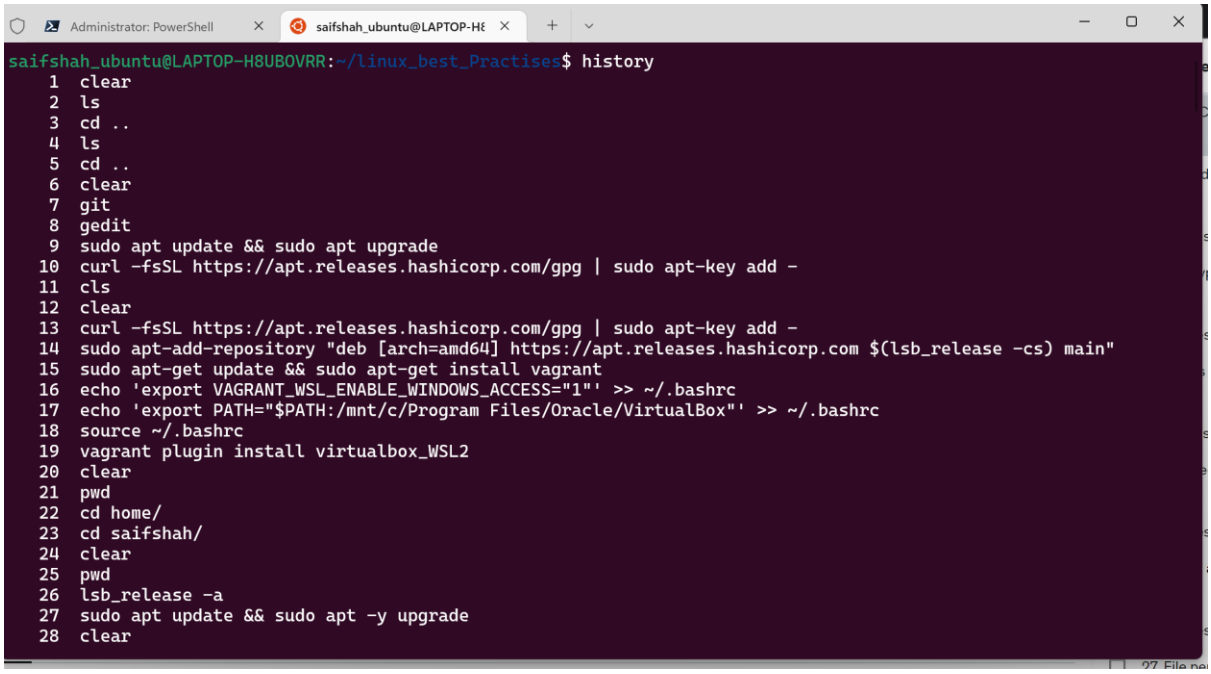


Fig. Shows Complete History

Vim Editor

VI Visual display editor

VIM Visual display editor improved

This is command mode editor for files.

Other editors in Linux are emacs, gedit vi editor is most popular

It has 3 modes:

1 Command Mode

2 Insert mode (edit mode)

3 extended command mode

Note: When you open the vim editor, it will be in the command mode by default

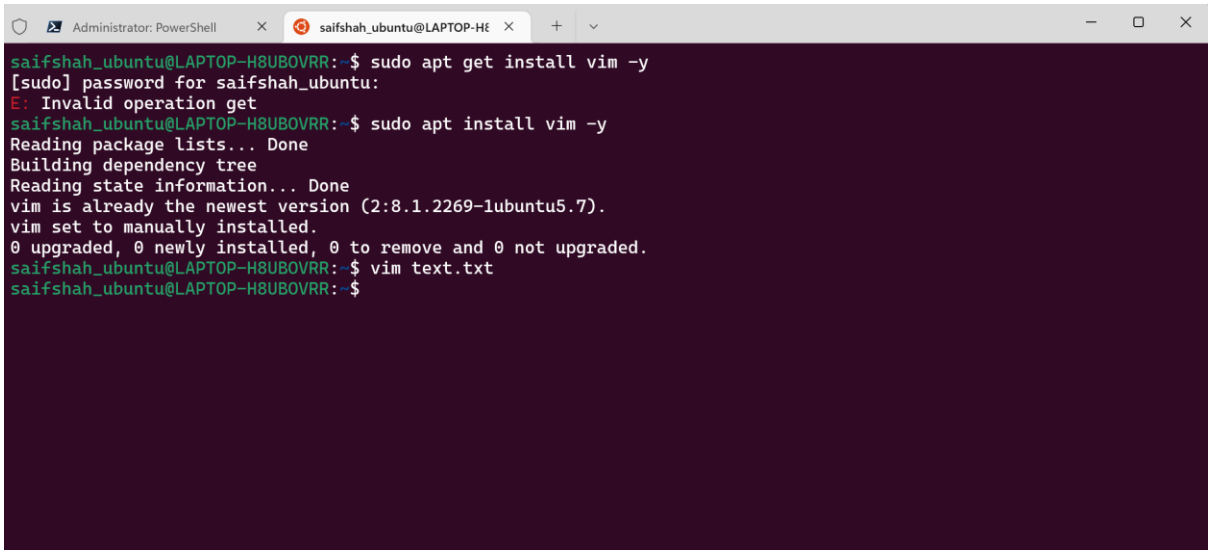


Fig. Vim Complete Text File Editor Installation

Creating First File of Linux:

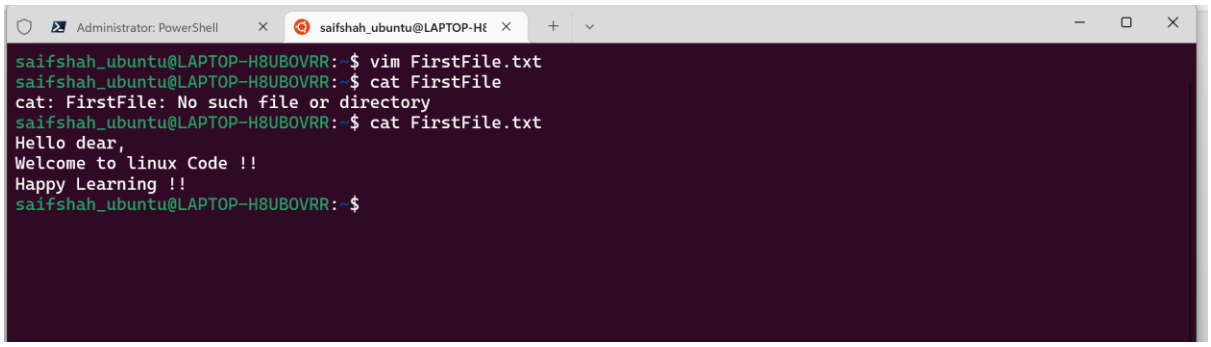


Fig. Successful Created First Linux File

Creating Lines with: se nu

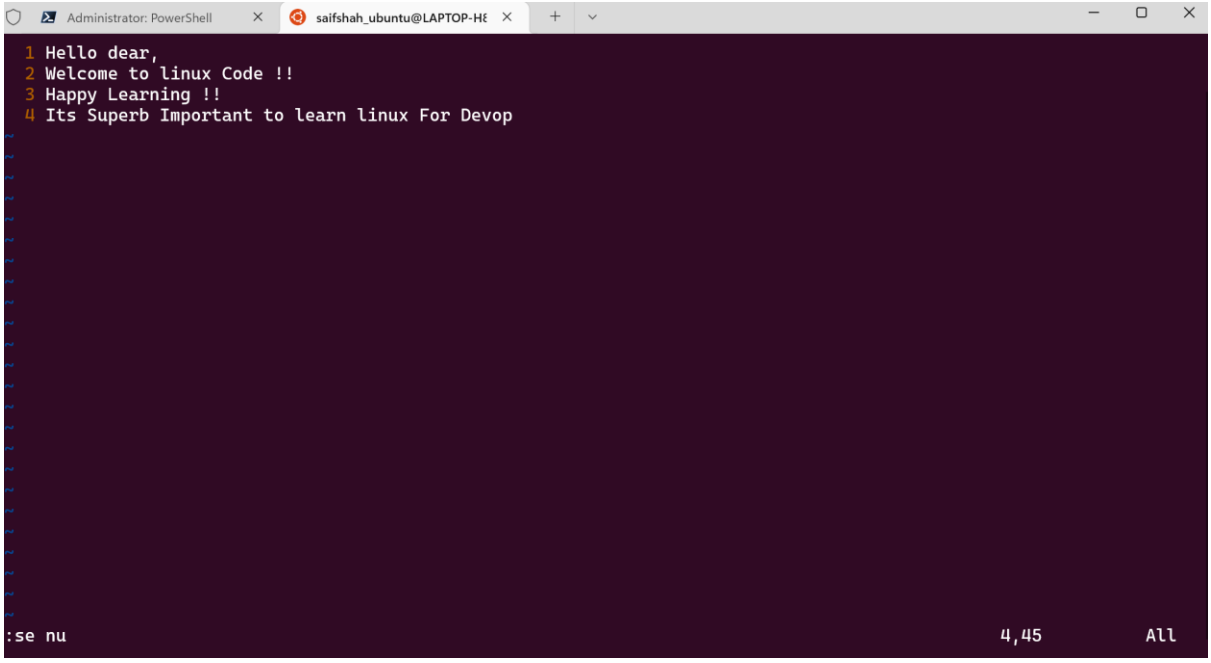


Fig. See all lines in Linux

Copying Files with: y

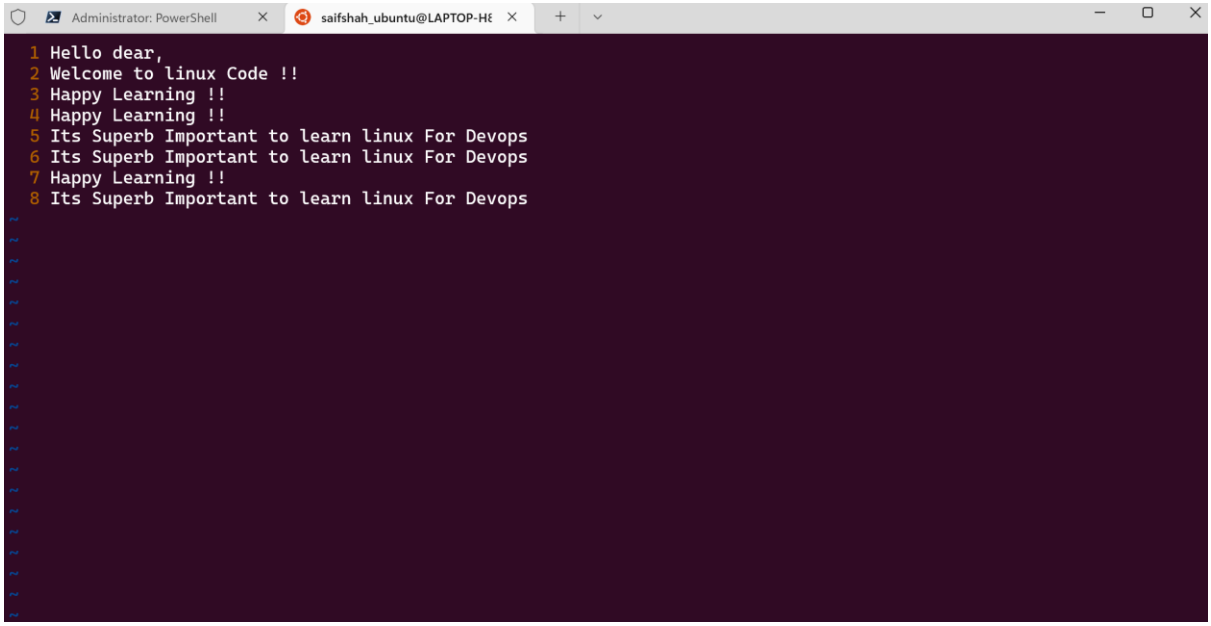


Fig. Copying Lines (Happy Learning!!)

Deleting lines with: d

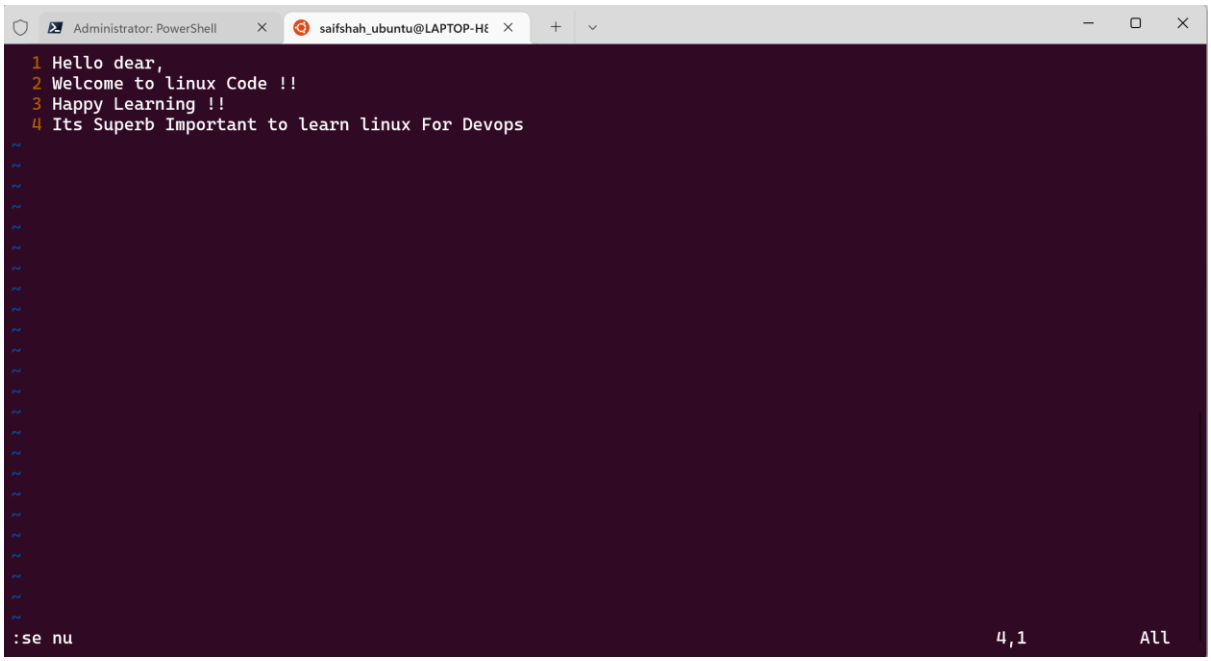


Fig. Deleting Files (Happy Learning which is twice)

