

# 1. Linux Introduction

## What is Open Source?

- Open source: software and source code available to all
- The Free Software Foundation specifies four freedoms
  - The freedom to run the program for any purpose.
  - The freedom to study and modify the source code
  - The freedom to redistribute the program
  - The freedom to create derivative programs
- Many open-source licenses exist, each with different particulars

## Linux Origins

- 1984: The GNU Project and the Free Software Foundation
  - Creates open source version of UNIX utilities
  - Creates the General Public License (GPL)
    - Software license enforcing open source principles
- 1991: Linus Torvalds
  - Creates open source, UNIX-like kernel, released under the GPL
  - Ports some GNU utilities, solicits assistance online
- Today:
  - Linux kernel + GNU utilities = complete, open source, UNIX-like operating system
    - Packaged for targeted audiences as *distributions*

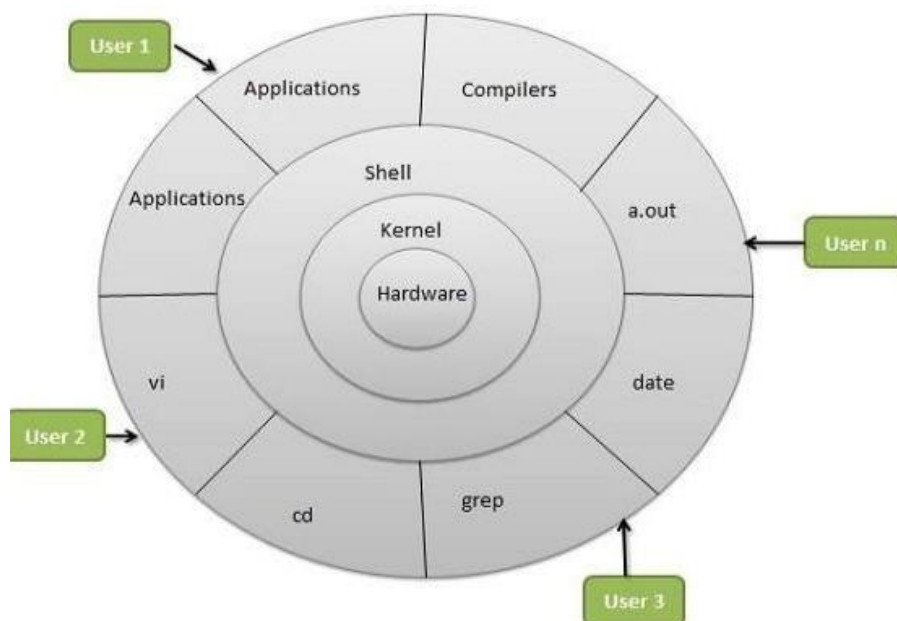
# Linux principles

- Everything is a file (including hardware)
- Small, single-purpose programs
- Ability to chain programs together to perform complex tasks
- Avoid captive user interfaces
- Configuration data stored in text

## 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



## Some Important Directories

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

## 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:** For more detailed information please refer to "DEVOPS METHODOLOGIES" by Praveenkumar Kundarapu.

## 2. Basic Commands

→ Open Terminal

→ Know where you are? Present Working Directory

```
ubuntu@ip-172-31-30-133:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-30-133:~$
```

→ Create a directory/folder in your home directory.

```
ubuntu@ip-172-31-30-133:~$ mkdir linux
ubuntu@ip-172-31-30-133:~$
```

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

```
ubuntu@ip-172-31-30-133:~$ cd linux
ubuntu@ip-172-31-30-133:~/linux$
```

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

```
ubuntu@ip-172-31-30-133:~/linux$ mkdir vmdir
ubuntu@ip-172-31-30-133:~/linux$ mkdir testdir
ubuntu@ip-172-31-30-133:~/linux$ mkdir devopsdir
ubuntu@ip-172-31-30-133:~/linux$ ls
devopsdir  testdir  vmdir
ubuntu@ip-172-31-30-133:~/linux$
```

