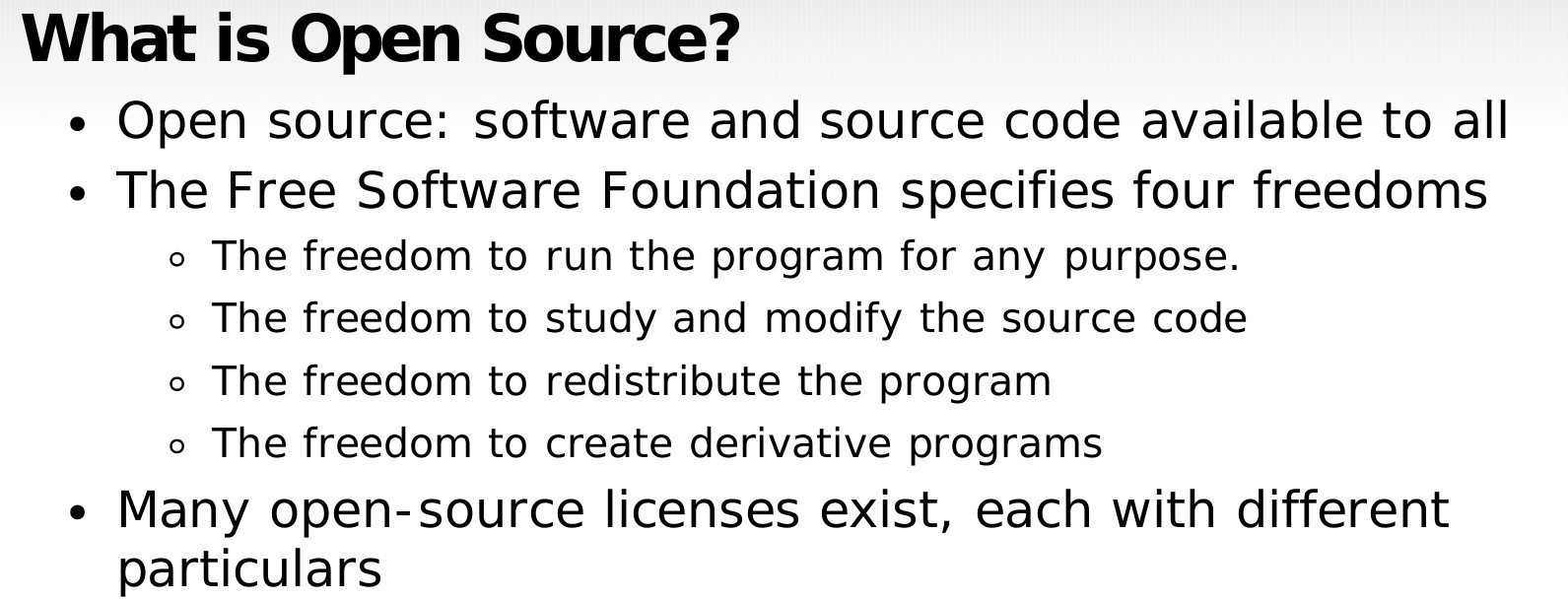
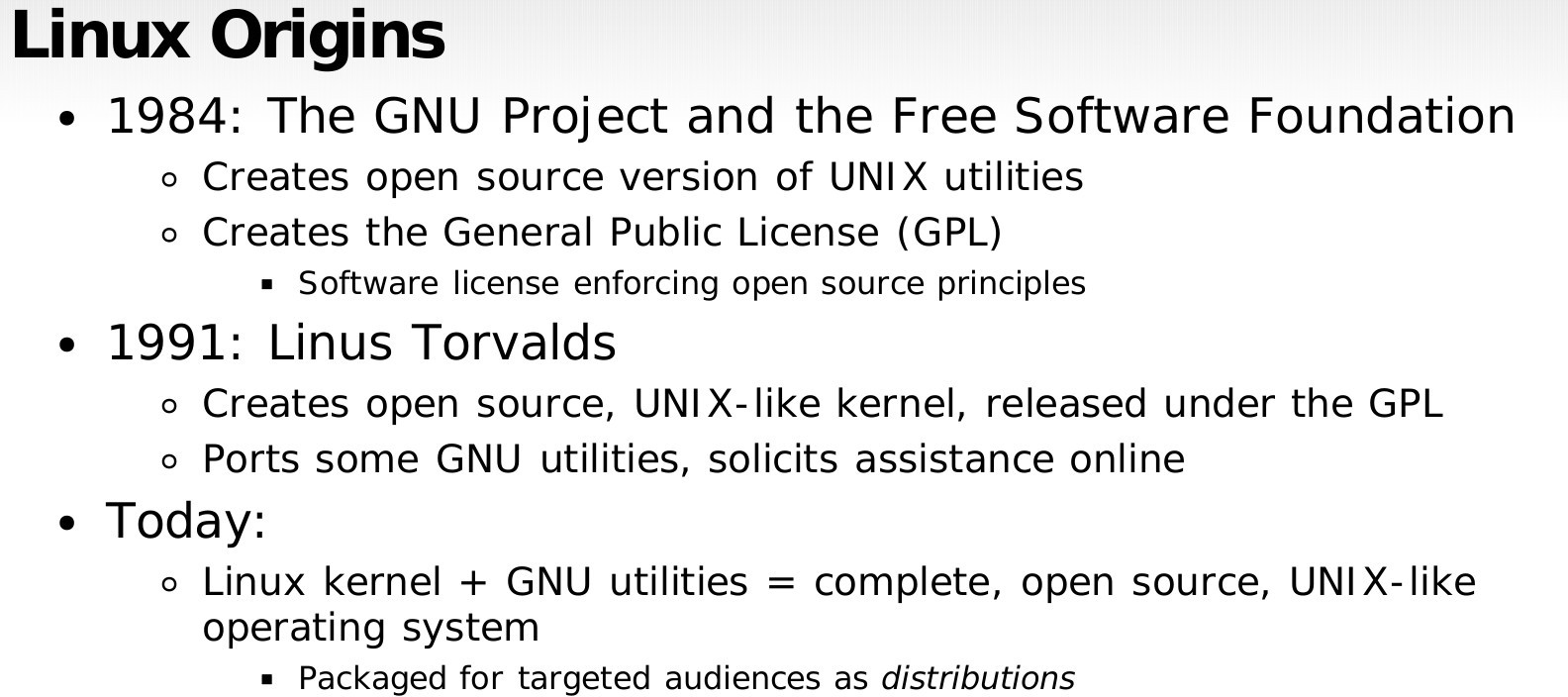
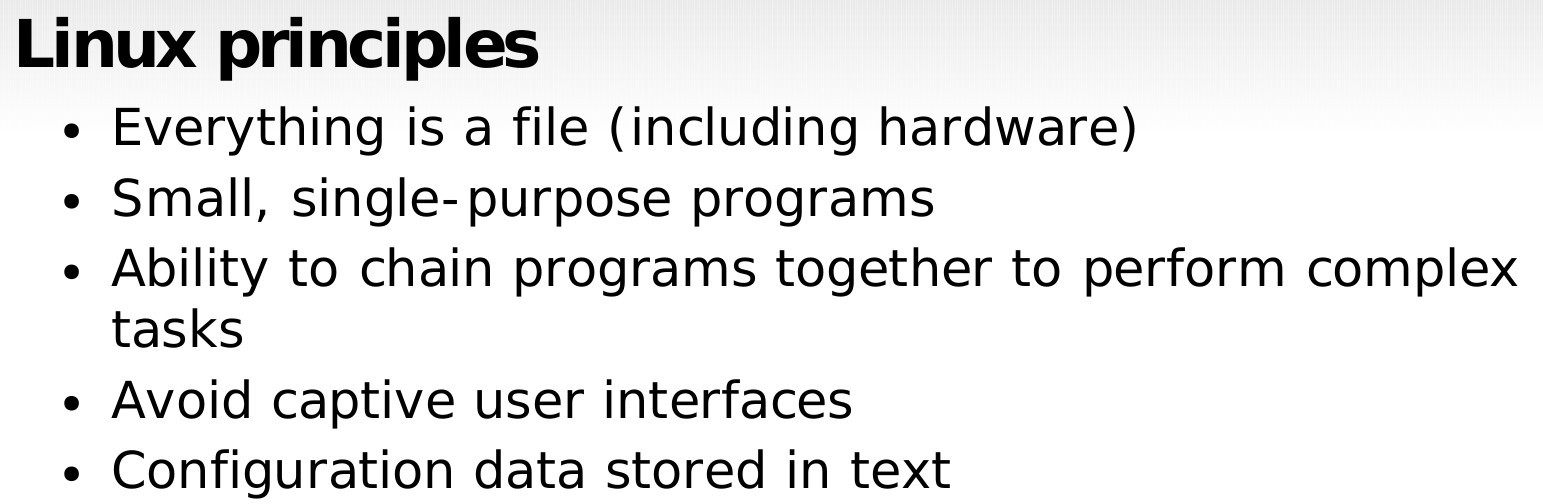
# Linux Introduction



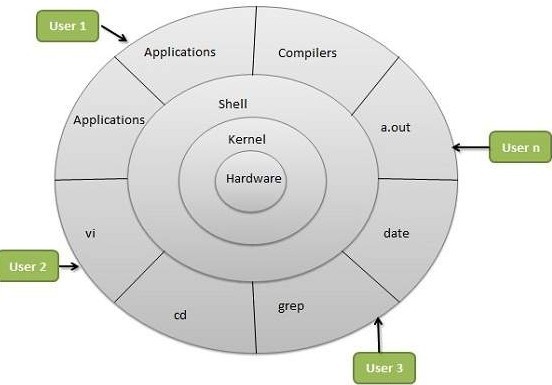


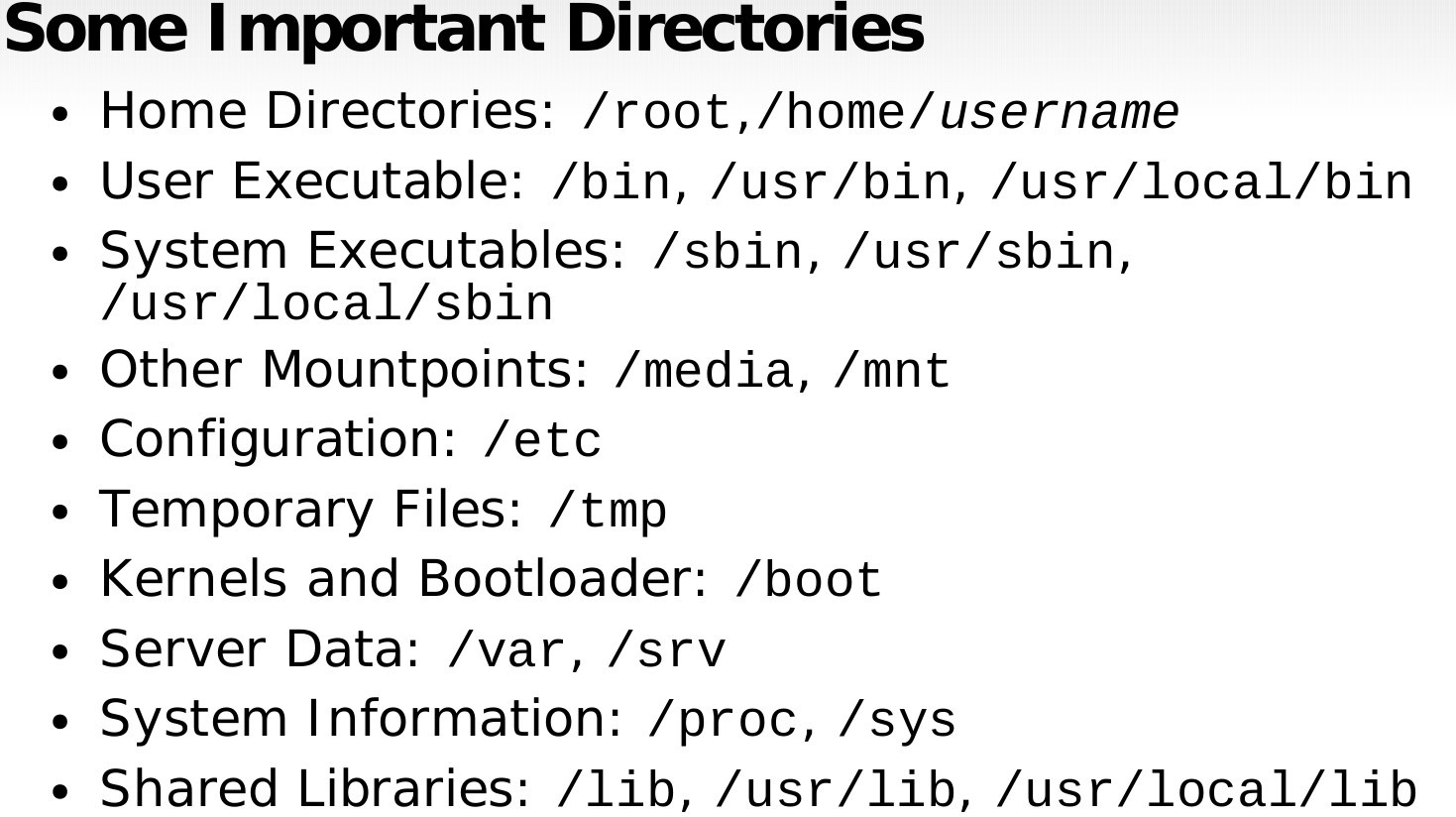


**Why Linux?**

* OpenSource.
* Community support.
* Heavily customizable.
* Most Servers runs on Linux.
* DevOps most of the tools implements on Linux only.
* Automation
* Secure.

# Architecture of Linux





**Diffrent 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 & Centos Debian based :- Ubuntu Server

## Diffrence between RPM based and Debian based.

From user’s point of view, there isn’t much difference in these tools. The RPM and DEB formats are both just archive files, with some metadata attached to them. They are both equally arcane, have hardcoded install paths and only differ in subtle details. DEB files are installation files for Debian based distributions. RPM files are installation files for Red Hat based distributions. Ubuntu is based on Debian’s package manage based on APT and DPKG. Red Hat, CentOS and Fedora are based on the old Red Hat Linux package management system, RPM.

**DEB or .deb (Debian based softwares)**

DEB is the extension of the Debian software package format and the most often used name for such binary packages. DEB was developed by Bedian.

**Example:** Google chrome software

**Package name:** google-chrome-stable\_current\_amd64**.deb Installation:** dpkg -i google-chrome-stable\_current\_amd64**.deb RPM or .rpm (Red Hat based softwares.)**

It is a package management system. The name RPM variously refers to the .rpm file format, files in this format, software packaged in such files, and the package manager itself. RPM was intended primarily for Linux distributions; the file format is the baseline package format of the Linux Standard Base. RPM was developed by Community & **Red Hat**.