Undo Changes with: u

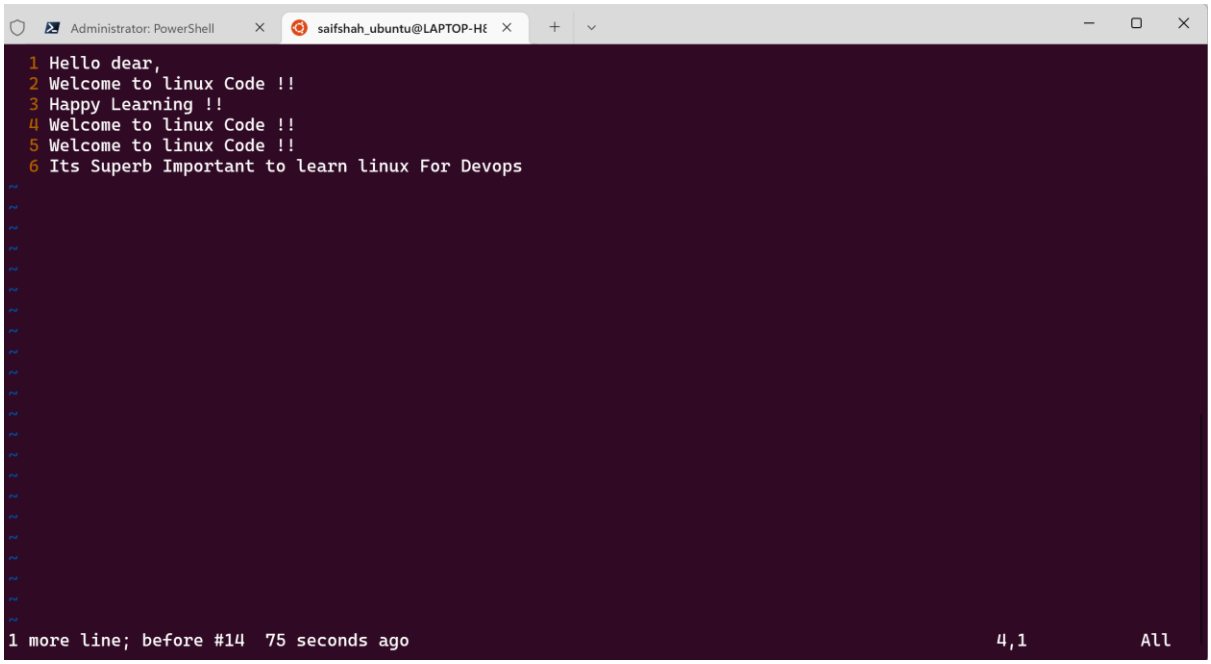


Fig. Undo all Commands

- **Cut Paste: d - Cut p – Paste**

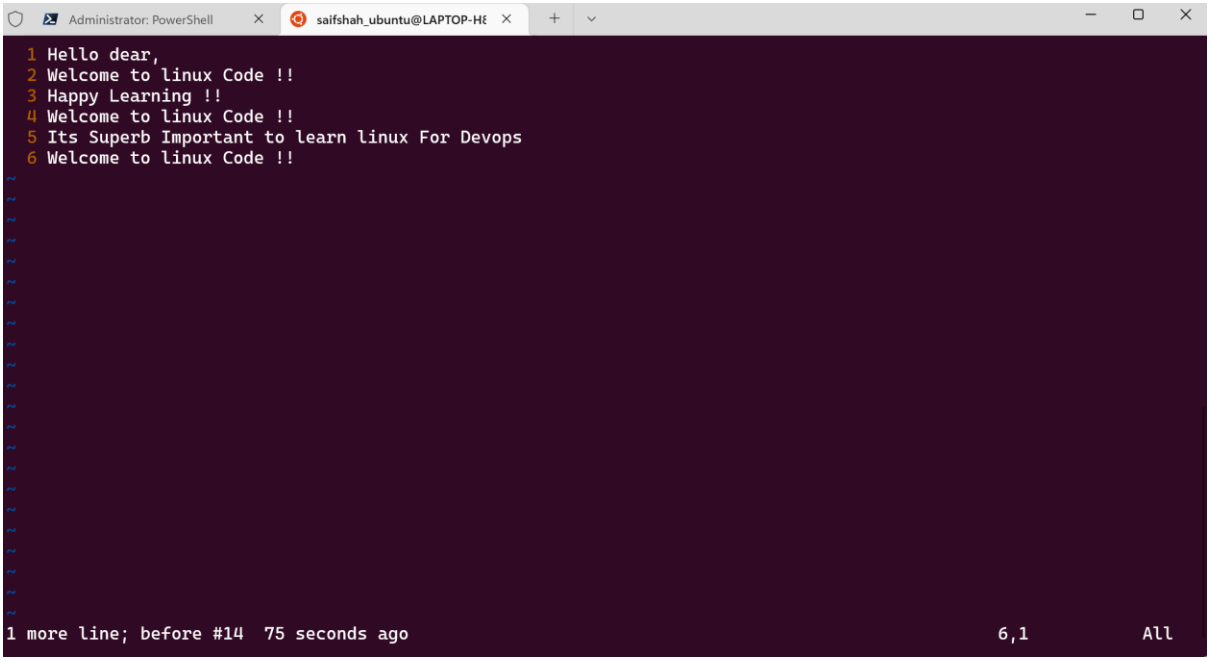


Fig. Cut paste Lines (Welcome to Linux Code)

