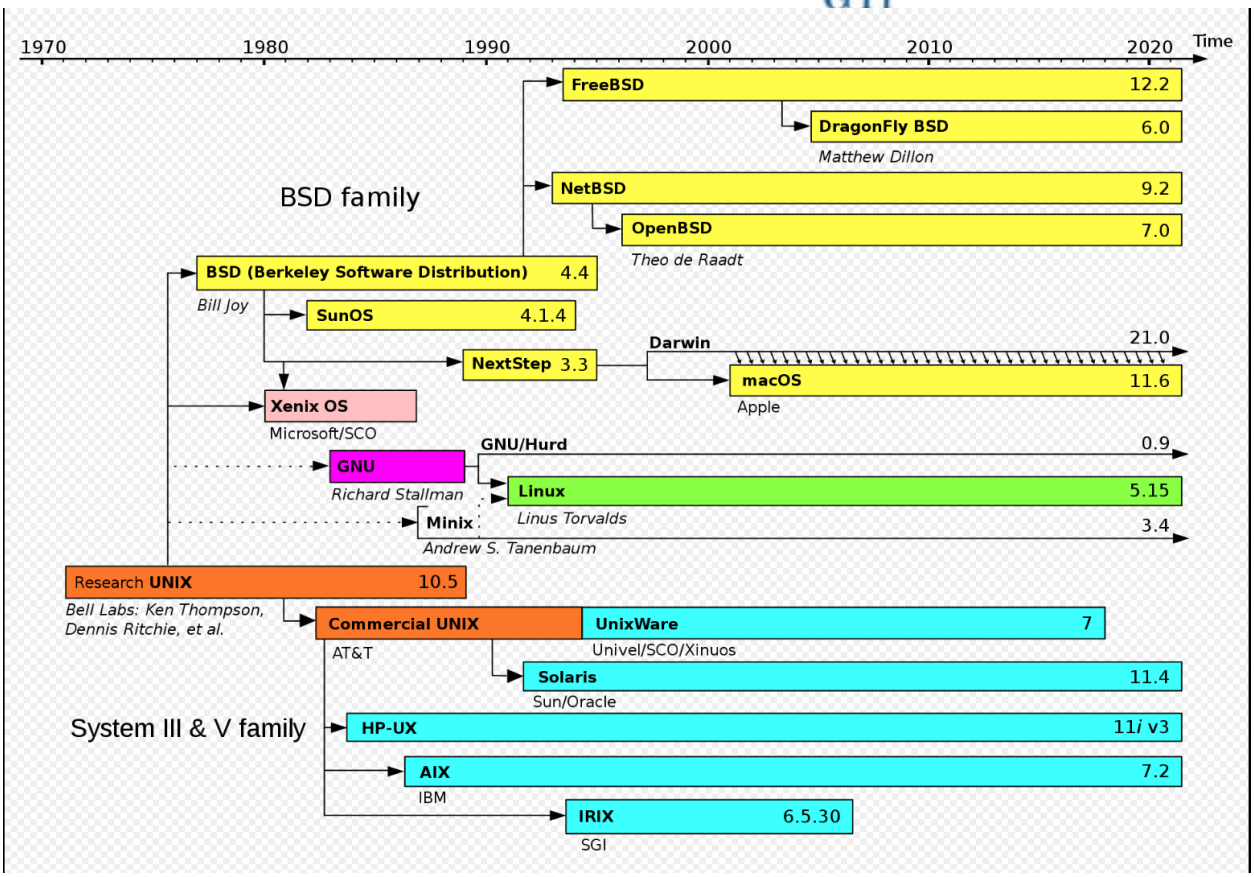
**Linux Basic and Bash Scripting**

Linux is an Operating system like your windows and Mac OS. There are different types of Linux systems, all Linux systems are built on top of a Linux kernel. Most of its tools came from GNU projects owned by GPL. The development of Linux system started in the 90s. The system has evolved to a more acceptable and accessible OS as open-source software.

