AWS Solution Architect Associate Certification Training – Module 2

2. Fundamentals of Linux

Introduction to Operating system

- An operating system is a program that controls the execution of application programs and acts as an interface between the user of a computer and the computer hardware.

- A more common definition is that the operating system is the one program running at all times on the computer (usually called the kernel), with all else being application programs.
- An operating system is concerned with the allocation of resources and services, such as memory, processors, devices and information. The operating system correspondingly includes programs to manage these resources, such as a traffic controller, a scheduler, memory management module, I/O programs, and a file system.

Examples of Operating System are –

- Windows (GUI based, PC)
- GNU/Linux (Personal, Workstations, ISP, File and print server, Three-tier client/Server)
- MacOS (Macintosh), used for Apple's personal computers and work stations (MacBook, iMac).
- Android (Google's Operating System for smartphones/tablets/smart watches)
- iOS (Apple's OS for iPhone, iPad and iPod Touch)

Distributions of Linux OS

A Linux distribution (often abbreviated as distro) is an operating system made from a software collection, which is based upon the Linux kernel and often, a package management system. Linux users usually obtain their operating system by downloading one of the Linux distributions, which are available for a wide variety of systems ranging from embedded devices.

List of Linux Distributions

1. Mint: Linux Mint is an Ubuntu-based distribution. It provides out-of-the-box solution and include browser, media supports Java and lots of other components. Features custom desktop and menus and it is compatible with Ubuntu software repositories.
2. Debian: Formally known as Debian GNU/Linux, Debian is free operating system that uses the Linux kernel. It is supported by programmers worldwide who has created more than 50,000 packages under Debian project.
3. Ubuntu: "Ubuntu" means "humanity to others" in African language. Ubuntu is free Linux operating system for desktops. Community and professional support is available.
4. OpenSUSE: S.u.S.E. is a German acronym for "Software und System-Entwicklung" (software and systems development). OpenSUSE is based on Linux and it is the community project sponsored by SUSE and others companies.
5. Manjaro: Manjaro is a user-friendly Linux distribution based on Arch Linux (i686/x86-64 general-purpose GNU/Linux distribution). It focus on user-friendliness and accessibility, available in both 32 and 64 bit versions and it is suitable for newcomers as well as experienced Linux users.

6. Fedora: Fedora is a Linux distribution based on GNU/Linux developed by the community-supported Fedora Project and owned by Red Hat.
7. Elementary: elementary Linux is based on Ubuntu desktop distribution with some custom apps including Music, Photos, Videos, Calendar along with Epiphany web browser.
8. Zorin: Linux alternative to windows, Zorin OS is an Ubuntu based Linux distribution. It has windows like interface and allows to run many program similar to windows and has an application which lets user run windows programs. Ideal for beginners starting with Linux OS. Distribution Zorin OS.
9. CentOS: Centos is named after **C**ommunity **ENT**erprise **O**perating **S**ystem.
   CentOS is open source enterprise class operating system build Red Hat Enterprise Linux by Red Hat.
10. Arch: Arch Linux is based on GNU/Linux available in i686- and x86_64-optimised Linux distribution and provided ability to build custom installation and new packages and share them by Arch Linux Repository.

Linux Vs Windows

What is windows operating system?

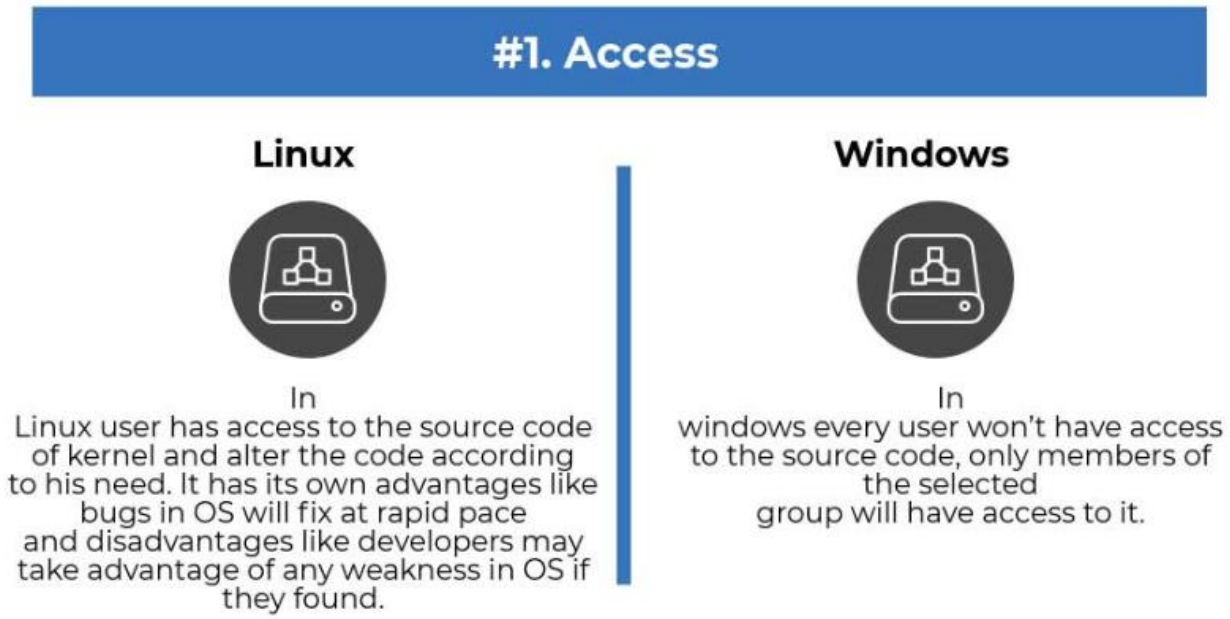
Windows is a series of operating systems, computer operating system (OS) developed by Microsoft for personal desktops/devices or computer (PC). Each operating system comes with a graphical user interface (GUI) with a desktop which allows a user to view all files, videos etc. The first version of Windows OS is released in 1985 which is a simple GUI, an extension of existing disk operating system (MS-DOS) and major release as my point of view was 1995 consumer release which has integrated windows and DOS with built-in internet support. Most of the PC is currently running on Windows operating system only. It is designed to run on x86 hardware such as AMD, Intel processors. So windows OS comes with almost all company who made PC's or laptops. Latest Windows OS version is Windows 10 which is currently ruling the market.

What is Linux Operating system?