⇨ **Searching: /**

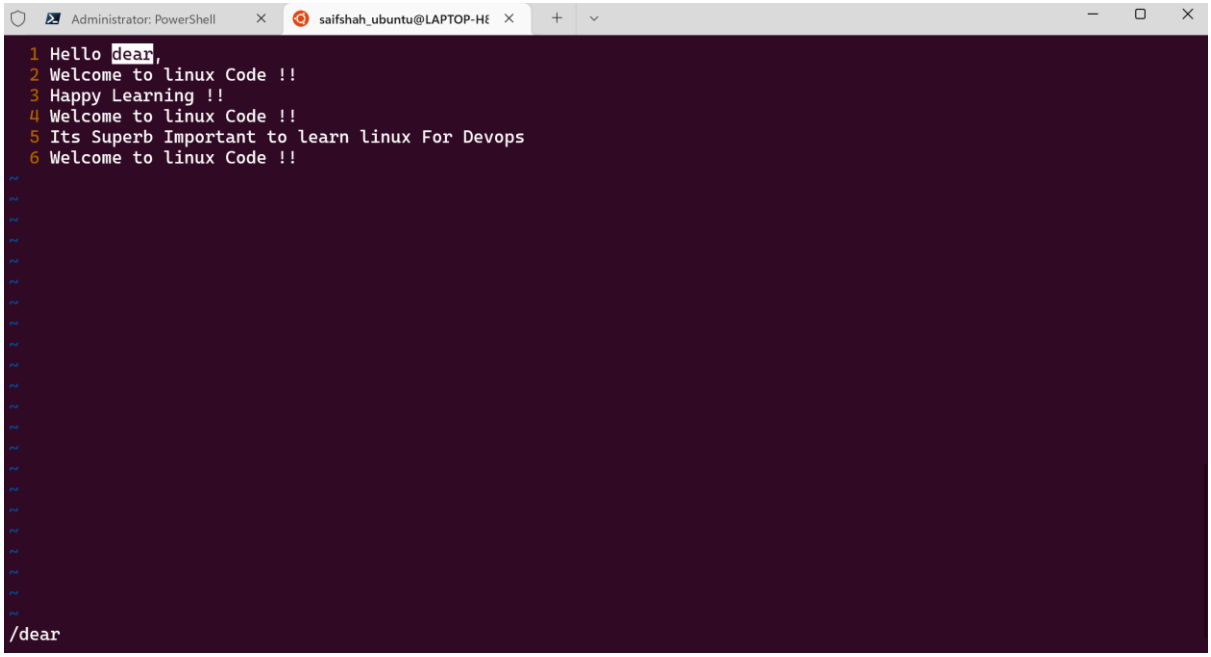


Fig. Searching the word (dear) in Lines

File System in Linux

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

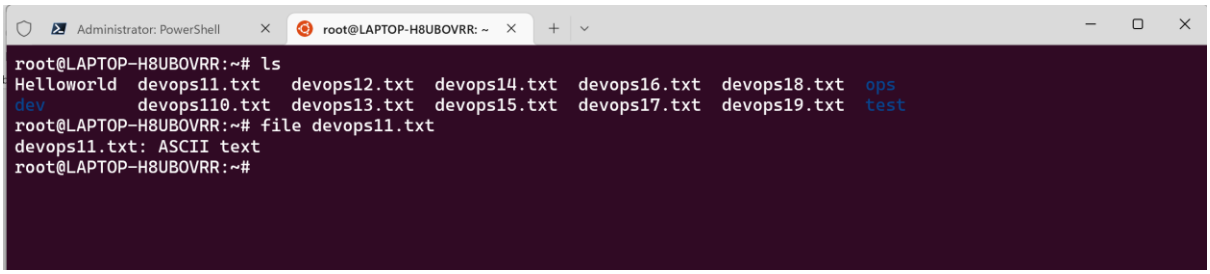


Fig. regular file

Fig. Binary File

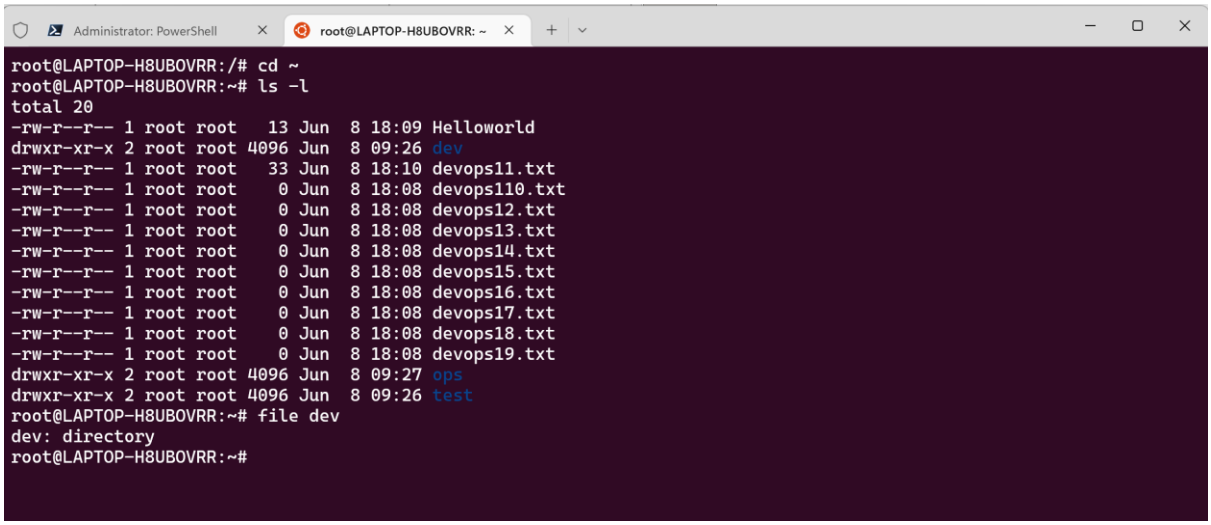


Fig. Directory File

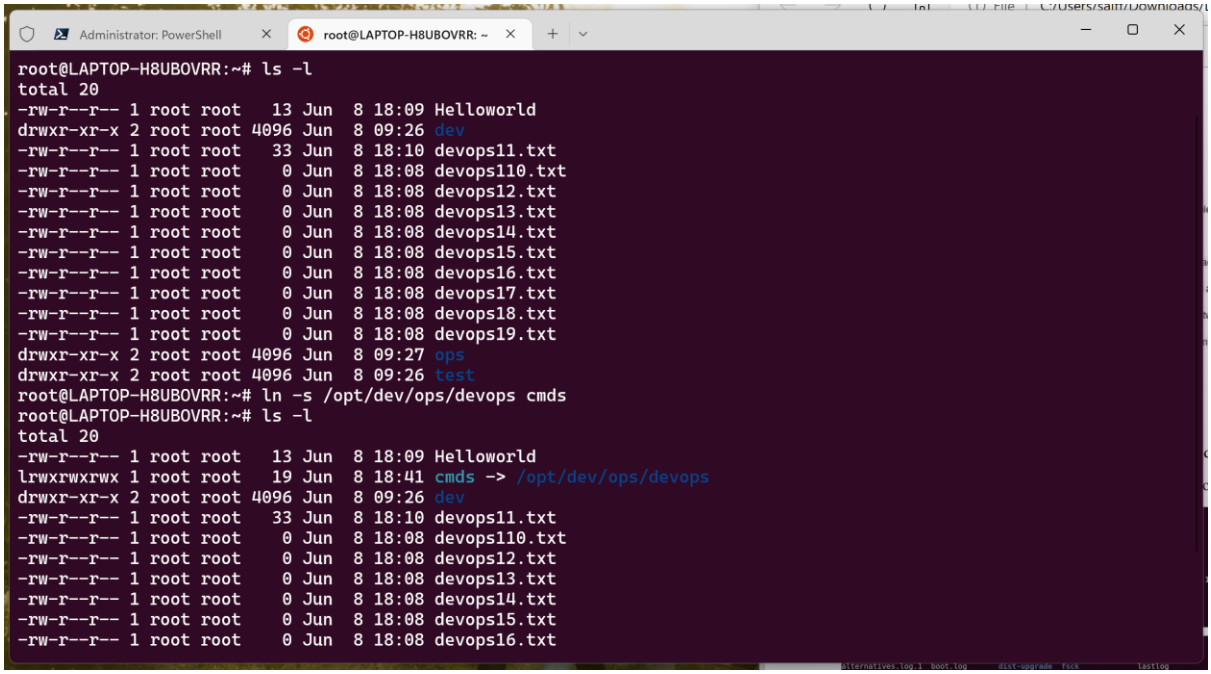


Fig. Creates Linking in Files

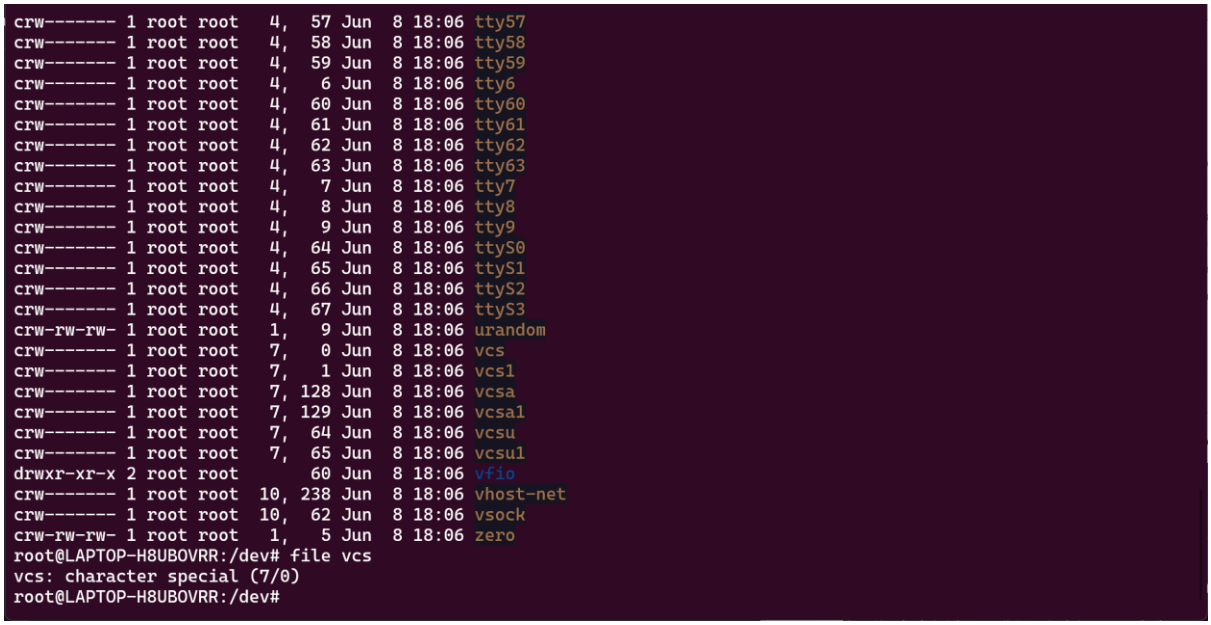


Fig. Character Files in dev directory

Fig. Block Files in dev directory

Time Stamp in File: ls -lt

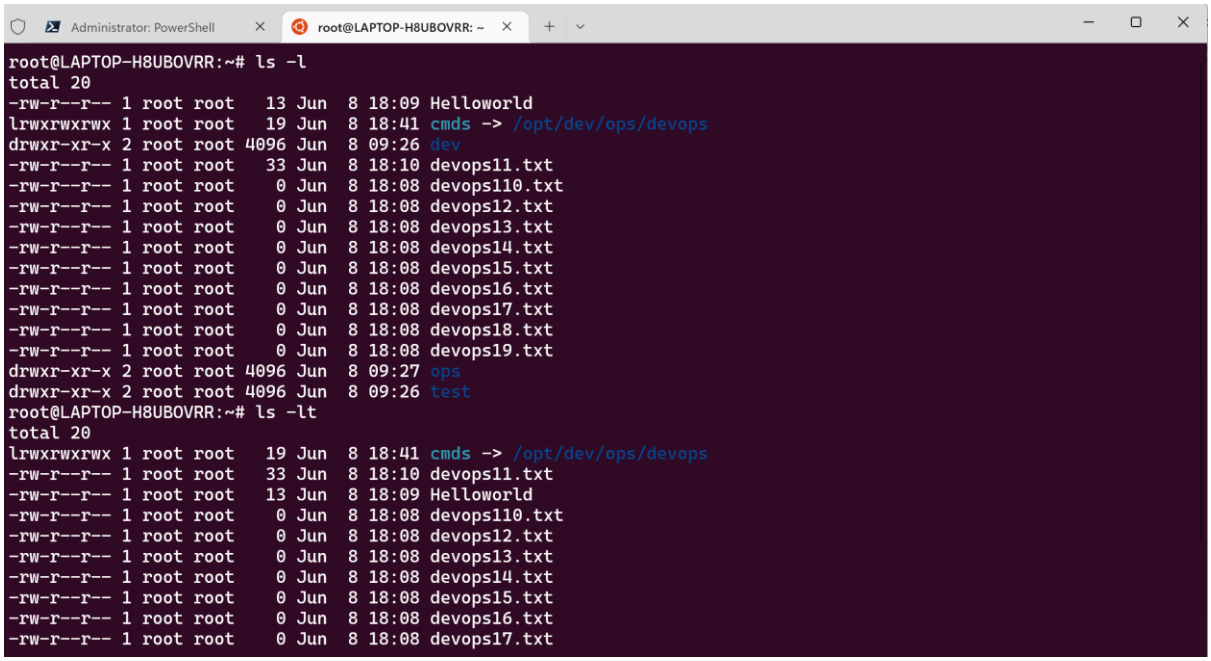


Fig. Timestamp file at Top

Time Stamp in File Reverse ls-ltr

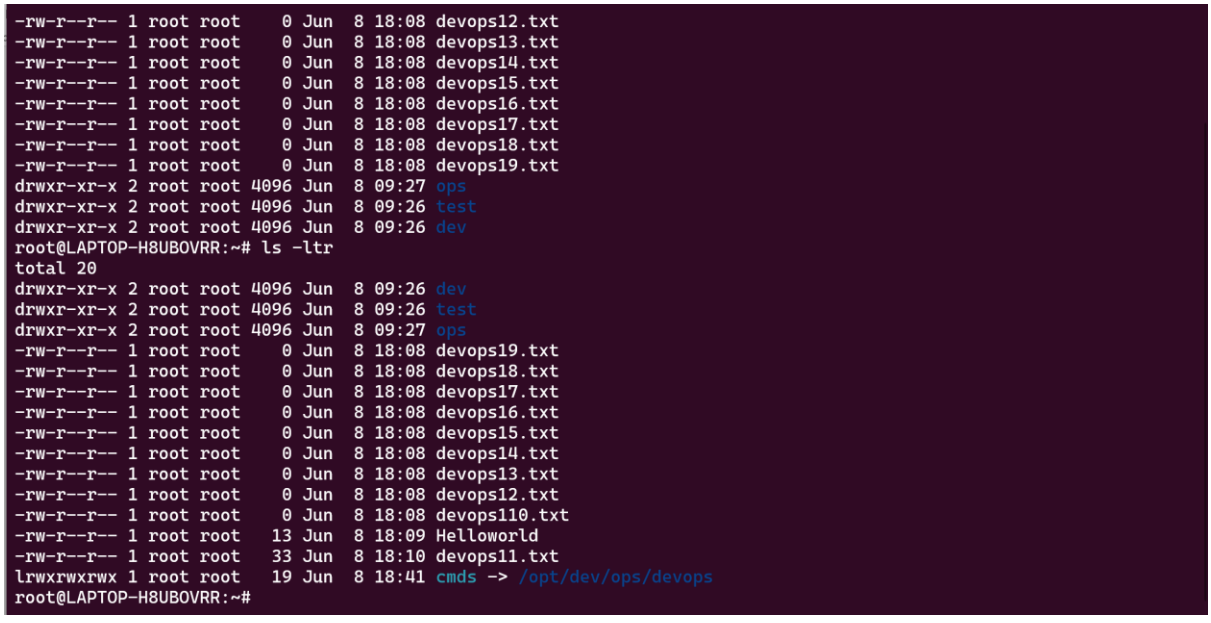


Fig. Timestamp file at Bottom

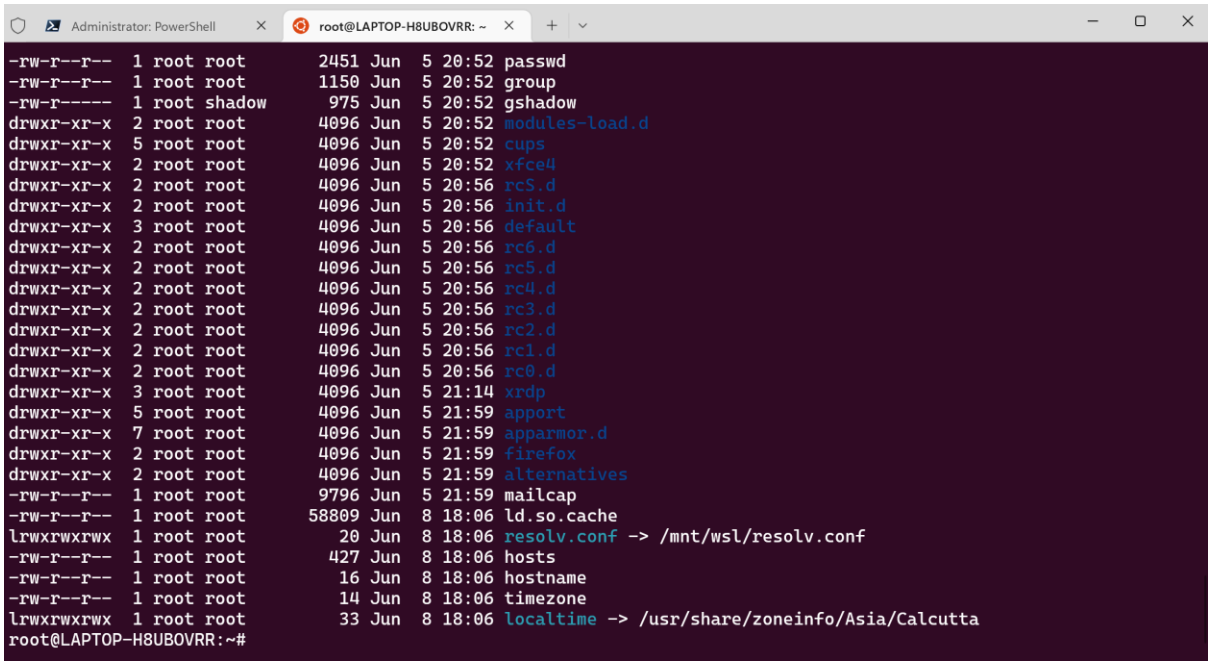


Fig: ls -ltr /etc/ Timestamp file at Bottom

Changing the Host Name:

Fig. Host Name Changes

Quiz:

Good job!

Question 1:

What command is used to find file type

- type <filepath>
- check <filepath>
- **file <filepath>**
- filetype <filepath>

Good job!

Question 2:

In -s <orignalFilePath> <LinkFilePath> command is used to create soft link.

- **True**
- False

Filters & IO redirectionscommand

Grep:

- Grep command is used to find texts from any text input

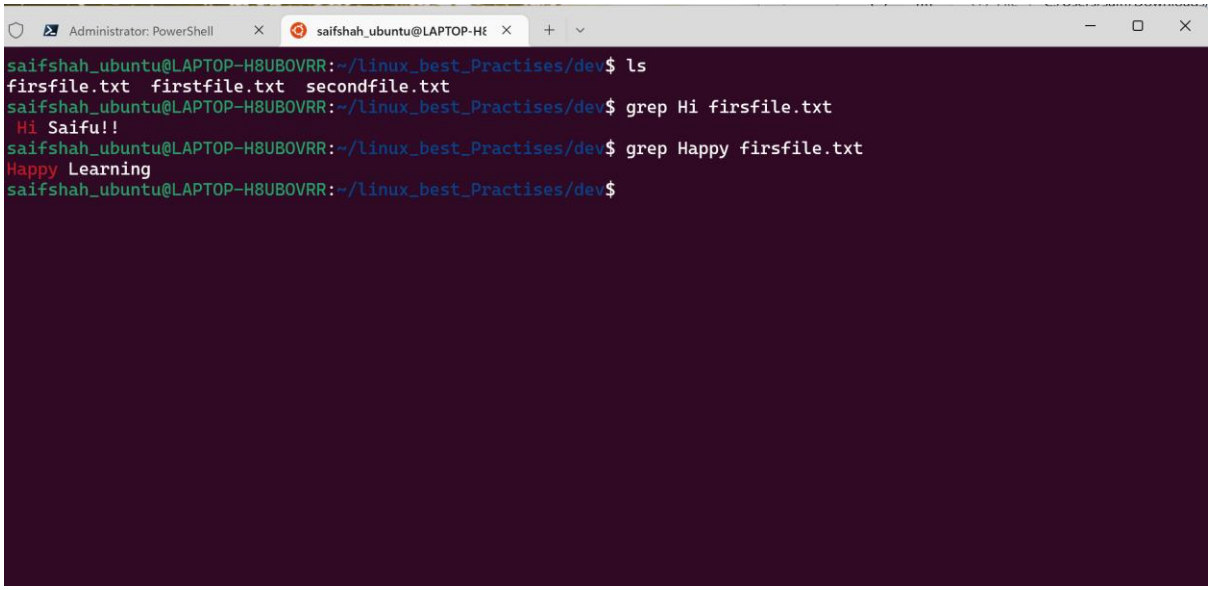


Fig. grep commands for finding text from any text input

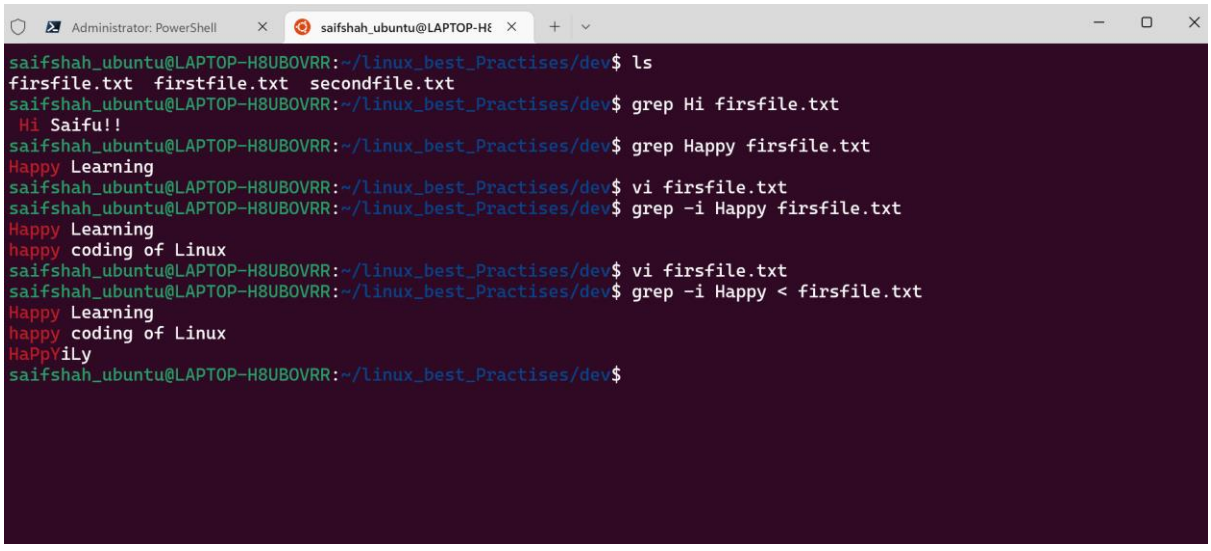


Fig. To ignores case sensitive in grep commands

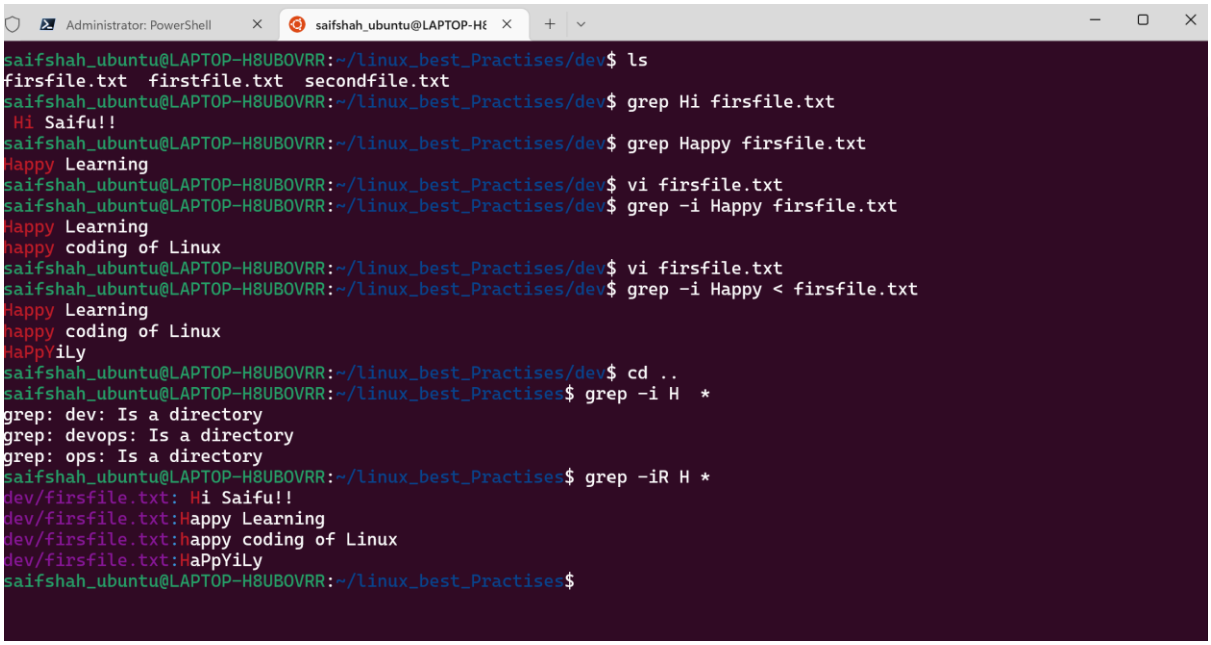


Fig. grep searching in the directory **-iR** commands

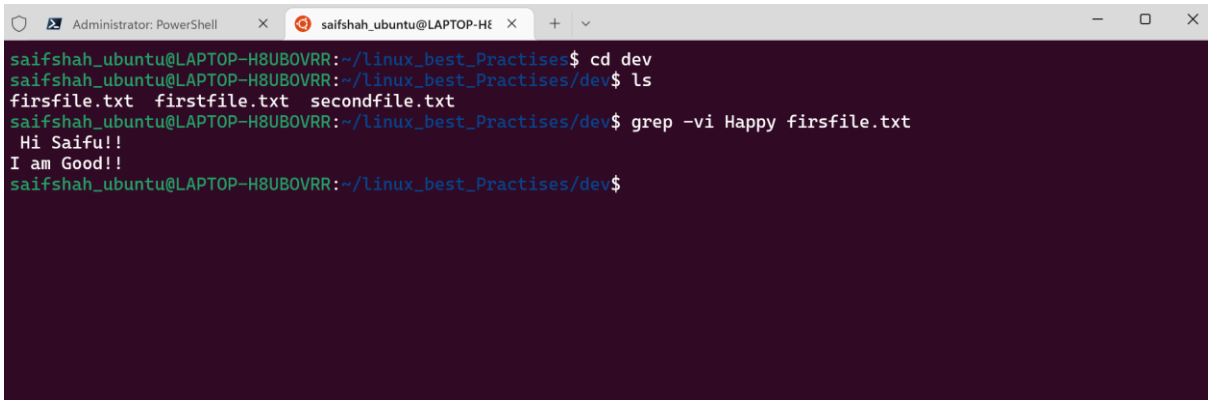


Fig. Reverse Searching with **-vi** commands

Reading the Content:

saifshah_ubuntu@LAPTOPH8UBOVRR:~/linux_best_Practises/dev$ less firsfile.txt

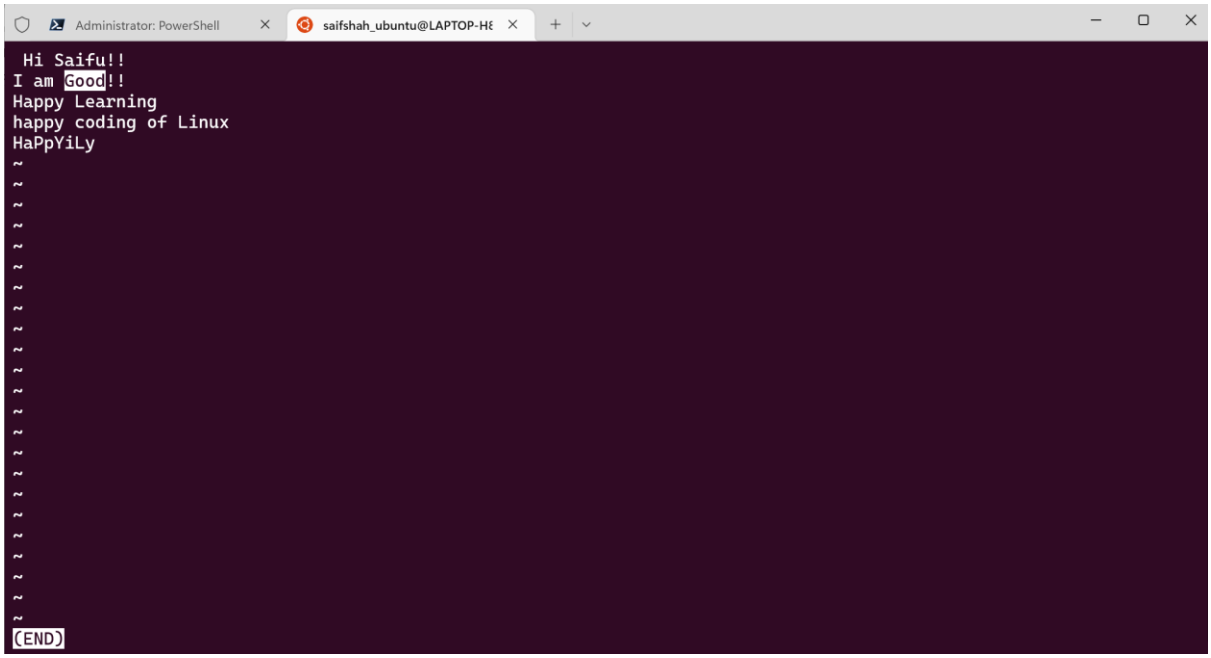


Fig. Reading the Content and Searching for (Good)

saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/dev$ more firsfile.txt

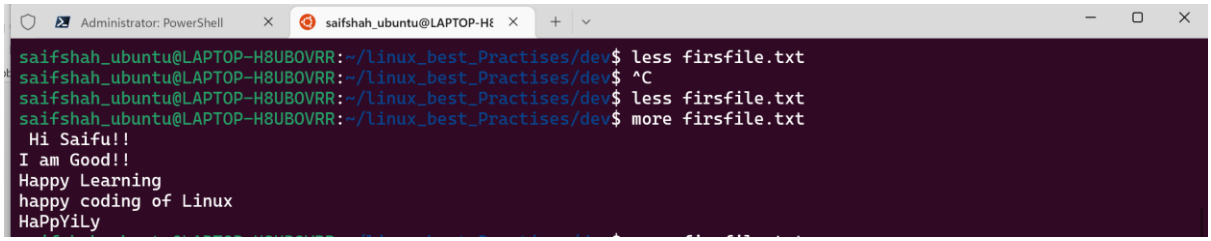


Fig. Reading the content with more command

head

- **Head:** It’s Used to display top 10 lines of files

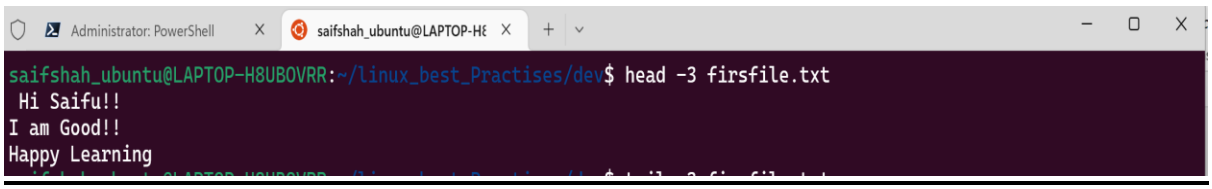


Fig. head to display first 3 lines from files

- **Tail**: It’s Used to display last 10 lines of files

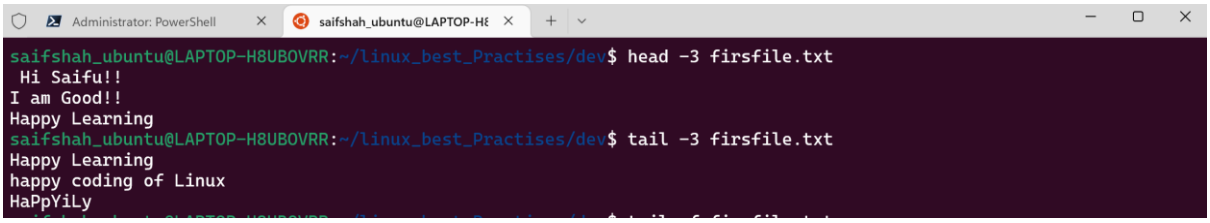


Fig. tail to display last 3 lines from files

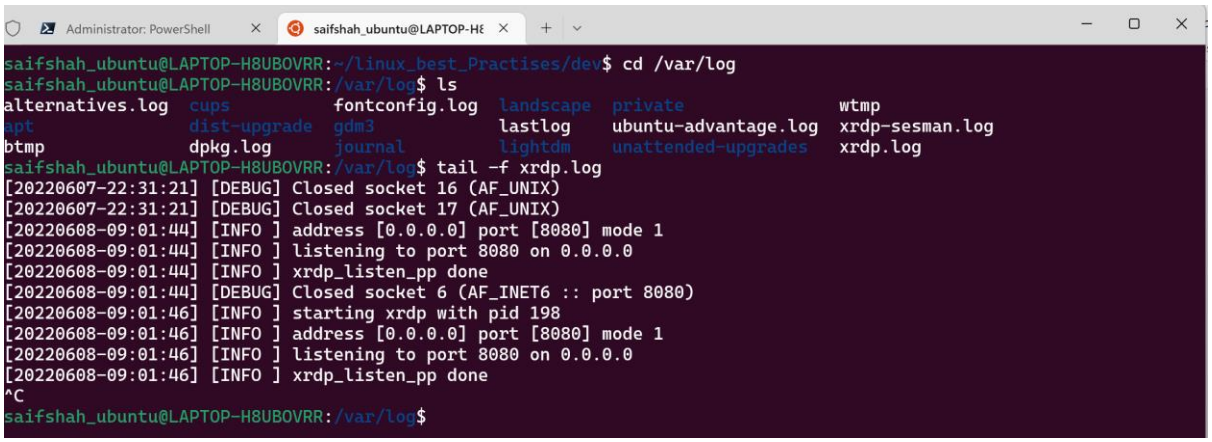


Fig. tail to check **-f** for newly added code in logs file

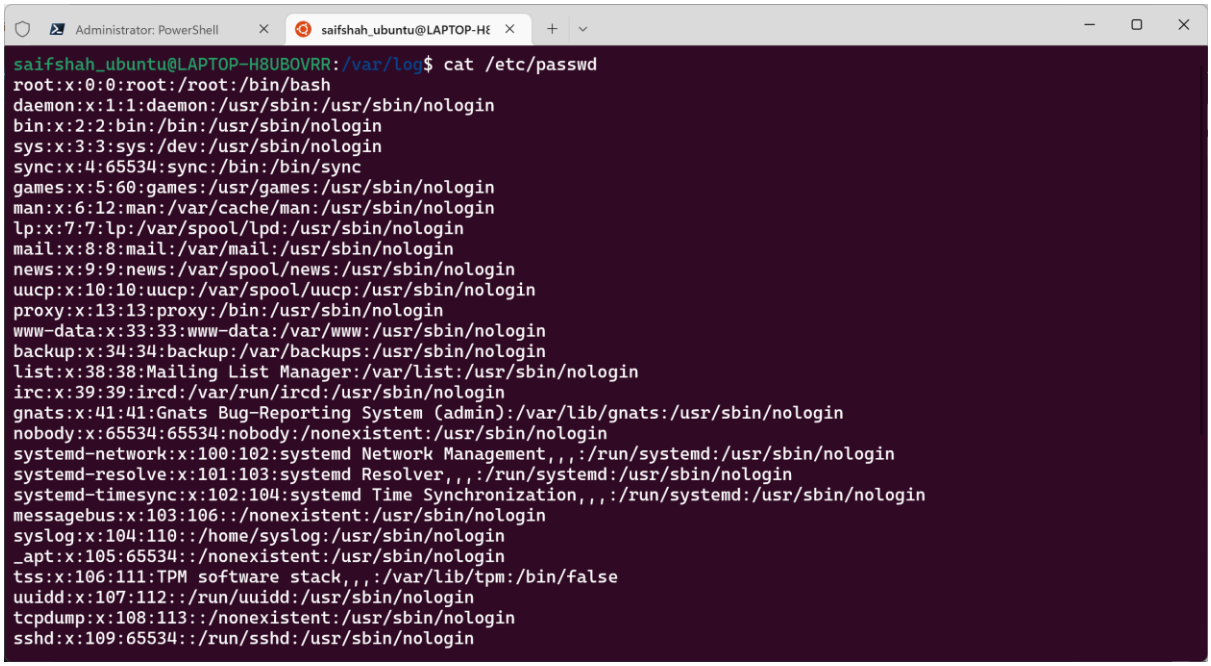


Fig. Shows full information of passwd file

saifshah_ubuntu@LAPTOP-H8UBOVRR:/var/log$ cut -d: -f1 /etc/passwd

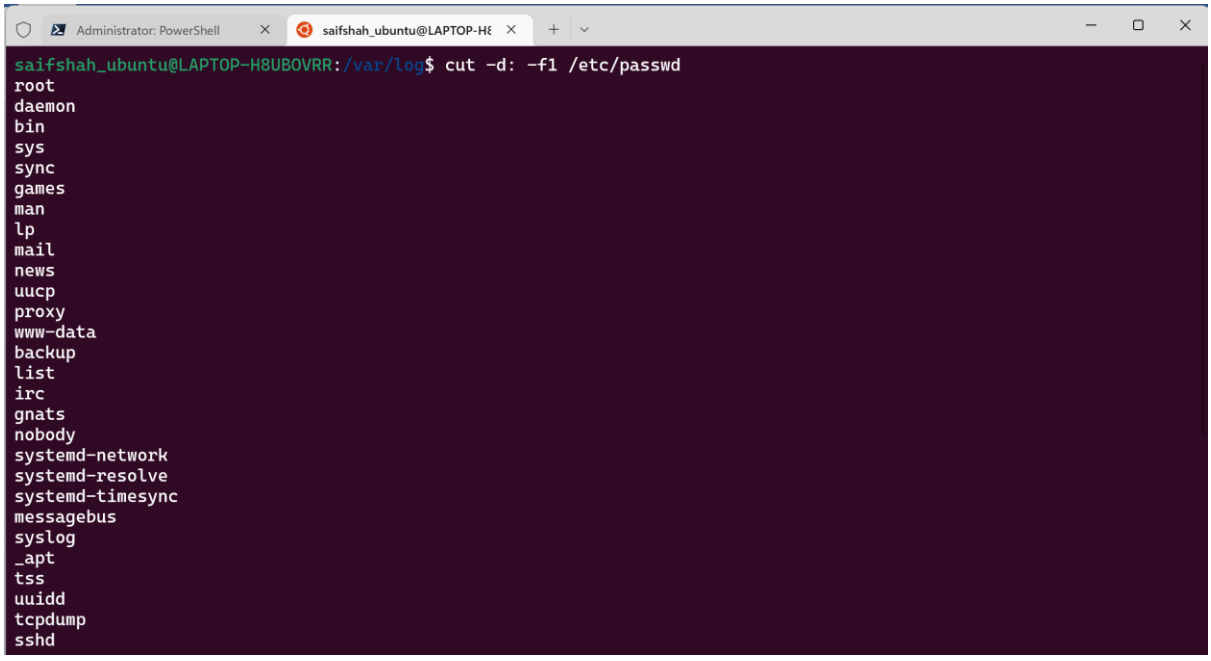


Fig. Shows first column of passwd file

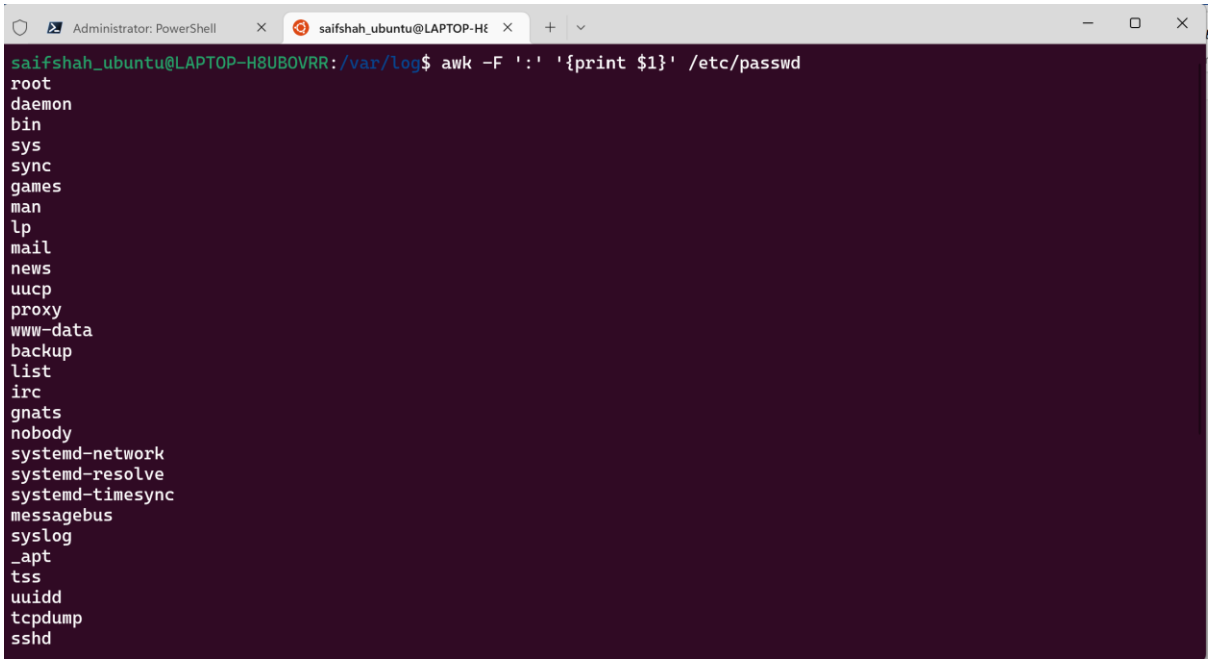


Fig. awk is useful when there is no separator in file

Searching and Replacing File:

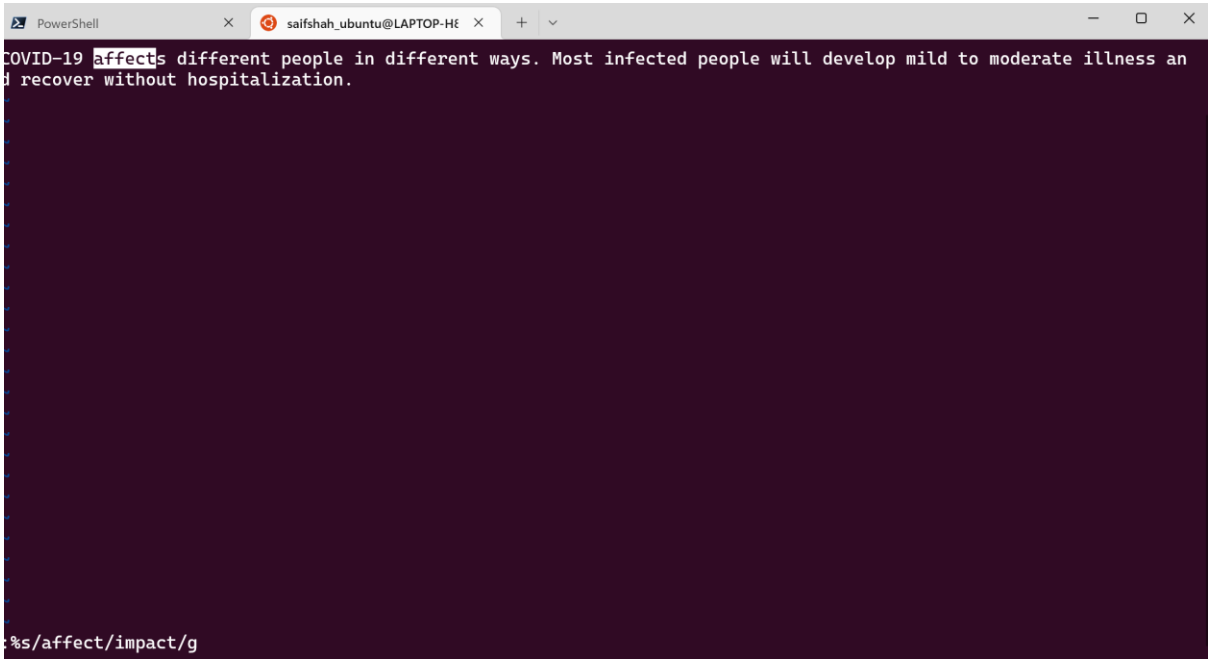


Fig. Searching the file

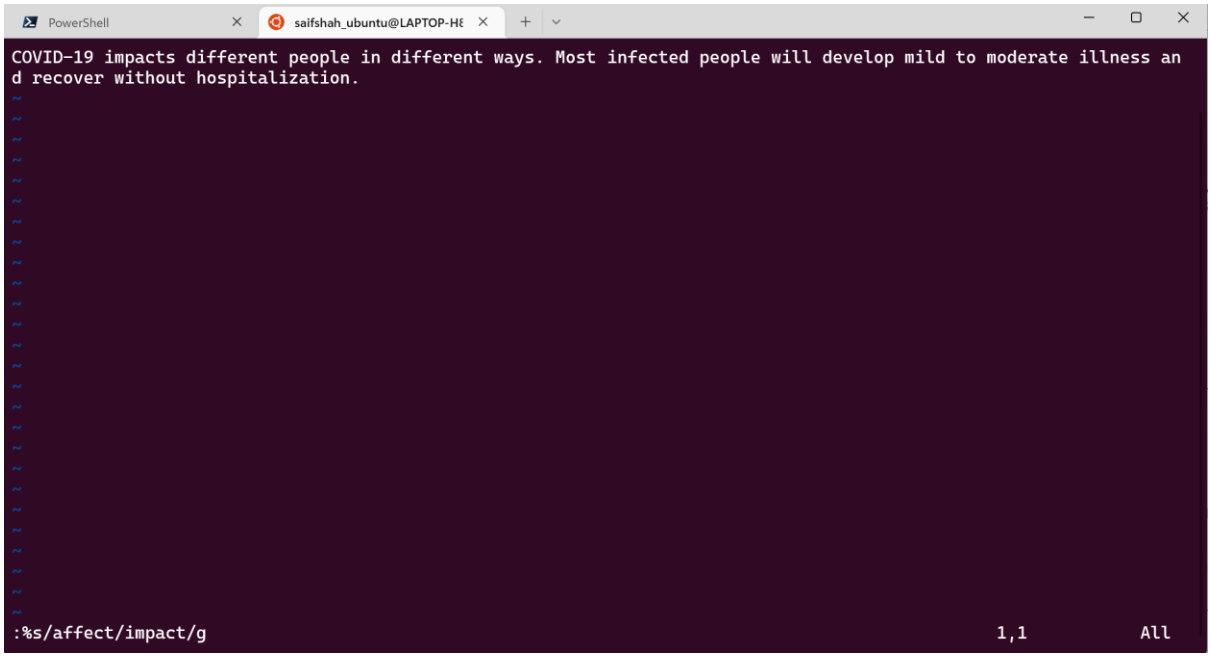


Fig. Replacing with enter

- **Sed**: It's stands for stream editor, which is used to search a word in the file & replace it with word required to be in the output

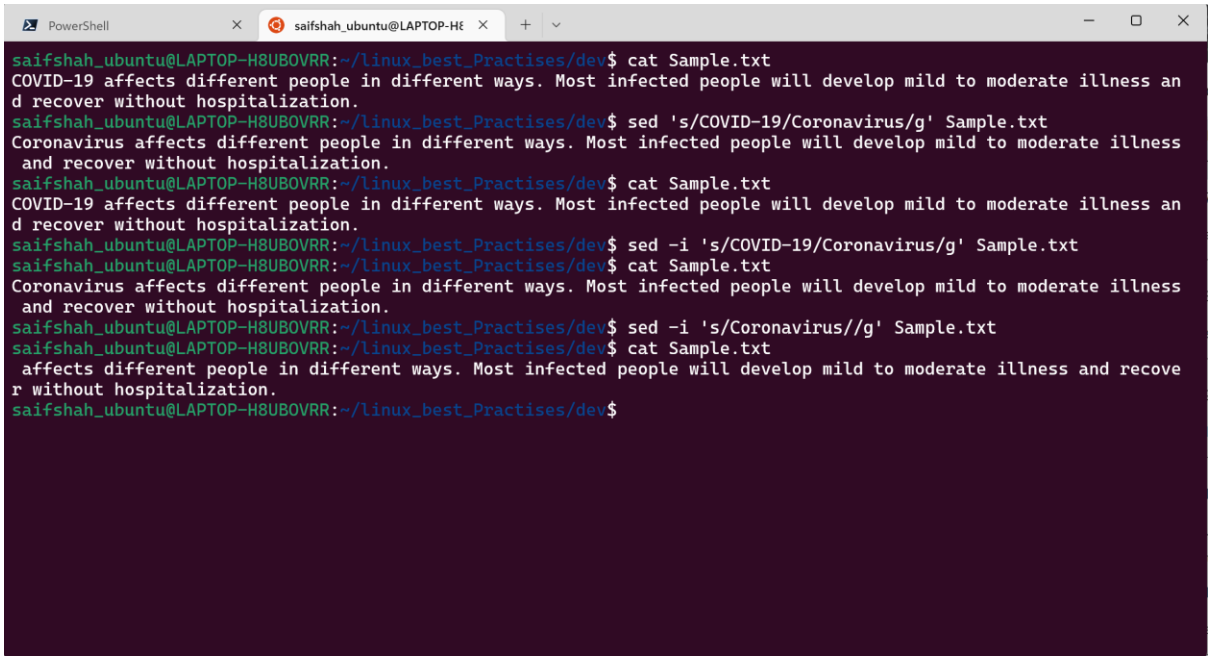


Fig. Stream Editor with all operations

Quiz:

Good job!

Question 1:

`grep <text> * -R` command will search for the text in all the file, in all the subdirectories.

- (●) **true**
- ( ) false

Good job!

Question 2:

What is the command to find last 20 lines of the file

- ( ) head 20 <filepath>
- ( ) tail <filepath>
- ( ) tail twenty <filepath>
- (●) **tail 20 <filepath>**

IO redirectionscommand

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

⇨ Create a File name called devops tools with content

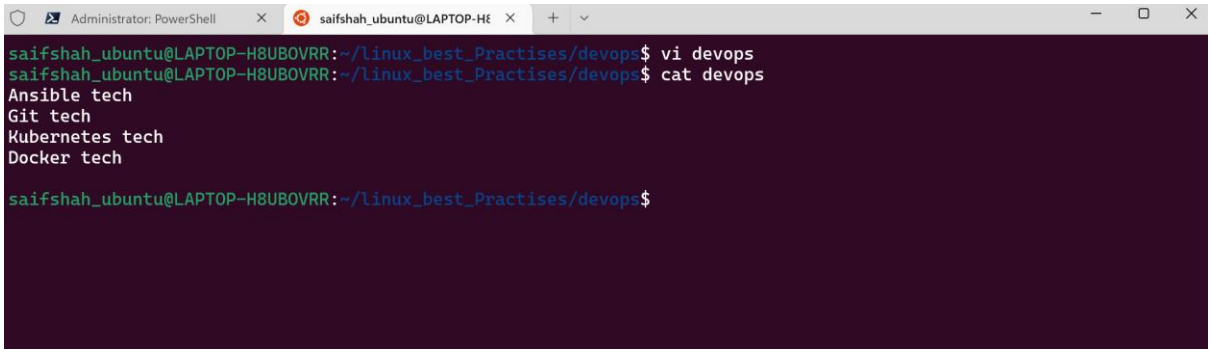


Fig. Creating devops tool with content

⇨ Search for text “tech” replace it with “tools” and redirect the output into a new file

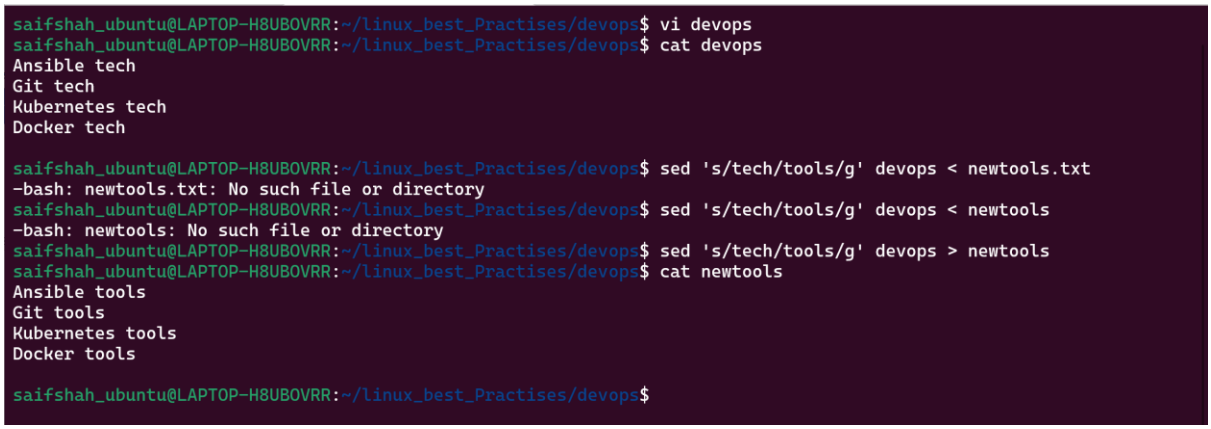


Fig. Searching and replacing with IO redirection

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

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

```
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$ tail -4 /etc/passwd
saned:x:120:129::/var/lib/saned:/usr/sbin/nologin
colord:x:121:130:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
gdm:x:122:131:Gnome Display Manager:/var/lib/gdm3:/bin/false
lightdm:x:123:132:Light Display Manager:/var/lib/lightdm:/bin/false
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$ tail -4 /etc/passwd >> newtools
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$ cat newtools
Ansible tools
Git tools
Kubernetes tools
Docker tools

saned:x:120:129::/var/lib/saned:/usr/sbin/nologin
colord:x:121:130:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
gdm:x:122:131:Gnome Display Manager:/var/lib/gdm3:/bin/false
lightdm:x:123:132:Light Display Manager:/var/lib/lightdm:/bin/false
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$
```

Fig. Appending another output in same file

```
Administrator: PowerShell | saifshah_ubuntu@LAPTOP-H8 | Settings
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$ echo "############################" >> Text2.txt
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$ uptime >>Text2.txt
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$ echo "############################" >> Text2.txt
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$  df -h >> Text2.txt
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$ cat Text2.txt
 Hello
############################
Thu Jun  9 12:28:04 IST 2022
############################
 12:28:22 up 34 min,  0 users,  load average: 0.00, 0.00, 0.00
############################
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdc        251G  2.7G  236G   2% /
tools           147G   89G   59G  61% /init
none            1.9G     0  1.9G   0% /dev
tmpfs           1.9G     0  1.9G   0% /sys/fs/cgroup
none            1.9G  4.0K  1.9G   1% /run
none            1.9G     0  1.9G   0% /run/lock
none            1.9G     0  1.9G   0% /run/shm
none            1.9G     0  1.9G   0% /run/user
drivers         147G   89G   59G  61% /usr/lib/wsl/drivers
lib             147G   89G   59G  61% /usr/lib/wsl/lib
none            1.9G  4.0K  1.9G   1% /mnt/wsl
none            1.9G  108K  1.9G   1% /mnt/wslg
none            1.9G   76K  1.9G   1% /mnt/wslg/versions.txt
none            1.9G   76K  1.9G   1% /mnt/wslg/doc
drvfsa          147G   89G   59G  61% /mnt/c
drvfsa          330G   62G  268G  19% /mnt/d
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$
```

Fig. Basic Commands of Linux redirected

⇨ Redirecting only error to a file “2>>”.

```
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$ uptimer 2>> Text3.txt
saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$ cat Text3.txt

Command 'uptimer' not found, did you mean:

  command 'uptime' from deb procps (2:3.3.16-1ubuntu2.3)
  command 'uptimed' from deb uptimed (1:0.4.2-1)

Try: sudo apt install <deb name>
```

Fig. redirecting only errors

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

saifshah_ubuntu@LAPTOP-H8UBOVRR:~/linux_best_Practises/devops$ freee -m &>> Text3.txt

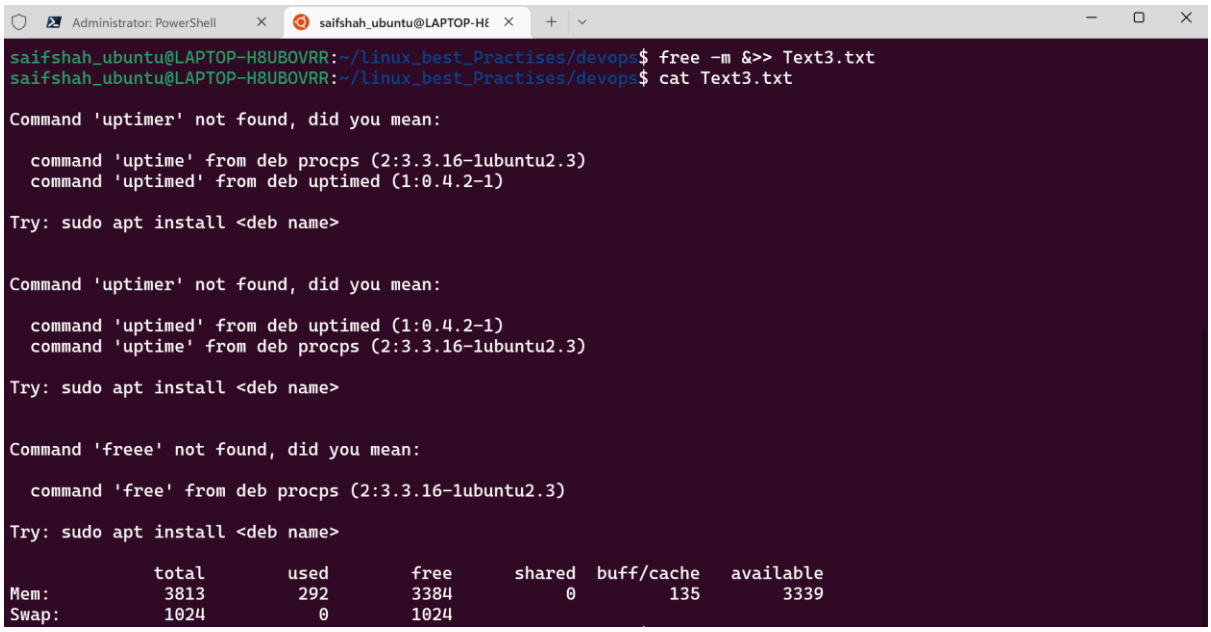


Fig. redirecting all output to the file

Piping:

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

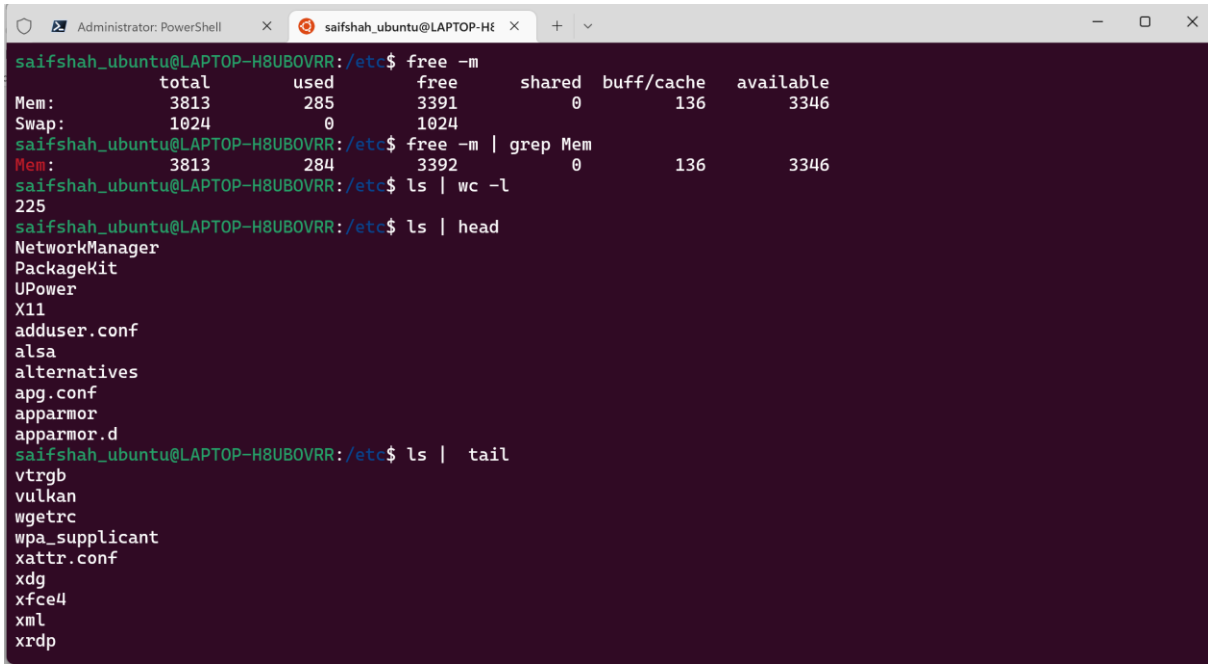


Fig. Piping with all commands

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

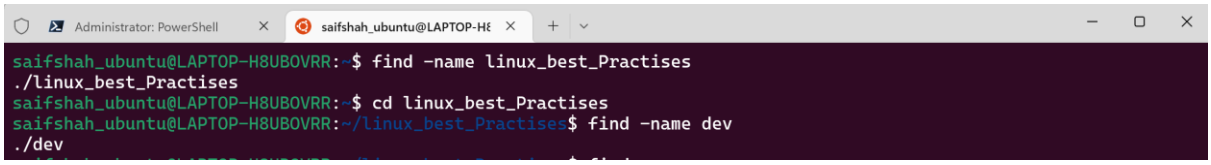


Fig. find command

Options that can be used with find command:

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

Quiz:

Good job!
2>> is to append standard ERROR not Output

Question 4:

2>> is to append standard Output to a file.

