**EXPERIMENT 1**

**Introduction**: History, Linux Foundation, Linux Requirements, Linux Components, Distributions, Features, Linux Architecture, Kernel, Difference between Windows and Linux

**Linux History**

Linux is a free open source computer operating system initially developed for Intel **x86**-based personal computers. It has been subsequently ported to many other hardware platforms. Linus Torvalds was a student in Helsinki, Finland, in 1991 when he started a project: writing his own operating system kernel. He also collected together and/or developed the other essential ingredients required to construct an entire operating system with his kernel at the center. This soon became known as the Linux kernel.  In 1992, Linux was re-licensed using the General Public License (GPL) by GNU (a project of the Free Software Foundation (FSF) which promotes freely available software) which made it possible to build a worldwide community of developers. By combining the kernel with other system components from the GNU project, numerous other developers created complete systems called Linux Distributions in the mid-90. The Linux distributions created in the mid-90’s provided the basis for fully free computing and became a driving force in the open source software movement. In 1998, major companies like IBM and Oracle announced support for the Linux platform and began major development efforts as well. Today, Linux powers more than half of the servers on the Internet, the majority of smart-phones (via the Android system which is built on top of Linux), and nearly all of the world’s most powerful supercomputers.

**Linux Foundation**

Since its inception in 1991, Linux has grown to become a major force in computing. The **Linux Foundation** is a nonprofit organization that sponsors the work of Linux creator Linus Torvalds. It was founded in 2000 and its mission is to promote, protect, and advance Linux. The Linux Foundation is supported by leading technology companies and thousands of individual members from around the world and marshal’s the resources of its members and the open source development community to ensure that Linux remains free and technically advanced.

The Linux Foundation is active on many fronts.

* Produces technical events throughout the world.
* Develops and delivers training programs.
* Hosts major collaborative projects and industry initiatives.
* Manages kernel.org where the official versions of the Linux kernel are released.
* Runs the popular website linux.com

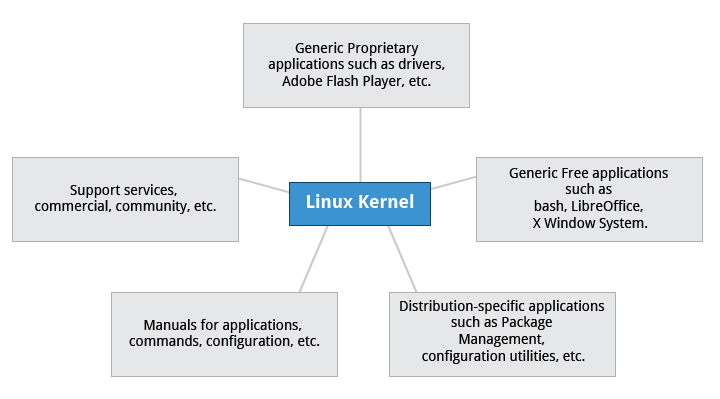
**Linux Foundation Events**

The Linux Foundation hosts conferences and other events throughout the world which bring community members together in person. These events:

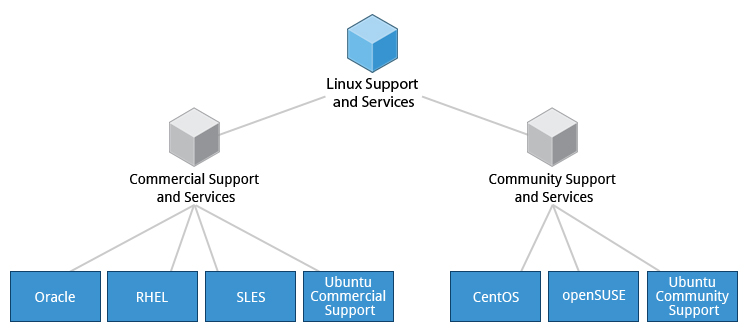
* Provide an open forum for development of the next kernel release.
* Bring together developers and system administrators to solve problems in a real-time environment.
* Host workgroups and community groups for active discussions.
* Connect end users, system administrators, and kernel developers in order to grow Linux use in the enterprise.
* Encourage collaboration among the entire community.
* Provide an atmosphere that is unmatched in its ability to further the platform.

