

# Linux Basics Handbook

KUSHAL PRASAD JOSHI

## TABLE OF CONTENTS

Introduction .....	3
Users in Linux.....	3
Text Editors .....	3
Vim .....	3
Basic File System .....	3
Paths in Linux.....	3
Linux Terminal.....	5
Opening Terminal.....	5
Using a Keyboard Shortcut:.....	5
Through the Applications Menu: .....	5
Using the Run Dialog: .....	5
Terminal Shortcuts .....	6
Basic Linux Commands .....	7
Introducing Commands .....	7
Working With Directories .....	7
Working With Files.....	8
Chmod Calculator .....	9
Executing Software.....	9
Process Manipulation .....	9
Changing Hostname.....	10
Changing Domain Name.....	10
Configuring Apache Server.....	10
Changing Apache Port .....	10
Software Installation Without Apt .....	11
From Debian Files .....	11
From GitHub .....	11
Error Resolving in Linux Apt .....	12
Root Access .....	12
Check Your Connection.....	12
Editing Source List.....	12
Using Fix-Broken .....	12
Removing Apt List.....	12

Running Multiple Commands in Single Terminal .....	13
Semicolon (;).....	13
And (&&) .....	13
Or (  ) .....	13

## INTRODUCTION

### USERS IN LINUX

1. Regular user (Home directory)
2. Root user (Full access) [ADMIN] (known as superuser of system)
3. Service user (Service access) (known as sever user)

#### NOTE:

- If a regular user wants root access, he can use '**sudo**' before any command to use root power.
- To get full access to root he can use '**sudo su**' command.

### TEXT EDITORS

#### VIM

Vim is a text editor software in Linux.

Some commands used in it are:

- **I** : insert
- **Esc** : exit insert mode
- **:wq** : exit with saving
- **:q!** : exit without saving

### BASIC FILE SYSTEM

<b>/bin</b>	→ Basic programs (ls, cd, mv, etc.)
<b>/sbin</b>	→ System programs (fdisk, sysctl, mkfs, etc.)
<b>/etc</b>	→ Configuration files (default values)
<b>/temp</b>	→ Temporary files
<b>/usr/bin</b>	→ Applications (apt, nmap, etc.)
<b>/usr/share</b>	→ Personal directories of users
<b>/root</b>	→ Home directory of super user [ADMIN]

### PATHS IN LINUX

Below is an example of a directory structure:

```
Desktop
├── IntelliJ idea.desktop
```

```
├─ Joshi
├─ Kushal
|   └─ Birthday
|       └─ Happy
|           └─ hero.txt
├─ Prasad
|   └─ Hello
|       └─ Sir
├─ firefox-esr.desktop
└─ libreoffice-startcenter.desktop
```

**NOTE:**

- **gedit /Desktop/Kushal/Happy/hero.txt** is an absolute path. (You can access absolute paths from anywhere.)
- **gedit hero.txt** is a relative path. (You need to be inside Happy folder i.e. /Desktop/Kushal/Happy to access relative path.)

## LINUX TERMINAL

The terminal is a command-line interface that allows you to interact with your Linux system using text-based commands.

### OPENING TERMINAL

Here are some common methods to open the terminal:

---

#### USING A KEYBOARD SHORTCUT:

- On many Linux distributions (especially those using GNOME), you can press **Ctrl + Alt + T** to launch the terminal quickly.

---

#### THROUGH THE APPLICATIONS MENU:

- Open the **Applications** or **Activities** menu.
- Search for **Terminal** (or the specific terminal emulator like *GNOME Terminal* or *Console*).
- Click on the terminal application to open it.

---

#### USING THE RUN DIALOG:

- Press **Alt + F2** to open the Run Command dialog.
- Type `gnome-terminal` (or `console` or another terminal command, depending on your environment) and press Enter.

Once the terminal is open, you can execute commands to manage your system, install software, navigate directories, and much more. The terminal is a powerful tool that can greatly enhance your efficiency on Linux.