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

```
ubuntu@ip-172-31-30-133:~/linux$ touch f1 f2 f3 f4
ubuntu@ip-172-31-30-133:~/linux$ ls
devopsdir  f1  f2  f3  f4  testdir  vmdir
ubuntu@ip-172-31-30-133:~/linux$
```

→ **Reconfirm your location in your system.**

```
ubuntu@ip-172-31-30-133:~/linux$ pwd
/home/ubuntu/linux
ubuntu@ip-172-31-30-133:~/linux$ ls
devopsdir  f1  f2  f3  f4  testdir  vmdir
ubuntu@ip-172-31-30-133:~/linux$
```

**NOTE:** For more detailed information please refer to “DEVOPS METHODOLOGIES” by Praveenkumar Kundarapu.

# 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/Ubuntu/linux/**

**/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/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.

```
ubuntu@ip-172-31-30-133:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-30-133:~$ cd linux/
ubuntu@ip-172-31-30-133:~/linux$ ls
devopsdir  f1  f2  f3  f4  testdir  vpsdir
ubuntu@ip-172-31-30-133:~/linux$ pwd
/home/ubuntu/linux
ubuntu@ip-172-31-30-133:~/linux$ cd devopsdir/
ubuntu@ip-172-31-30-133:~/linux/devopsdir$ pwd
/home/ubuntu/linux/devopsdir
ubuntu@ip-172-31-30-133:~/linux/devopsdir$
```

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

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

```
ubuntu@ip-172-31-30-133:~/linux$ ls
devopsdir f1 f2 f3 f4 testdir vpdire
ubuntu@ip-172-31-30-133:~/linux$ mkdir devopsdir/ansible
ubuntu@ip-172-31-30-133:~/linux$ mkdir /home/ubuntu/linux/devopsdir/aws
ubuntu@ip-172-31-30-133:~/linux$ ls devopsdir/
ansible aws
ubuntu@ip-172-31-30-133:~/linux$
```

→ **Copying files into directory.**

```
ubuntu@ip-172-31-30-133:~/linux$ pwd
/home/ubuntu/linux
ubuntu@ip-172-31-30-133:~/linux$ ls
devopsdir f1 f2 f3 f4 testdir vpdire
ubuntu@ip-172-31-30-133:~/linux$ cp f1 testdir/
ubuntu@ip-172-31-30-133:~/linux$ cd testdir/
ubuntu@ip-172-31-30-133:~/linux/testdir$ ls
f1
ubuntu@ip-172-31-30-133:~/linux/testdir$
```

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

```
ubuntu@ip-172-31-30-133:~/linux/testdir$ cd ../
ubuntu@ip-172-31-30-133:~/linux$ cd
ubuntu@ip-172-31-30-133:~$ cd -
/home/ubuntu/linux
ubuntu@ip-172-31-30-133:~/linux$ pwd
/home/ubuntu/linux
ubuntu@ip-172-31-30-133:~/linux$ ls
devopsdir f1 f2 f3 f4 testdir vpdire
ubuntu@ip-172-31-30-133:~/linux$ cp -rvfp testdir/ vpdire/
'testdir/' -> 'vpdire/testdir'
'testdir/f1' -> 'vpdire/testdir/f1'
ubuntu@ip-172-31-30-133:~/linux$ ls vpdire/
testdire
ubuntu@ip-172-31-30-133:~/linux$
```



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

```
ubuntu@ip-172-31-30-133:~/linux$ pwd
/home/ubuntu/linux
ubuntu@ip-172-31-30-133:~/linux$ ls
devopsdir f1 f2 f3 f4 testdir vpdire
ubuntu@ip-172-31-30-133:~/linux$ mv devopsdir/ vpdire/
ubuntu@ip-172-31-30-133:~/linux$ ls
f1 f2 f3 f4 testdir vpdire
ubuntu@ip-172-31-30-133:~/linux$ ls vpdire/
devopsdir testdire
ubuntu@ip-172-31-30-133:~/linux$
ubuntu@ip-172-31-30-133:~/linux$ mv f3 f4 vpdire/
ubuntu@ip-172-31-30-133:~/linux$ ls
f1 f2 testdire vpdire
ubuntu@ip-172-31-30-133:~/linux$
```