**Linux distributions**

* Linux kernel is the core of a computer operating system. A full **Linux distribution** consists of the kernel plus a number of other software tools for file-related operations, user management, and software package management. Each of these tools provides a small part of the complete system. Each tool is often its own separate project, with its own developers working to perfect that piece of the system.
* Examples of other essential tools and ingredients provided by distributions include the **C/C++** compiler, the **gdb** debugger, the core system libraries applications need to link with in order to run, the low-level interface for drawing graphics on the screen as well as the higher-level desktop environment, and the system for installing and updating the various components including the kernel itself.



**Services Associated with Distributions**

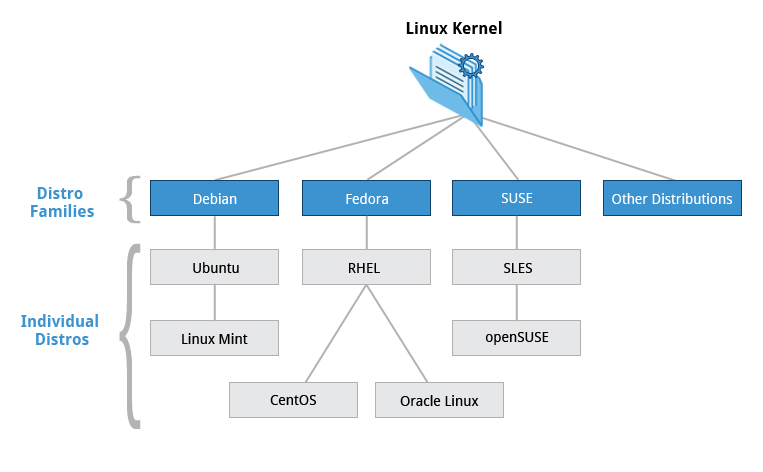


* A vast variety of Linux distributions cater to different audiences and organizations depending on their specific needs. Large commercial organizations tend to favor the commercially supported distributions from Red Hat, SUSE and Canonical (Ubuntu).
* CentOS is a popular free alternative to Red Hat Enterprise Linux (RHEL). Ubuntu and Fedora are popular in the educational realm. Scientific Linux is favored by the scientific research community for its compatibility with scientific and mathematical software packages. Both CentOS and Scientific Linux are binary-compatible with RHEL; i.e., binary software packages in most cases will install properly across the distributions.
* Many commercial distributors, including Red Hat, Ubuntu, SUSE, and Oracle, provide long term fee-based support for their distributions, as well as hardware and software certification. All major distributors provide update services for keeping your system primed with the latest security and bug fixes, and performance enhancements, as well as provide online support resources.

**Linux Distributions and Software Requirements**

There are many distribution families and hundreds of distributions available. Families and their distributions that are fairly widely used and are broadly representative of their respective family are:

* Debian Family Systems (such as Ubuntu)
* SUSE Family Systems (such as openSUSE)
* Fedora Family Systems (such as CentOS)



**Fedora Family**

**Fedora** is the community distribution that forms the basis of Red Hat Enterprise Linux (RHEL), CentOS, Scientific Linux, and Oracle Linux .Fedora contains significantly more software than Red Hat’s enterprise version. One reason for this is that a diverse community is involved in building Fedora; it is not just a company.

Once installed, CentOS is also virtually identical to Red Hat Enterprise Linux, which is the most popular Linux distribution in enterprise environments.

**Key Facts of the Fedora Family**

* The Fedora family is upstream for CentOS, RHEL, and Oracle Linux.
* The Linux kernel 2.6.32 is used in RHEL/CentOS 6.x
* It supports hardware platforms such as x86, x86-64, Itanium, PowerPC, and IBM System z.
* It uses the RPM-based yum package manager (we cover in more detail later) to install, update, and remove packages in the system.
* RHEL is widely used by enterprises which host their own systems.

**SUSE Family**

The relationship between **SUSE,**SUSE **Linux Enterprise Server (SLES),** and **openSUSE**is similar to the one described between **Fedora, Red Hat Enterprise Linux,** and**CentOS**. In this  **openSUSE** 12.3 as the reference distribution for the **SUSE** family is available to end users at no cost. The two products are extremely similar, and material that covers **openSUSE** can typically be applied to **SLES** with no problem.