**Example:** Google chrome software

**Package Name:** google-chrome-stable-57.0.2987.133-1.x86\_64.rpm

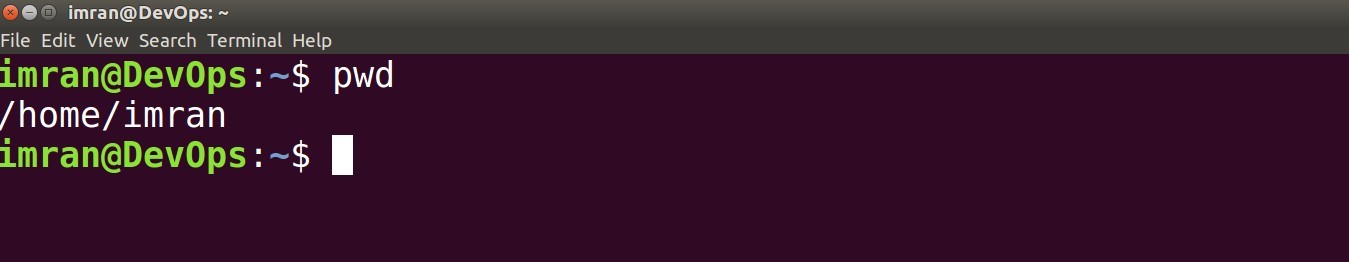
**Installation:** rpm -ivh google-chrome-stable-57.0.2987.133-1.x86\_64.rpm

**NOTE: You will also encounter diffrent commands, packages and service names while using both kinds of distros.**

* **Open Terminal**

# Basic Commands

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



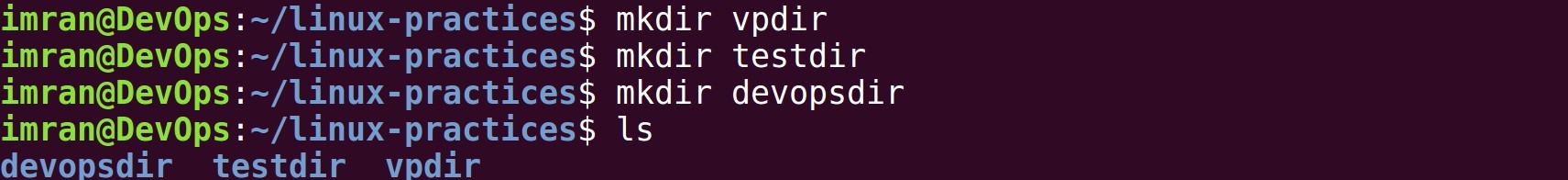
* **Create a directory/folder in your home directory.**



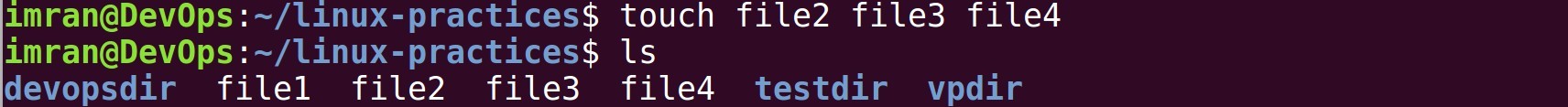
* **Change your current working directory to linux-practices(Go to linux-practices folder).**



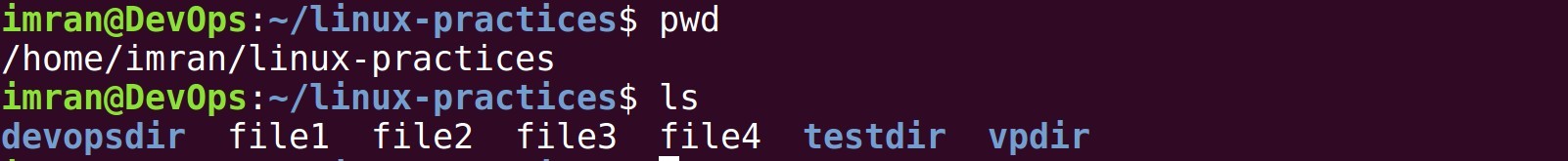
* **Create some more directories and list them with “ls” command.**



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



* **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:**

**/home/imran/linux-practices/**

**/var/**[**ftp**](http://www.linuxnix.com/ftps-server-configuration/)**/pub**

[**/etc**](http://www.linuxnix.com/linux-directory-structure-explainedetc-folder/)**/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/imran and I want to change directory to /home/imran/linux-practices. I can use relative path concept to change directory to linux-practices and also devopsdir directory.



If you see all these paths did not start with / directory.

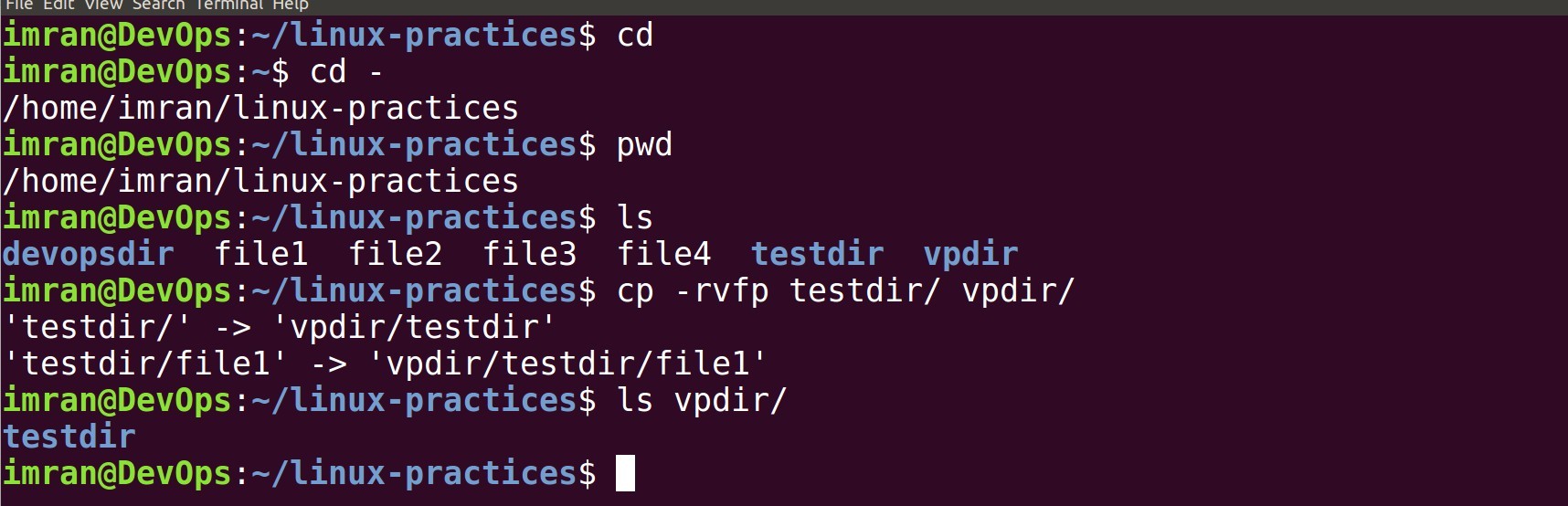
* **Creating directories in devopsdir directory with absolute and relative path.**



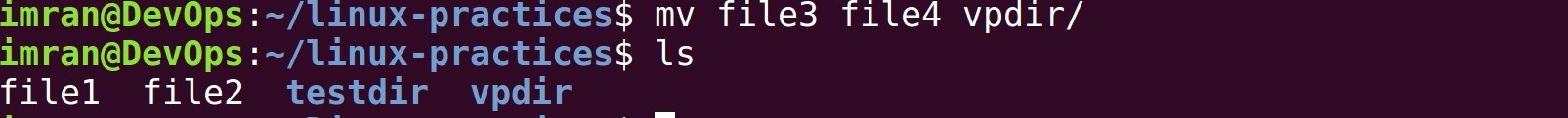
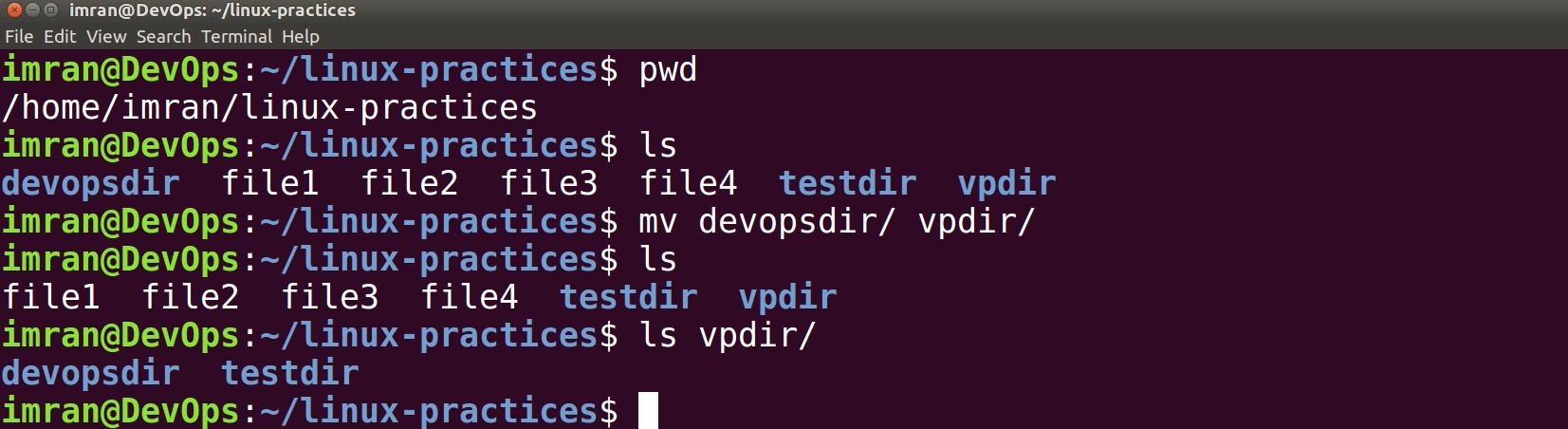
* **Copying files into directory.**



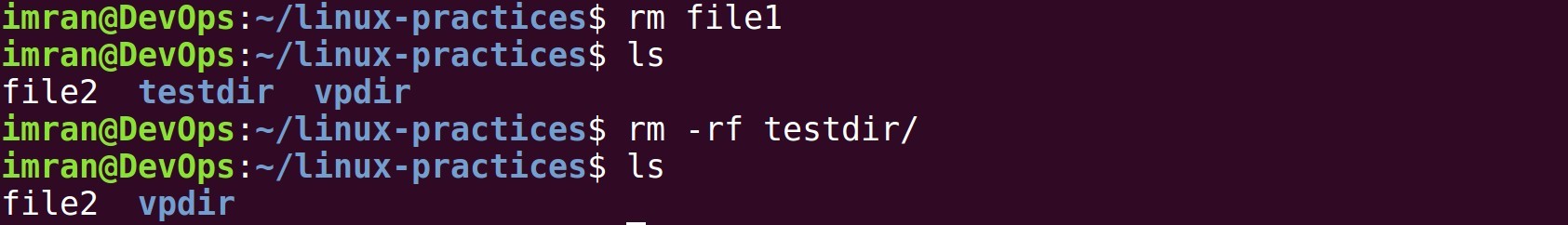
* **Copying directories from one location to another.**