→ **Removing files and directories.**

```
ubuntu@ip-172-31-30-133:~/linux$ rm f1
ubuntu@ip-172-31-30-133:~/linux$ ls
f2 testdire vpdire
ubuntu@ip-172-31-30-133:~/linux$ rm -rf testdire/
ubuntu@ip-172-31-30-133:~/linux$ ls
f2 vpdire
ubuntu@ip-172-31-30-133:~/linux$ mkdir praveen
ubuntu@ip-172-31-30-133:~/linux$ ls
f2 praveen vpdire
ubuntu@ip-172-31-30-133:~/linux$ rm -rf
ubuntu@ip-172-31-30-133:~/linux$ rm -rf praveen/
ubuntu@ip-172-31-30-133:~/linux$ ls
f2 vpdire
ubuntu@ip-172-31-30-133:~/linux$
```

**NOTE:** For more detailed information please refer to “DEVOPS METHODOLOGIES” by Praveenkumar Kundarapu.

### 3. VIM EDITOR

#### → Install vim editor

```
ubuntu@ip-172-31-30-133:~/linux$ sudo apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
vim is already the newest version (2:7.4.1689-3ubuntu1.2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-30-133:~/linux$
```

#### → Open up a file in vim editor

```
ubuntu@ip-172-31-30-133:~/linux$ vim firstfile.txt
```

#### → Hit i to enter into insert mode



#### => type few lines => hit Esc



=> type :wq

ubuntu@ip-172-31-30-133: ~/linux

```
third line  
and  
fourth line
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
:wq
```

=> Enter.

→ Read file with cat command.

```
ubuntu@ip-172-31-30-133:~/linux$ cat firstfile.txt  
my name is praveenkumar  
devops engineer  
third line  
and  
fourth line  
ubuntu@ip-172-31-30-133:~/linux$
```

## 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.

### Command Mode:

gg	To go to the beginning of the page
G	To go to end of the page
w	To move the cursor forward, word by word
b	To move the cursor backward, word by word
nw	To move the cursor forward to n words (5W)
nb	To move the cursor backward to n words (5B)
u	To undo last change (word)

U	To undo the previous changes (entire line)
Ctrl+R	To redo the changes
yy	To copy a line
nyy	To copy n lines (5yy or 4yy)
p	To paste line below the cursor position
P	To paste line above the cursor position
dw	To delete the word letter by letter (like Backspace)
x	To delete the word letter by letter (like DEL Key)
dd	To delete entire line
ndd	To delete n no. of lines from cursor position(5dd)
/	To search a word in the file

### **Extended Mode: ( Colon Mode)**

Extended Mode is used for save and quit or save without quit using "Esc" Key with ":"

Esc+:w	To Save the changes
Esc+:q	To quit (Without saving)
Esc+:wq	To save and quit
Esc+:w!	To save forcefully
Esc+wq!	To save and quit forcefully
Esc+:x	To save and quit
Esc+:X	To give password to the file and remove password
Esc+:20(n)	To go to line no 20 or n
Esc+: se nu	To set the line numbers to the file
Esc+:se nonu	To Remove the set line numbers

## ls syntax

```
$ ls [options] [file|dir]
```

## ls command options

ls command main options:

option	description
ls -a	list all files including hidden file starting with '.'
ls --color	colored list [=always/never/auto]
ls -d	list directories - with '*'
ls -F	add one char of */=>@  to enteries
ls -i	list file's inode index number
ls -l	list with long format - show permissions
ls -la	list long format including hidden files
ls -lh	list long format with readable file size
ls -ls	list with long format with file size
ls -r	list in reverse order
ls -R	list recursively directory tree
ls -s	list file size
ls -S	sort by file size
ls -t	sort by time & date
ls -X	sort by extension name