**Key Facts about the SUSE Family**

* SUSE Linux Enterprise Server (SLES) is upstream for openSUSE.
* The Linux kernel 3.11 is used in openSUSE 12.3.
* It uses the RPM-based zypper package manager (we cover in more detail later) to install, update, and remove packages in the system.
* It includes the YaST (Yet another System Tool) application for system administration purposes.
* SUSE is widely used in the retail sector.

**Debian Family**

The Debian distribution is upstream for several other distributions including Ubuntu, and Ubuntu is upstream for Linux Mint and others. It is commonly used on both servers and desktop computers. Debian is a pure open source project and focuses on one key aspect, that is, stability. It also provides the largest and most complete software repository to its users. Ubuntu aims at providing a good compromise between long term stability and ease of use. Since Ubuntu gets most of its packages from Debian’s stable branch, Ubuntu also has access to a very large software repository.

**Key Facts about the Debian Family**

* The Debian family is upstream for Ubuntu, and Ubuntu is upstream for Linux Mint and others.
* The Linux kernel 3.13 is used in Ubuntu 14.04.
* It uses the DPKG-based apt-get package manager (we cover in more detail later) to install, update, and remove packages in the system.
* Ubuntu has been widely used for cloud deployments.
* Ubuntu is built on top of Debian, it uses the Unity graphical interface, is GNOME-based and differs quite a bit visually from the interface on standard Debian as well as other distributions.

While choosing a Linux distribution system, you will notice that the technical differences are mainly about package management systems, software versions, and file locations. Once you get a grasp of those differences it becomes relatively painless to switch from one Linux distribution to another.

**Terminology**

Kernel: Interface between the hardware and the user applications. It is the core part of the system.

Distribution: Collection of software making up a Linux-based OS.

Boot Loader: Program that boots the operating system.

Service: Program that runs as a background process.

File system: Method for storing and organizing files.

X window System: Toolkit for developing graphical subsystem on nearly all linux systems.

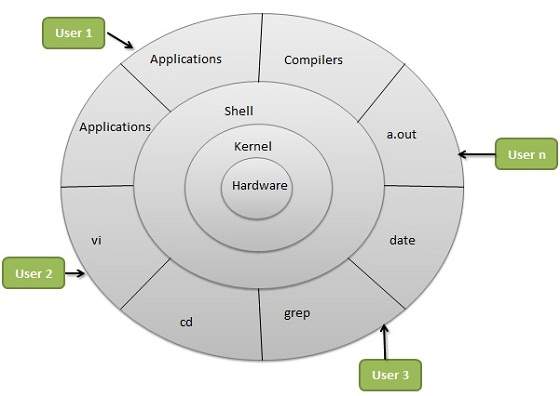
Desktop Environment: GUI on top of the operating system.

Command Line: Interface for typing commands on top of the OS.

Shell: Command line interpreter that interprets the commands and instructs the kernel to perform required tasks and commands.

**Linux System Architecture**

The Linux Operating System’s architecture primarily has these components: the Kernel, Hardware layer, System library, Shell and System utility.

=

The kernel is the core part of the operating system, which is responsible for all the major activities of the LINUX operating system. This operating system consists of different modules and interacts directly with the underlying hardware. The kernel offers the required abstraction to hide application programs or low-level hardware details to the system. The types of Kernels are as follows:

* Monolithic Kernel
* Micro kernels
* Exo kernels
* Hybrid kernels

2. System libraries are special functions, that are used to implement the functionality of the operating system and do not require code access rights of kernel modules.

3. System Utility programs are liable to do individual and specialized-level tasks.

4. Hardware layer of the LINUX operating system consists of peripheral devices such as RAM, HDD, and CPU.

5. The shell is an interface between the user and the kernel, and it affords services of the kernel. It takes commands from the user and executes kernel’s functions. The Shell is present in different types of operating systems, which are classified into two types: command line shells and graphical shells.

The command line shells provide a command line interface, while the graphical line shells provide a graphical user interface. Though both shells perform operations, but the graphical user interface shells perform slower than the command line interface shells. Types of shells are classified into four:

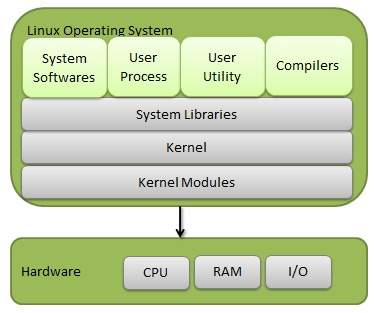
* Korn shell
* Bourne shell
* C shell
* POSIX shell

Linux is one of popular version of UNIX operating System. It is open source as its source code is freely available. It is free to use. Linux was designed considering UNIX compatibility. Its functionality list is quite similar to that of UNIX.