* **Moving files from one location to another.**

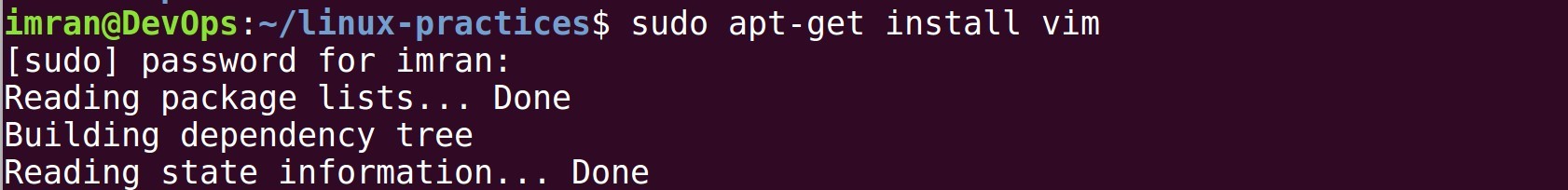


* **Removing files and directories.**



* **Install vim editor.**

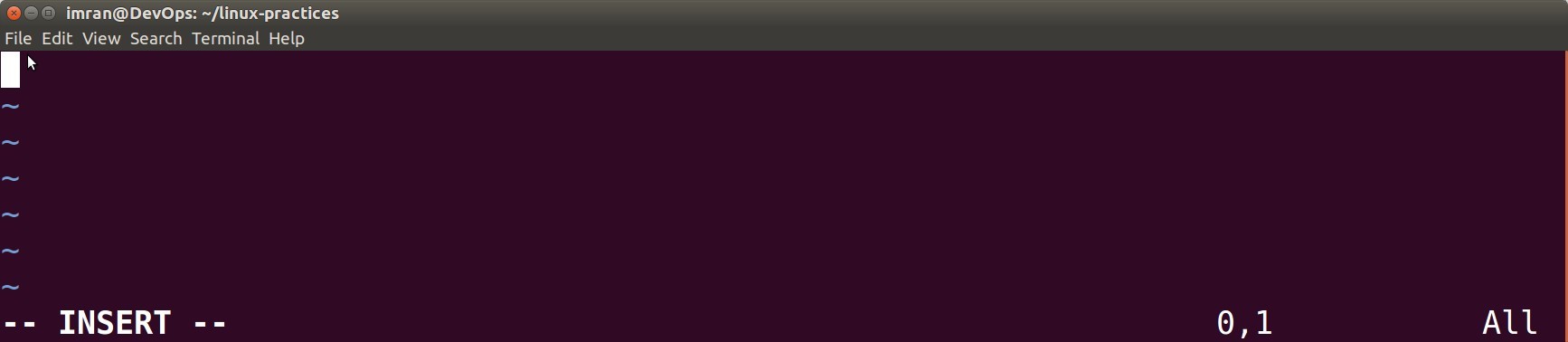
# VIM EDITOR



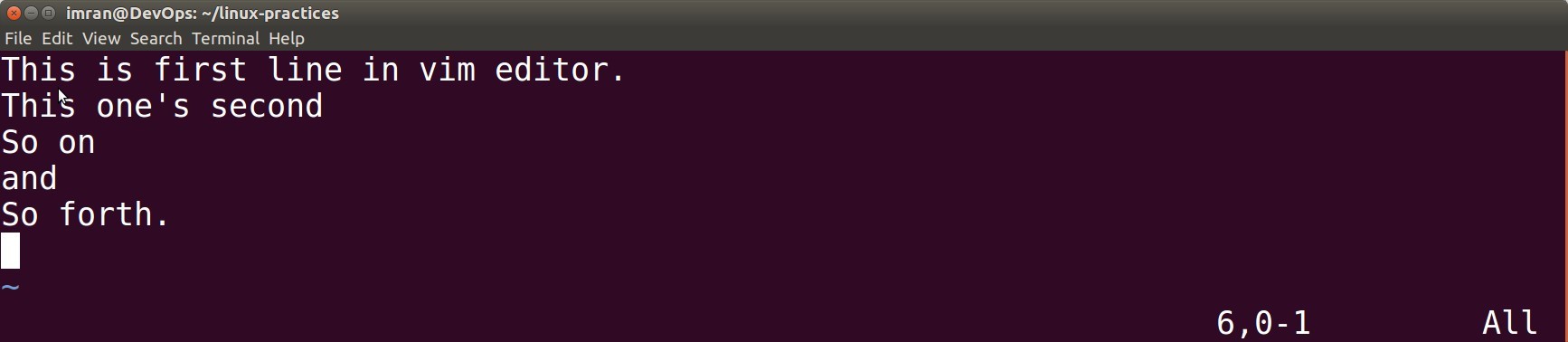
* **Open up a file in vim editor**



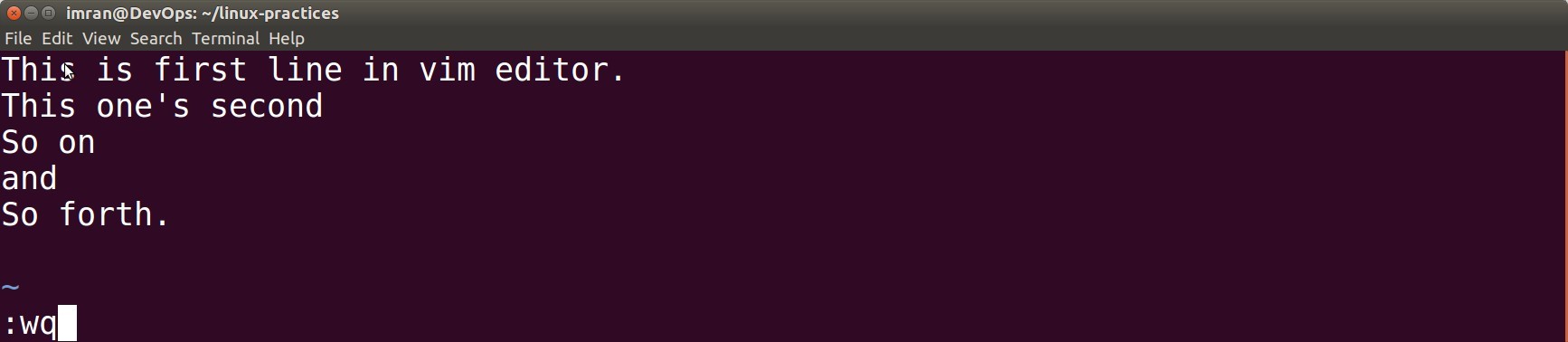
* **Hit i to enter into insert mode**



**=> type few lines => hit Esc**

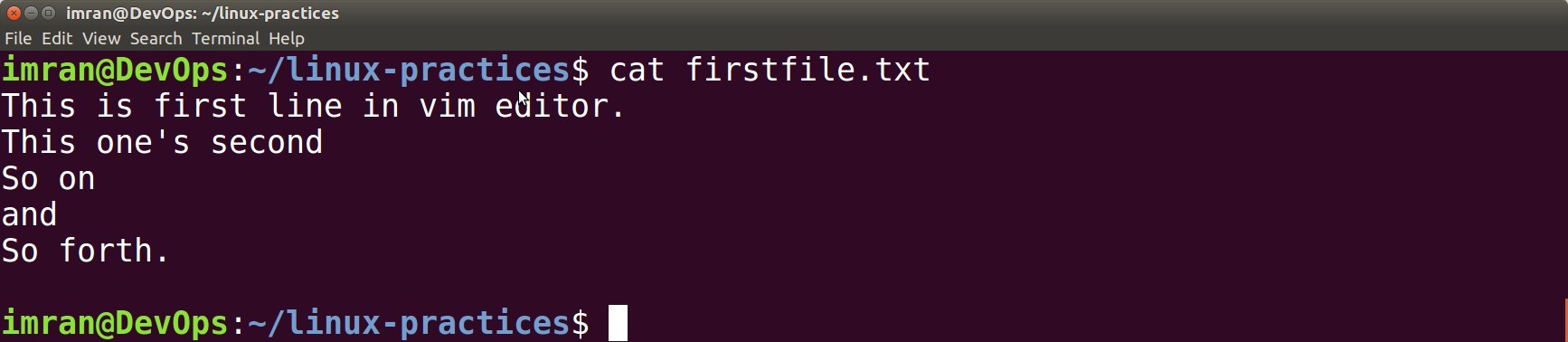


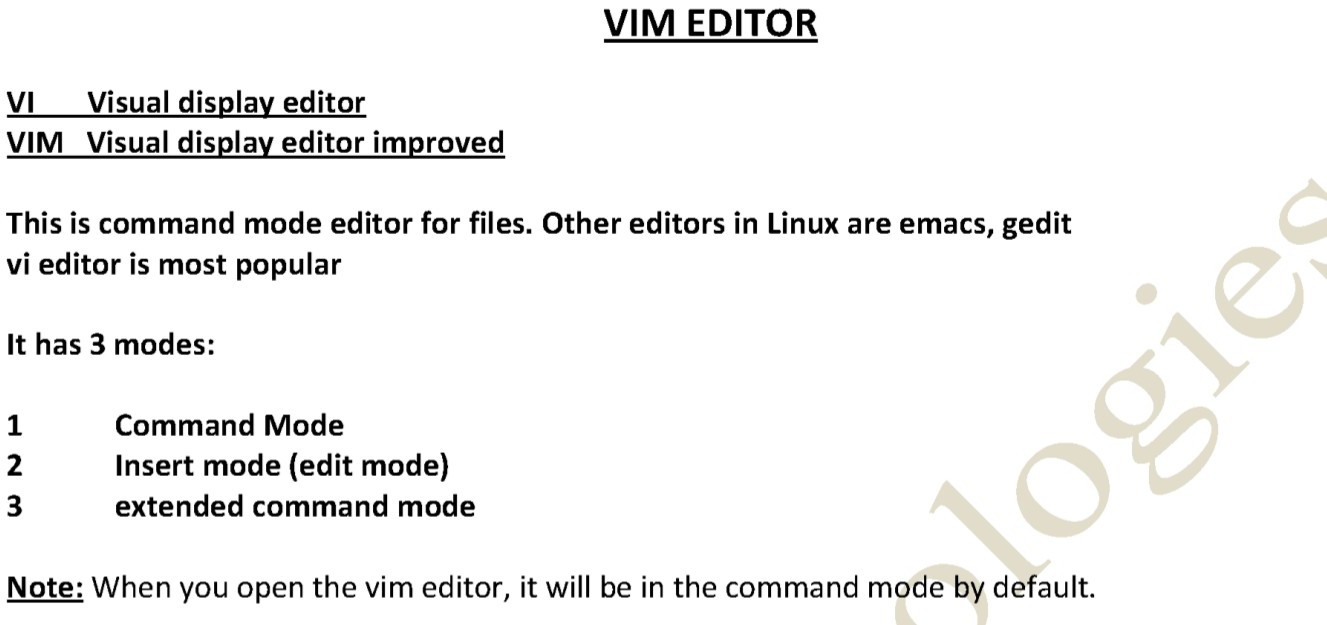
**=> type :wq**

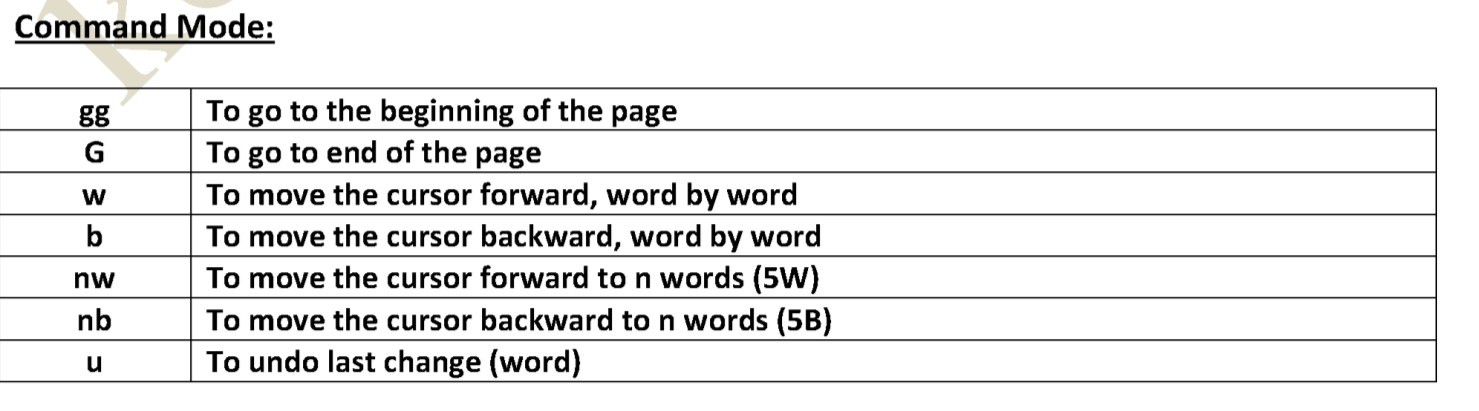
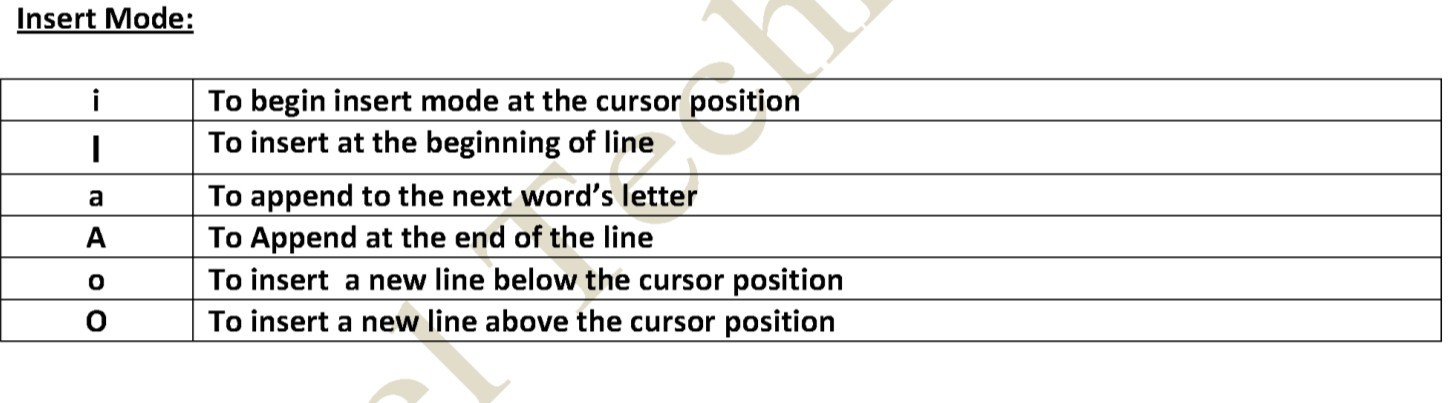