# Types of files 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

## 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

```
ubuntu@ip-172-31-30-133:~/linux$ ls
f2  firstfile.txt  vpdир
ubuntu@ip-172-31-30-133:~/linux$ ls /var/log/
amazon  btmp          dist-upgrade  kern.log  syslog
apt     cloud-init.log  dpkg.log      lastlog   unattended-upgrades
auth.log cloud-init-output.log  fsck        lxd       wtmp
ubuntu@ip-172-31-30-133:~/linux$ ln -s /var/log/ logdir
ubuntu@ip-172-31-30-133:~/linux$ ls -l
total 8
-rw-rw-r-- 1 ubuntu ubuntu    0 Oct 23 21:35 f2
-rw-rw-r-- 1 ubuntu ubuntu   67 Oct 23 22:10 firstfile.txt
lrwxrwxrwx 1 ubuntu ubuntu    9 Oct 23 22:23 logdir -> /var/log/
drwxrwxr-x 4 ubuntu ubuntu 4096 Oct 23 21:57 vpdир
ubuntu@ip-172-31-30-133:~/linux$ ls logdir
amazon  apt  auth.log  btmp  cloud-init.log  cloud-init-output.log  dist-upgrade  dpkg.log  fsck  kern.log  lastlog  lxd  syslog  unattended-upgrades  wtmp
ubuntu@ip-172-31-30-133:~/linux$
```

## 4. Filter & IO redirection command.

### Grep

grep command is used to find texts from any text input. Passwd

file: stores information about all the users in the system

```
ubuntu@ip-172-31-30-133:~/linux$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
```

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

```
ubuntu@ip-172-31-30-133:~/linux$ grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
ubuntu@ip-172-31-30-133:~/linux$
```

→ Linux is case sensitive, Root is different than root. Ignoring case in grep with -i option.

```
ubuntu@ip-172-31-30-133:~/linux$ grep Root /etc/passwd
ubuntu@ip-172-31-30-133:~/linux$ grep -i Root /etc/passwd
root:x:0:0:root:/root:/bin/bash
ubuntu@ip-172-31-30-133:~/linux$
```

→ 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



**Note:** -press **Enter** key to scroll down line by line (or)

Use **d** to go to next page

Use **b** to go to previous page

Use **/** to search for a word in the file

Use **v** to go vi mode where you can edit the file and once you save it you will back to less command

- **more**

**more** is exactly same like **less**

**Ex:** #more /etc/passwd

**Note:** -press **Enter** key to scroll down line by line (or)

Use **d** to go to next page

Use **/** to search for a word in the file

Use **v** to go vi mode where you can edit the file and once you save it you will back to more command

- **head**

It is used to display the top **10 lines** of the file.

**Ex:** # head /etc/passwd

```
[root@ktlinux ~]# head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
```

- **tail**

It is used to display the **last 10 lines** of the file

#tail /etc/passwd

```
[root@ktlinux ~]# tail /etc/passwd
apache:x:48:48:Apache:/var/www:/sbin/nologin
nslcd:x:65:55:LDAP Client User:/:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
pulse:x:496:494:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
visitor:x:500:500:visitor:/home/visitor:/bin/bash
ktuser:x:501:501:/home/ktuser:/bin/bash
```

- **cut**

# **cut -d -f filename** (where d stands for delimiter ex. :, " " etc and f stands for field)

```
[root@ktlinux ~]# cut -d: -f1 /etc/passwd
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
uucp
```

To delimit spaces and print the field

#**cut -d " " -f1 filename**

- **sed**

**sed** stands for **stream editor**, which is used to search a word in the file and replace it with the word required to be in the output

**Note:** it will only modify the output, but there will be no change in the original file.

#**sed 's/searchfor/replacewith/g' filename**

```
[root@ktlinux ~]# cat ktfile
Welcome to Kernel Tech
[root@ktlinux ~]# sed 's/Tech/Technologies/g' ktfile
Welcome to Kernel Technologies
[root@ktlinux ~]# cat ktfile
Welcome to Kernel Tech
```