### TERMINAL SHORTCUTS

<b>Tab</b>	→ Autocomplete names
<b>Double Tab</b>	→ For selection
<b>Ctrl + C</b>	→ Abort process manually
<b>Ctrl + L</b>	→ Same as clear command
<b>Ctrl + D</b>	→ Same as exit command
<b>Ctrl + Alt + D</b>	→ Minimize all terminals/commands (same for reverting)
<b>Ctrl + U</b>	→ Clear line
<b>Ctrl + Z</b>	→ Suspend for background (Programs run on background.)

**Ctrl + A** → Position starting point

**Ctrl + E** → Position ending point

**Shift + Ctrl + C** → Copy in terminal

**Shift + Ctrl + V** → Paste in terminal

#### TERMINAL SHORTCUTS

**Ctrl + U** → clear line

→ suspend for background (Programs run on background.)

## BASIC LINUX COMMANDS

### INTRODUCING COMMANDS

1. **help** : Shows you basic commands and their uses.
2. **man** : Shows you complete manual of that command or program.
3. **ls** : List all the folders and files of a directory.
4. **ls -a** : Shows all files and folders.
5. **ls -l** : Shows permissions of file & DOB & user / groups.
6. **ls -R** : List all directories and subdirectories. (Recursive)
7. **cd** : Change directory. (Enter into directory / node)
8. **pwd** : Present working directory.
9. **clear** : Clean the terminal.
10. **history** : Shows history of commands.
11. **echo** : To print on terminal.
12. **printf** : To print on terminal.
13. **mkdir** : make a new directory in current location.

#### NOTE:

- To make any file hidden use . before file name. (e.g. **.file**)
- **ls /home/kali/Downloads** can be used.
- **cd /home/kali/Downloads** can be used.
- **-h** gives help.
- **cd ..** is used for backspace in directories.
- ' ' for character and " " for string is recommended.
- **mkdir Kushal Prasad Joshi** forms three different directories.
- To make a single directory use **mkdir "Kushal Prasad Joshi"** (double quote) or **mkdir Kushal\ Prasad\ Joshi** (escape character).

### WORKING WITH DIRECTORIES



1. **dir** : Same as ls.
2. **mkdir** : Create a directory (folder).
3. **cp** : Copy a file or folder.
4. **mv** : Move a file or folder.
5. **rm** : Remove. (Delete a file or folder.)

### NOTE:

- `mkdir /home/.....`
- `cp filename /home/.....`
- `mv filename /home/.....`
- `rm filename /home/.....`
- `rm filename`
- `rm -r folder`
- `rm /home/.....`

## WORKING WITH FILES

1. **sudo su root** : Grant root privileges.
2. **cat** : Shows content of a file.
3. **nano** : Linux file editor.
4. **gedit** : File editor software.
5. **chmod** : change directories and files permissions.

### NOTE:

- `cat /home/.....`
- Nano is command line text editor.
- Gedit is graphical text editor.
- **ls -l** gives detailed information about files and folders.
- **chmod +wxr filename** is used to add permissions.
- **chmod -wxr filename** is used to remove permissions.

- **sudo su** can be used instead of **sudo su root** in newer versions of Linux.

---

## CHMOD CALCULATOR

- **chmod filename ch-number** (Changes file permissions)
- **chgroup** (Changes group)

### NOTE:

- The format is (Owner Group Public).

## EXECUTING SOFTWARE

1. **./filename** : Execute shell file.
2. **bash filename** : Execute shell programs.
3. **apt-get update** : Update packages list. (URL's)
4. **apt-get upgrade** : Update all installed software.
5. **apt-get install software\_name** : Install a particular software.
6. **apt-get update software\_name** : Update packages for a particular software.
7. **apt-get upgrade software\_name** : Update a particular software.