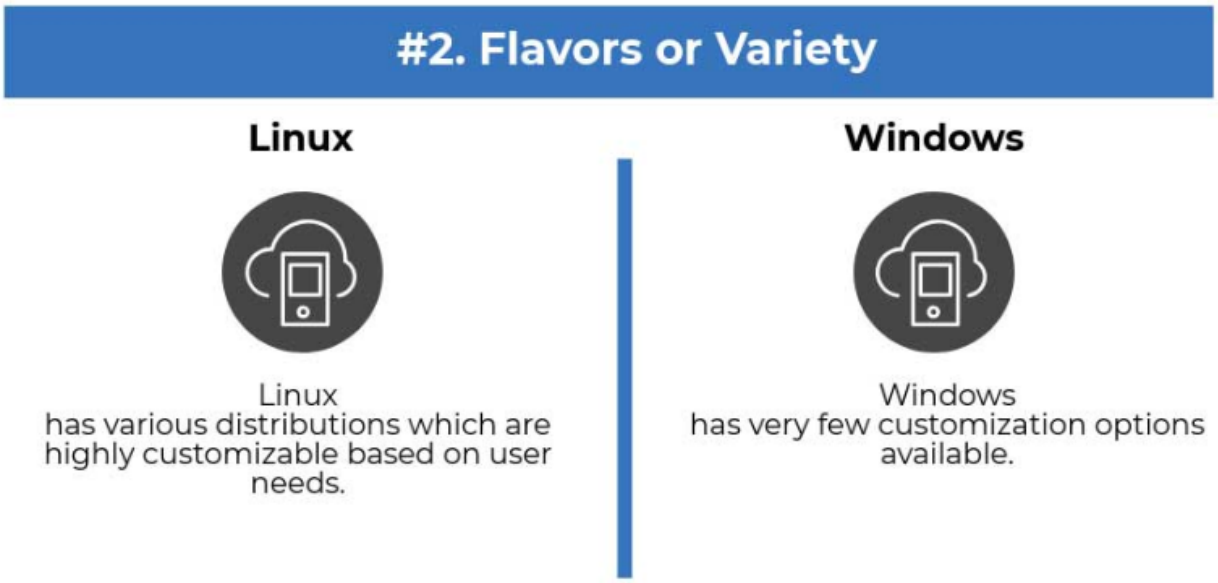
Linux is an open source operating system based on UNIX, created in 1991. It is software which sites underneath of all other software on a computer. Users can modify the existing code and create distributions from it as it is an open source operating system. System also comes with a graphical user interface (GUI) with some necessary software's which are used on a daily basis. Linux is mostly used as a server – as most of the web pages over the internet are generated from Linux servers and also used in desktop computers, mobile devices, gaming consoles, digital storing devices, eBook readers, cameras, video recorders have Linux running.

Head to Head Comparison between Linux vs Windows

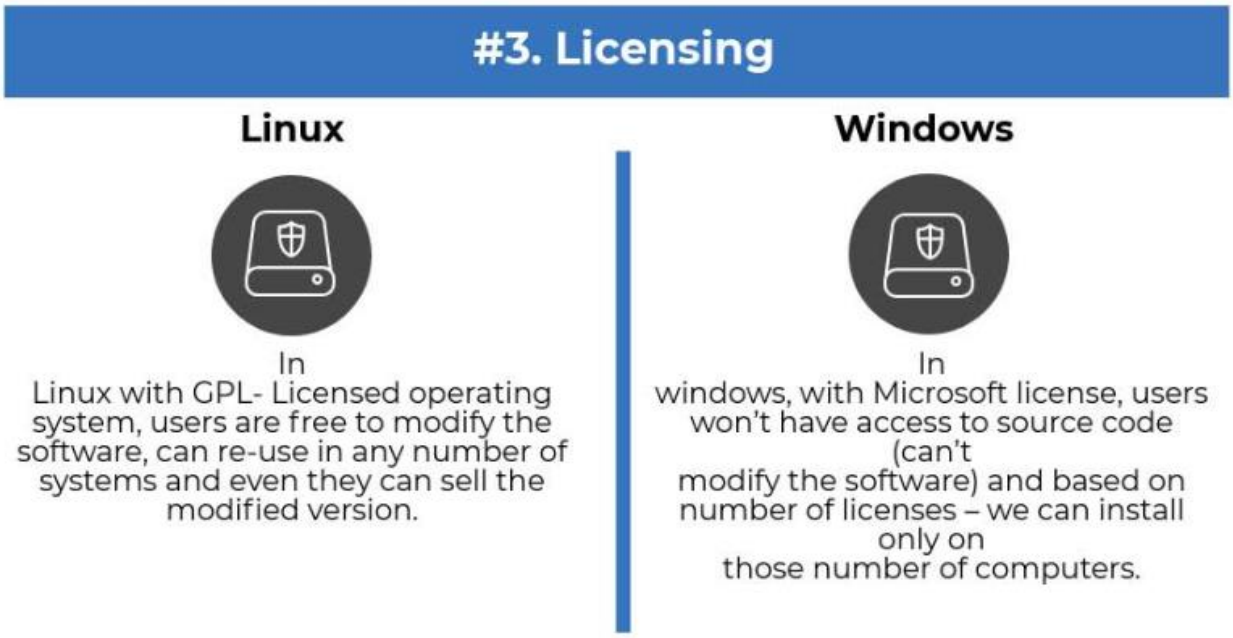
1. Access:



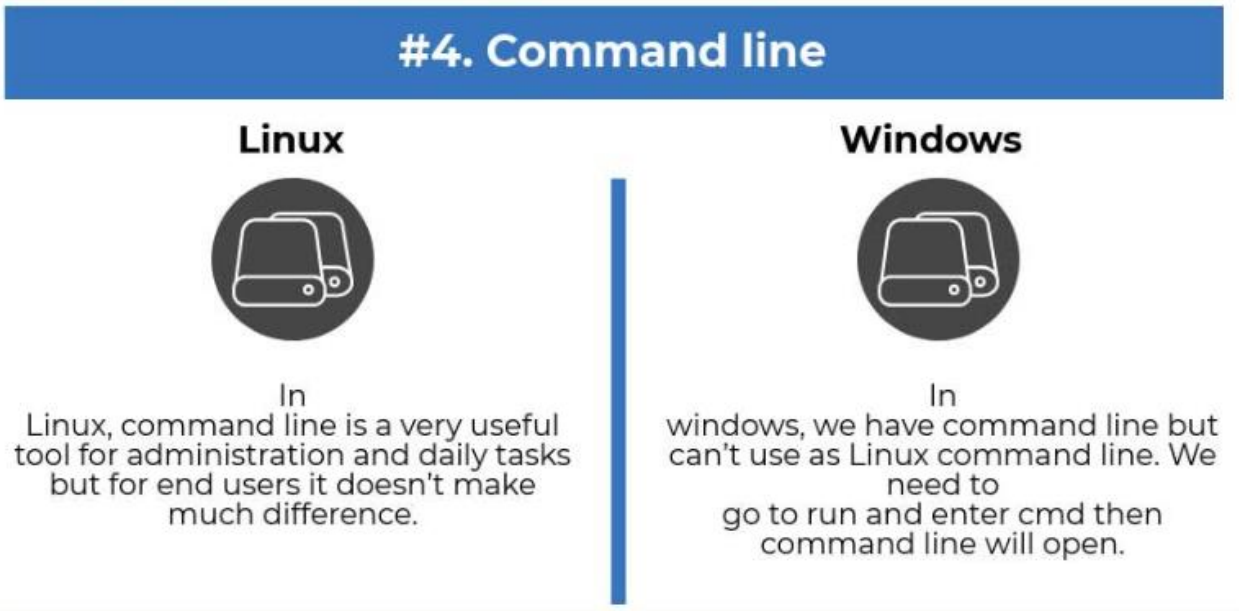
2. Flavors or Variety



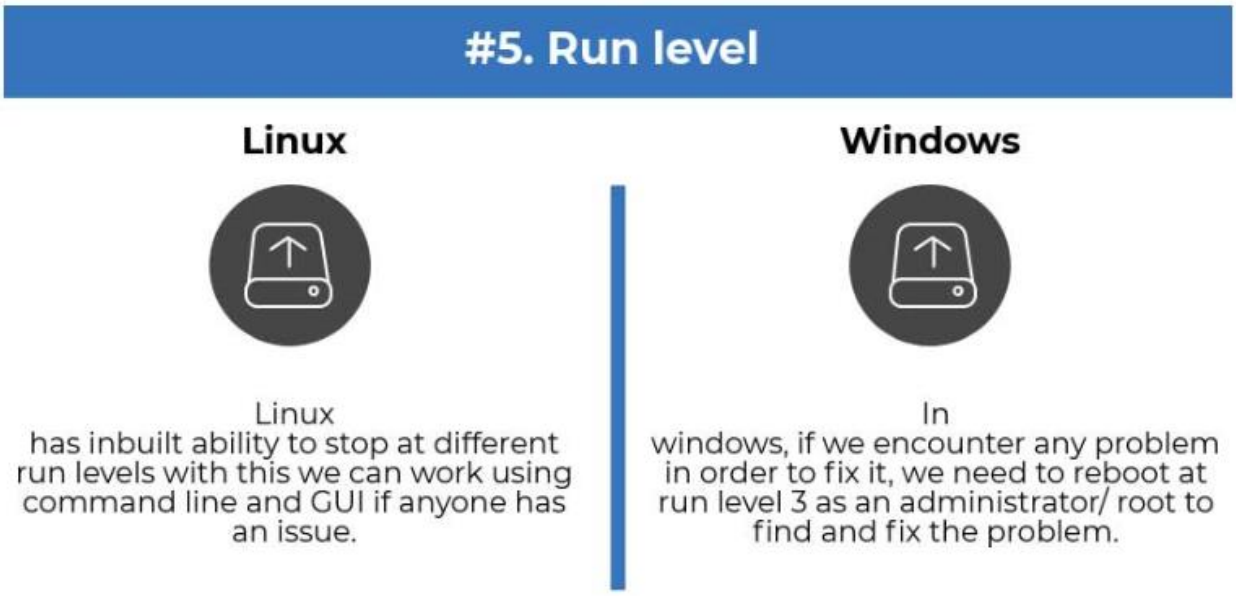
3. Licensing



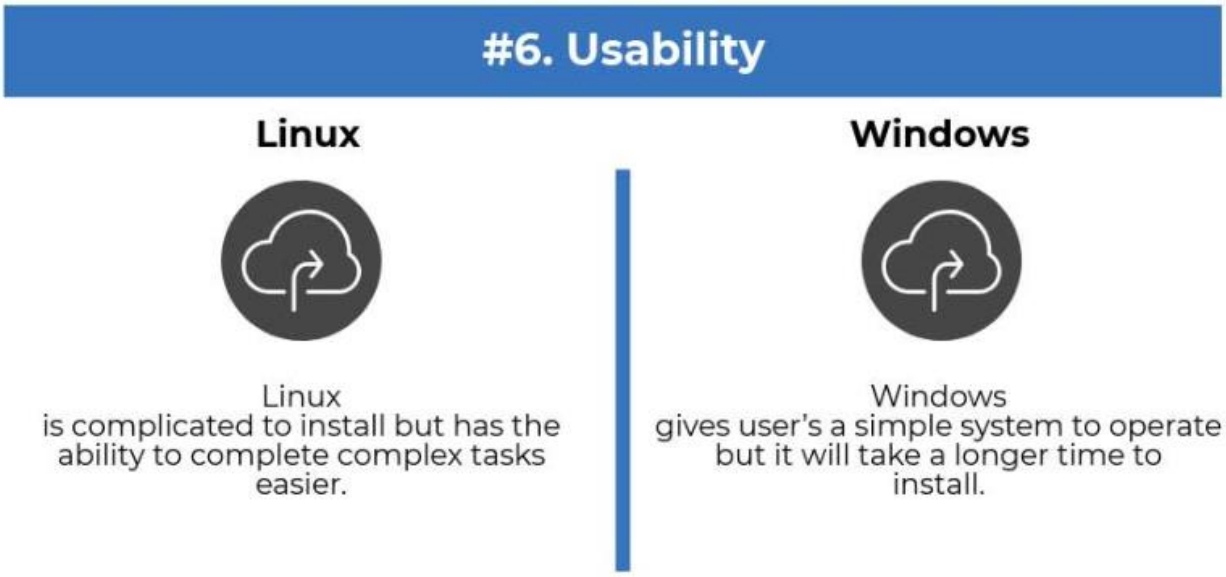
4. Command line



5. Run level

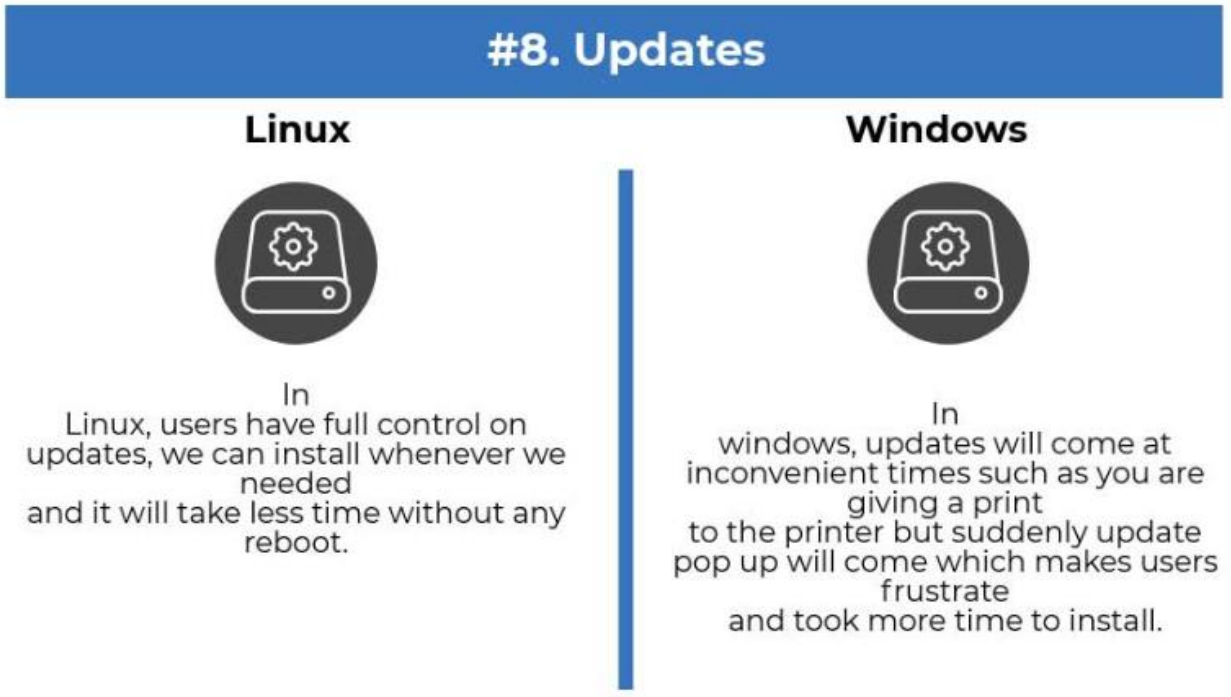


6. Usability

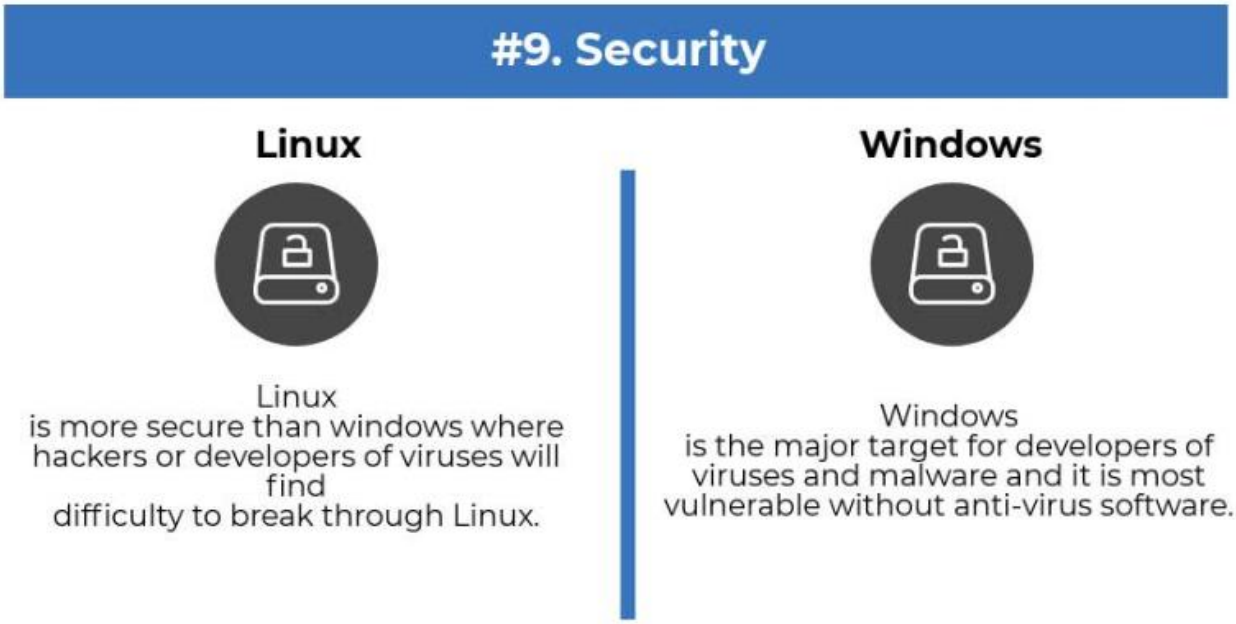


7. Support

8. Updates



9. Security



Key Differences between Linux vs Windows

- Linux is open source operating system whereas Windows OS is commercial.
- Linux has access to source code and alters the code as per user need whereas Windows does not have access to source code.
- Linux will run faster than windows latest editions even with a modern desktop environment and features of the operating system whereas windows are slow on older hardware.
- Linux distributions don't collect user data whereas Windows collect all the user details which lead to privacy concern.
- Linux is more reliable then windows as in Linux we can kill application if they hung through x kill command whereas, in windows, we need to try multiple times to kill it.
- Linux supports a wide variety of free software's than windows but windows have a large collection of video game software.
- In Linux software cost is almost free as all programs, utilities, complex applications such as open office are free but windows also have many free programs and utilities but most of the programs are commercial.
- Linux is highly secure because it's easy to identify bugs and fix whereas Windows has a large user base and becomes a target for developers of viruses and malware.
- Linux is used by corporate organizations as servers and operating system for security purpose at Google, Facebook, twitter etc. whereas windows are mostly used by gamers and business users.
- Linux and windows have same priority over hardware and driver support in the present situation.

Shell Environments

- The shell is the command interpreter in an operating system such as UNIX or GNU/Linux, it is a program that executes other programs. It provides a computer user an interface to the Unix/GNU Linux system so that the user can run different commands or utilities/tools with some input data.
- When the shell has finished executing a program, it sends an output to the user on the screen, which is the standard output device. For this reason, it is referred to as the "command interpreter".
- The shell is much more than just a command interpreter, it is also a programming language of its own with complete programming language constructs such as conditional execution, loops, variables, functions and many more.