## I/O redirection

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 named devopstools with below content.**

```
ubuntu@ip-172-31-30-133:~/linux$ cat >> devopstools
chef tech
ansible tech
git tech
docker tech
aws tech
```

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

```
ubuntu@ip-172-31-30-133:~/linux$ sed 's/tech/tool/g' devopstools
chef tool
ansible tool
git tool
docker tool
aws tool
ubuntu@ip-172-31-30-133:~/linux$ sed 's/tech/tool/g' devopstools > newtools.txt
ubuntu@ip-172-31-30-133:~/linux$ cat newtools.txt
chef tool
ansible tool
git tool
docker tool
aws tool
ubuntu@ip-172-31-30-133:~/linux$
```

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

```
ubuntu@ip-172-31-30-133:~/linux$ tail /etc/passwd >> newtools.txt
ubuntu@ip-172-31-30-133:~/linux$ cat newtools.txt
chef tool
ansible tool
git tool
docker tool
aws tool
syslog:x:104:108::/home/syslog:/bin/false
apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd:/bin/false
messagebus:x:107:111::/var/run/dbus:/bin/false
uidd:x:108:112::/run/uidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
pollinate:x:111:1::/var/cache/pollinate:/bin/false
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
praveen:x:1001:1001:::/home/praveen:/bin/bash
ubuntu@ip-172-31-30-133:~/linux$
```

## 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.

```
ubuntu@ip-172-31-30-133:~/linux$ ls
devopstools f2 firstfile.txt logdir newtools.txt vpdire
ubuntu@ip-172-31-30-133:~/linux$ ls | head -3
devopstools
f2
firstfile.txt
ubuntu@ip-172-31-30-133:~/linux$ ls | grep logdir
logdir
ubuntu@ip-172-31-30-133:~/linux$ cat /etc/passwd | grep root
root:x:0:0:root:/root:/bin/bash
ubuntu@ip-172-31-30-133:~/linux$
```

## Find

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

```
ubuntu@ip-172-31-30-133:~/linux$ find /home/ubuntu/ -name newtools.txt
/home/ubuntu/linux/newtools.txt
ubuntu@ip-172-31-30-133:~/linux$
```

Options that can be used with find command:

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

## 5. Users & 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 user ID number ( the UID)
- Users name and UID are stored in **/etc/passwd**
- User's password is stored in **/etc/shadow** in encrypted form.
- Users are assigned a **home directory** and a program that is run when they login (**Usually a shell**)
- Users cannot read, write or execute each other's files without permission.

#### Types of users In Linux and their attributes:

TYPE	EXAMPLE	USER ID (UID)	GROUP ID (GID)	HOME DIRECTORY	SHELL
Super User	Root	0	0	/root	/bin/bash
System User	ftp, ssh, apache nobody	1 to 499	1 to 499	/var/ftp , etc	/sbin/nologin
Normal User	Visitor, ktuser,etc	500 to 60000	500 to 60000	/home/user name	/bin/bash

## In Linux there are three types of users.

### 1. Super user or root user

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

### 2. System user

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

### 3. Normal user

Normal users are the users created by root user. They are normal users like Rahul, Musab etc. Only the root user has the permission to create or remove a user.

## Whenever a user is created in Linux things created by default:-

- 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

```
[root@ktlinux ~]# head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

The above fields are

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

\$ 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 (force add existing group)	=> Add group "admin"



\$ 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)

```
ubuntu@ip-172-31-30-133:~/linux$ sudo groupadd deck
ubuntu@ip-172-31-30-133:~/linux$ sudo usermod -G deck sam
```

\$ sudo passwd sam => Changing password for other user

\$ passwd => Changing your own password.

\$ sudo userdel -r sam => Delete user sam  
(force,file removal)

\$ sudo groupdel deck => Deletes a group

## 6.File permissions

### Viewing Permissions from the Command-Line

- File permissions may be viewed using **ls -l**