- ○ True
- ◉ False

Good job!

Question 1:

Output redirection symbol is ___

- [x] >
- [ ] <
- [ ] =>
- [ ] <=

Good job!

Question 2:

Input Redirection symbol is ___

- [ ] >
- [x] <
- [ ] <=
- [ ] =>

Users and Groups

Users

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 UserID number (the UID)
- Users name and UID are stored in **/etc/passwd**
- Users 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 without permission.

Type of User

TYPE	EXAMPLE	USER ID(ID)	GROUP ID(ID)	HOME DIR	SHELL
ROOT	root	0	0	/root	/bin/bash
REGULAR	Saifshah, ubuntu	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

In Linux there are three types of users.

1. Super user or root user

Super user or root user is the most powerful user. He is the administrator user.

2. System User

System users are the users created by the software or applications. For Example, if we install Apache, it will create a user apache. These kinds of users are also known as system users.

3. Normal User

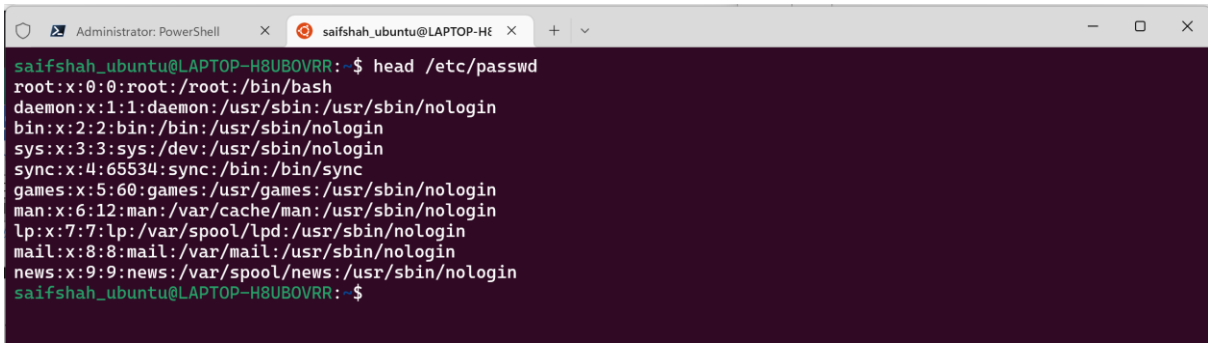
Normal users are the users created by the root user. They are normal users like saif, Sandeep, Amar, etc..

<u>Whenever a user is created in Linux things created by default:</u>

- A home directory is created(home/username)
- A mail box is created(var/spool/mail)
- Unique UID & GID are given to user

Passwd file

1. /etc/passwd



The above fields are

- root = name
- x = link to the password file i.e. /etc/shadow
- **0** or **1** = UID (user id)
- **0** or **1** = GID (group id)
- **root** or **daemon** = comments (brief information about user)
- **/root** or **/usr** = home directory of the user
- **/bin/bash** or **/sbin/nologin** = shell

Group file

2. /etc/group

The file /etc/group stores group information. Each line in this file stores one group entry.

Group name, group password, GID, group member

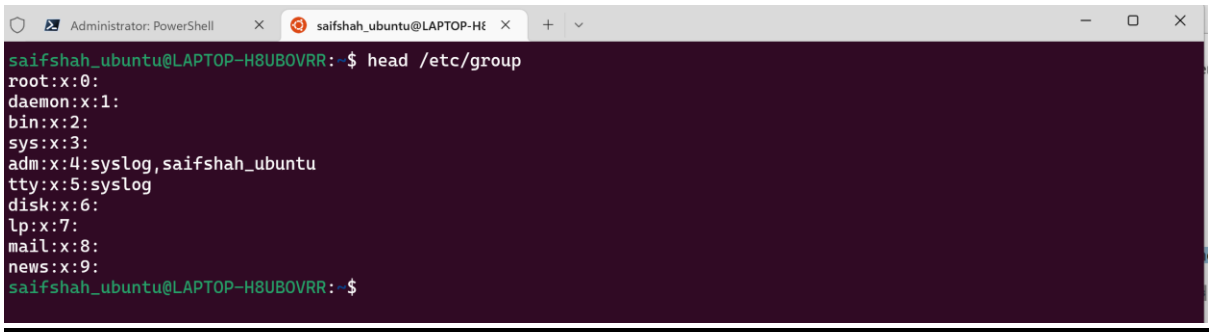


Fig. Group head

ADD USER, SET PASSWORD & SWITCH TO USER

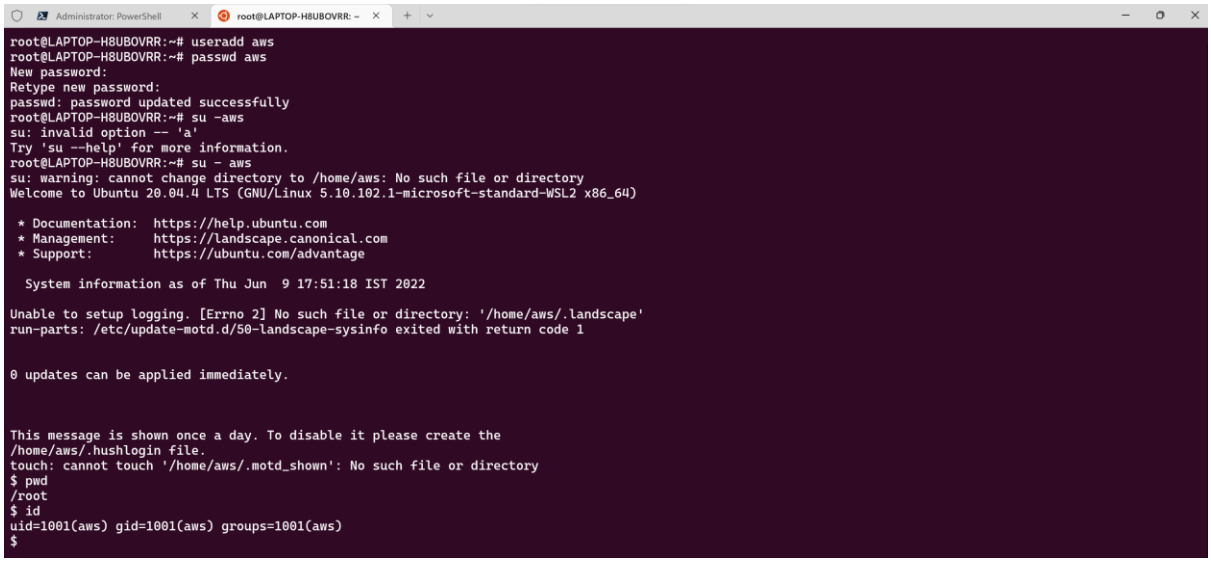


Fig. adding user, password setting and switch to user

ADD USER, GROUP & USER INTO GROUP

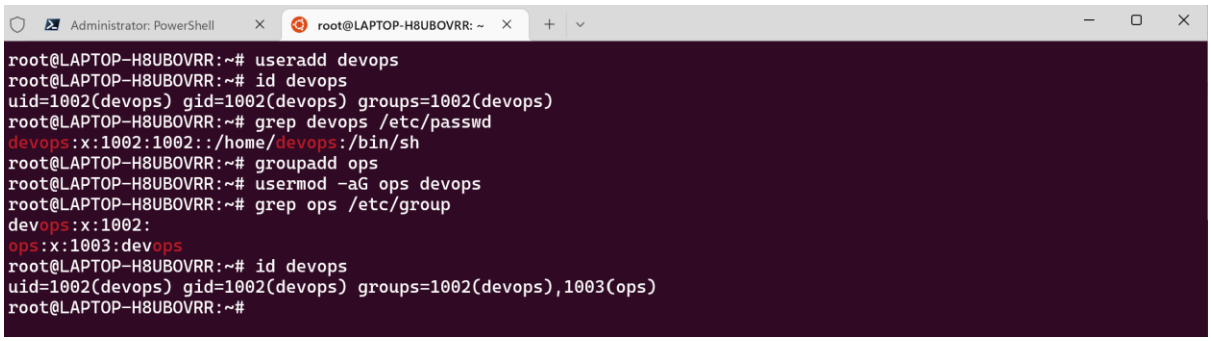


Fig. all commands check

DELETE USER and GROUP

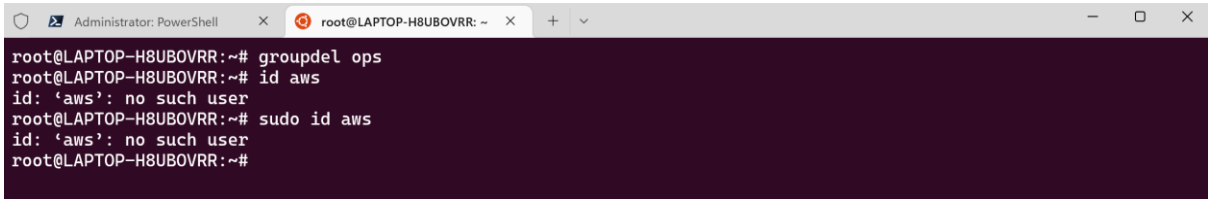
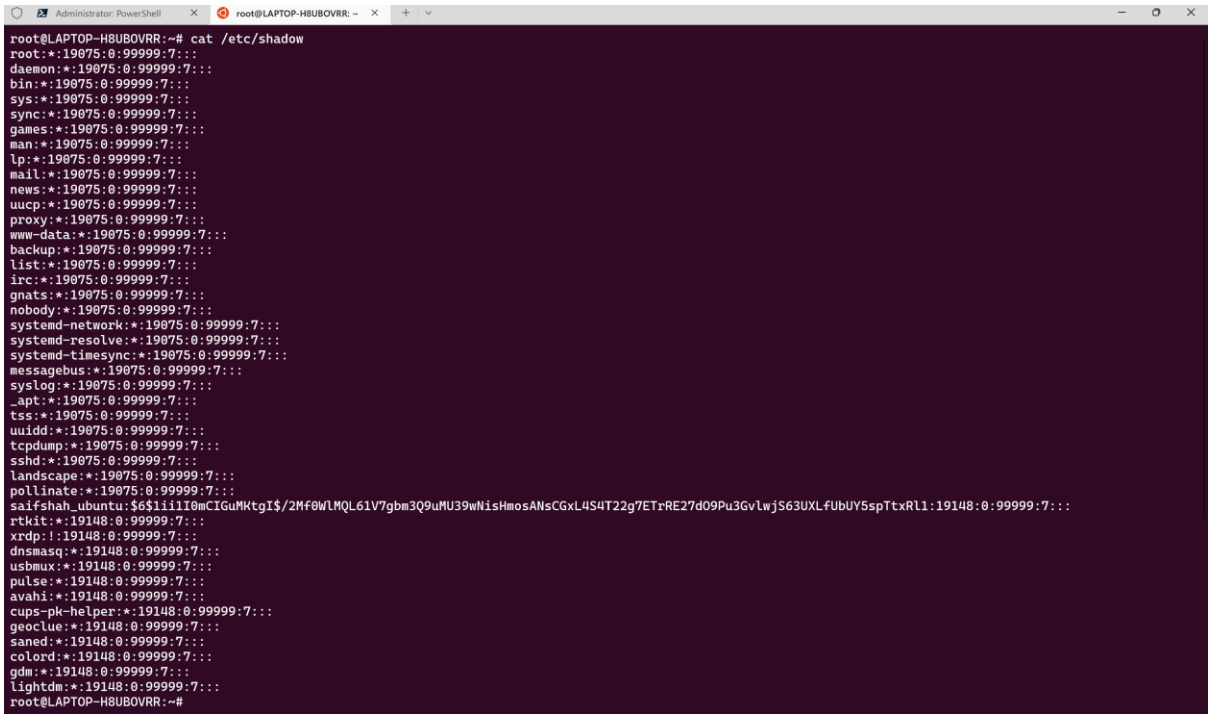


Fig. deleted users and groups

cat /etc/shadow



Users and Group cheat sheet

COMMANDS	DESCRIPTION
useradd	Creates user in RedHat
adduser	Creates user in ubuntu
id	Shows user info
groupadd	Creates group
usermod -G grpnam usrname	Adds user to group
passwd	set/reset password
userdel -r	removes user with home dir
groupdel	removes group
last	shows last login in system
who	who is logged into system
whoami	username
`lsof -u user`	`List files opened by user`

Fig. Users and Group cheat sheet

Quiz:

Good job!
User info file is /etc/passwd and Group info file is /etc/group

Question 1:

User info file is /etc/group and Group info file is /etc/passwd

- ○ True
- ◉ **False**

Good job!

Question 2:

lsof -u username will list files opened by user

- ◉ **True**
- ○ False

Good job!

Question 3:

How to reset password of a user ?

- ○ Any User can reset any users password by using `passwd username` command
- ○ Only user can reset its own password by using `passwd username` command
- ◉ **root user can reset any users password by using `passwd username` command, also user can reset its own password by running just `passwd` command.**

Good job!

Question 4:

How to switch between users in Linux cli?

- (•) **su - username**
- ( ) sudo - username
- ( ) sudo -i

Good job!

Question 5:

Users password is stored(encrypted) in /etc/shadows file.

- (•) **True**
- ( ) False

File Permissions

Viewing Permission from command – Line: ls -l

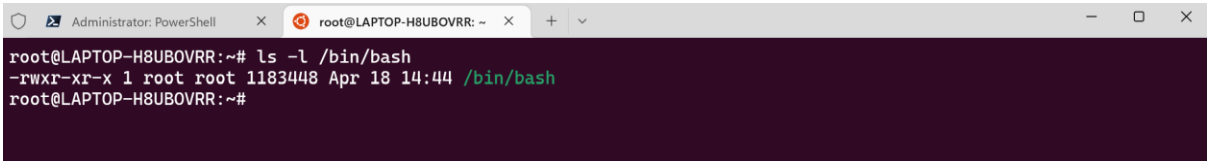


Fig. view permission

- Four Symbol are used when display permissions

- : Type of file directory, links, characters, etc

r :Permission to read a file or list's the directory content

w : Permission to write a file or creates & remove file from directory

x : Permission to execute a program and changing the directory and do a long listing to a directory

- : no permission in place of (r, w and x)

Changing the File Ownership

- Only root can change the 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...