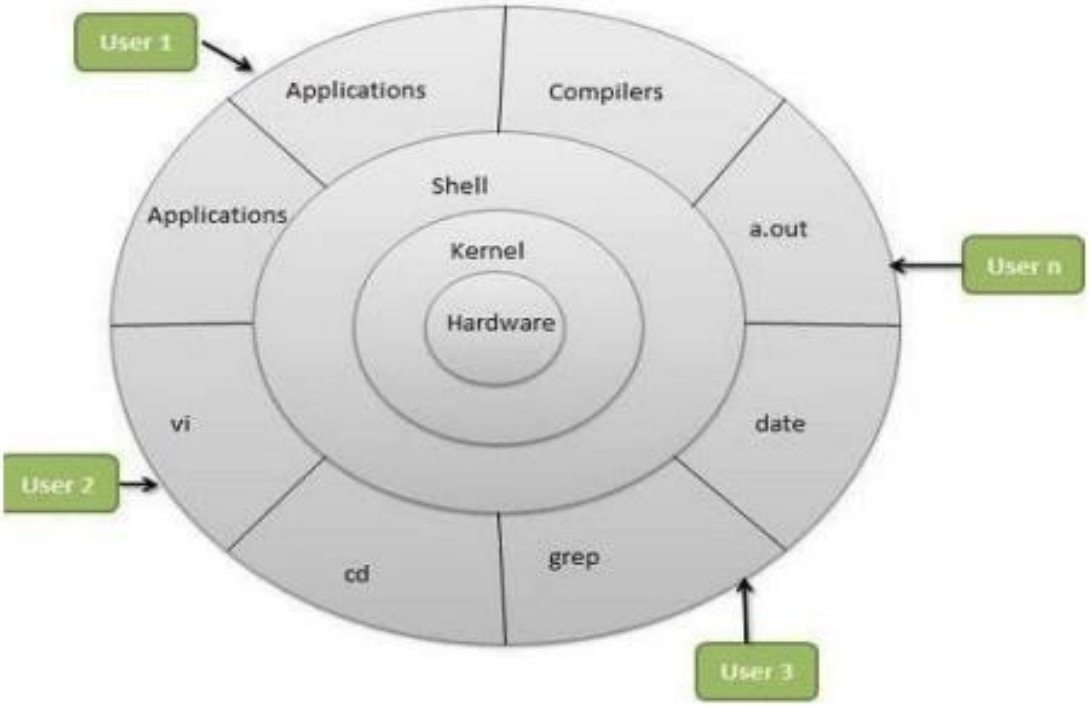
**Linux Timeline**

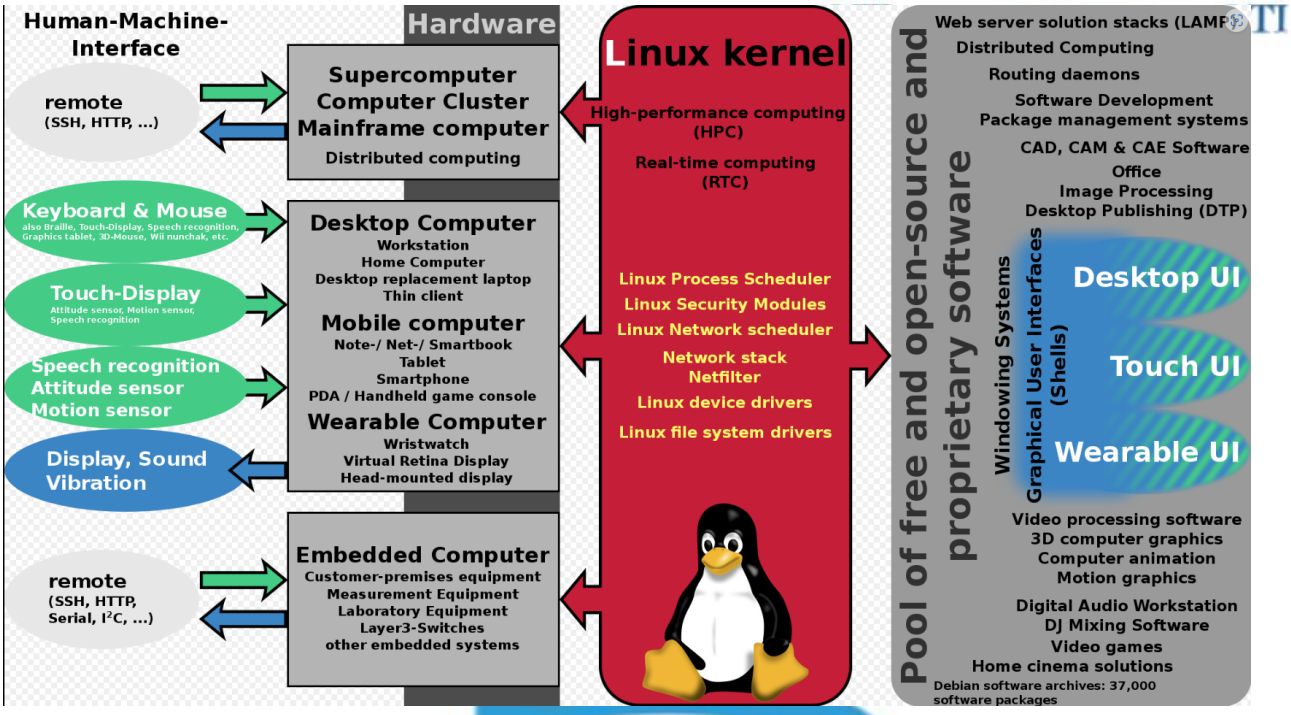



Linux OS manages the communication between your software and hardware. The OS is made up of several pieces and include

1. Bootloader
2. Kernel
3. Init system
4. Daemons
5. Graphical server
6. Desktop environment
7. Application

# Architecture of Linux






Linux is an Open source with extensive community support, and the software is offered for free. There are different types of Linux system some are listed below

**Why Linux?**

- Open Source.
- Community support.
- Heavily customizable.
- Most Servers run on Linux.
- DevOps tools mostly rely on Linux systems.
- Automation
- Secure. Architecture of Lin

**Client (Desktop):**

Linux Mint,
Debian, ubuntu,
openSUSE,
fedora etc.

**Server:**

Red Hat,

Ubuntu,

Centos,

SUSE Enterprise etc.

Most used Linux system currently in IT industry.

**RPM-based**: RHEL & Centos

**Debian based**: Ubuntu Server

Difference between RPM-based and Debian based.

From user's point of view, there isn't much difference between 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 management 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 software's)

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

**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 software.)

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.rp

**Reference :** What is Linux? - Linux.com

## Basic Linux Command

**To start exploring a Linux system, here are some very important directories to know.**

1. **Home Directories: /root , /home/username**
2. **Kernels and Bootloader: /boot**
3. **System information : /sys , /proc**
