Linux Distributions

An operating system based on the Linux Kernel is called a Distribution or Distro

There are hundreds of distributions out there, many of them were built for a specific purpose
e.g. for Web servers, Switches & Routers.
A light weight version for smart phones is Android
Linux systems can run for years together without a reboot.
Linux is a clone of Unix.
Linux is a free to use Kernal for programmers.

ArchLinux	Popular for development
ContOS	Most used Linux distributions. Most used for enterprises and web servers A FREE enterprise class OS and based on RedHat enterprise distro
Gentoo	Source based distribution, you need to configure core before you use it.
Linux Mint	A desktop OS; The fourth most used OS
Ubuntu	The third most popular desktop OS after MS Windows and Apple Mac It is based on Debian Linux distro

Installation Methods

1.

USB stick	Need Ubuntu ISO	www.ubuntu.com/download/desktop
	.iso Linux files on the USB Stick	www.pendrivelinux.com

Files

- Files are ordered in a Tree structure, starting with Root directory.
- Root is noted with a forward slash (/)
- This Root directory can be considered as start of the file system and branches out various other sub-directories Viz…
  - bin
  - boot
  - etc
  - dev
- In Linux or Unix, everything is a File.
  - Directories
  - Files
  - Mouse, Printer and Keyboard etc.,

File Types:

General Files	Also known as ordinary files. e.g. image, video, text or program.
	ASCII or Binary format
Directory Files	• A warehouse for other file types. • You can have a directory file within a directory. • Analogous to Folder in Windows. • CDROM is not a drive in Linux, it is a folder. • Removable media files are also not drives, but directories.
	bin folder is similar to "Program Files" in windows
Device Files	e.g. Printers, Hard drives, CDROM etc., are represented with a drive letter in windows. All the device files reside in the "dev" directory. All these file types require permissions to read/execute them (Access restrictions)

File Name Convention

- In windows, you can't have two files with same name in the same folder.
- In Linux, two files can have same name, provided the difference in cases (Upper/Lower)
- For every user, a directory is created under **Home** directory.
- A regular user (standard user) can't save files/folder outside his directory.
- A regular user does not have access to the directories of other users (Very similar to the Users' folder in Windows)
- **Note**: A working directory (Home directory) of a user can be changed using some commands in Linux
- Unix/Linux users a tree like hierarchical file system
- There are **no drives in Linux**, unlike Windows.
- Peripherals like hard drives, cd rom, printers are also considered files in Linux/Unix

Users

Regular user	• The one that gets created when you first install Ubuntu. • The root folder gets created under /home/<User> • As regular user, you will not have access to other users. • Regular accounts are also called **Standard Accounts** in Ubuntu Desktop
Super User (root)	Is also created at the time of installation. It is the **Super User** Can access restricted files, install software and has administrative privileges.
Service User	Ubuntu as server, offers many services viz. Apache, Squid This account is for the security of your computer. NOTE: You will not see service accounts in Ubuntu Desktop version but in server version.

CLI

How to launch CLI in Ubuntu:

1. Click on Dash, type "*terminal*"
2. Press **Ctrl + Alt + T**

Default Terminal Command:
e.g. ubuntu@instance-1:~$

- First part is user name (before '@')
- Second part is host name or computer name (after '@')
- ':' (colon) is a simple separator
- '~' (tilde) sign indicates that user is working in the home directory. If you change the directory, this sign will vanish.
- '$' sign suggests that you are working as regular user.
- When you are a master or root user, '#' sign will be displayed in place of '$' sing.

pwd	Displays the **Print Working Directory**. The moment you boot your system, you will get into your home directory.
cd	Change directory
	cd/cd ~ # takes you to the **home directory**
	cd / # takes you to the **root directory, a space has to be given between cd and /** Note: root is represented in Linux with '/' (forward slash), similar to 'c:\' in windows
	cd .. # to move one level up
Downloads path	/home/user/downloads in Linux is similar to C:\Users\Home\Downloads in windows

Absolute Vs. Relative path

Absolute Path	Absolute path is the complete address of a file or directory. We have to specify the full path to the file or directory. To navigate to /home/user/downloads directory, we give cd /home/raghu/downloads This is said to be absolute directory
Relative path	• Relative path is a relative location of a file or directory with respect to current directory. • Relative path helps avoid typing complete path • You don't need to specify the full path to navigate to a sub-directory or file in your current hierarchical tree.e.g. if you are already in /home/raghu, to navigate to downloads, you have to give.. • cd downloads