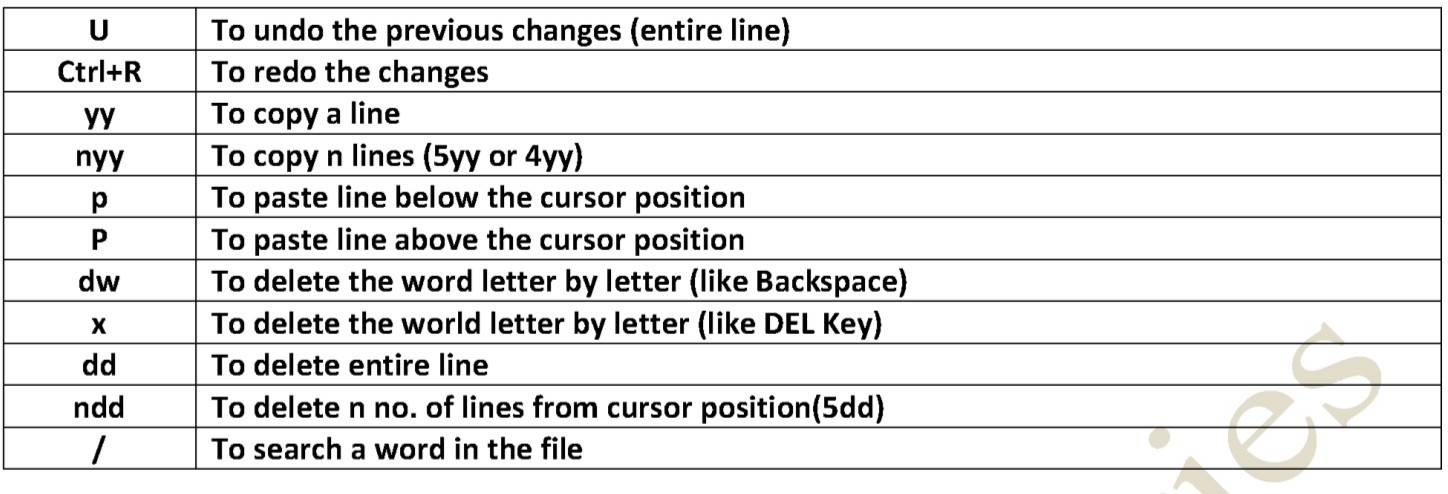
**=> Enter.**

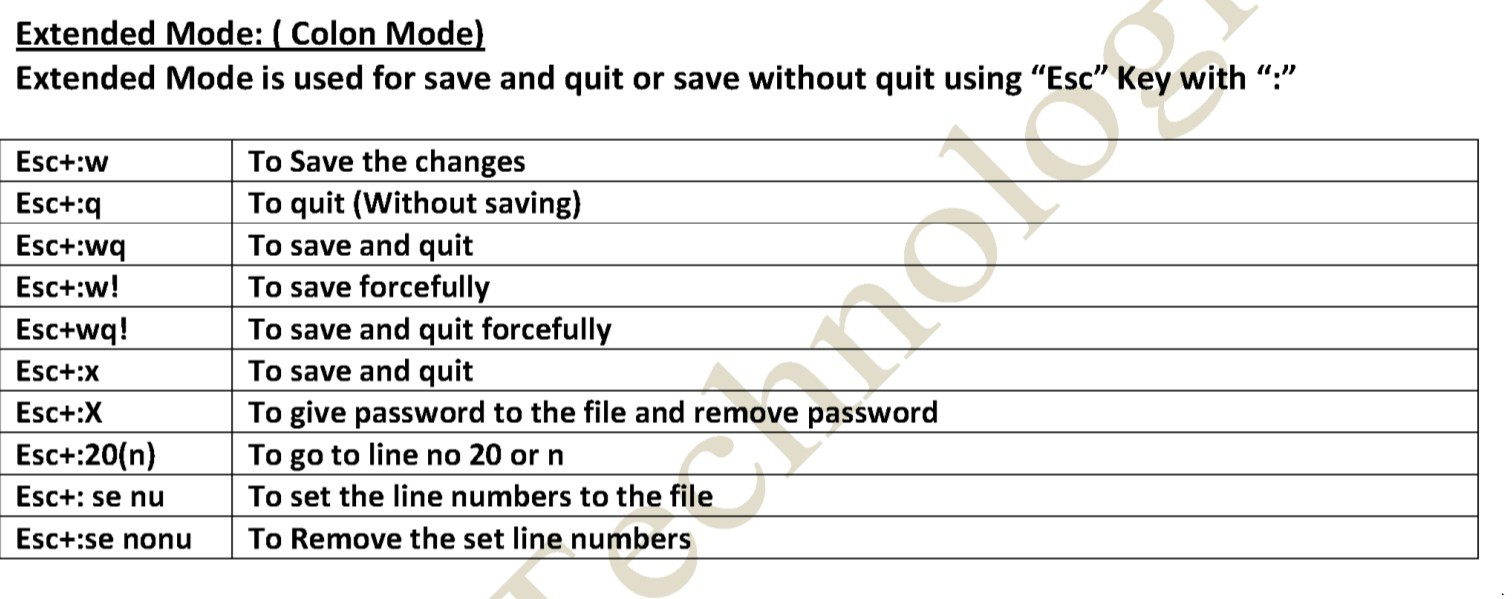
* **Read file with cat command.**

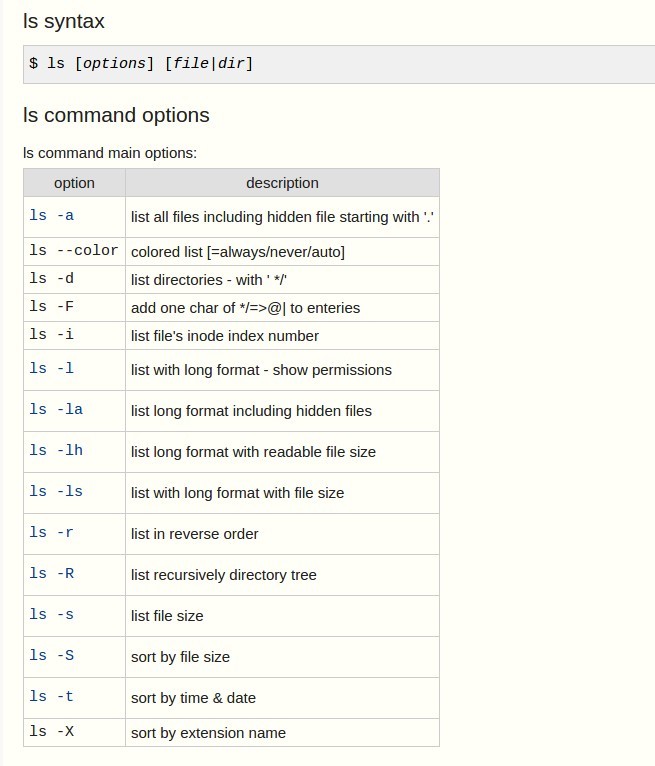




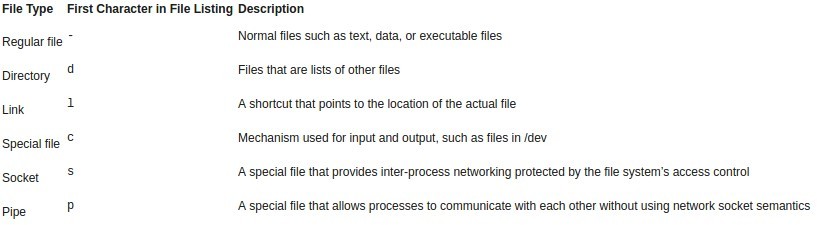








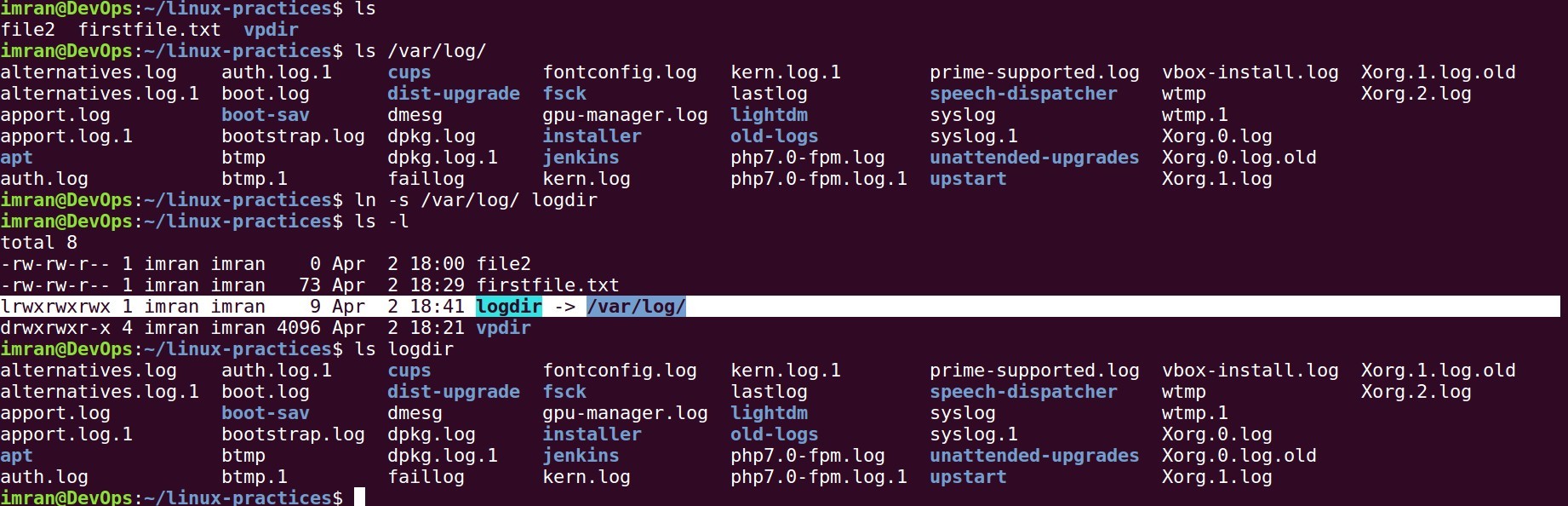
**Types of files in linux.**



**Symbolic links**

Symbolic links are like desktop shortcuts we use in windows.

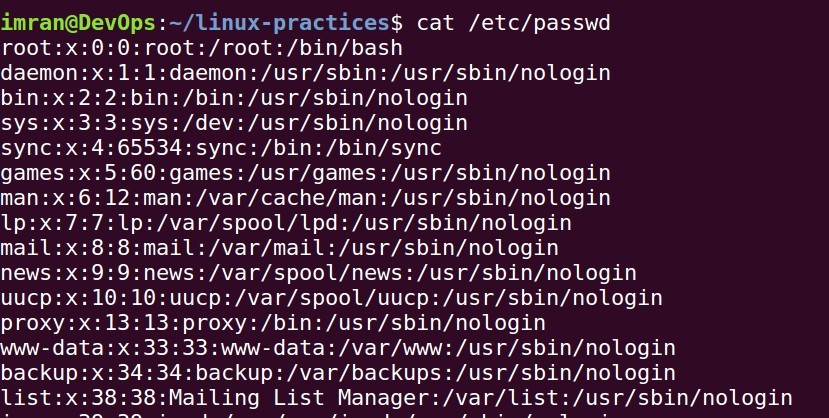
Create a soft link for /var/log directory in our current working directory.



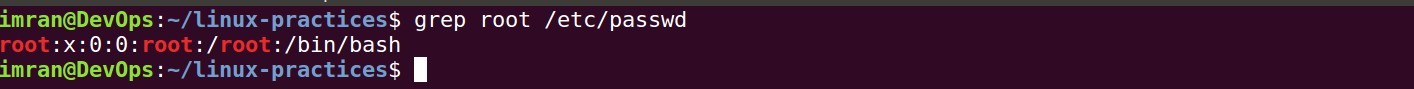
**Grep**

# Filter & IO redirection command.

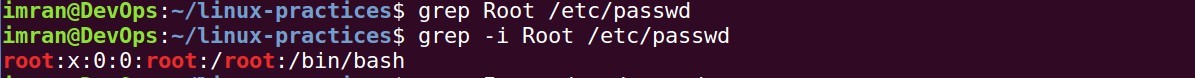
grep command is used to find texts from any text input. Passwd file: stores information about all the users in the system



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



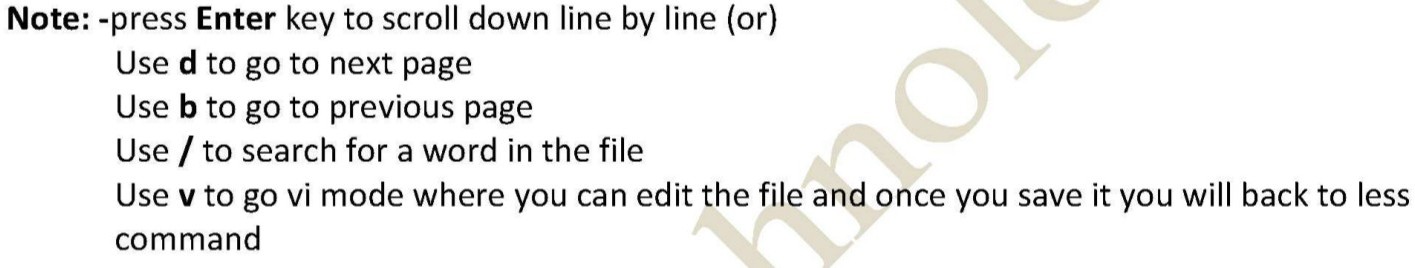
* **Linux is case sensetive, Root is diffrent that root. Ignoring case in grep with -i option.**



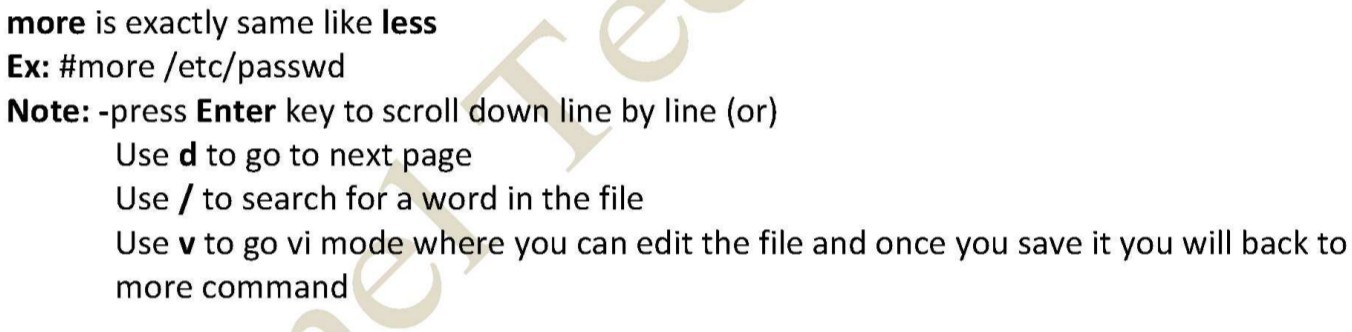
* **To display things except the given word use -v option**

**Filter Commands**

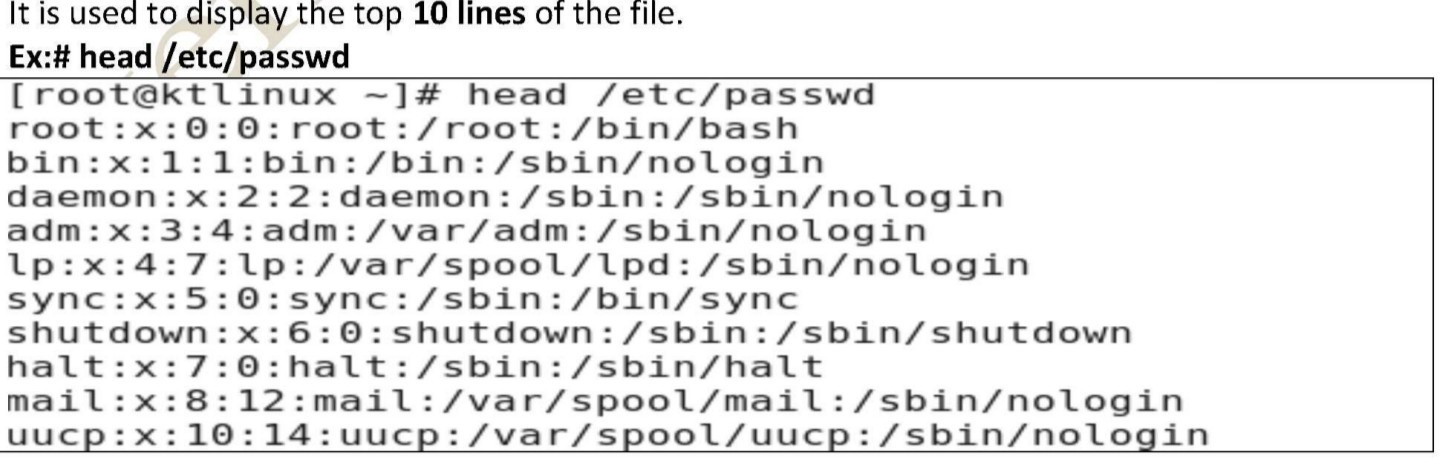
* **less: Displays file content page wise or line wise. Ex: less /etc/passwd**