4. **Server data : /var , /srv**
5. **Configuration : /etc**
6. **Other Mountpoints /media , /mnt**

7. **System executables : /sbin , /usr/sbin, /usr/local/sbin**
8. **Users executables : /bin , /usr/bin , /usr/local/bin**
9. **Shared libraries : /lib , usr/lib, /usr/local/lib**
10. **Temporary files : /tmp**

**Centos**

```
[vagrant@localhost ~]$ cd /
[vagrant@localhost /]$ ls -la
total 20
dr-xr-xr-x.  18 root    root     239 Oct 13 22:03 .
dr-xr-xr-x.  18 root    root     239 Oct 13 22:03 ..
lrwxrwxrwx.   1 root    root       7 Feb 12  2022 bin -> usr/bin
dr-xr-xr-x.   5 root    root    4096 Feb 12  2022 boot
drwxr-xr-x.  19 root    root    3040 Dec 17 04:29 dev
drwxr-xr-x.  79 root    root    8192 Dec 17 04:29 etc
drwxr-xr-x.   3 root    root      21 Feb 12  2022 home
lrwxrwxrwx.   1 root    root       7 Feb 12  2022 lib -> usr/lib
lrwxrwxrwx.   1 root    root       9 Feb 12  2022 lib64 -> usr/lib64
drwxr-xr-x.   2 root    root       6 Apr 11  2018 media
drwxr-xr-x.   2 root    root       6 Apr 11  2018 mnt
drwxr-xr-x.   3 root    root      39 Feb 12  2022 opt
dr-xr-xr-x. 127 root    root       0 Dec 17 04:29 proc
dr-xr-x---.   3 root    root     149 Feb 12  2022 root
drwxr-xr-x.  26 root    root     840 Dec 17 04:29 run
lrwxrwxrwx.   1 root    root       8 Feb 12  2022 sbin -> usr/sbin
drwxr-xr-x.   2 root    root       6 Apr 11  2018 srv
dr-xr-xr-x.  13 root    root       0 Dec 17 04:29 sys
drwxrwxrwt.  10 root    root    4096 Dec 17 04:30 tmp
drwxr-xr-x.  13 root    root     155 Feb 12  2022 usr
drwxrwxrwx.   1 vagrant vagrant    0 Dec 17 04:28 vagrant
drwxr-xr-x.  19 root    root     267 Feb 12  2022 var
```