Commands - 1

Command	Description	Example
pwd	Print working directory	
ls	List of directories	ls Ls -a to see all the hidden files
clear		
cd	Change directory	cd git cd .. Takes you back to the previous directory cd / takes you to the root directory
mkdir	Create a directory	mkdir testfolder
rmdir	Remove directory	rmdir testfolder removes empty directory only rmdir -rf testfolder removes all files and then the directory also
cp	Copy a file or directory	cp <filename to be copied> <new filename> cp testfile /home copies to the destination folder
man	Manual pages for shell commands	man ls
rm	Removes a file from current/given directory	rm testfile
cat	outputs the contents of a file either to the shell, another file that already exists, or a file that does not yet exist.	cat /tmp/foo.txt
chown	change file owner and group (This command changes the user and/or group ownership of each given file.)	# chown raghu raghu.tar.gz
chmod	change file mode bits (This command changes the file mode bits of each given fil e according to mode)	chmod 777 raghu.tar.gz chmod +x *.sh "+x" key make all the .sh files executable.
chgrp	change group ownership (This command change the group of each file to group)	# chgrp raghu raghu.tar.gz
sudo	Also referred to as superuser do, a sudo command allows you to run other commands with administrative privileges. This command is especially useful for modifying files in a directory that a user wouldn't necessarily have access to.	
sudo apt-get update	Updates your repository	
sudo reboot	Reboots your machine	
shutdown	Shut down the computer from terminal	shutdown -h
find	find searches the file located at / (This command find searches the directory tree rooted at each given file)	find / name raghu.txt -print
compgen	User information	compgen -u displays all users Compgen -g displays all user groups
adduser	Adds a user	sudo adduser raghu
userdel	Deletes a user	sudo userdel username sudo rm -r home/username to delete user's directory
usermod	To modify a user name	usermod -l new_username old_username usermod -aG sudo raghu ----adds user to sudo group
passwd	Changes password for user accounts	Password raghu -d deletes a user password
ifconfig	configure a network interface (This command is used to configure the kernel-resident network interfaces) # ifconfig	

chfn	Changing user information	sudo chfn raghu
su	To become another user during login session	su ----defaults to becoming super user su raghu
groupadd	Creates a new group account	groupadd dev
sudo passwd	To change the root passwd	
lsb_release -a	To know the Ubuntu version	
systemctl	systemctl enable <SERVICE NAME>	# Restarts a service only if it is running. systemctl try-restart <SERVICE NAME> # Reloads configuration if it's possible. systemctl reload <SERVICE NAME> # try to reload but if it's not possible restarts the service systemctl reload-or-restart <SERVICE NAME> #You can use it to find out about a service status: systemctl status <SERVICE NAME>
top	Shows the list of processes running	

Commands - 2

$ sudo service ssh status	
sudo apt-get autoremove	remove any packages that aren't used or associated with any installed program
apt-get purge <package_name>	To remove a package completely
sudo apt-get upgrade	upgrade all packages
apt list --installed	Shows all the installed packages Apt list shows all the available packages
sudo apt-get upgrade [package name 1] [package name 2] … [package name n]	upgrade individual programs
sudo apt-get remove [package name 1] [package name 2] … [package name n]	get rid of a program, you can uninstall its associated packages.
sudo apt-get remove –purge [package name 1] [package name 2] … [package name n]	If you want to get rid of the configuration files and associated directories (usually in the user's home directory)
Service --status-all	Shows all the services installed

vi Editor commands

In vim there are 3 different modes:

- Insert - allows typing and editing as normal
- Visual - used for selecting copy/paste etc.
- Normal - used for commands

To get back to Normal mode, you can always press `esc`.

Once you are at Normal mode Press `:` to begin your command (you'll see it appear in the bottom left). The following commands are related to quiting vim:

- `:q` - quit if no changes were made
- `:q!` - quit and destroy any changes made
- `:wq` - write changes (save) and quit
- `:x` - similar to `:wq`, only write the file if changes were made, then quit

Commands - 3

df	Shows the amount of disk space used and available on Linux file systems.	
du	Display the amount of disk space used by the specified files and for each subdirectory.	
btrfs fi df /device/	Show disk space usage information for a btrfs based mount point/file system.	
sudo init 6	To restart the machine	
netstat -tulpno	To see all the tcp/udp listening ports and number	
	netstat -tulpno| grep :80 # to see what is running on port 80	
Apt-get install tree	Installs tree	
tree	Shows the complete tree structure	
sudo su -	Will switch to root	
	sudo sed -i 's/PasswordAuthentication/yes/' /etc/ssh/sshd_config sudo sed -i 's/prohibit-password/yes/' /etc/ssh/sshd_config	
ls -al ~/.ssh	Check existing ssh keys	
find	find <startingpoint> <options> <search term> find / <options> <search term> #To start searching the whole drive find . -name game #current folder find ~ -name game e.g. find / -name packages # to search all the folders/files that have name "package" find / -name *.mp3	
Process status	ps -aud | less ps -ef | grep java	