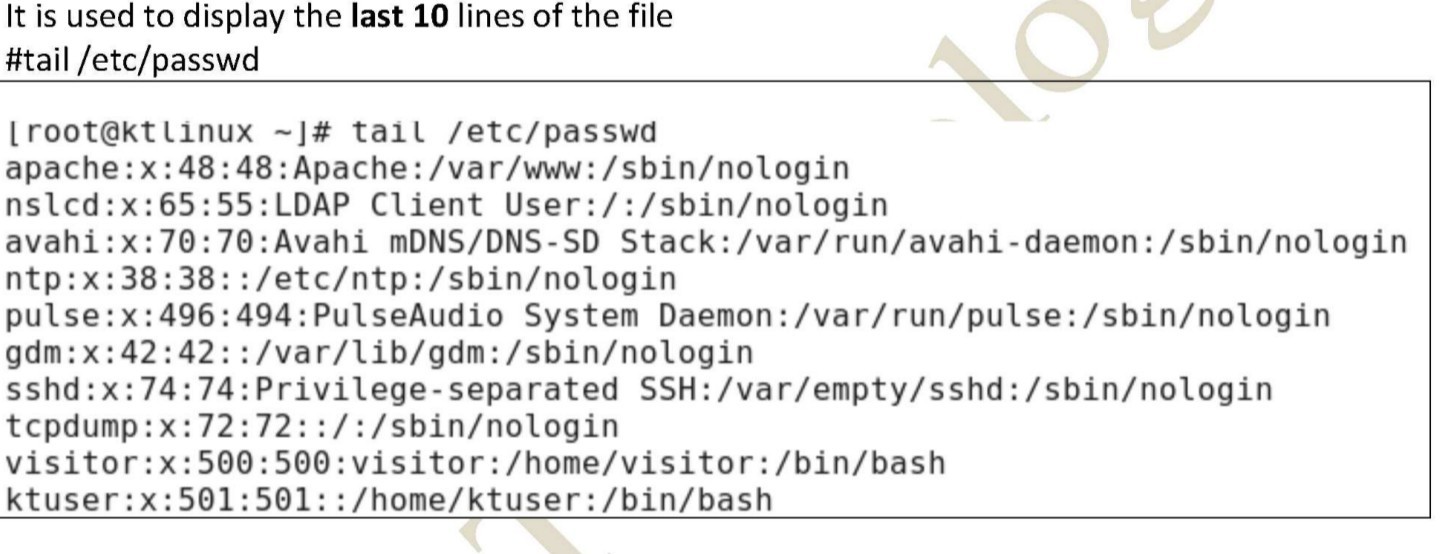
* **more**



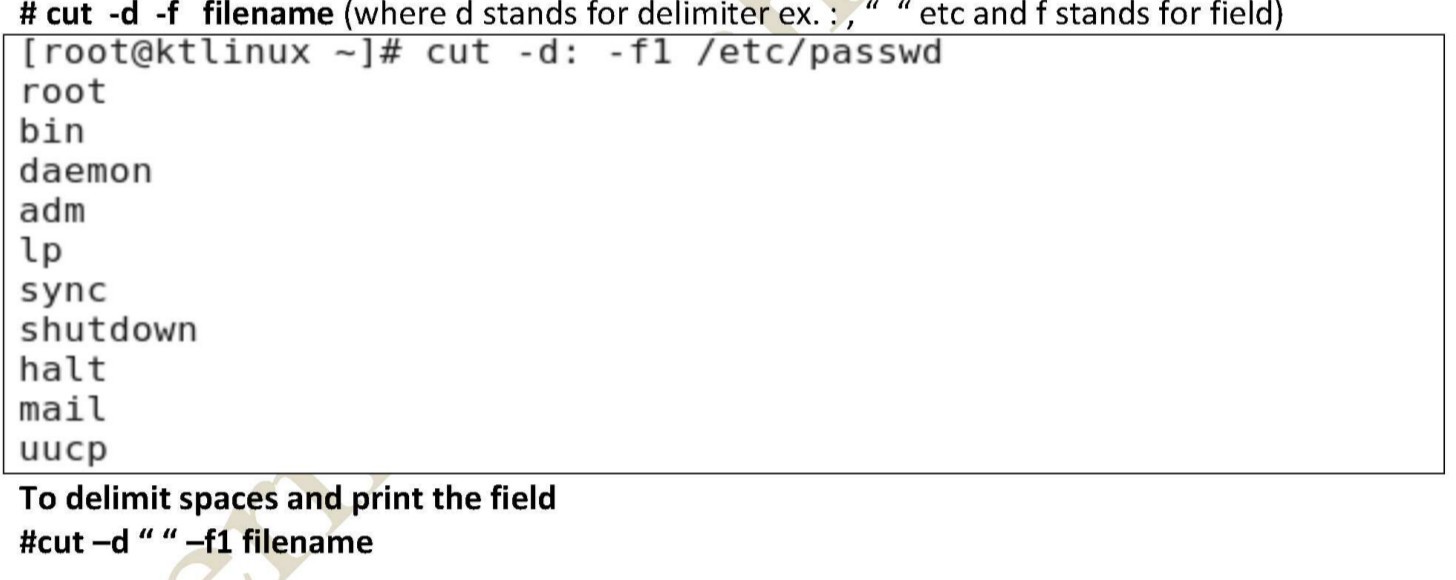
* **head**



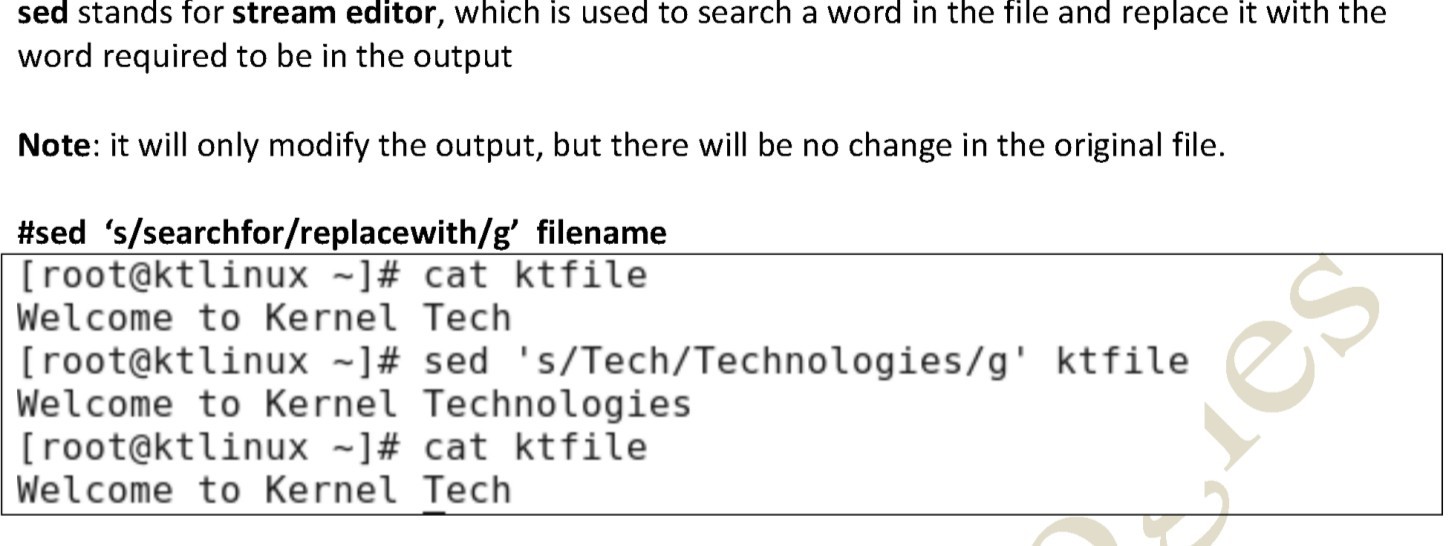
* **tail**



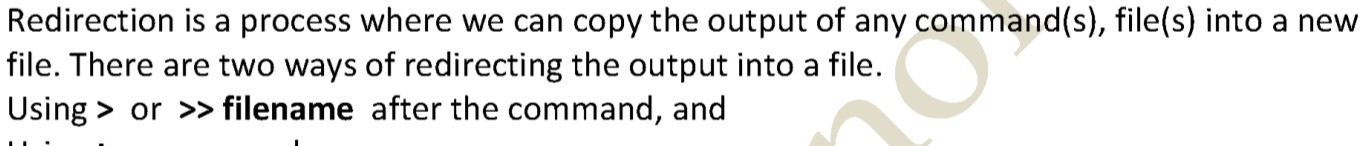
* **cut**



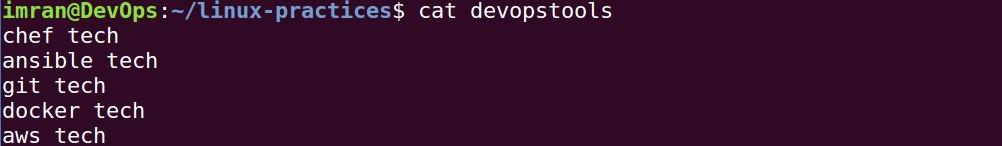
* **sed**



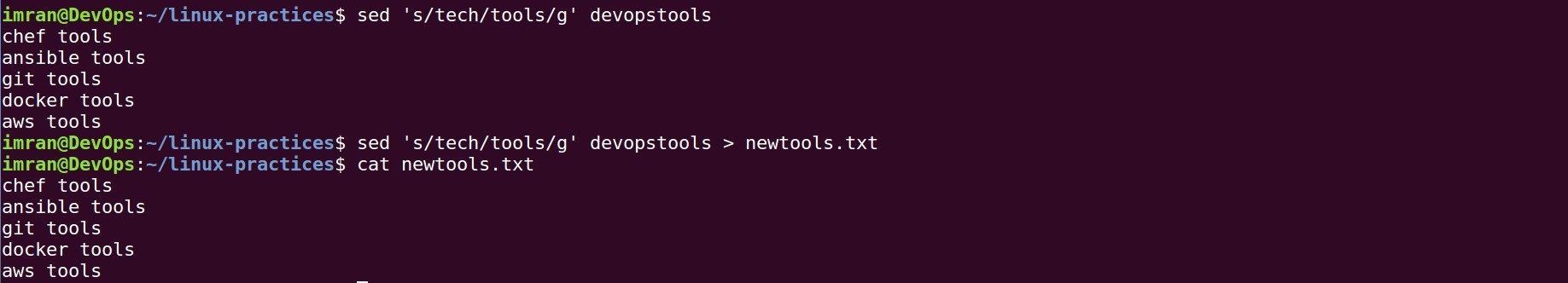
## I/O redirection

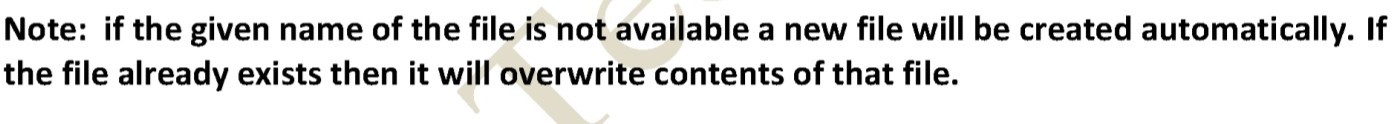


* **Create a file named devopstools with below content.**

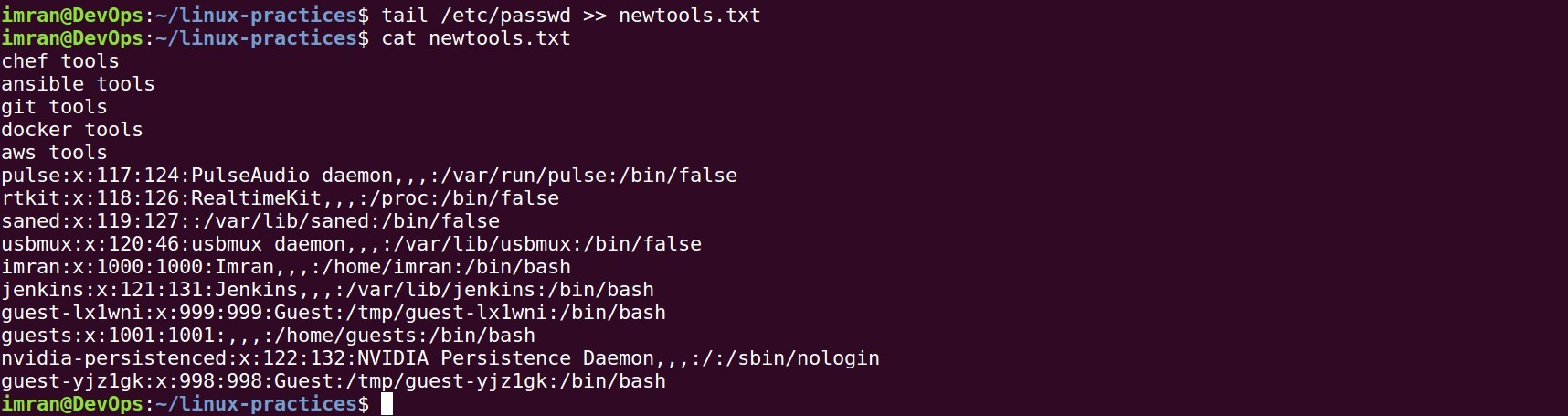


* **Search for text “tech” replace it with “tools” and redirect output to a new file.**



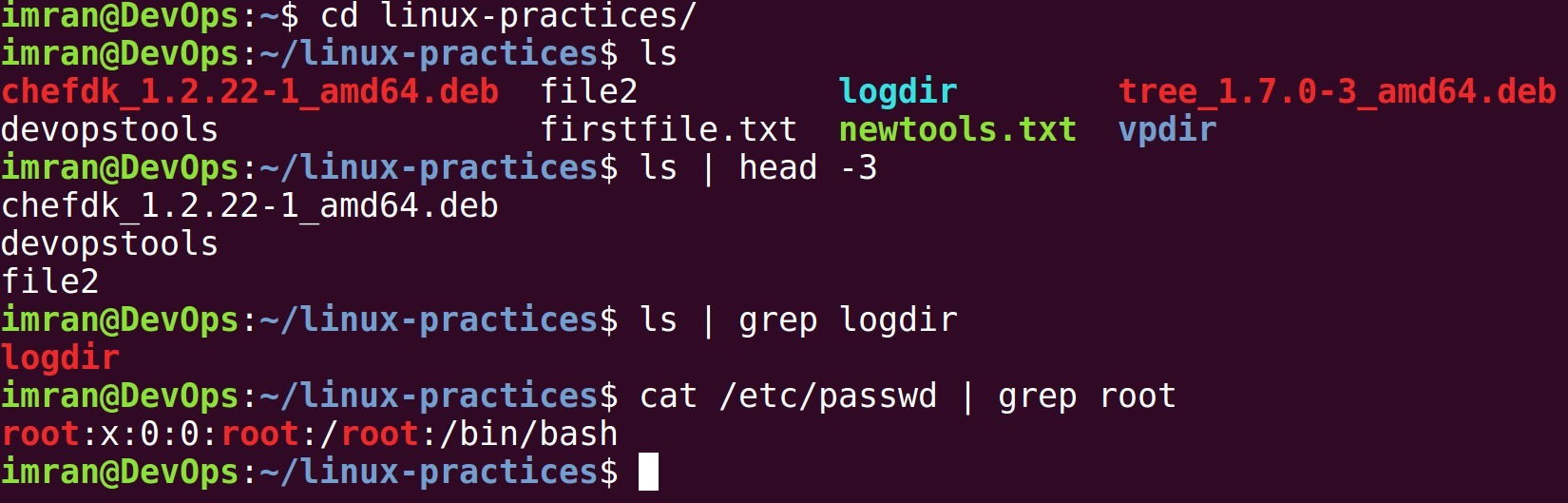


* **Appending another output in same file with “>>” .**



## Piping

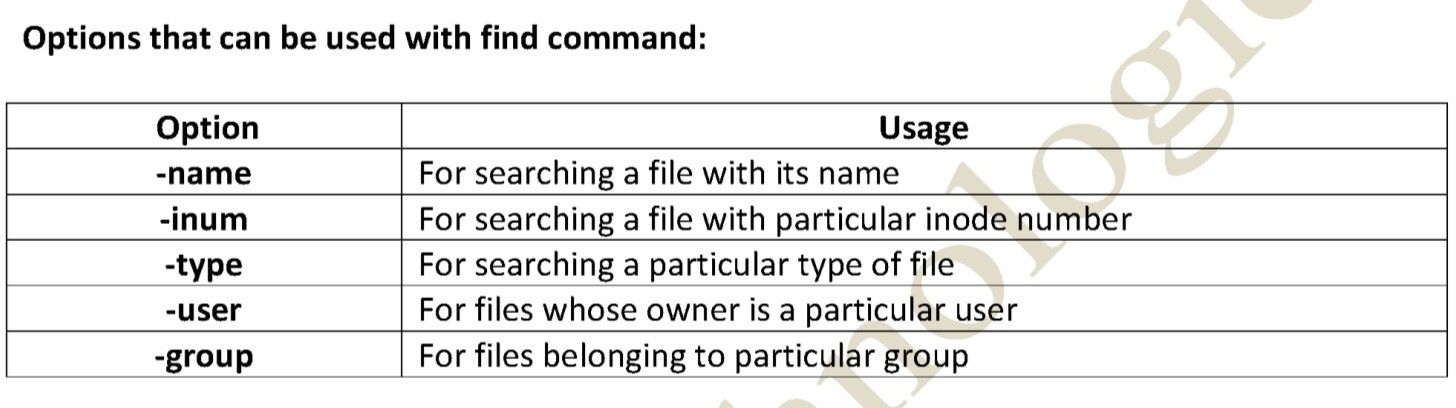
So far we've dealt with sending data to and from files. Now we'll take a look at a mechanism for sending data from one program to another. It's called piping and the operator we use is ( | ). What this operator does is feed the output from the program on the left as input to the program on the right.