**Ubuntu**

```
vagrant@ubuntu-bionic:/sys$ cd ../
vagrant@ubuntu-bionic:/$ ls -la
total 88
drwxr-xr-x  24 root    root     4096 Dec 17 02:23 .
drwxr-xr-x  24 root    root     4096 Dec 17 02:23 ..
drwxr-xr-x   2 root    root     4096 Oct 13 15:52 bin
drwxr-xr-x   3 root    root     4096 Oct 13 15:52 boot
drwxr-xr-x  15 root    root     3640 Dec 17 02:23 dev
drwxr-xr-x  92 root    root     4096 Dec 17 02:23 etc
drwxr-xr-x   4 root    root     4096 Oct 26 01:37 home
lrwxrwxrwx   1 root    root       34 Oct 13 15:52 initrd.img -> boot/initrd.img-4.15.0-194-generic
lrwxrwxrwx   1 root    root       34 Oct 13 15:52 initrd.img.old -> boot/initrd.img-4.15.0-194-generic
drwxr-xr-x  21 root    root     4096 Oct 13 16:01 lib
drwxr-xr-x   2 root    root     4096 Oct 13 15:50 lib64
drwx------   2 root    root    16384 Oct 13 15:53 lost+found
drwxr-xr-x   2 root    root     4096 Oct 13 15:49 media
drwxr-xr-x   2 root    root     4096 Oct 13 15:49 mnt
drwxr-xr-x   2 root    root     4096 Oct 13 15:49 opt
dr-xr-xr-x 121 root    root        0 Dec 17 02:22 proc
drwx------   3 root    root     4096 Oct 26 01:37 root
drwxr-xr-x  26 root    root      900 Dec 17 02:23 run
drwxr-xr-x   2 root    root     4096 Oct 13 16:01 sbin
drwxr-xr-x   2 root    root     4096 Oct 26 01:38 snap
drwxr-xr-x   2 root    root     4096 Oct 13 15:49 srv
dr-xr-xr-x  13 root    root        0 Dec 17 02:24 sys
drwxrwxrwt  10 root    root     4096 Dec 17 02:23 tmp
drwxr-xr-x  10 root    root     4096 Oct 13 15:49 usr
drwxrwxrwx   1 vagrant vagrant     0 Oct 26 01:37 vagrant
drwxr-xr-x  14 root    root     4096 Oct 26 01:38 var
lrwxrwxrwx   1 root    root       31 Oct 13 15:52 vmlinuz -> boot/vmlinuz-4.15.0-194-generic
lrwxrwxrwx   1 root    root       31 Oct 13 15:52 vmlinuz.old -> boot/vmlinuz-4.15.0-194-generic
```

**Preparing your system for Operations**

Create users, Groups and assign user to a group. In any systems users should be managed by groups. In linux when a user is created by default a corresponding group is also created. But this group can be modified. In this case we will create devops group and assign appropriate authorization, and then create users and add the users to devops group.

Command to create group  (For Ubuntu use adduser)

Sudo useradd <username>
Sudo groupmod -G <username>

Gorup and Passwd File

/etc/group
/etc/passwd
The details is explained thus

Root ➔ Name
X ➔ Link to Password file.  /etc/shadow
0 or 1 ➔
0 or 1 ➔

Root or bin ➔ comment (Info about the user)
/root or /bin ➔ home directory of the user
/bin/bash or /sbin/nologin ➔ shell

Types of users

| Type | Example | User Id | Group id | Home Dir | Shell |
|---|---|---|---|---|---|
| **Root** | **root** | **0** | **0** | **/root** | **/bin/bash** |
| **Regular** | **Vagrant, devops** | **1000 to 60000** | **1000 to 60000** | **/home/<user_id>** | **/bin/bash** |
| **Service** | **ssh, ftp, apache** | **1 to 999** | **1 to 999** | **/var/<>service_name>** | **/sbin/nologin** |

**User and Group Commands**

| 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` |

The /etc/shadow file This file stores users' password and password related information. Just like /etc/passwd file, this file also uses an individual line for each entry.