**Components of Linux System**

Linux Operating System has primarily three components

* **Kernel** − Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.
* **System Library** − System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features. These libraries implement most of the functionalities of the operating system and do not requires kernel module's code access rights.
* **System Utility** − System Utility programs are responsible to do specialized, individual level tasks.



**Kernel Mode vs User Mode**

Kernel component code executes in a special privileged mode called **kernel mode** with full access to all resources of the computer. This code represents a single process, executes in single address space and do not require any context switch and hence is very efficient and fast. Kernel runs each processes and provides system services to processes, provides protected access to hardware to processes.

Support code which is not required to run in kernel mode is in System Library. User programs and other system programs works in **User Mode** which has no access to system hardware and kernel code. User programs/ utilities use System libraries to access Kernel functions to get system's low level tasks.

### Features of Linux Operating System

The main features of Linux operating system are

**Portable:** Linux operating system can work on different types of hardware’s as well as Linux kernel supports the installation of any kind of hardware platform.

**Open Source:** Source code of LINUX operating system is freely available and, to enhance the ability of the LINUX operating system, many teams work in collaboration.

**Multiuser:** Linux operating system is a multiuser system, which means; multiple users can access the system resources like RAM, Memory or Application programs at the same time.

**Multiprogramming:** Linux operating system is a multiprogramming system, which means multiple applications can run at the same time.

**Hierarchical File System:** Linux operating system affords a standard file structure in which system files or user files are arranged.

**Shell:** Linux operating system offers a special interpreter program that can be used to execute commands of the OS. It can be used to do several types of operations like call application programs, and so on.

**Security:** Linux operating system offers user security systems using authentication features like encryption of data or password protection or controlled access to particular files.

| **BASIS FOR COMPARISON** | **LINUX** | **WINDOWS** |
| --- | --- | --- |
| Cost | Free of cost | Expensive |
| Open source | Yes | No |
| Customizable | Yes | No |
| Security | More secure | Vulnerable to viruses and malware attacks. |
| Booting | Either primary or logical partition. | Only primary partition. |
| Separation of the directories using | Forward slash | Back slash |
| Filenames | Case sensitive | Case insensitive |
| File system | EXT2, EXT3, EXT4, Reisers FS, XFS and JFS | FAT, FAT32, NTFS and ReFS |
| Type of kernel used | Monolithic kernel | Microkernel |
| Efficiency | Effective running efficiency | Lower than Linux |

**Difference between LINUX and UNIX**

The primary difference is that **Linux**and **Unix**are two different Operating Systems though they both have some common commands.

Let’s talk about **Linux** OS first.

* The source code of Linux is freely available to it’s users.
* Linux primarily uses Graphical User Interface with an optional Command Line Interface.
* Linux OS is portable and can be executed in different hard drives.
* Linux is very flexible and can be installed on most of the home based PCs.
* Linux is used on home based PCs, Mobile Phones, Desktops, etc.
* Different versions of Linux are Ubuntu, Linux Mint, RedHat, Solaris, etc.
* Linux installation is economical and doesn’t require much specific and high end hardware.
* The file systems supported by Linux are as follows: xfs, ramfs, nfs, vfat, cramfsm, ext3, ext4, ext2, ext1, ufs, autofs, devpts, ntfs.
* Linux is development by an active Linux Community worldwide.

Now comes the **UNIX** OS.

* The source code of Unix is not available for the general public.
* Unix primarily uses Command Line Interface.
* Unix is not portable.
* Unix has a rigid environment of the hardware. Hence, cannot be installed on every other machine.
* Unix is mainly used in Server Systems, Mainframes and High End Computers.
* Different Versions of Unix are AIS, HP-UX, BSD, Iris, etc.
* Unix installation is comparatively costlier as it requires more specific hardware circuitry.
* The file systems supported by Unix are as follows: zfs, js, hfx, gps, xfs, vxfs.
* Unix is developed by AT&T Developers.