- That is why the Unix/GNU Linux shell is more powerful compared to the Windows shell.

5 most frequently used shells for Linux

I. **Bash shell:** Bash stands for Bourne Again Shell and it is the default shell on many Linux distributions today. It is also a sh-compatible shell and offers practical improvements over sh for programming and interactive use which includes:
   - Command line editing
   - Job control
   - Unlimited size command history
   - Shell functions and alias
   - Unlimited size index arrays
   - Integer arithmetic in any base from two to sixty-four

II. **Tcsh/Csh Shell:** Tcsh is enhancement C shell, it can be used as a interactive login shell and shell script command processor.
   Tcsh has the following features:
   - C like syntax
   - Command line editor
   - Programmable word and file name completion
   - Spelling correction
   - Job control

III. **Ksh Shell:** Ksh stands for "Korn shell" and was designed and developed by David G. Korn. It is a complete, powerful, high-level programming language and also an interactive command language just like many other Unix/GNU Linux shells.

IV. **Zsh Shell:** Zsh is designed to be interactive and it incorporates many features of other Unix/GNU Linux shells such as bash, tcsh and ksh. It is also a powerful scripting language just like the other shells available. Though it has some unique features that include:
   - Filename generation
   - Startup files
   - Login/Logout watching
   - Closing comments
   - Concept index

- Variable index
- Functions index
- Key index and many more that you can find out in man pages

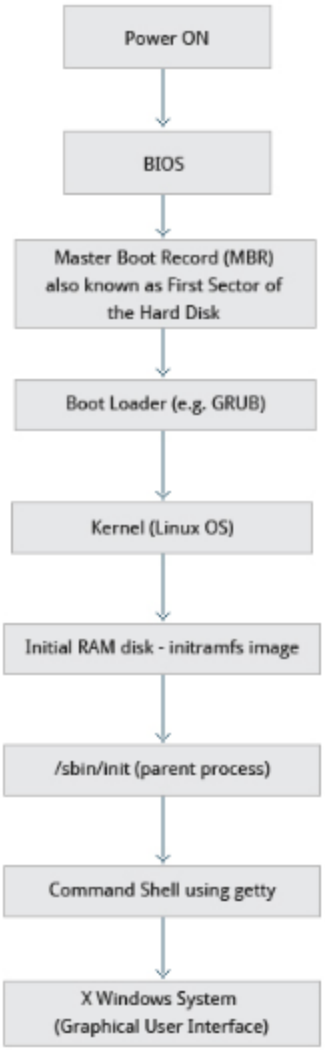
V. **Fish:** Fish in full stands for “friendly interactive shell” and was authored in 2005. It was intended to be fully interactive and user friendly, just like the other shells, it has some pretty good features that include:
    - Man page completions
    - Web based configuration
    - Auto-suggestions
    - Fully scriptable with clean scripts
    - Support for term256 terminal technology

Linux Booting and process

The Linux boot process is the procedure for initializing the system. It consists of everything that happens from when the computer power is first switched on until the user interface is fully operational.

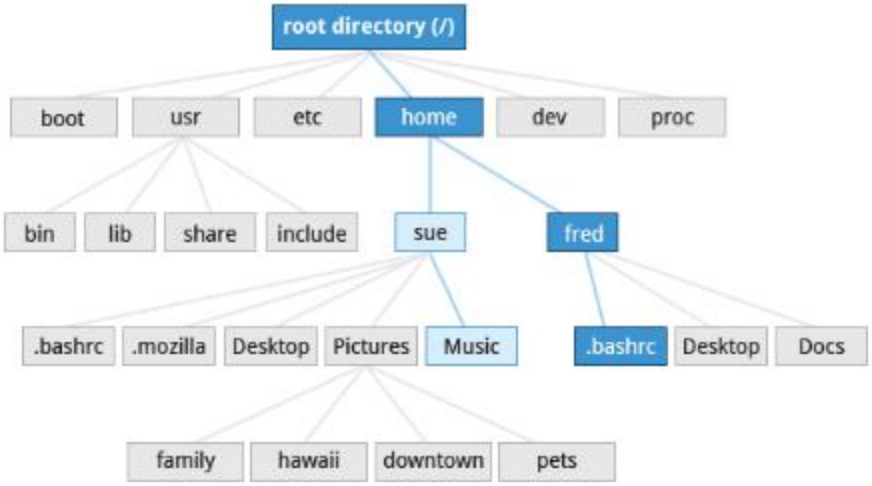
Having a good understanding of the steps in the boot process may help you with troubleshooting problems, as well as with tailoring the computer's performance to your needs.