1. Username 2. Encrypted password 3. Number of days when password was last changed 4. Number of days before password can be changed 5. Number of days after password must be changed 6. Number

of days before password expiry date to display the warning message 7. Number of days to disable the account after the password expiry 8. Number of days since the account is disabled 9. Reserved field

2. cat /etc/shadow (to show the file contents)

File Type

1. Regular Files (regular files such as text, data or executables)

2. Directory d (File that are list of other files)

3. Link l (A shortcut that point to the location for the actual file)

4. special file c (Mechanism use for input and output. such as files in /dev)

5. socket s (a special file that provides inter-process networking protected by the system's access control)

6. Pipe p ( A special file that allows process to communicate with each other without using network socket semantics)

file <file_name> (to know a file type)

file /bin/pwd

file /dev/dm-1

ln -s <path_to_be_link> <linkname> (to create a link)

Filter and IO redirection

1. grep (to find test from any test input)
cd /home/root
grep firewall anaconda-ks.cfg (grep --help to see usage)
grep -i firewall * (this will search for the keyword firewall in all files in the current directory. you can use grep -iR to search sub directories))
grep -R SELINUX /etc/*
less <flie_name> (this is a reader)
more
head
tail -f (this shows dynamic content. If any changes happen to the file, you will see it)
tail -f /var/log/yum.log

cat /etc/passwd (get all user names)
cut -d: -f1 /etc/passwd (you can try f3, f4 and see result) this is good for file that has proper seprator
awk -F':' ' {print $1} ' /etc/passwd

vi<filename> (on Open, usde %s<content_name_to_replace>/<new_content>/g the g does a global replace)

sed 's<content_name_to_replace>/<new_content>/g' <filename> (this will not make the change. Using -i will effect the change))

**I/O Redirection**

uptime > /tmp/sysinfo.txt
free -m > /tmp/meminfo.txt (Use >> to append)
df -h (This shows hard disk partition)
echo  (This is to print)
> /dev/null (output rediretced to the dev/null can not be seen or view. Ex yum install htppd -y > /dev/null)
cat /dev/null > /tmp/sysinfo.txt (This action deletes the sysinfo.txt content)
<bad_command> 2> /tmp/sysinfo.txt (standard error. 1 is a standard output and & any knid of output)

**Piping**

ls | wc -l
ls | grep <file_name> or wild card
tail -20 /var/log/messages | grep -i vagrant  (fetching vagrant in the message file)
free -m | grep Mem
ls -l | head
find /etc -name host*  (You can also find at the root / level, but it can slow down your system)

**Processes**

Top
ps aux
ps -ef
ps -ef | grep <service_name> | grep -v 'grep' | awk ' {print $2} ' xargs kill -9
ps -ef | grep httpd | grep -v 'grep' | awk ' {print $2} ' xargs kill -9

Achieving

tar -czvf home.tar.gz /home/vagrant (achieving the home/vagrant directory)
file <file_name>
tar -xzvf home.tar.gz  (to extract the archived. You may need to move the file before extracting))
tar -xzvf home.tar.gz -C /opt/  (To archived to opt directory )
zip -r home.zip /home/vagrant
unzip home.zip

**File Permission**

## 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*** **...**

**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

**SUDO**

sudo gives power to a normal user to execute commands which are 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

**How to mount a file system**

Filesystem are external storage you can attached to you OS, they are either read only, read, write only or write only.

Commands

1. lsblk -o NAME,FSTYPE,LABEL,SIZE,MOUNTPOINT : list mount point on a server
2. lshw -C disk -short : to show moubnt point in ubuntu server
3. gdisk /dev/sdb , then select option n, then w, and then y to exit
4. mkfs.ext4 /dev/sdb
5. lsblk -f

To mount a drive (Manually)

6. mkdir /mnt/devops
7. mount /dev/sdb /mnt/disk2/ : mount new disk to the drive

Auto Mount a drive

6. blkid | grep sdb : to get the disk uuid

7. vi /etc/fstab

8. added the outpu from #6 (UUID=7823f91c-4ad3-4a96-b783-9d90b1fa275f /mnt/devops ext4 defaults 0 1

9. mount -a

10. lsblk -o NAME,FSTYPE,LABEL,SIZE,MOUNTPOINT