**Find**

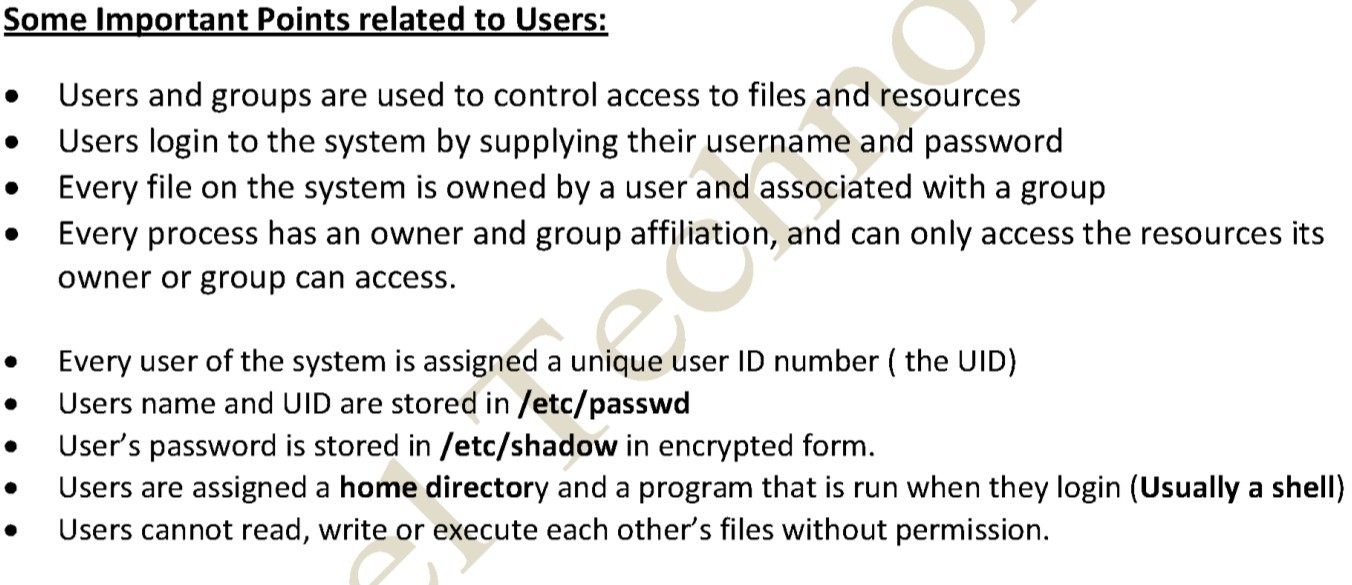


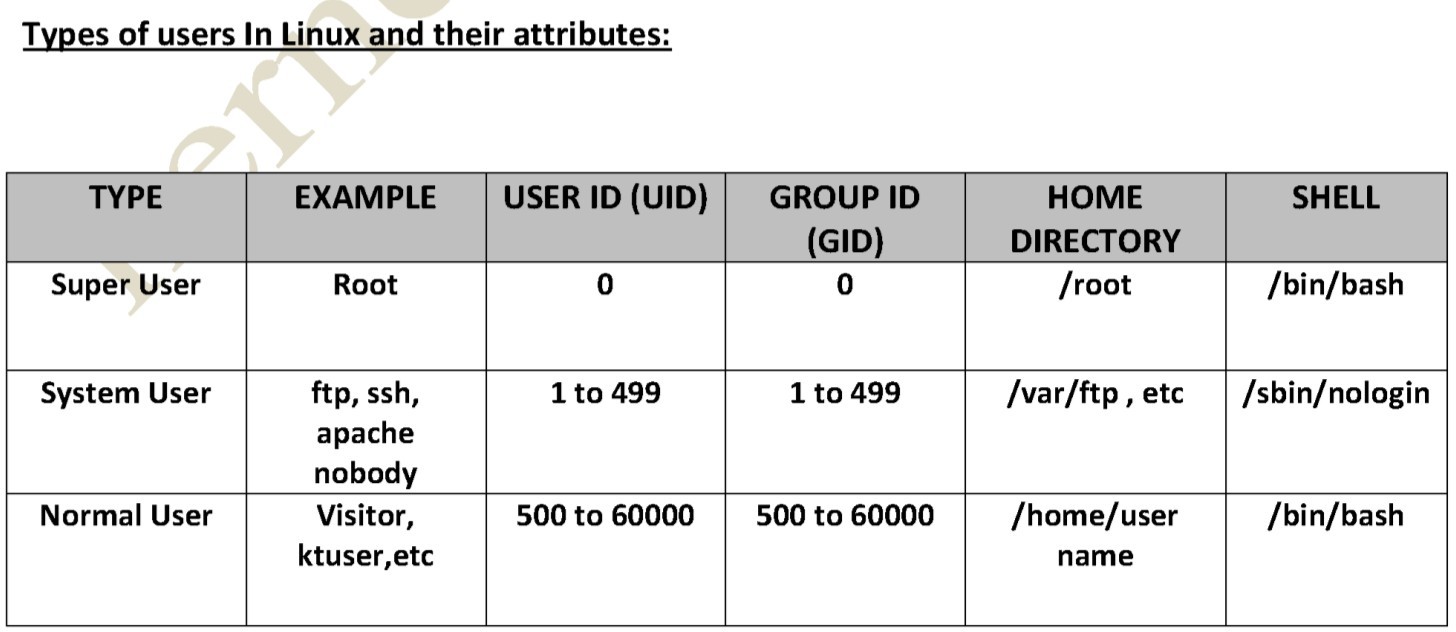


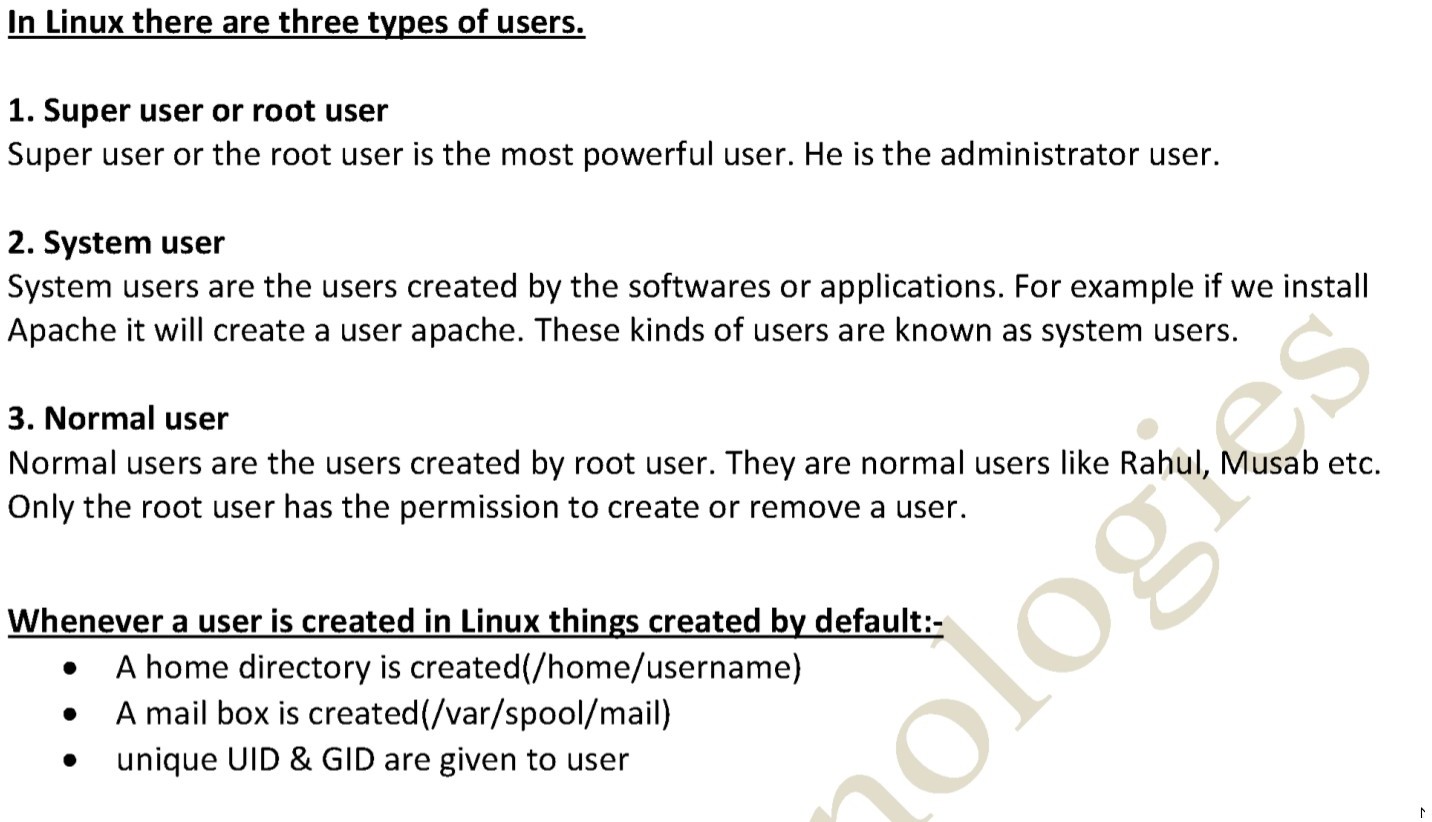


# Users & Groups.

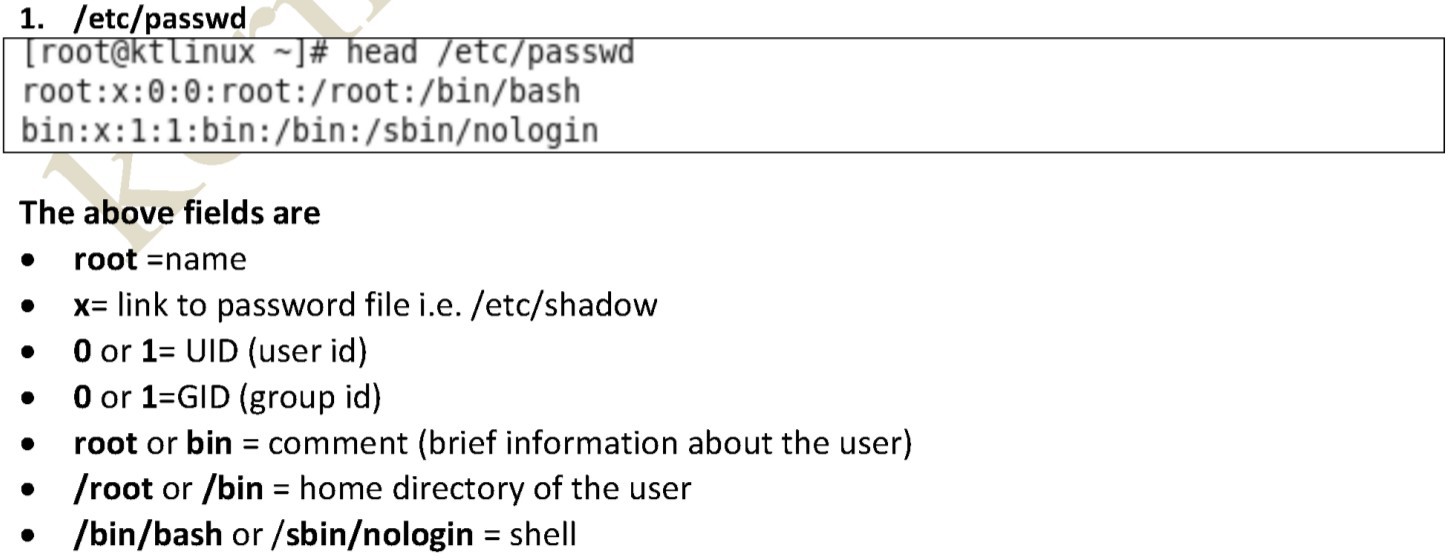
**USERS**







**Passwd file**



$ id

with login and group

=> Show the active user id

$ last

system (few more examples)

=> Show last logins on the

$ who

system(real user who logged in)

=> Show who is logged on the

$ sudo groupadd admin

=> Add group "admin"

(force add existing group)



$ sudo adduser sam => Create user "sam" and add to group "admin"(here read all parameter)

$ sudo usermod

=> Modify user

information(mostly useful for linux system admins)

$ sudo passwd sam

=> Changing password for other user

$ passwd

=> Changing your own password.

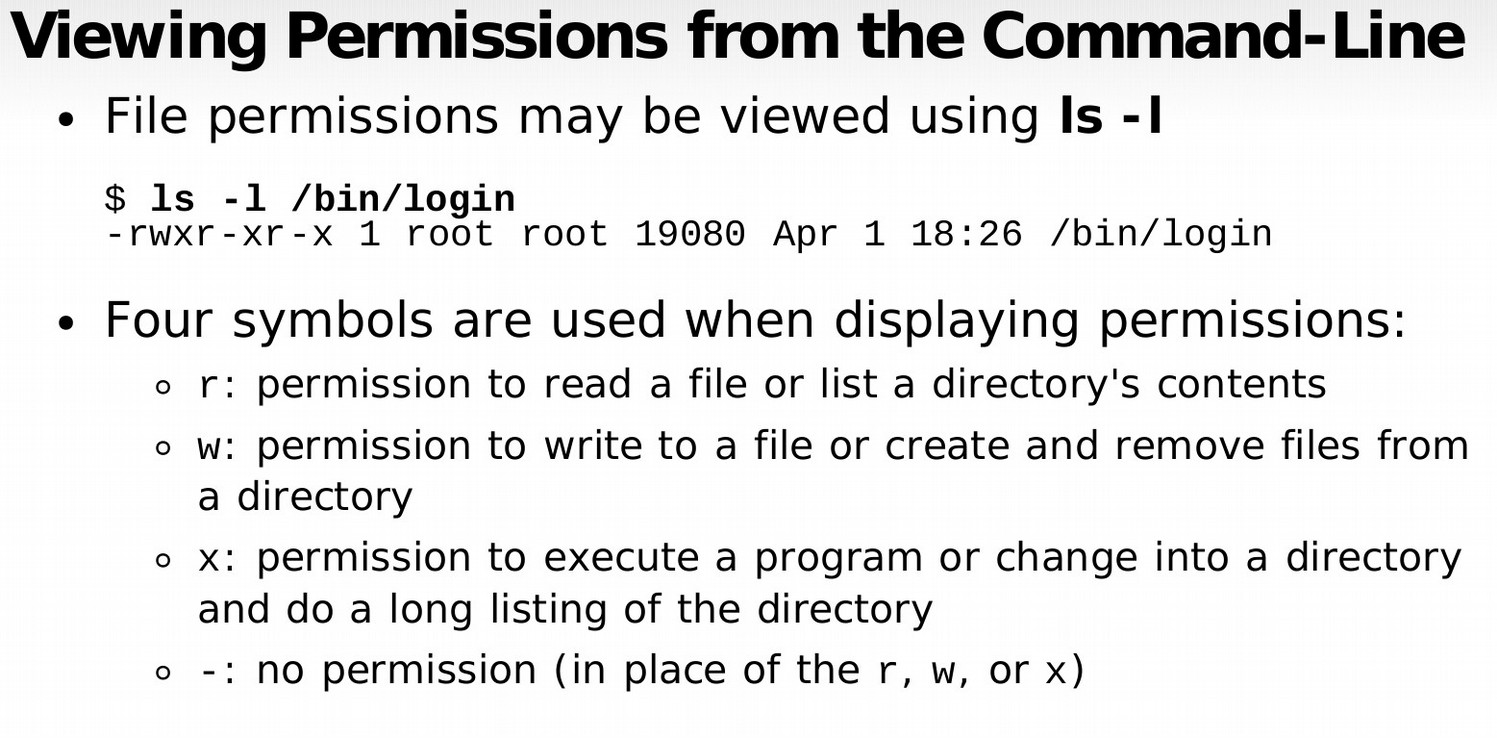
$ sudo userdel -r sam (force,file removal)