On the other hand, the boot process can be rather technical, and you can start using Linux without knowing all the details.



The Boot Process

The directory structure of Linux

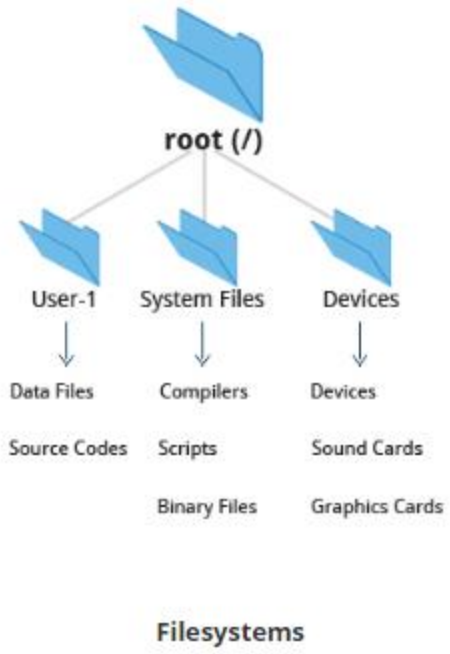


Files, file types and permissions

Introduction to Filesystems

In Linux (and all UNIX-like operating systems) it is often said “Everything is a file”, or at least it is treated as such. This means whether you are dealing with normal data files and documents, or with devices such as sound cards and printers, you interact with them through the same kind of Input/output (I/O) operations. This simplifies things: you open a “file” and perform normal operations like reading the file and writing on it.

On many systems (including Linux), the filesystem is structured like a tree. The tree is usually portrayed as inverted, and starts at what is most often called the root directory, which marks the beginning of the hierarchical filesystem and is also sometimes referred to as the trunk, or simply denoted by **/**. The root directory is *not* the same as the root user. The hierarchical filesystem also contains other elements in the path (directory names), which are separated by forward slashes (*/*), as in **/usr/bin/emacs**, where the last element is the actual file name.



Filesystems

Linux supports a number of native filesystem types, expressly created by Linux developers, such as:

- ✓ ext3
- ✓ ext4
- ✓ squashfs
- ✓ btrfs.

It also offers implementations of filesystems used on other alien operating systems, such as those from:

- ✓ Windows (ntfs, vfat)
- ✓ SGI (xfs)
- ✓ IBM (jfs)
- ✓ MacOS (hfs, hfs+)

Many older, legacy filesystems, such as FAT, are also supported.

It is often the case that more than one filesystem type is used on a machine, based on considerations such as the size of files, how often they are modified, what kind of hardware they sit on and what kind of access speed is needed, etc. The most advanced filesystem types in common use are the journaling varieties: ext4, xfs, btrfs, and jfs. These have many state-of-the-art features and high performance, and are very hard to corrupt accidentally.

Ownership of Linux files

Each file and directory on your Linux/Unix system is assigned 3 types of owner, given below.

User: A user is the owner of the file. By default, the person who created a file becomes its owner. Hence, a user is also sometimes called an owner.

Group: A user- group can contain multiple users. All users belonging to a group will have the same access permissions to the file. Suppose you have a project where a number of people require access to a file. Instead of manually assigning permissions to each user, you could add all users to a group, and assign group permission to file such that only this group members and no one else can read or modify the files.

Other: Any other user who has access to a file. This person has neither created the file, nor he belongs to a usergroup who could own the file. Practically, it means everybody else. Hence, when you set the permission for others, it is also referred as set permissions for the world.

Now how does **Linux distinguish** between these three user types so that a user 'A' cannot affect a file which contains some other user 'B's' vital information/data. It is like you do not want your colleague, who works on your Linux computer, to view your images. This is where **Permissions** set in. and they define **user behavior.**

Let us understand the **Permission system** on Linux.

Permissions

Every file and directory in your UNIX/Linux system has following 3 permissions defined for all the 3 owners discussed above.

- Read: This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists it's content.
- Write: The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory. Consider a scenario where you have to write permission on file but do not have write permission on the directory where the file is stored. You will be able to modify the file contents. But you will not be able to rename, move or remove the file from the directory.
- Execute: In Windows, an executable program usually has an extension ".exe" and which you can easily run. In Unix/Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code (provided read & write permissions are set), but not run it.

Basic Linux commands and Administration

Basic operations

In this section, we will discuss how to accomplish basic operations from the command line. These include how to log in and log out from the system, restart or shut down the system, locate applications, access directories, identify absolute and relative paths, and explore the filesystem.

Logging In and Out

An available text terminal will prompt for a username (with the string **login:**) and password. When typing your password, nothing is displayed on the terminal (not even a ***** to indicate that you typed in something), to prevent others from seeing your password. After you have logged into the system, you can perform basic operations.

Once your session is started (either by logging into a text terminal or via a graphical terminal program), you can also connect and log into remote systems via the Secure Shell (SSH) utility. For example, by typing **ssh username@remote-server.com**, SSH would connect securely to the remote machine and give you a command line terminal window, using passwords (as with regular logins) or cryptographic keys (a topic we will not discuss) to prove your identity.

Rebooting and Shutting Down

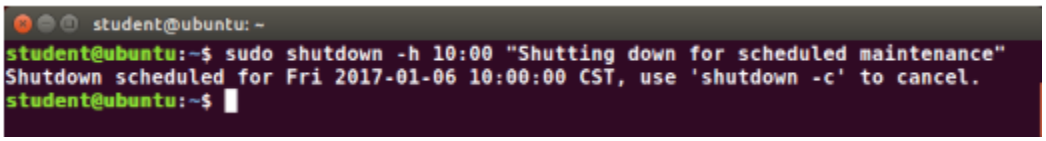
The preferred method to shut down or reboot the system is to use the shutdown command. This sends a warning message, and then prevents further users from logging in. The init process will then control shutting down or rebooting the system. It is important to always shut down properly; failure to do so can result in damage to the system and/or loss of data. The halt and poweroff commands
issue shutdown -h to halt the system; reboot issues shutdown -r and causes the machine to reboot

instead of just shutting down. Both rebooting and shutting down from the command line requires superuser (root) access.

When administering a multiuser system, you have the option of notifying all users prior to shutdown, as in:

sudo shutdown -h 10:00 "Shutting down for scheduled maintenance."

Note: On Ubuntu systems, the shutdown message is not broadcast to users currently on the system. This would appear to be a bug, as it violates the man page for shutdown.