```
$ ls -l /bin/login
-rwxr-xr-x 1 root root 19080 Apr 1 18:26 /bin/login
```

- Four symbols are used when displaying permissions:
  - r: permission to read a file or list a directory's contents
  - w: permission to write to a file or create and remove files from a directory
  - x: permission to execute a program or change into a directory and do a long listing of the directory
  - -: no permission (in place of the r, w, or x)

### Changing File Ownership

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

```

ubuntu@ip-172-31-28-92:~$ sudo -i
root@ip-172-31-28-92:~# adduser india ✓
Adding user `india' ...
Adding new group `india' (1002) ...
Adding new user `india' (1002) with group `india' ...
Creating home directory `/home/india' ...
Copying files from `/etc/skel' ...
Enter new UNIX password: ✓
Retype new UNIX password: ✓
passwd: password updated successfully
Changing the user information for india
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] y ✓
root@ip-172-31-28-92:~# █

```

## Changing Permissions - Symbolic Method

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

## 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**



```

ubuntu@ip-172-31-30-133:~/linux$ ls -l
total 16
-rw-rw-r-- 1 ubuntu ubuntu 53 Oct 23 22:45 devopstools
-rw-rw-r-- 1 ubuntu ubuntu 0 Oct 23 21:35 f2
-rw-rw-r-- 1 ubuntu ubuntu 67 Oct 23 22:10 firstfile.txt
lrwxrwxrwx 1 ubuntu ubuntu 9 Oct 23 22:23 logdir -> /var/log/
-rw-rw-r-- 1 ubuntu ubuntu 519 Oct 23 22:58 newtools.txt
drwxrwxr-x 4 ubuntu ubuntu 4096 Oct 23 21:57 vpdire
ubuntu@ip-172-31-30-133:~/linux$ chmod u+x newtools.txt
ubuntu@ip-172-31-30-133:~/linux$ ls -l newtools.txt
-rwxrw-r-- 1 ubuntu ubuntu 519 Oct 23 22:58 newtools.txt
ubuntu@ip-172-31-30-133:~/linux$ chmod o-r newtools.txt
ubuntu@ip-172-31-30-133:~/linux$ ls -l newtools.txt
-rwxrw---- 1 ubuntu ubuntu 519 Oct 23 22:58 newtools.txt
ubuntu@ip-172-31-30-133:~/linux$ chmod 700 newtools.txt
ubuntu@ip-172-31-30-133:~/linux$ ls -l newtools.txt
-rwx----- 1 ubuntu ubuntu 519 Oct 23 22:58 newtools.txt
ubuntu@ip-172-31-30-133:~/linux$ chmod 755 newtools.txt
ubuntu@ip-172-31-30-133:~/linux$ ls -l newtools.txt
-rwxr-xr-x 1 ubuntu ubuntu 519 Oct 23 22:58 newtools.txt
ubuntu@ip-172-31-30-133:~/linux$ █

```

## 7. 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

```

ubuntu@ip-172-31-30-133:~/linux$ id
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),109(netdev),110(lxd)
ubuntu@ip-172-31-30-133:~/linux$ sudo -i █
[sudo] password for █
root@DevOps:~# id
uid=0(root) gid=0(root) groups=0(root)
→ root@DevOps:~# █
→ root@ip-172-31-30-133:~# █

```

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

→ **Adding user india in sudoers list.**

```
root@ip-172-31-28-92:~# export EDITOR=vim
root@ip-172-31-28-92:~# visudo

# User privilege specification
root    ALL=(ALL:ALL) ALL
kumar   ALL=(ALL:ALL) NOPASSWD: ALL
india   ALL=(ALL:ALL) NOPASSWD: ALL

# Change to no to disable tunnelled clear text passwords
PasswordAuthentication yes
```

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

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

→ 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
kumar   ALL=(ALL:ALL) NOPASSWD: ALL
india   ALL=(ALL:ALL) NOPASSWD: ALL
```

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

```
ubuntu@ip-172-31-30-133:~$ su praveen
Password:
praveen@ip-172-31-30-133:/home/ubuntu$
```

→ Become a root user from praveen user login.

```
ubuntu@ip-172-31-30-133:~$ su praveen
Password:
praveen@ip-172-31-30-133:/home/ubuntu$
```

## 8. Software Management.

→ Download package from internet.