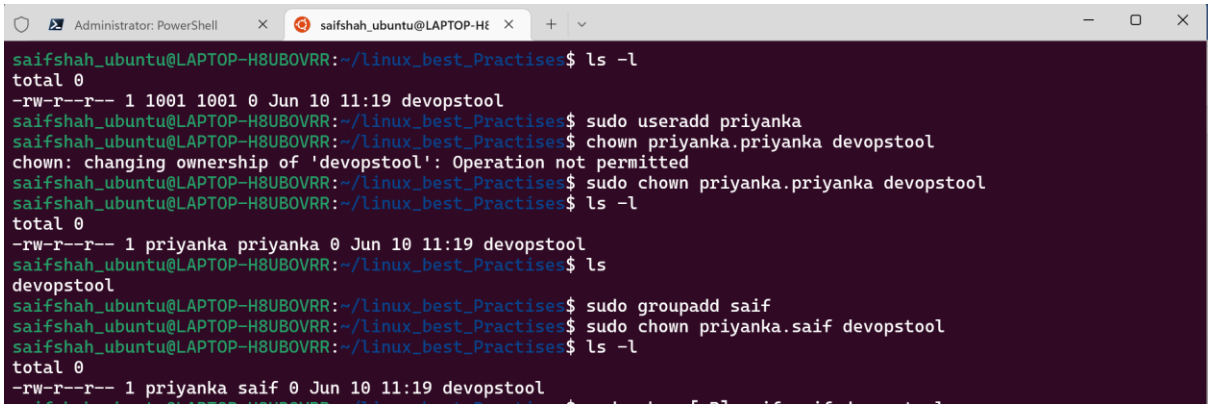


Fig. Changing File Permission

Changing Permissions - Symbolic Method

- To change access modes:
  - **chmod [-OPTION]... mode[,mode]** ***filel 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*

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**

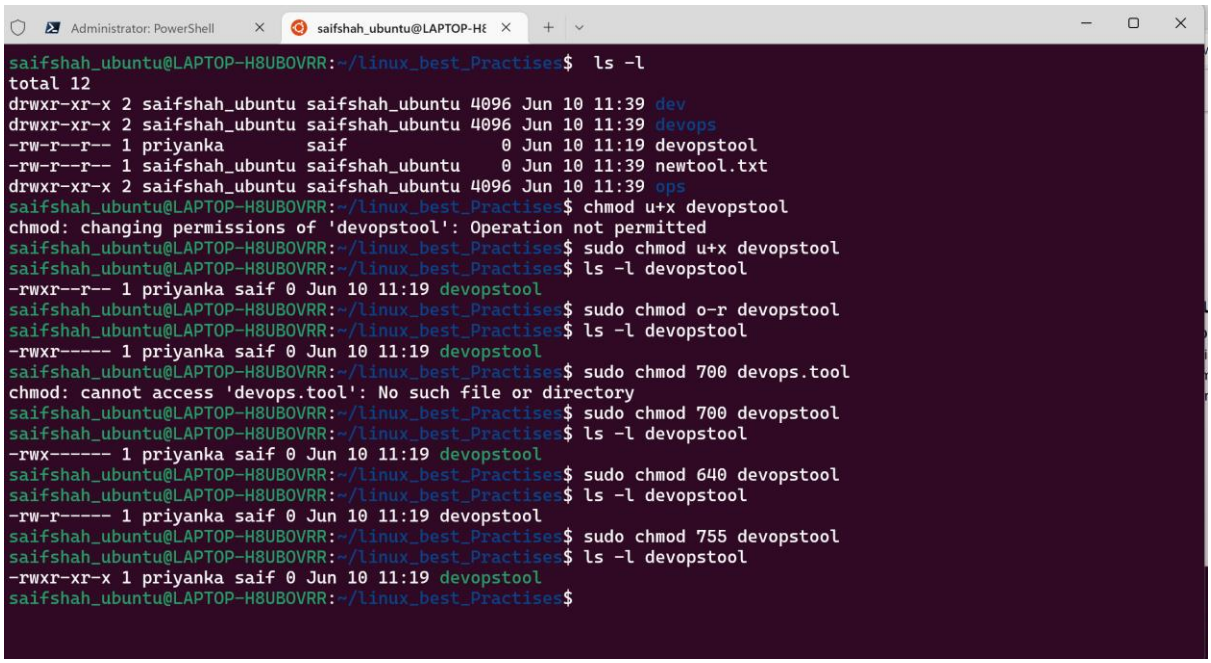


Fig. Changing permission Symbolic & Numeric Method

Sudo

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

If a user has already full sudoers privileges, it can become a root user anytime.

⇨ sudo -i changes from normal user to root user

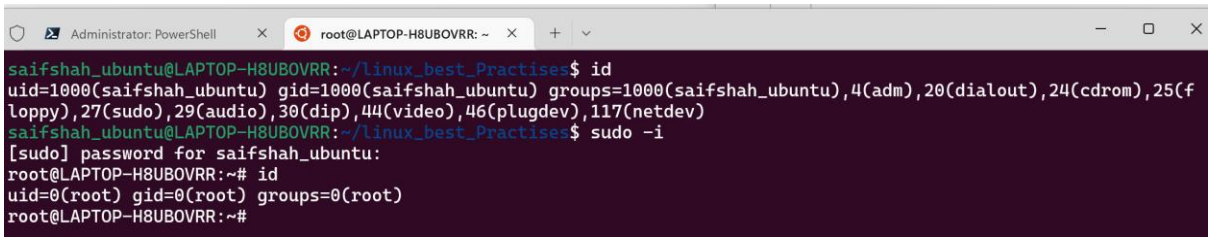
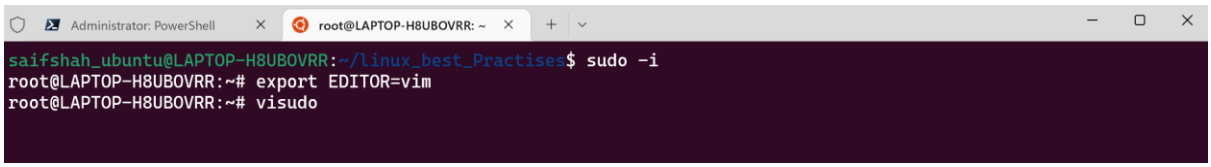


Fig. Sudo Commands to switch from normal to root user

Note: User saifshah was already a sudo user with full privilege

⇨ Adding a user pari in sudoers list



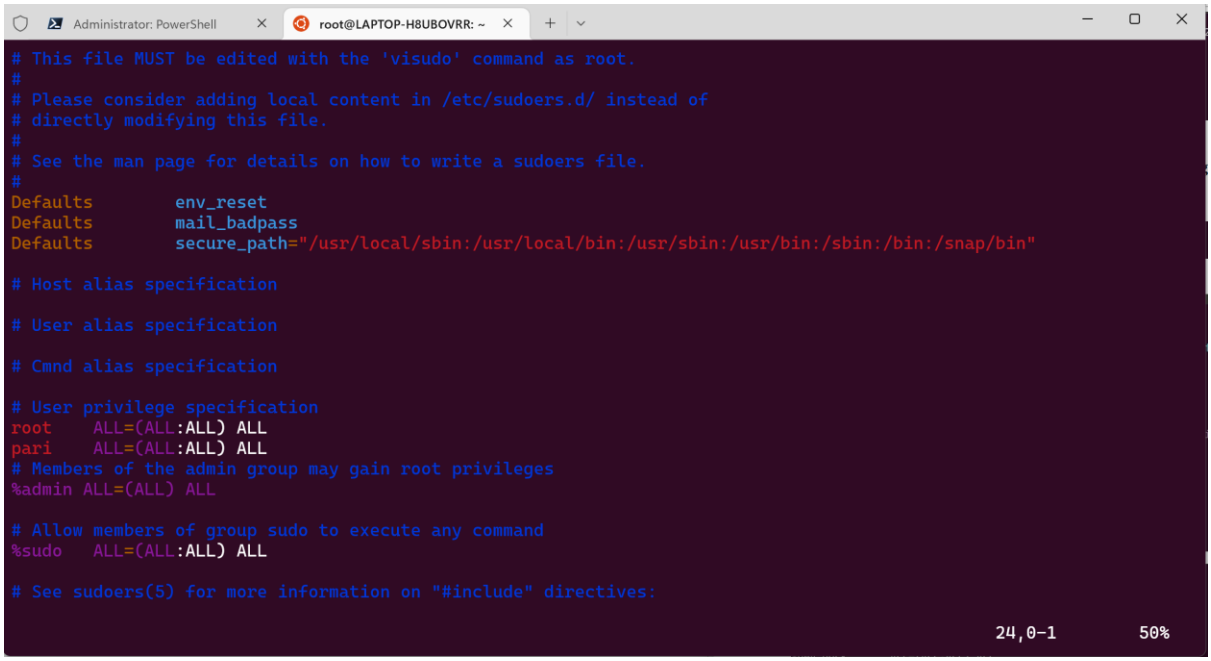


Fig. adding user in sudoers list

⇨ **Like a user a group can also be added into sudoers list.**

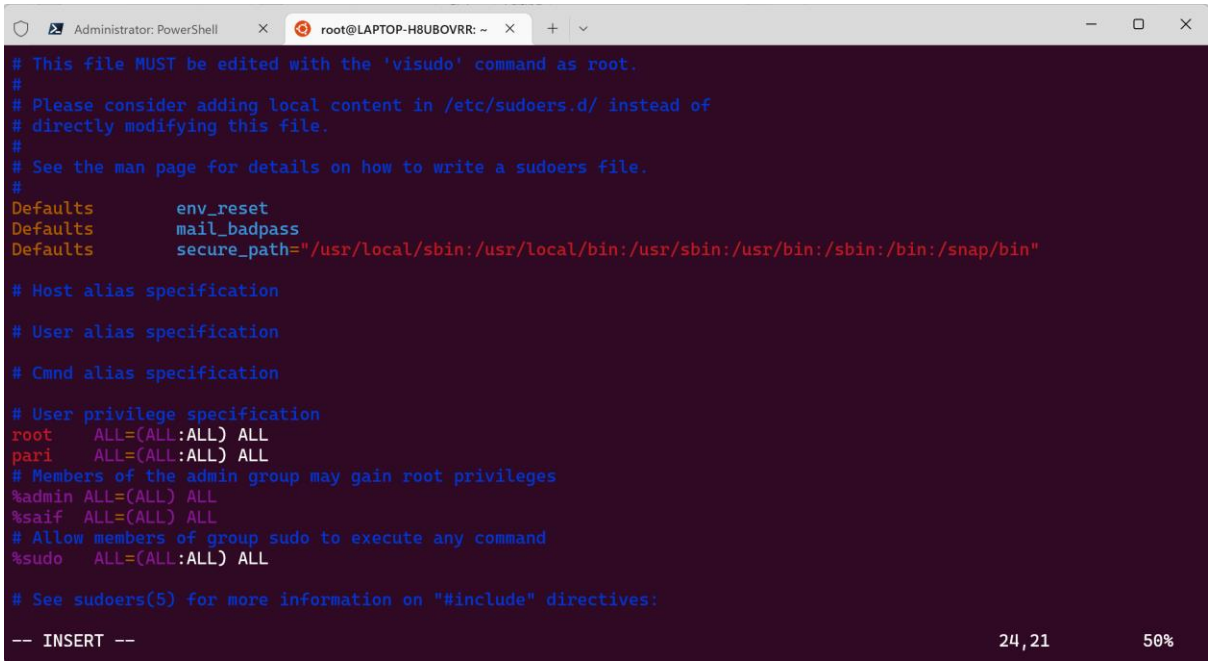


Fig. adding groups in sudoers list

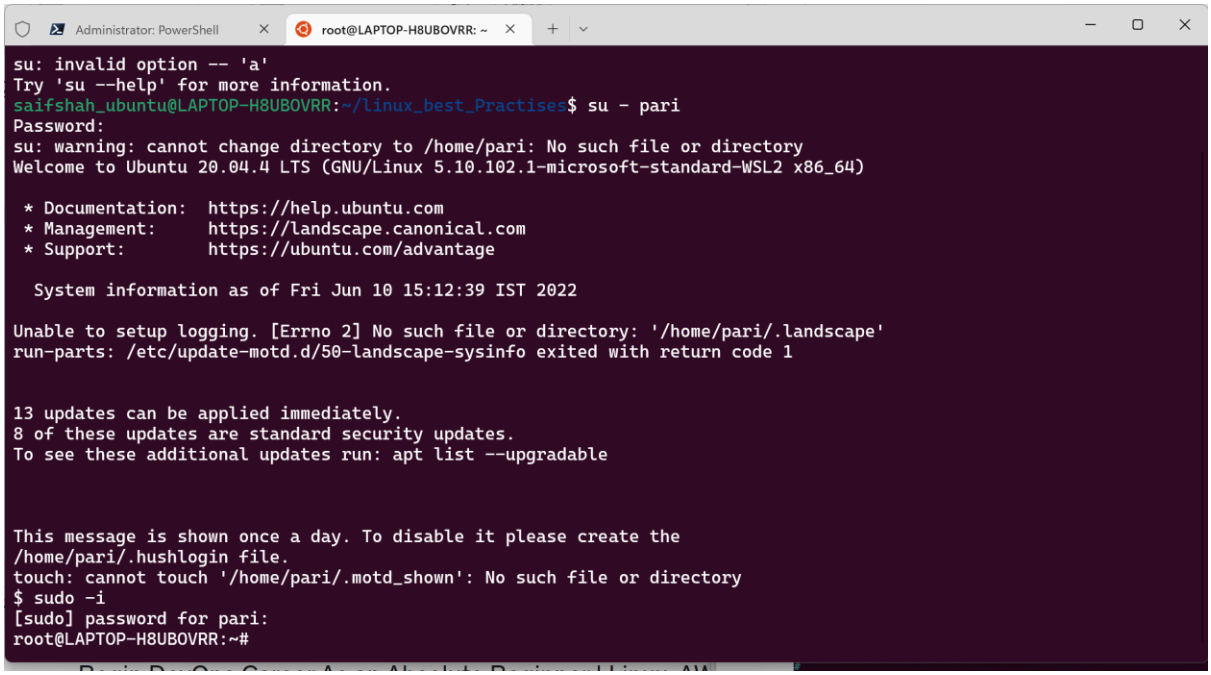


Fig. login with sudoers users

⇨ **Every time you enter sudo command it asks your own password. To turn that off use NOPASSWD in sudoers file.**