Locating Applications

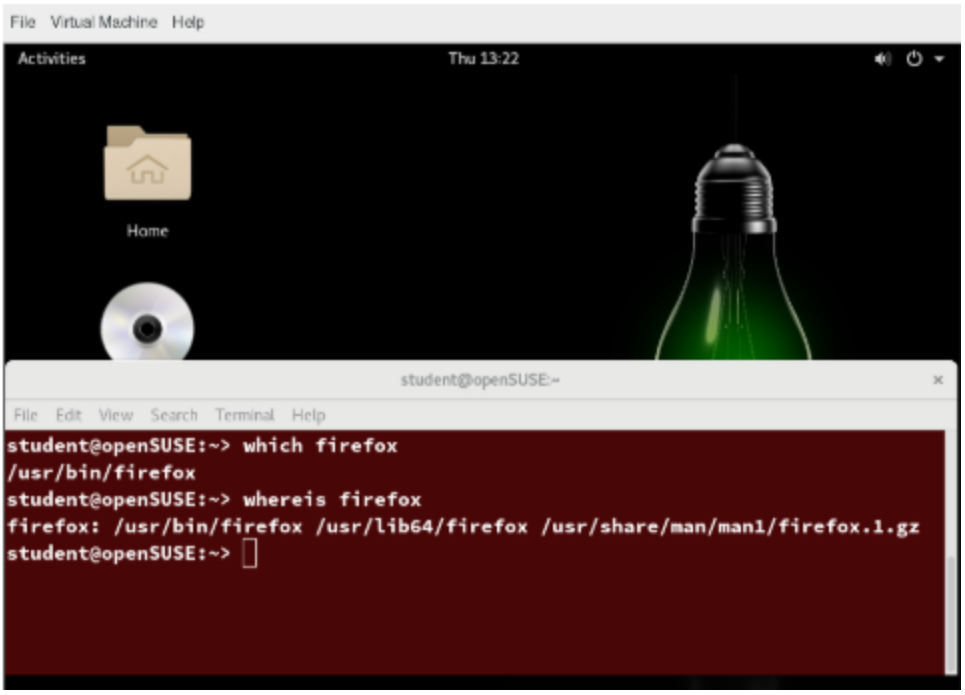
Depending on the specifics of your particular distribution's policy, programs and software packages can be installed in various directories. In general, executable programs and scripts should live in the **/bin**, **/usr/bin**, **/sbin**, **/usr/sbin** directories, or somewhere under **/opt**. They can also appear in **/usr/local/bin** and **/usr/local/sbin**, or in a directory in a user's account space, such as **/home/student/bin**.

One way to locate programs is to employ the **which** utility. For example, to find out exactly where the diff program resides on the filesystem:

```
$ which diff
/usr/bin/diff
```

If **which** does not find the program, **whereis** is a good alternative because it looks for packages in a broader range of system directories:

```
$ whereis diff
diff: /usr/bin/diff /usr/share/man/man1/diff.1.gz /usr/share/man/man1p/diff.1p.gz
```



Accessing Directories

When you first log in to a system or open a terminal, the default directory should be your Home directory. You can print the exact path of this by typing **echo $Home.** Many Linux distributions actually open new graphical terminals in **$HOME/Desktop**. The following commands are useful for directory navigation:

Command	Result
`pwd`	Displays the present working directory
`cd ~` or `cd`	Change to your home directory (shortcut name is ~ (tilde))
`cd ..`	Change to parent directory (`..`)
`cd -`	Change to previous directory (`-` (minus))

Understanding Absolute and Relative path

There are two ways to identify paths:

- Absolute path: An absolute pathname begins with the root directory and follows the tree, branch, until it reaches the desired directory or file. Absolute paths always start with **/**.
- Relative path: A relative pathname starts from the present working directory. Relative paths never start with **/**.

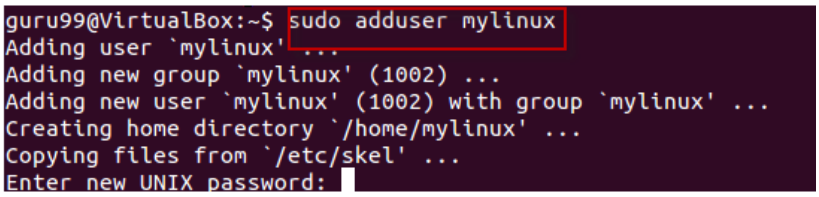
User Management

As Linux is a multi-user operating system, there is a high need of an administrator, who can manage user accounts, their rights, and the overall system security.

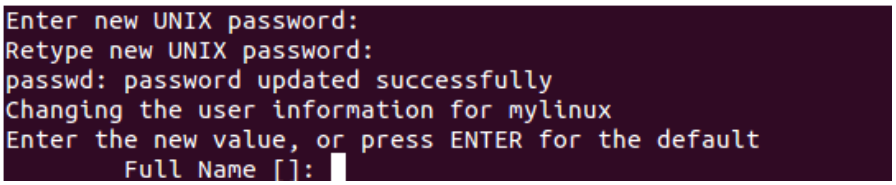
Creating a user:

In Linux, every user is assigned an individual account which contains all the files, information, and data of the user. You can create multiple users in a Linux operating system. The steps to creating a user are:

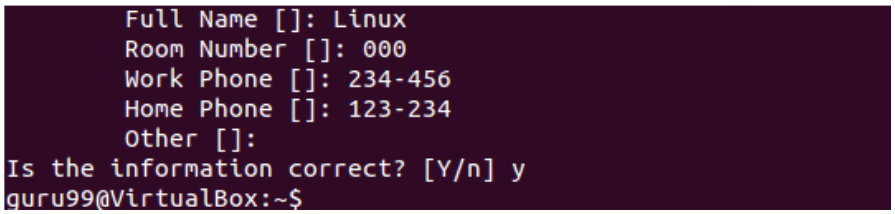
Step 1) Use command sudo adduser



Step 2) Enter password for the new account and confirm



Step 3) Enter details of the new user and press Y



Deleting, disabling account

For disabling an account using Terminal, remove the password set on the account.

sudo passwd -l ‘username’

```
guru99@VirtualBox:~$ sudo passwd -l ubuntu
[sudo] password for guru99:
passwd: password expiry information changed.
guru99@VirtualBox:~$
```

To delete an account, use the command –

sudo userdel -r 'username'

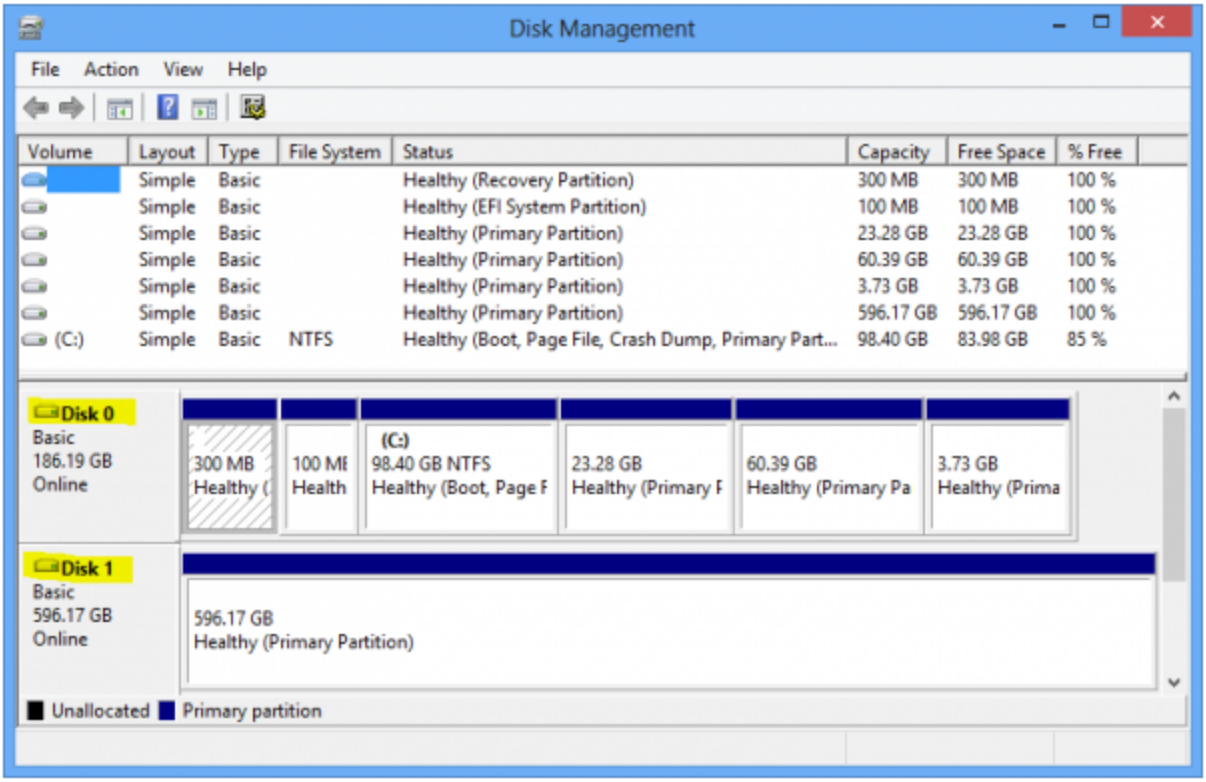
```
guru99@VirtualBox:~$ sudo userdel -r ubuntu
```

Optional. The -r parameter removes the user's home directory and mail spool file if they exist.