# wget [http://archive.ubuntu.com/ubuntu/pool/universe/t/tree/tree\\_1.7.0-3\\_amd64.deb](http://archive.ubuntu.com/ubuntu/pool/universe/t/tree/tree_1.7.0-3_amd64.deb)

```
ubuntu@ip-172-31-30-133:~/linux$ #wget http://archive.ubuntu.com/ubuntu/pool/universe/t/tree/tree_1.7.0-3_amd64.deb
ubuntu@ip-172-31-30-133:~/linux$
```

#

**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)

```

## 9. SEARCH

```

$ grep pattern files                                # Search for pattern in files
(you will this command often)

$ grep -r pattern dir                               # Search recursively for
pattern in dir

$ locate file                                         # Find all instances of file

$ find /home/tom -name 'index*'                      # Find files names that start
with "index"(10 find examples)

$ find /home -size +10000k                            # Find files larger than
10000k in /home

```

## 10. LOGIN (SSH AND TELNET)

```

$ ssh user@host                                     # Connect to host as user
(secure data communication command)

$ ssh -p port user@host                             # Connect to host using
specific port

```

```
$ telnet host                                # Connect to the system
using telnet port
```

## 11. 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/               # Synchronize
source to destination

$ rsync -avz /home/apps linuxide@192.168.10.1:/backup #
Synchronize files/directories between the local and remote system
with compression enabled
```

## 12. DISK USAGE

```
$ df -h                                     # Show free space on mounted
filesystems (commonly used command)

$ df -i                                     # Show free inodes on mounted
filesystems

$ fdisk -l                                  # Show disks partitions sizes and
types (fdisk command output)

$ du -ah                                    # Display disk usage in human
readable form (command variations)

$ du -sh                                    # Display total disk usage on the
current directory

$ findmnt                                   # Displays target mount point for
all filesystem (refer type,list,evaluate output)

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

## 13. 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
```

## 14. 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
```

## 15. COMPRESSION / ARCHIVES

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

$ tar xf file.tar # Extract the files from
file.tar

$ tar czf file.tar.gz files # Create a tar with gzip
compression

$ gzip file # Compress file and renames it
to file.gz (untar gzip file)
```

## 16. PROCESS RELATED

```
$ ps # Display your currently active
processes (many parameters to learn)
```

\$ ps aux   grep 'telnet'	# Find all process id related to telnet process
\$ 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 bringing them to foreground (bg and fg command)	# Resumes suspended jobs without
\$ fg foreground	# Brings the most recent job to
\$ fg n	# Brings job n to the foreground

## 17. SYSTEM

\$ uname -a information	=> Display linux system
\$ uname -r information (refer uname command in detail)	=> Display kernel release
\$ 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 + screenshots)	=> Who you are logged in as
\$ finger user (many options of finger command)	=> Display information about user

## 18. HARDWARE

\$ dmesg messages (dmesg many more options)	=>Detected hardware and boot
\$ 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 hardware configuration of the system	=>Displays information on
\$ lsblk information in Linux (sudo yum install util-linux-ng)	=>Displays block device related
\$ 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
```

## 19. 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                          => Display I/O statistics (2sec
Intervals) (more examples)

$ 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'         => Monitor all traffic on port
80 ( HTTP )

$ lsof                              => List all open files belonging
to all active processes.(sysadmin favorite command)

$ lsof -u testuser                  => List files opened by specific
User

$ free -m                           => Show amount of RAM (daily
usage command)

$ watch df -h                      => Watch changeable data continuously(interesting
linux command)
```

**NOTE: For more detailed information please refer to “DEVOPS METHODOLOGIES” by Praveenkumar Kundarapu**