File Commands

	LISTING FILES
ls	To check the list of files/sub-folders in the current directory Directories are denoted in blue and files in white color
ls -R	shows the files/sub-folders not only in the directories, but in the sub-directories.
ls -al	To see all the details about the files/folders e.g. drwxr - xr -x 22 raghu dev 4096 2017-05-04 17:10 .bash_history First column (drwxr - xr -x 22) shows file types & access permissions Second column (22) shows the memory blocks Third column (raghu) shows the owner of the file Forth column (dev) displays the user group of the owner Fifth column (4096) shows the size in bytes Sixth column (2017-05-04 17:10) shows the date & time created Seventh column (.bash_history) shows the file/directory name
ls -a	Shows the hidden files **Note:** '.' (period symbol) at the beginning of the file/folder indicates that it is a hidden file e.g. .config
	CREATION AND VIEWING FILES
cat command	Helps in… • Display • Copy • Combine • Create text files **NOTE**: Only text files can be viewed, combined using cat command.
Create a file	cat > file1 # terminal prompts you to enter the content **Ctrl + d** to quit the write mode and go back to the command prompt. e.g. cat > file1
Viewing a file	cat <file name> e.g. cat file1
Combine files	cat filename_1 filename2 > filename-3 e.g. cat file1 file2 > file3 Concatenates filename_1 and filename_2 and outputs into a filename_3
	DELETING FILES
rm	To delete a file rm <filename>
	MOVING FILES
mv	Moves file from one location to another location. mv <filename> <new file location> e.g. mv file3 /home/raghu/files You can perform this operation with SUDO. The moment you type in the above command with 'sudo', it performs the same action with super user privileges. It asks for the current user password. It logs all the activities you are performing as SUDO for administrative purposes. The password you type will be retained for 15 minutes and you will not be prompted for password again.
Rename files	mv <filename> <new filename> e.g. mv file1 file5

User Management

If for some reason you wish to enable the root account, simply give it a password	sudo passwd
☑ Adding user using **adduser** command #create a new user or update default new user information #By default, a group will also be created for the new user #this command adds a user along with home dir., password and other details.	adduser username
	ls -ld /home/username #To verify your current user home directory permissions
deluser #Deleting an account does not remove their respective home folder. Remember, any user added later on with the same UID/GID as the previous owner will now have access to this folder if you have not taken the necessary precautions.	deluser usernme #deletes a user deluser username group #removes a user from a group
useradd This command adds a user without adding home dir. And password	useradd username
To verify your user home directory permissions	ls -ld /home/username
Creates a user along with a home dir.	useradd -m username
Creates a user, adds to a group and sets the	useradd -G dev -d /home/git -m -s /bin/bash git
To temporarily lock or unlock a user account	sudo passwd -l username sudo passwd -u username
To add or delete a personalized group	sudo addgroup groupname sudo delgroup groupname
☑ To add a user to a group	sudo adduser username groupname e.g. adduser raghu dev adduser raghu sudo
To modify a user	usermod
	usermod -aG sudo raghu ----adds user to sudo group
	usermod -l new_username old_username

File Permissions

Authorization Levels

1. Ownership
    a. User is the owner of the file who created it
    b. Group: All users belonging to the group will have same permission (read/modify)
    c. Any other who has access to the file
        i. Does not own the file
        ii. Does not belong to a user group
2. Permissions
    a. Read, write and execute (Exe)
    r = read
    w = write
    x = execute
    '-' no permission

e.g. -rw-rw-r-- 1 raghu dev 4096
'-' indicates it is a file. Otherwise it would start with 'd' if it is a directory.
First rw- indicates the permissions of a **owner** (here it is read, write but no execute)
Second rw- is for the **group**.
Third --r is for **other**. He/she can only read the file

Changing file/directory permissions

chmod	Changes file permissions. Using this command we can set read/write/execute permissions on a file/directory for the owner/group/world
Absolute mode	Filer permissions are represented in numeric way.
Symbolic mode	Change permissions for a specific user

Number	Permission Type	Symbol
0	No permission	---
1	Execute	--X
2	Write	-w-
3	Write + Execute	-wx
4	Read	r--
5	Read + execute	r-x
6	Read + write	rw-
7	Read + write + execute	rwx

e.g. chmod 764 test	Sets permissions on read, write and execute for the user Read + write for the group Read for the world

Symbolic

Operator	Description
+	Adds a permission to a file/directory
-	Removes permission
=	Sets the permission and overwrites the permission set earlier

User denotations

u	User/owner
g	Group
o	Other
a	all

e.g. chmod o=rwx test #sets read, write and execute permissions for the other user.
chmod g+x sets execute permission for the group
Chmod u-r removes read permission for the user

Changing ownership and Group

chown user <filename>

e.g. sudo chown root file1.txt # changes the owner of file1.txt to root

chown user:group <filename>

e.g. sudo chown git:dev test1.txt # changes the owner name and group name

chgrp <groupname> <filename>

e.g. sudo chgrp dev file1.txt #changes the group name to dev.

Or

sudo chown raghu:dev file1.txt