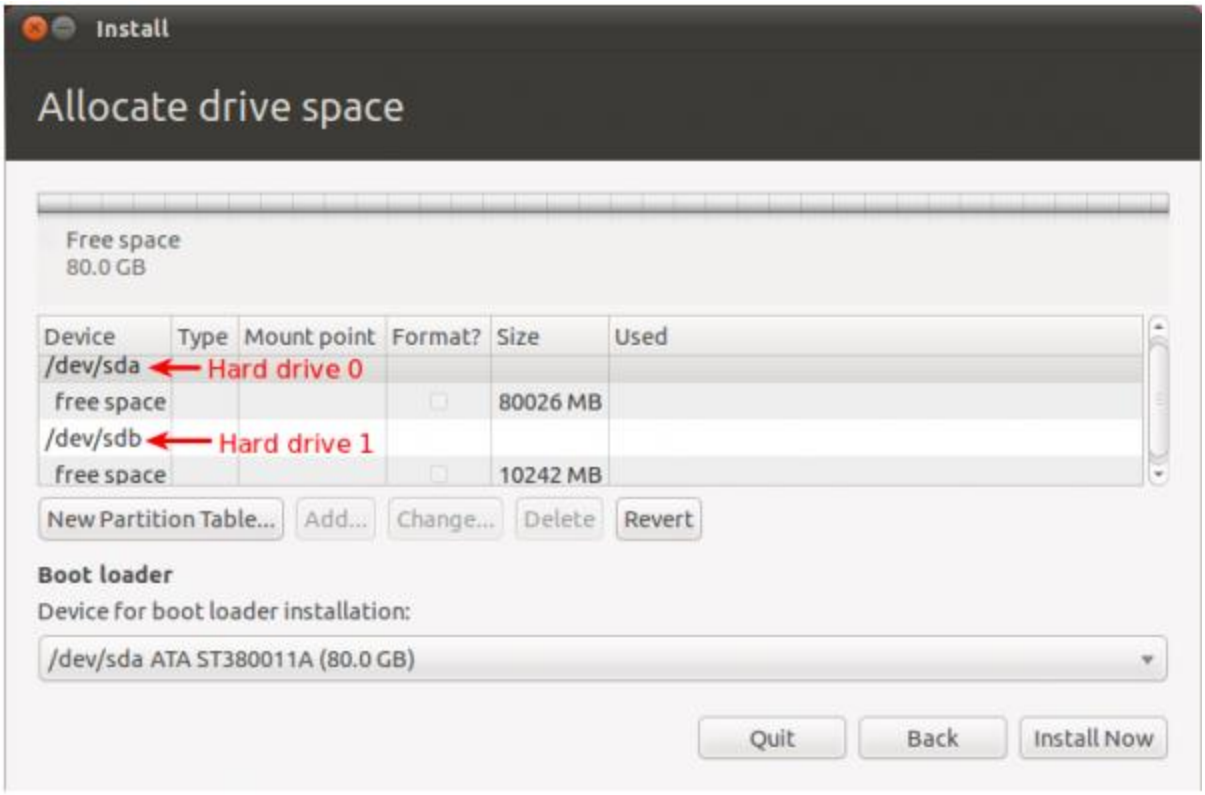
Storage – Disks and partitions

1. Hard Drive Naming Convention: The first thing you need to know is this: There's no C or D drive in in Linux. There are equivalents, but when you come across a reference to a hard drive in Linux, you'll typically see something like /dev/sda, /dev/sdb, /dev/sdc, … etc. The "dev" is short for device, and, in this case, a block storage device. The "sd" is short for SCSI mass-storage driver. (SCSI stands for Small Computer System Interface.) For the rest of this article, the "/dev/" part will be dropped, so all references to hard drives (and partitions) will start with the last part only.

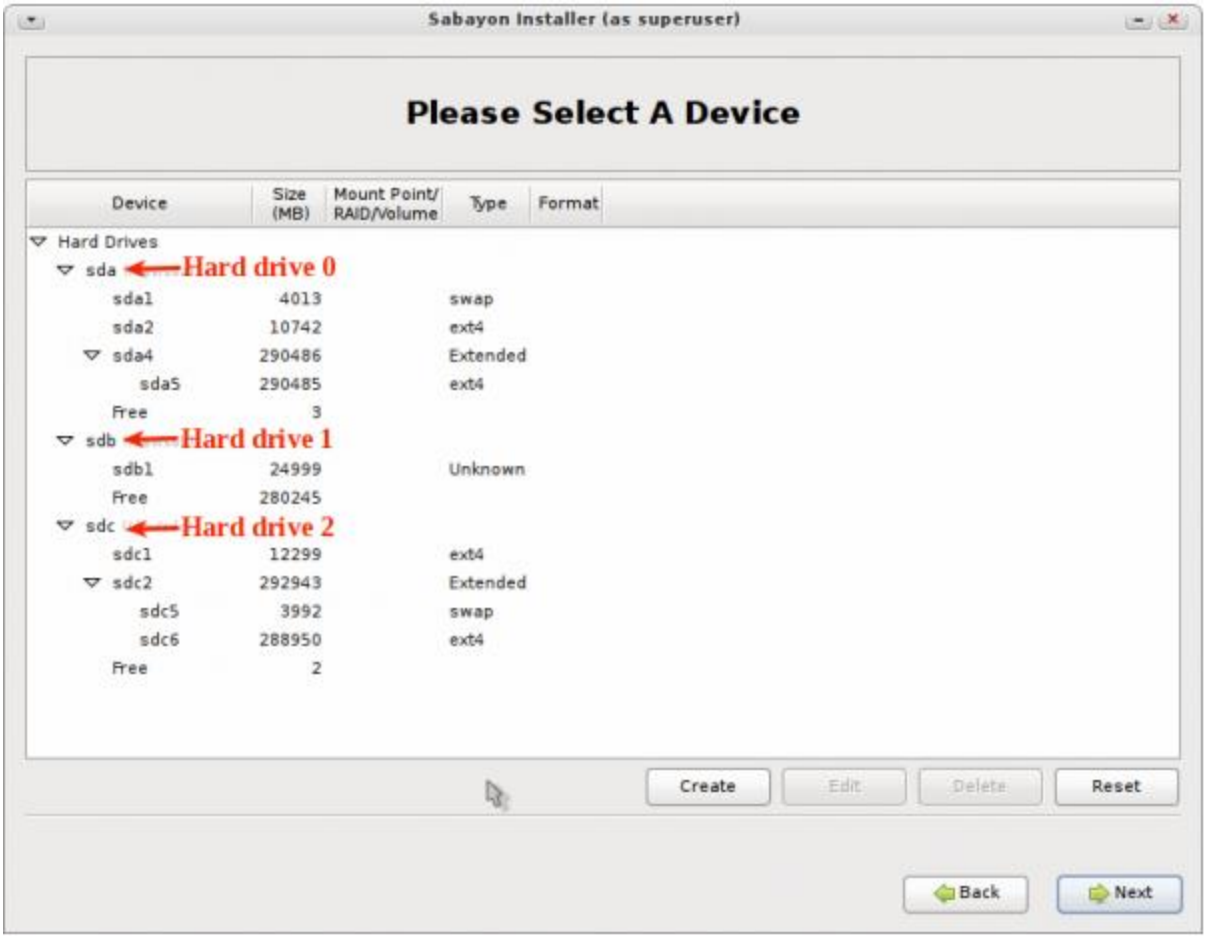
Let's begin by taking a look at how hard drives are represented in Windows. Below figure was taken from a Windows 8 installation on a computer with two hard drives attached – **Disk 0** and **Disk 1**.