### NOTE:

- **apt-get update** updates the store.
- **apt-get upgrade** updates all the packages, programs, tools, etc.
- In the newer version of Linux, **apt** can be used instead of **apt-get**.

## PROCESS MANIPULATION

1. **top** : Showing Linux processes consuming more resources.
2. **ps** : print present processes.
3. **ps -a** : print all processes including background processes.
4. **kill** : Terminate process manually.
5. **who** : Who are logged on and what they are doing.
6. **whoami** : Displays the username of current user.

7. **touch** : Create an empty file.

#### NOTE:

- shortcuts in the tab (process) shown are: PID - process id; PR - priority; NI - priority number; VIRT - virtual; RES - resources used; SHR - shared memory; S - software status.
- **kill PID** (Terminate process with the PID given to kill.)
- **Ctrl + C** is used to stop ongoing process.

#### CHANGING HOSTNAME

- **sudo su** (Getting root access)
- **cd /etc** (Contains all software configuration)
- **gedit hostname** (Editing hostname file using gedit text editor)
- **reboot** (Restarting system)

#### CHANGING DOMAIN NAME

- **sudo su** (Getting root access)
- **cd /etc** (Contains all types of configurations)
- **gedit hosts** (Editing hosts file using gedit text editor)
- **service apache2 start** (Opening ports on Apache server)
- Now, go to browser and search for an Ip address and domain name. (e.g. Kushal:80)

#### CONFIGURING APACHE SERVER

- **service apache2 start** (Starting Apache server)
- **cd /var/www/html/** (Pages that server Apache is executing)
- **sudo gedit index.html** (Editing the html page for confirmation)

#### NOTE:

- **ifconfig** (Configure private ip) e.g. inet 192.168.78.141 (Using this Ip local area network can be connected.)
- Apache server always works on port 80 by default.

#### CHANGING APACHE PORT

- **cd /etc** (/etc contains all types of configuration files)

- **cd /apache2** (Entering apache2 folder)
- **gedit ports.conf** (Editing ports.conf file using gedit. Here, change the 80 into any number that will be a new port. e.g. 8080)
- Run **service apache2 restart** to restart your Apache server.
- Now go to browser and check your Ip. (192.168.78.141:80 in my case.)

## SOFTWARE INSTALLATION WITHOUT APT

---

### FROM DEBIAN FILES

- **cd Downloads** (Downloads folder contains all the downloaded software.)
- **dpkg -i filename** (Unpacking and installing Debian files.)

---

### FROM GITHUB

- **git clone URL**
- **cd folder**
- **./exe file**

### NOTE:

- Use the instructions given by GitHub to install programs and services.

## ERROR RESOLVING IN LINUX APT

### ROOT ACCESS

You need root access to download any software or service using apt command. So, confirm that you have root access. If you don't have root access, use **sudo su** command to get root access.

### CHECK YOUR CONNECTION

You need an internet connection to download any software or services using apt. So, confirm that you are connected to internet. If not, connect to the internet through available network.

### EDITING SOURCE LIST

- **cd /etc/apt**
- **gedit sources.list** (Confirm that line2 and line5 doesn't contain # sign because # indicates that line is commented.)
- **apt update** (Updating source list)

### USING FIX-BROKEN

- **apt-get install --fix-broken**

### REMOVING APT LIST

- **rm -rf /var/lib/apt/list/\***
- **apt-get update**

## RUNNING MULTIPLE COMMANDS IN SINGLE TERMINAL

### SEMICOLON (;)

Second command must work whether first command work or not. Second command is independent of first command and always works.

For example: **cd ; ls** (ls will work whether cd works or not.)

### AND (&&)

Runs first command first and second command second. If the first command fails, then the second command doesn't work.

For example: **cd && ls** (ls will work only if cd works.)

### OR (||)

Runs first command if it is true, else runs second command. The second command only runs if first command fails.

For example: **cd || ls** (ls will work if cd doesn't work.)