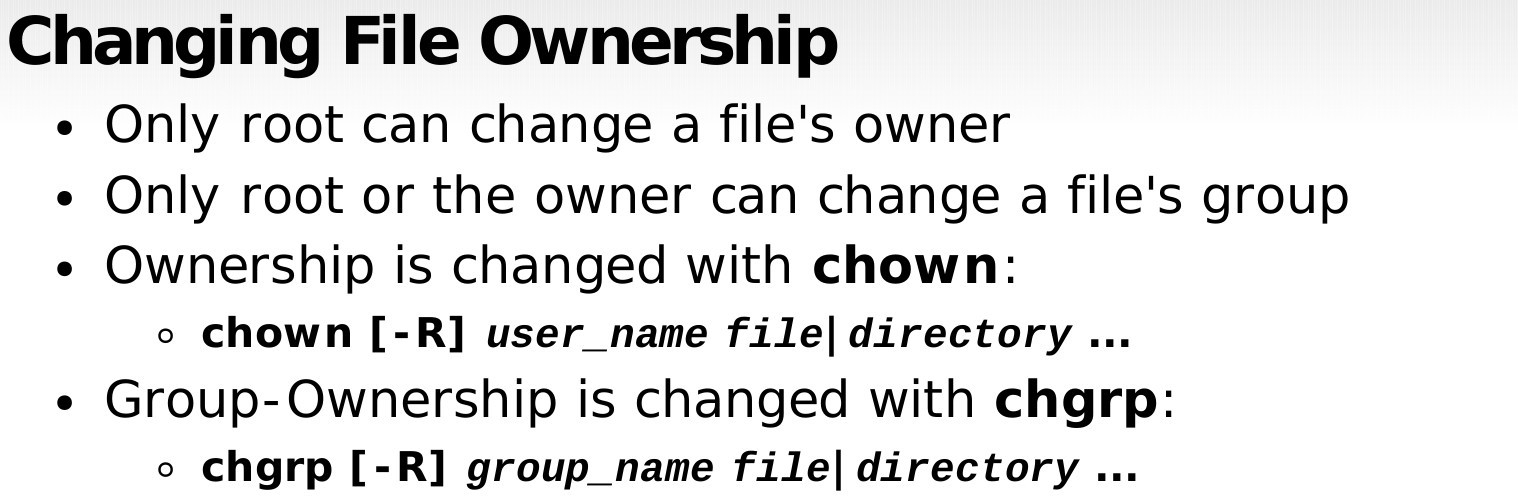
=> Delete user sam

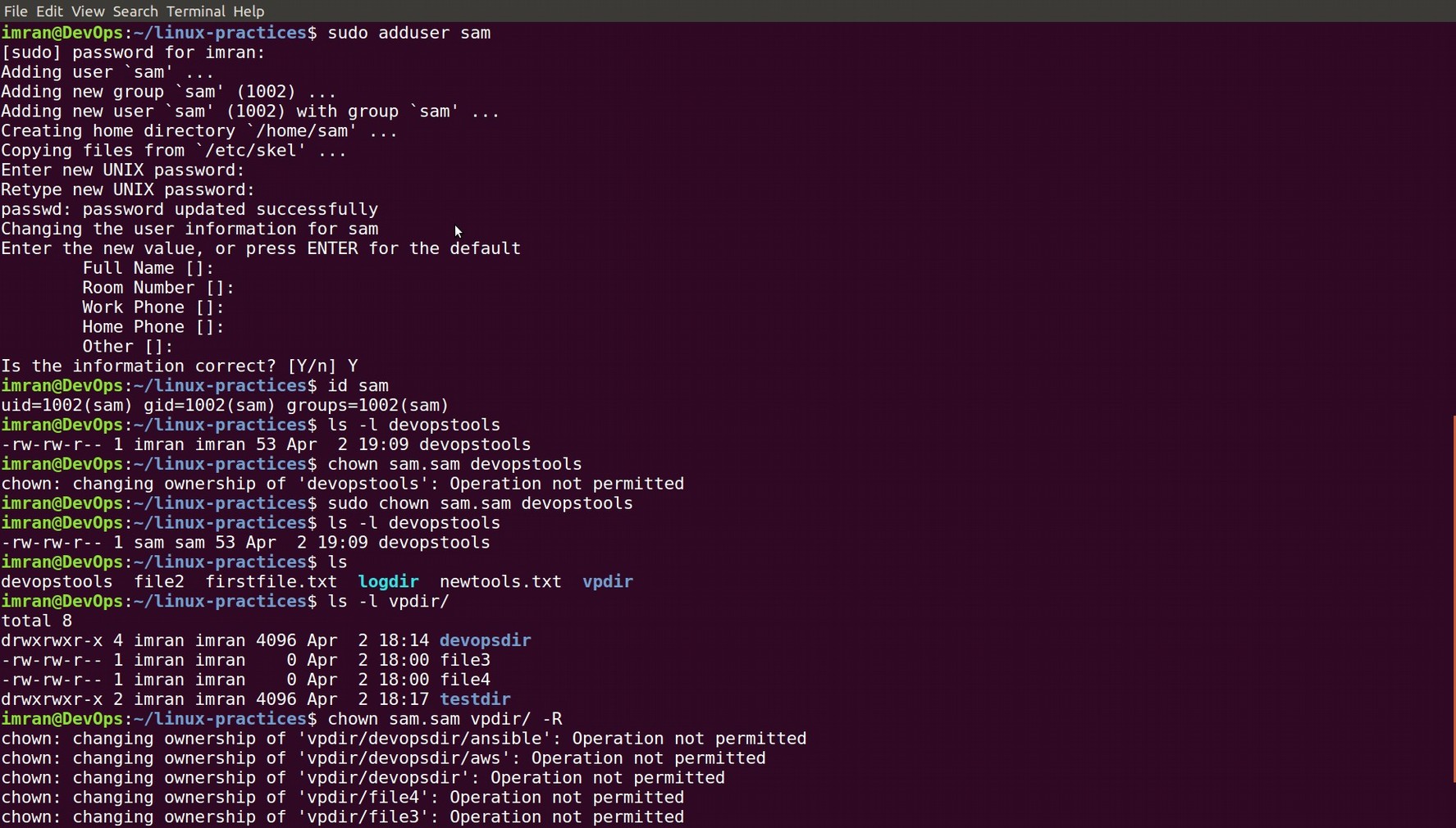
$ sudo groupdel deck

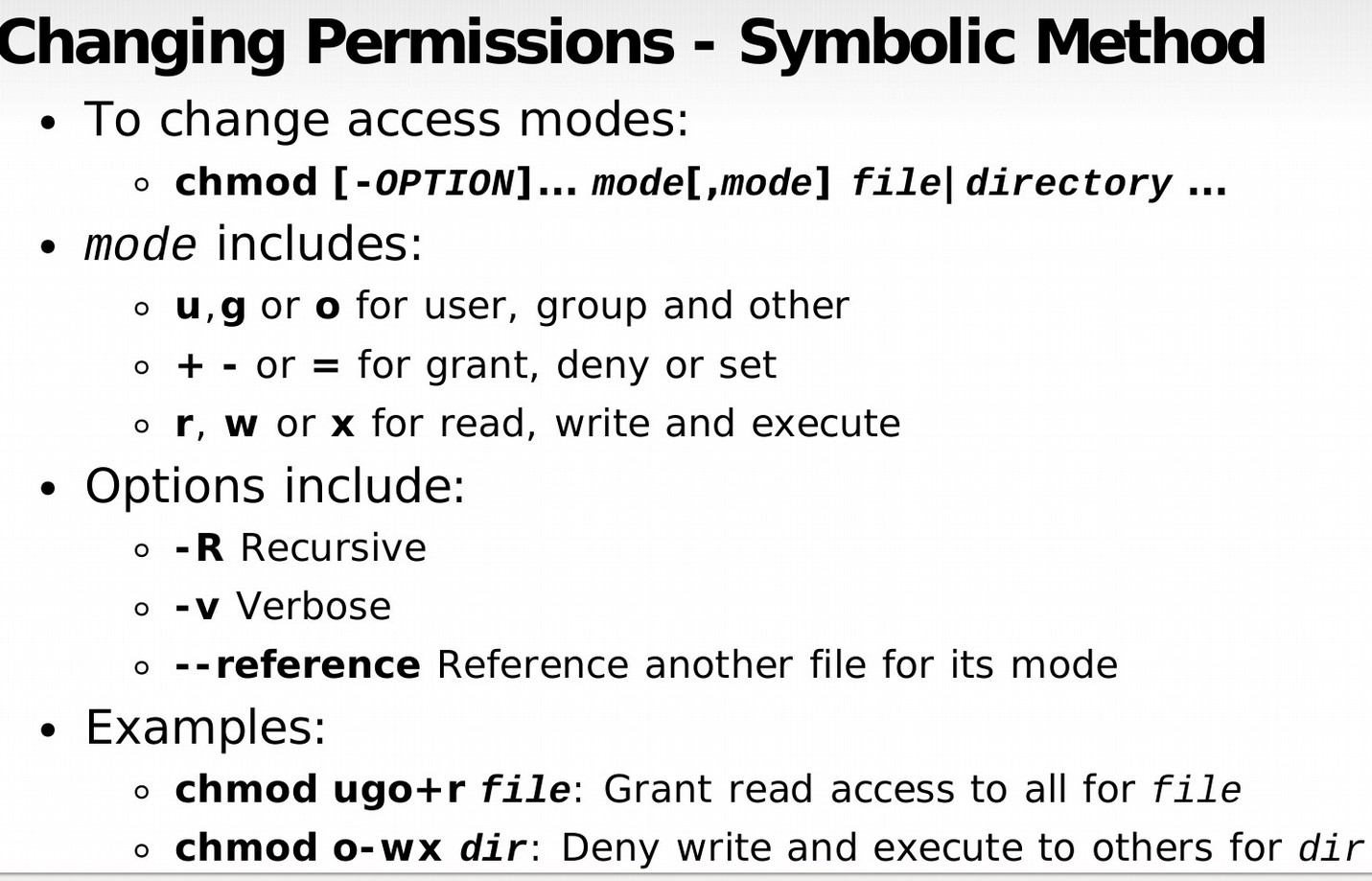
=> Deletes a group

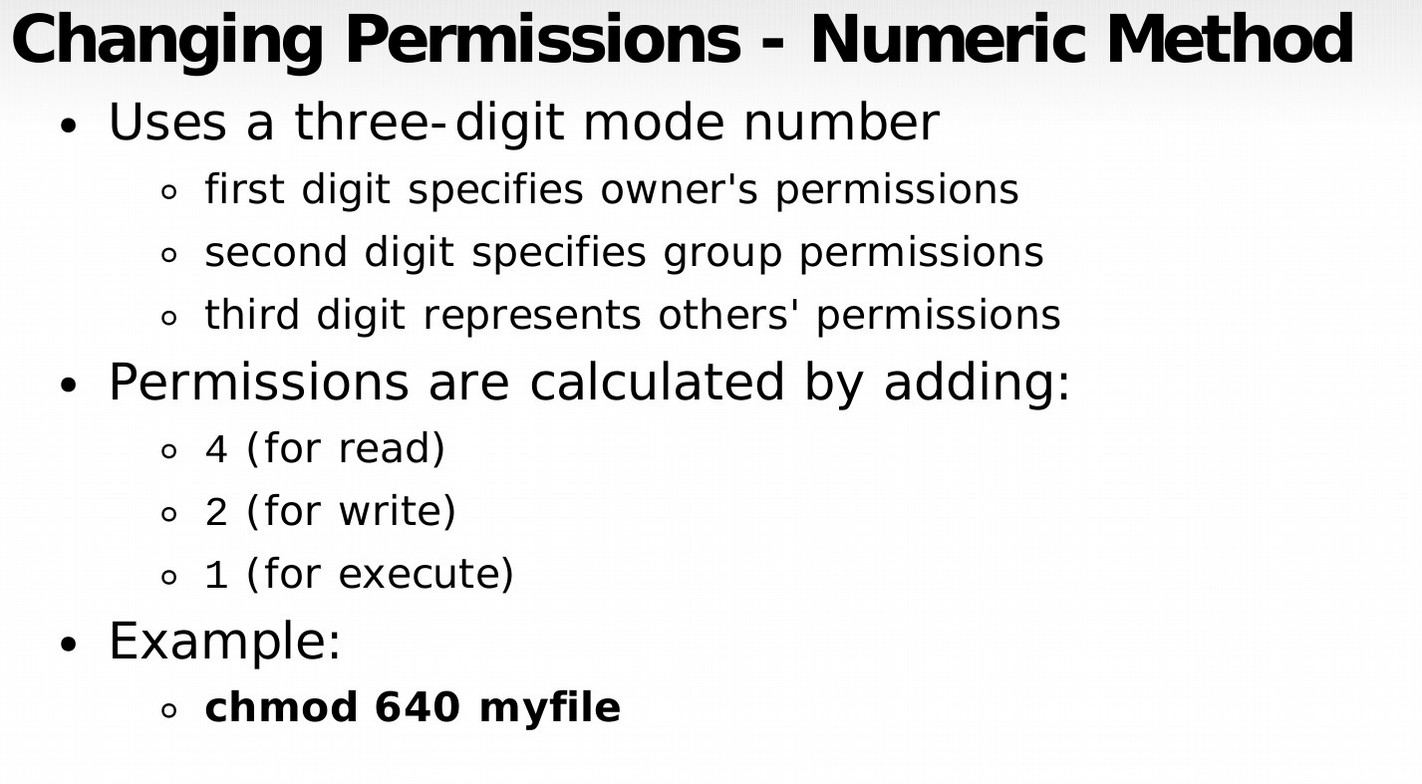
# File permissions

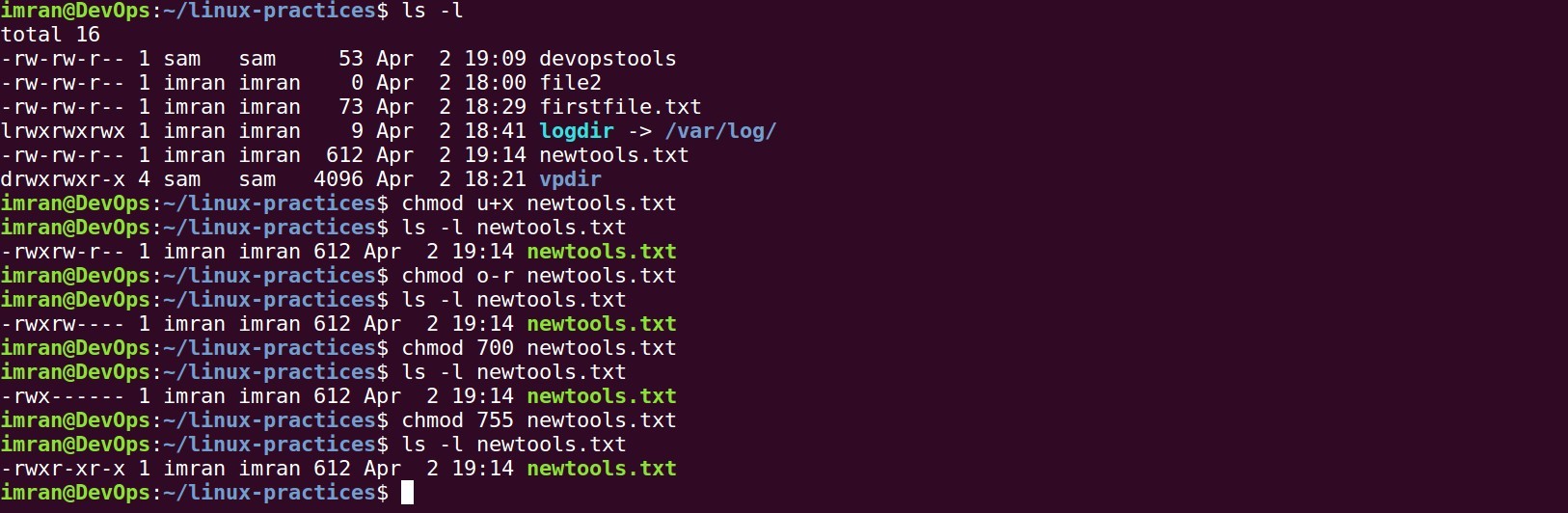










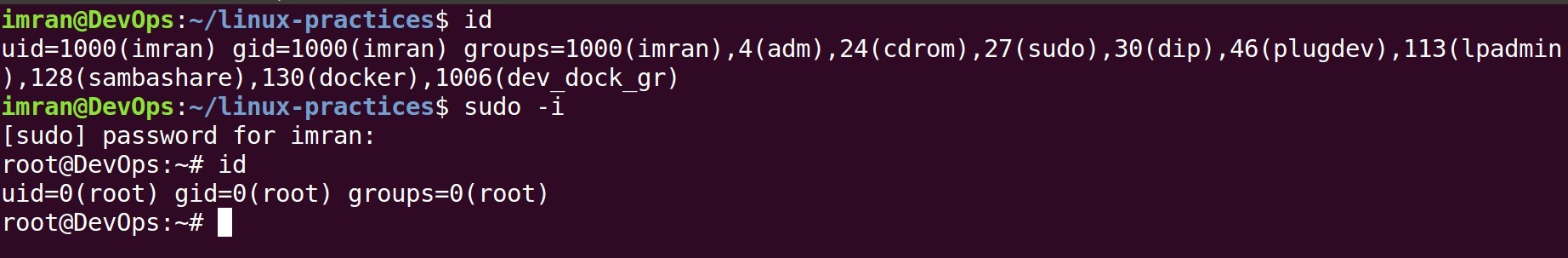


1. **Sudo**

sudo gives power to a normal user to execute commands which is owned by root user. Example shown below:

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

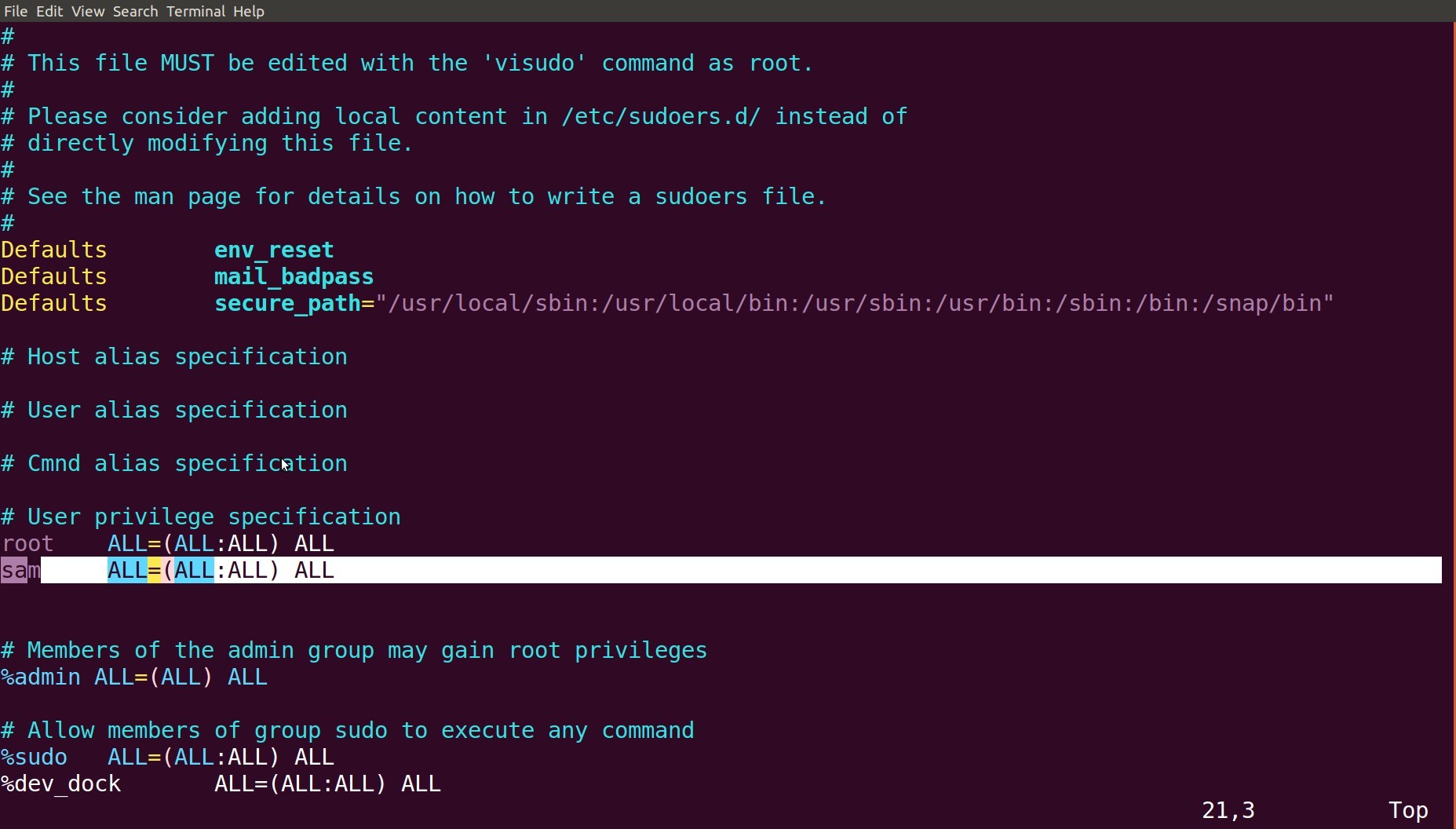
* **sudo -i changes from normal user to root user**



**Note: User imran was already a sudo user with full privilege.**

* **Adding user sam in sudoers list.**





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



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



* **Changing to any other user with “su -” command.**



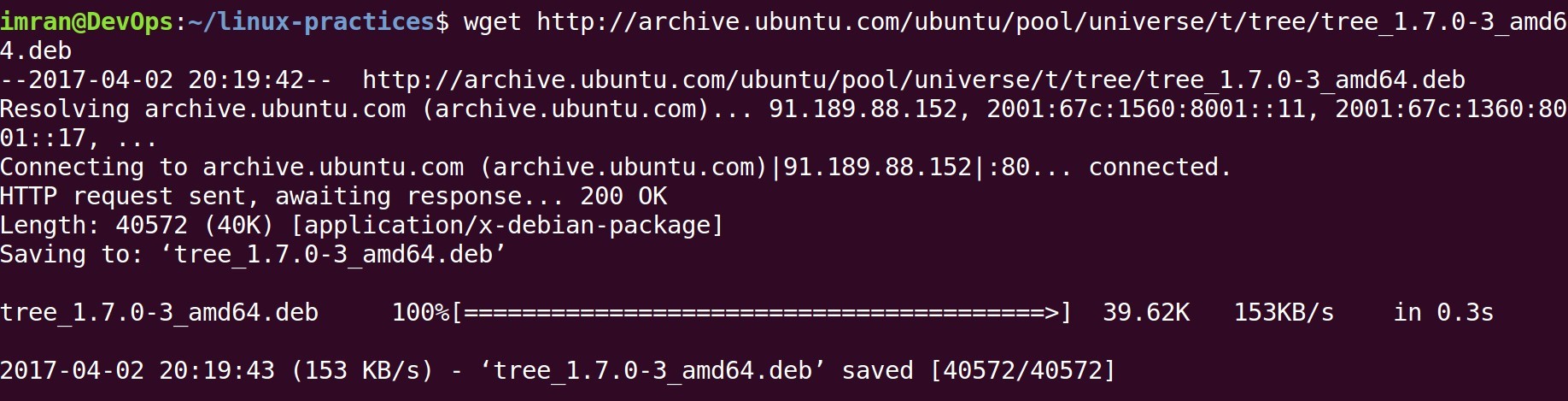
* **Become a root user from sam user login.**



# Software Management.

* **Download package from internet.**

# wget <http://archive.ubuntu.com/ubuntu/pool/universe/t/tree/tree_1.7.0-3_amd64.deb>



**REDHAT family(single package)**

$ sudo rpm -i pkgname.rpm

# Install rpm

based package (Installing, Uninstalling, Updating, Querying

,Verifying)

$ sudo rpm -e pkgname

# Remove package

**REDHAT family(Package with dependency from internet)**

$ sudo yum install package-name

#Install with dep

$ sudo yum remove package-name

# Remove package

**UBUNTU/DEBIAN**

$ sudo dpkg -i package-name

#Install deb pack

$ sudo dpkg -e package-name

#Remove deb pack

**UBUNTU/Debian(Package with dependency from internet)**

$ sudo apt-get install package-name

#Install with dep

$ sudo apt-get remove package-name

#Remove deb pack

**Install from source**

./configure make

make install (what it is)

1. **SEARCH**

$ grep pattern files

(you will this command often)

# Search for pattern in files

$ grep -r pattern dir pattern in dir

# Search recursively for

$ locate file

# Find all instances of file

$ find /home/tom -name 'index\*' with "index"(10 find examples)

# Find files names that start

$ find /home -size +10000k 10000k in /home

# Find files larger than

1. **LOGIN (SSH AND TELNET)**

$ ssh user@host

(secure data communication command)

# Connect to host as user

$ ssh -p port user@host specific port

# Connect to host using

$ telnet host using telnet port

# Connect to the system

1. **FILE TRANSFER**

**scp**

$ scp file.txt server2:/tmp

# Secure copy

file.txt to remote host /tmp folder

$ scp nixsavy@server2:/www/\*.html /www/tmp # Copy \*.html files from remote host to current system /www/tmp folder

$ scp -r nixsavy@server2:/www /www/tmp

# Copy all files

and folders recursively from remote server to the current system

/www/tmp folder

**rsync**

$ rsync -a /home/apps /backup/ source to destination

# Synchronize

$ rsync -avz /home/apps linoxide@192.168.10.1:/backup

#

Synchronize files/directories between the local and remote system with compression enabled

1. **DISK USAGE**

$ df -h

# Show free space on mounted

filesystems(commonly used command)

$ df -i filesystems

# Show free inodes on mounted

$ fdisk -l

types(fdisk command output)

# Show disks partitions sizes and

$ du -ah

# Display disk usage in human

readable form (command variations)

$ du -sh

current directory

# Display total disk usage on the

$ findmnt

# Displays target mount point for

all filesystem (refer type,list,evaluate output)

$ mount device-path mount-point # Mount a device

1. **DIRECTORY TRAVERSE**

$ cd .. # To go up one level of the directory tree(simple & most needed)

$ cd

# Go to $HOME directory

$ cd /test

# Change to /test directory

1. **SERVICES**

|  |  |  |
| --- | --- | --- |