Below figure shows how those hard drives would be represented in Linux. Where Windows sees Disk 0 and Disk 1, Linux takes a different approach. The first hard drive detected by a Linux system carries the *sda* label. In numerical terms, it is hard drive 0 (zero; counting begins from 0, not 1). The second hard drive is *sdb*, the third drive, *sdc*, etc. In the screenshot below, there are two hard drives detected by the installer – *sda* and *sdb*.



Below figure comes from a Linux system with three hard drives attached. So that's how hard drives are referenced in Linux – sda, sdb, sdc, sdd, sde, …, sdz…



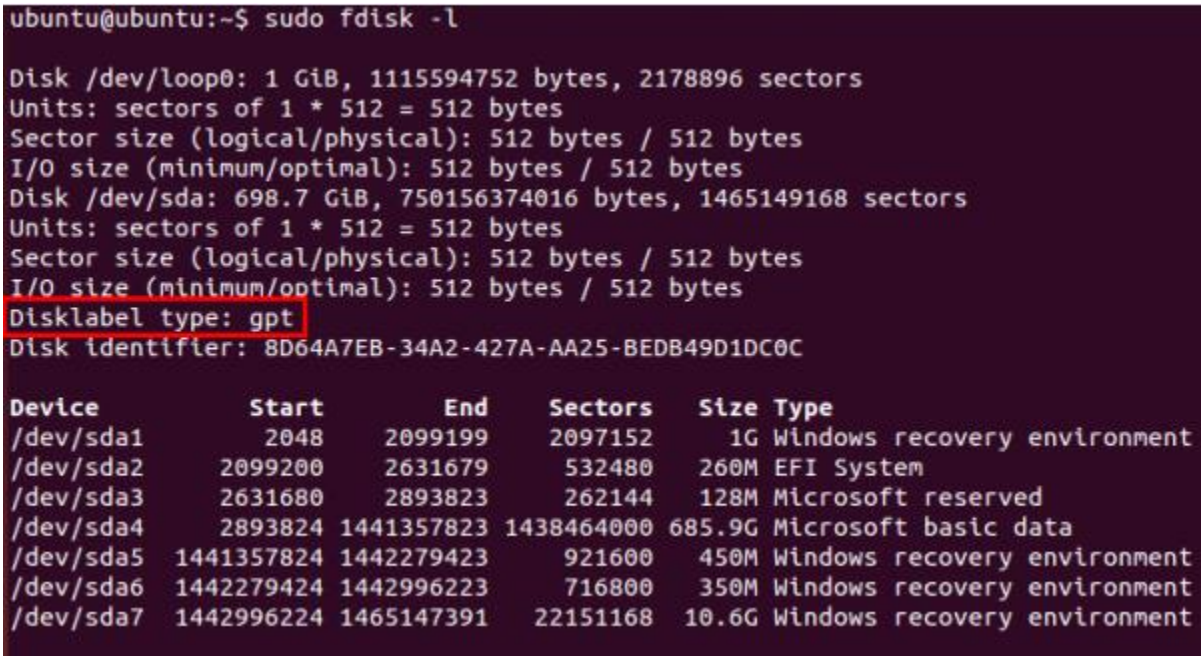
2. Partition Tables: In simple terms, a partition table describes the layout of partitions of a hard drive. There are two partition table standards – MBR (Master Boot Record) and GPT (GUID Partition Table). MBR, also know as **ms-dos**, is what you might call the first standard. GPT came much later.

The MBR partitioning scheme is what you'll find on older computers. Newer computers support both schemes, so it's still possible to use an MBR partitioning scheme on those computers. MBR's major limitations led to the development of GPT. Those limitations are:

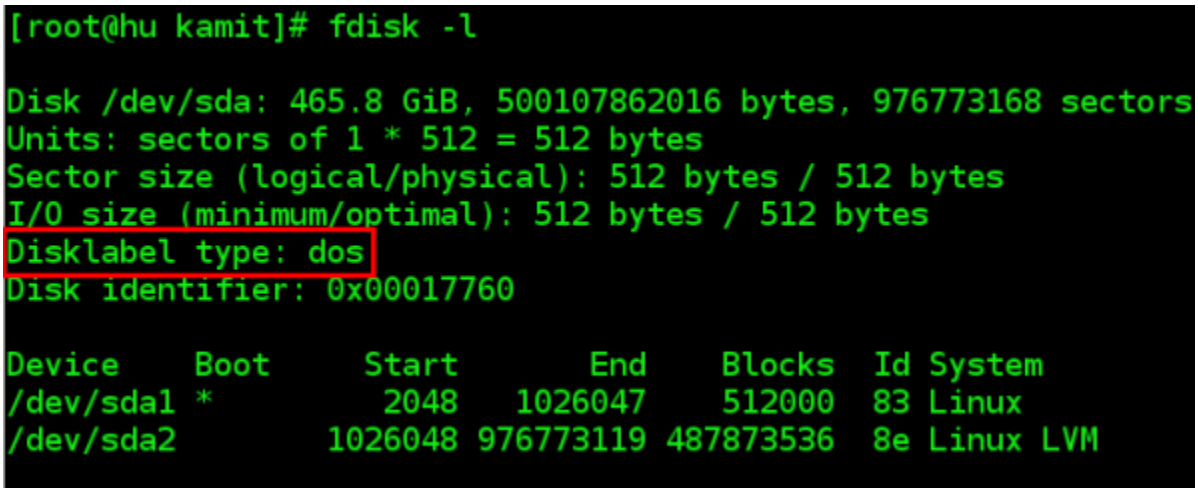
a. It does not allow the configuration of more than four main partitions. Those partitions are called primary partitions.
b. Disk partitions are limited to 2TB

Newer computers come with a replacement firmware for the old BIOS system called UEFI (Unified Extensible Firmware interface), and GPT is a part of the UEFI standard. If you bought a Windows 8 computer, it's most certainly installed on a GPT partitioning scheme. If you're already running a Linux distribution or attempting to install one on a recent Windows 7/8 computer, the easiest method of finding out what partitioning standard is used is to launch a shell terminal and type **sudo fdisk –l**

Below figure shows the output of sudo fdisk -l from a Ubuntu Linux installation. The Disklabel type: gpt line confirms that GPT is in use.



Below figure was taken from a Fedora Linux installation. As in above figure, the **Disklabel type: dos** line confirms what partitioning scheme is in use. In this case, it is MBR.



You can also tell whether GPT or MBR is in use by accessing the UEFI setup utility. Under the **Boot** menu, look for **PCI ROM Priority**. You should see two options – **EFI Compatible ROM** and **Legacy ROM**. The latter indicates MBR.