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
pari    ALL=(ALL:ALL) NOPASSWD: ALL
```

Fig. setting NOPASSWD in sudoers file

⇨ **Changing to any other users with “su- “command.**

⇨ **Become a root user from pari user login**

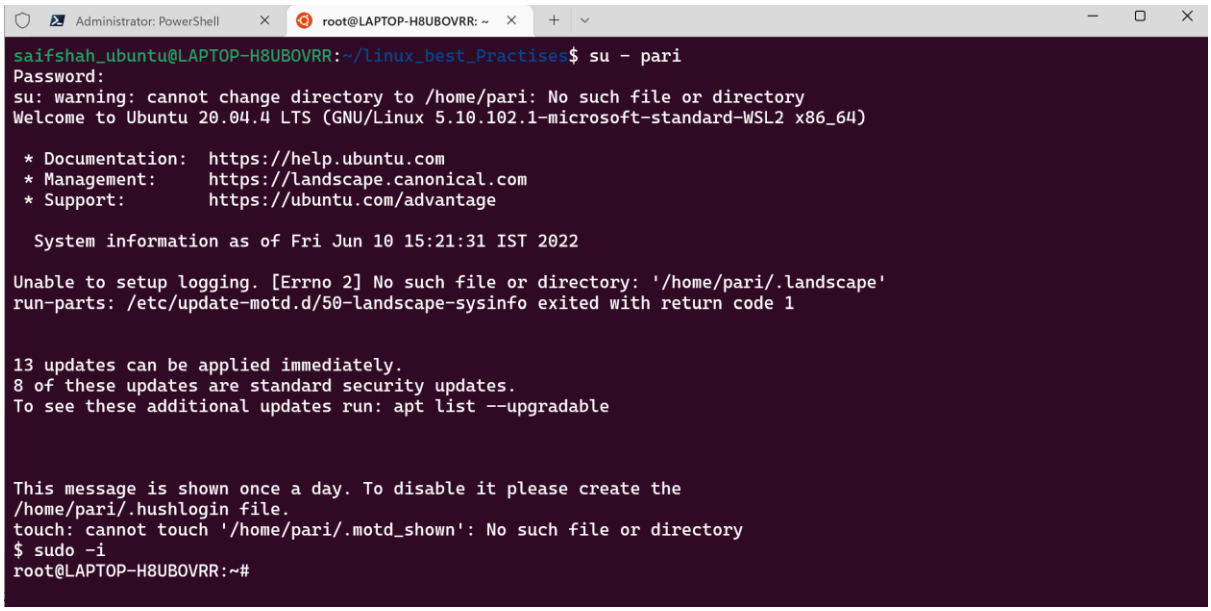
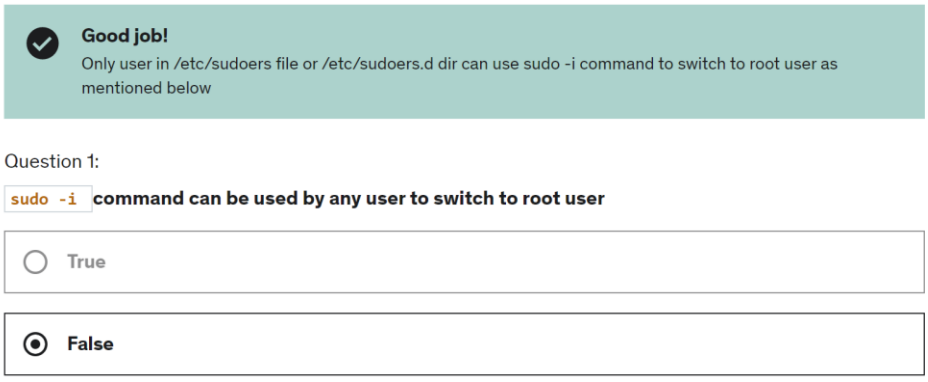


Fig. Changing to other users with su – command & become a root user from pari user login

Quiz:



Package Management

⇨ Download Packages from Internet

Install Tree

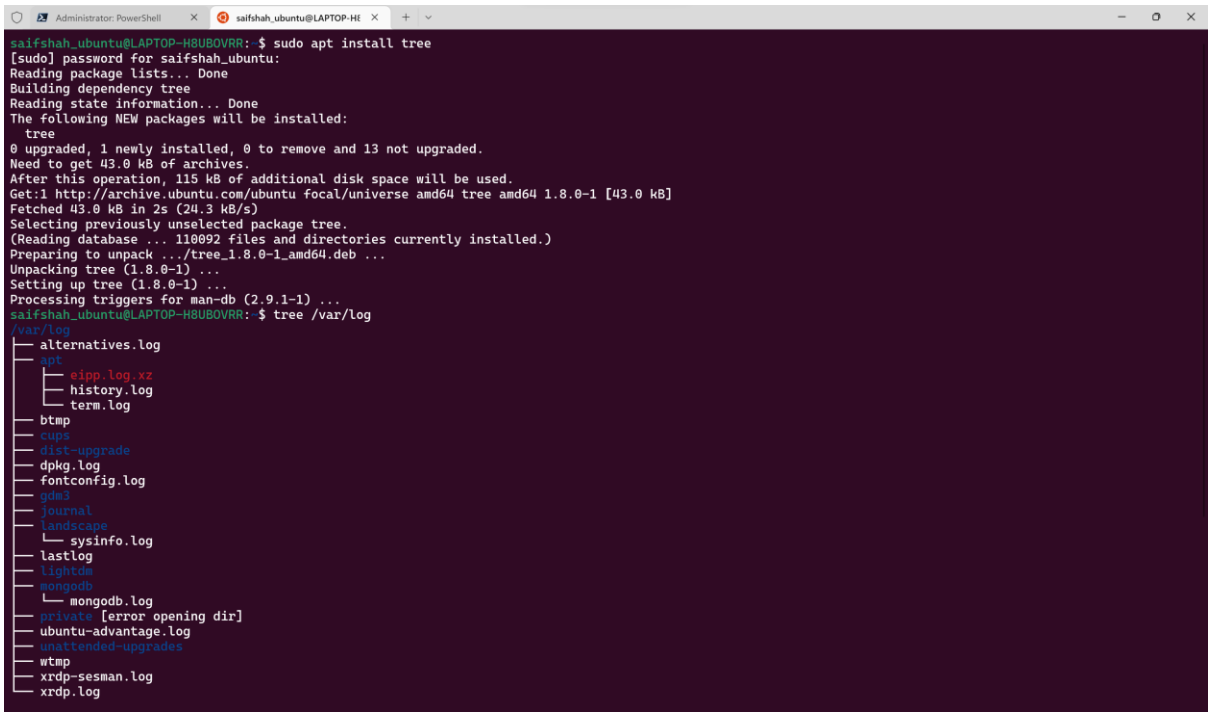


Fig. tree successful Installation

To Update all your packages

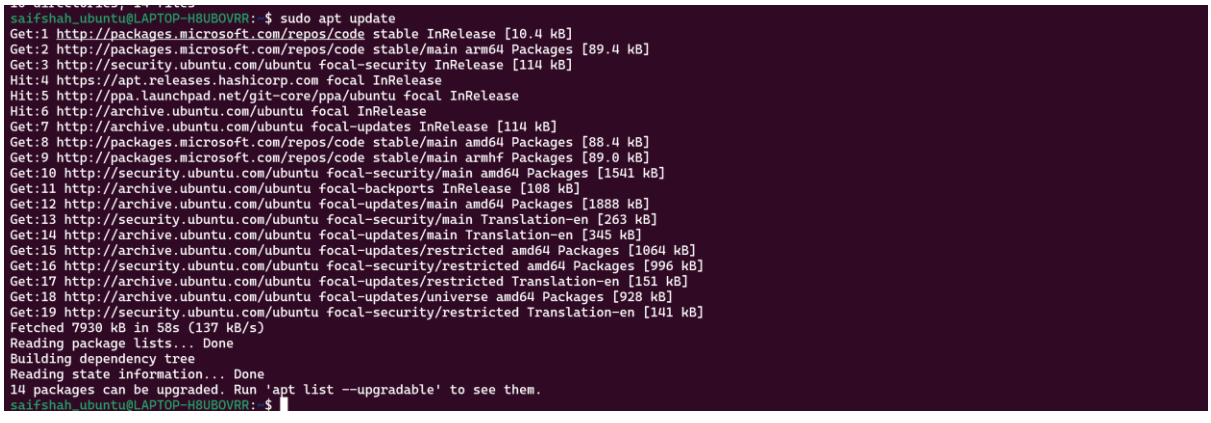
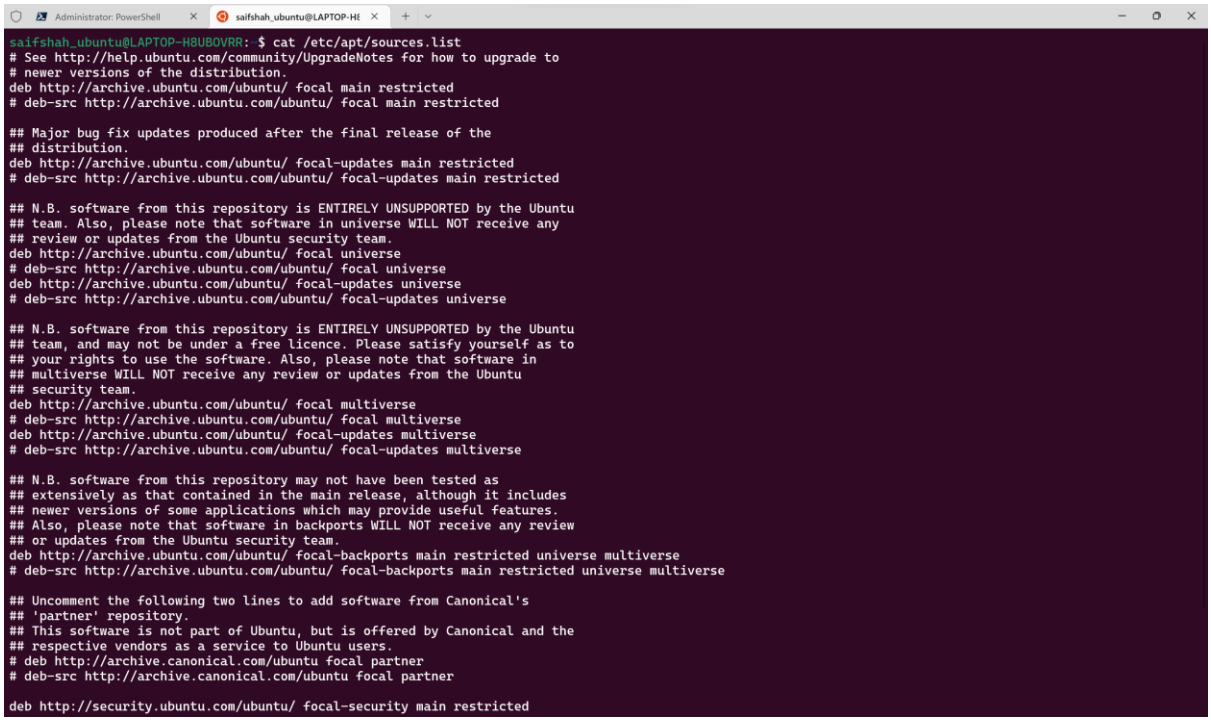
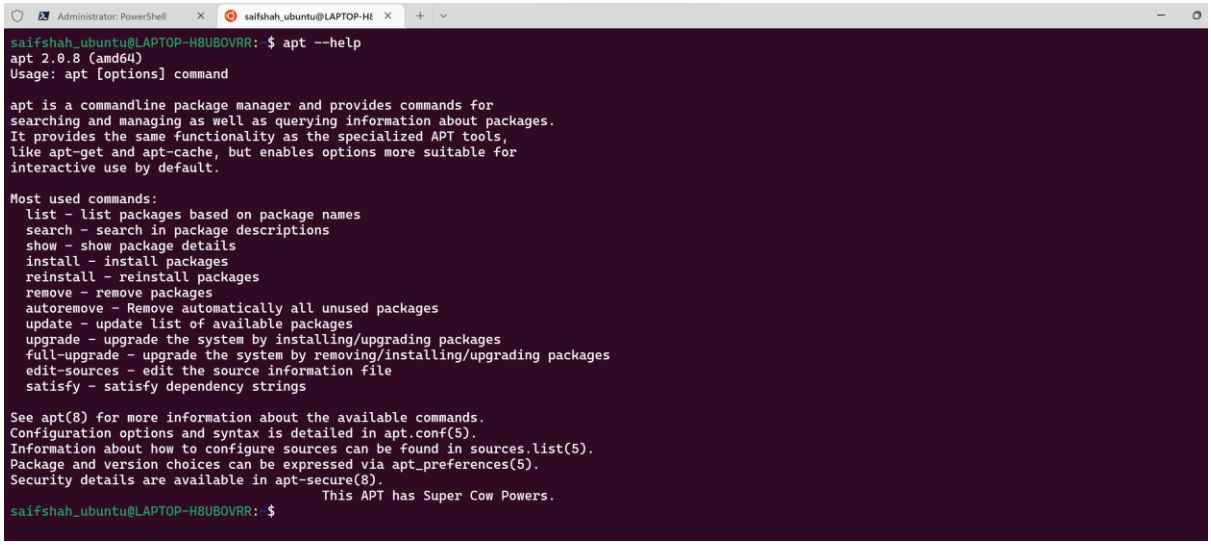


Fig. Update Packages

The **sources. list** file is a key factor in adding or upgrading applications to your Ubuntu installation. This is also used by your system for system updates. The file is basically the roadmap for your system to know where it may download programs for installation or upgrade.





To remove tree

root@LAPTOP-H8UBOVRR:~# apt remove tree

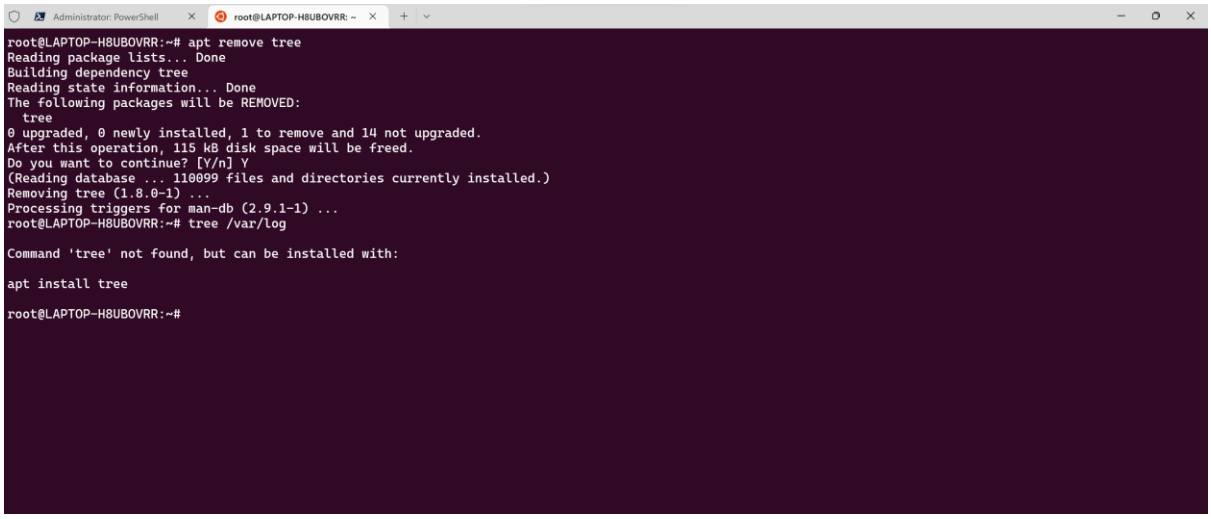


Fig. Removing packages

Ubuntu20 Commands		
apt commands cheatsheet	https://itsfoss.com/apt-command-guide/	
apt search PACKAGE	search from available repositories	
apt install PACKAGE -y	To Install Packages	
apt install apache2 -y	To Install apache2	
apt reinstall PACKAGE	To reinstall PACKAGE	
apt remove PACKAGE	To remove PACKAGE	
apt update	update all packages	
apt update PACKAGE	update only a package	

apt grouplist	List all available Group Packages	
apt groupinstall "GROUPNAME"	Installs all the packages in a group.	
apt repolist	List Enabled apt Repositories	
apt clean all	Clean apt Cache	
apt history	View History of apt	
apt show package name	Shows the information of package like version, size, source, repository etc	

Services:

$ sudo systemctl start apache2	*# Starts apache2 on ubuntu*
$ sudo systemctl stop apache2	*# Stops apache2 on ubuntu*
$ sudo systemctl restart apache2	*# Restart service*
$ sudo systemctl reload apache2	*# Reload conf*
$ sudo systemctl enable apache2	*# starts apache2 at boot time*
$ sudo systemctl disable apache2	*# stops apache2 at boot time*
$ sudo systemctl is-active apache2	*# Shows whether the service is active or not*
$ sudo systemctl is-enabled apache2	*# Shows whether the service is enabled or not*

Fig. Services in Ubuntu20

Process Related or Compression / Archives

```
$ tar cf home.tar  home  # Create tar named home.tar containing
home/ (ll tar examples)

$ tar xf
file.tar
file.tar                                  # Extract the files
from
$ tar czf
file.tar.gz                    fi         # Create a tar with
compression                    le         gzip
                               s
$ gzip file
to file.gz (untar gzip                    # Compress file and renames
file)                                     it
```

```
s
processes (many parameters to             # Display your currently
learn)                                    active
$ ps aux | grep
'telnet' telnet                           # Find all process id related
process                                   to
$ pmap
(kernel,user memory                       # Memory map of
etc)                                      process
$ top
(30                                       # Display all running
examples)                                 processes
$ kill pid
pid id (types of                          # Kill process with
signals)                                  mentioned
$ killall
proc                                      # Kill all processes named
                                          proc
$ pkill
processname its                           # Send signal to a process
name                                      with
s                                         # Resumes suspended jobs
bringing them to foreground (bg and fg
command)
$ fg
foregrou                                  # Brings the most recent job
nd                                        to
$ fg
n                                         # Brings job n to the
                                          foreground
```

Quiz:

Good job!

Question 1:

Difference between CentOS & Ubuntu Linux

- ○ Both are same just different name, different packaging.
- ○ CentOS is for Server and Ubuntu is for Desktop
- ◉ **The biggest difference between the two Linux distributions is that Ubuntu is based on the Debian architecture while CentOS is forked from Red Hat Enterprise Linux**

Good job!

Question 2:

apt command in ubuntu and yum in centos to manage packages

- ◉ **True**
- ○ False