|  | $ sudo service apache2 start # Starts apache2 on ubuntu  $ sudo service httpd start # Starts apache2 on Redhat  $ sudo service httpd stop  $ sudo service httpd restart # Restart services  $ sudo service httpd reload # Reload conf  $ chkconfig httpd on # starts httpd at boot time  $ chkconfig httpd off # stops httpd at boot time |  |
|  | | |
|  | | |

1. **COMPRESSION / ARCHIVES**

$ tar cf home.tar home # Create tar named home.tar containing home/ (11 tar examples)

$ tar xf file.tar file.tar

# Extract the files from

$ tar czf file.tar.gz compression

files

# Create a tar with gzip

$ gzip file

to file.gz (untar gzip file)

# Compress file and renames it

1. **PROCESS RELATED**

$ ps

# Display your currently active

processes (many parameters to learn)

$ ps aux | grep 'telnet' telnet process

# Find all process id related to

$ pmap

(kernel,user memory etc)

# Memory map of process

$ top

(30 examples)

# Display all running processes

$ kill pid

pid id (types of signals)

# Kill process with mentioned

$ killall proc

# Kill all processes named proc

$ pkill processname its name

# Send signal to a process with

$ bg

# Resumes suspended jobs without

bringing them to foreground (bg and fg command)

$ fg foreground

# Brings the most recent job to

$ fg n

# Brings job n to the foreground

1. **SYSTEM**

$ uname -a information

=> Display linux system

$ uname -r

=> Display kernel release

information (refer uname command in detail)

$ cat /etc/redhat\_release installed

=> Show which version of redhat

$ uptime

load (learn uptime command)

=> Show how long system running +

$ hostname

=> Show system host name

$ hostname -i

host (all options hostname)

=> Display the IP address of the

$ last reboot

(more examples last command)

=> Show system reboot history

$ date

(options of date command)

=> Show the current date and time

$ cal

more in cal)

=> Show this month calendar (what

$ w

more about w command)

=> Display who is online (learn

$ whoami

(example + sreenshots)

=> Who you are logged in as

$ finger user => Display information about user (many options of finger command)

1. **HARDWARE**

$ dmesg

=> Detected hardware and boot

messages (dmesg many more options)

$ cat /proc/cpuinfo

=> CPU model

$ cat /proc/meminfo

=> Hardware memory

$ cat /proc/interrupts per CPU per I/O device

=> Lists the number of interrupts

$ lshw

=> Displays information on

hardware configuration of the system

$ lsblk

=> Displays block device related

information in Linux (sudo yum install util-linux-ng)

$ free -m

MB) (free command in detail)

=> Used and free memory (-m for

$ lspci -tv

to find vendor ids)

=> Show PCI devices (very useful

$ lsusb -tv lsusb options)

=> Show USB devices (read more

$ lshal

their properties

=> Show a list of all devices with

$ dmidecode

BIOS (vendor details)

=> Show hardware info from the

$ hdparm -i /dev/sda

# Show info about disk sda

$ hdparm -tT /dev/sda

# Do a read speed test on disk sda

$ badblocks -s /dev/sda

# Test for unreadable blocks on

disk sda

1. **STATISTICS**

$ top => Display and update the top cpu processes (30 example options)

$ mpstat 1 => Display processors related statistics (learn mpstat command)

$ vmstat 2

=> Display virtual memory

statistics (very useful performance tool)

$ iostat 2

Intervals) (more examples)

=> Display I/O statistics (2sec

$ tail -n 500 /var/log/messages

=> Last 10 kernel/syslog

messages (everyday use tail options)

$ tcpdump -i eth1

=> Capture all packets flows on

interface eth1 (useful to sort network issue)

$ tcpdump -i eth0 'port 80'

80 ( HTTP )

=> Monitor all traffic on port

$ lsof

=> List all open files belonging

to all active processes.(sysadmin favorite command)

$ lsof -u testuser user

=> List files opened by specific

$ free -m usage command)

=> Show amount of RAM (daily

**$ watch df -h linux command)**

**=> Watch changeable data continuously(interesting**