**Viva Questions:**

* Name some of the distribution families available.
* Give the software requirements for installation of Ubuntu.
* What is Linux Foundation?
* What is a boot Loader?
* What do you mean by Kernel and Shell?
* Red hat linux belongs to which family?
* Ubuntu linux belongs to which family?
* What is the purpose of linux foundation?
* What kind of events is held by linux foundation?
* What do you mean by platform independence?

**Assignment:**

Briefly describe why Ubuntu is suitable distribution for installation in your practice labs?

**EXPERIMENT 2**

**Installation**: Configuration & Customizations of Linux, Linux Structure and Installation Linux file-system basics, the boot process, partitioning, dual boot.

**1. Requirements**

You'll need to consider the following before starting the installation:

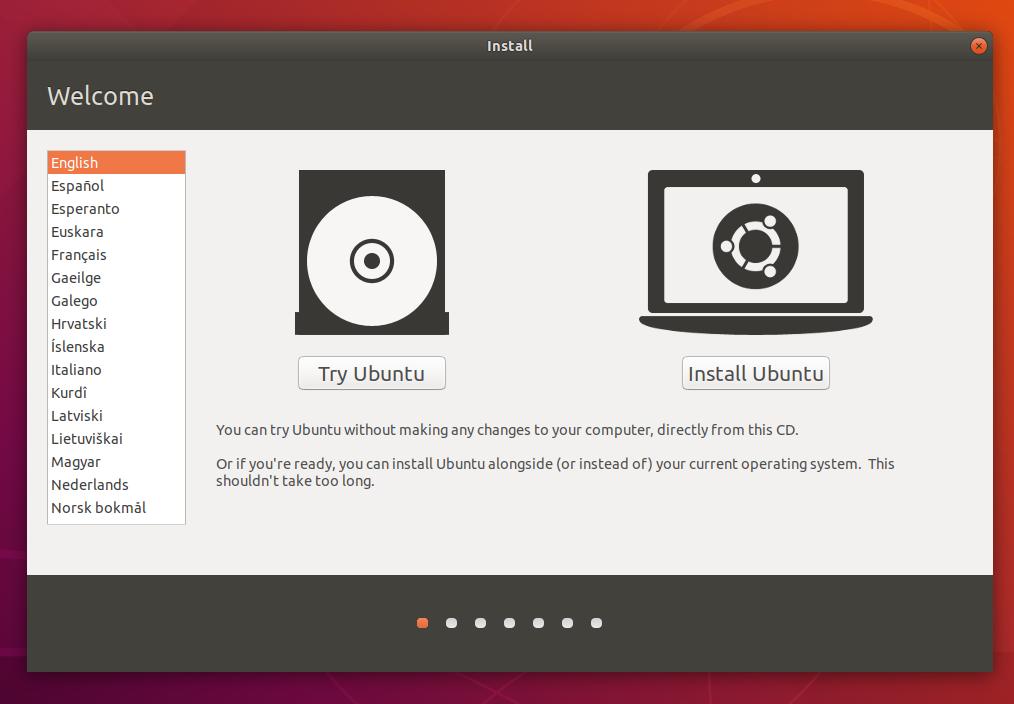
* Connect your laptop to a power source.
* Ensure you have at least 25GB of free storage space, or 5GB for a minimal installation.
* Have access to either a DVD or a USB flash drive containing the version of Ubuntu you want to install.
* Make sure you have a recent backup of your data. While it's unlikely that anything will go wrong, you can never be too prepared.

**2. Boot from DVD**

It's easy to install Ubuntu from a DVD. Here's what you need to do:

1. Put the Ubuntu DVD into your optical/DVD drive.
2. Restart your computer.

As soon as your computer boots up you'll see the welcome window.



**3. Boot from USB flash drive**

Most computers will boot from USB automatically. Simply insert the USB flash drive and either power on your computer or restart it. You should see the same welcome window we saw in the previous ‘Install from DVD' step, prompting you to choose your language and either install or try the Ubuntu desktop.

If your computer doesn't automatically boot from USB, try holding F12 when your computer first starts. With most machines, this will allow you to select the USB device from a system-specific boot menu.

F12 is the most common key for bringing up your system's boot menu, but Escape, F2 and F10 are common alternatives. If you're unsure, look for a brief message when your system starts - this will often inform you of which key to press to bring up the boot menu.

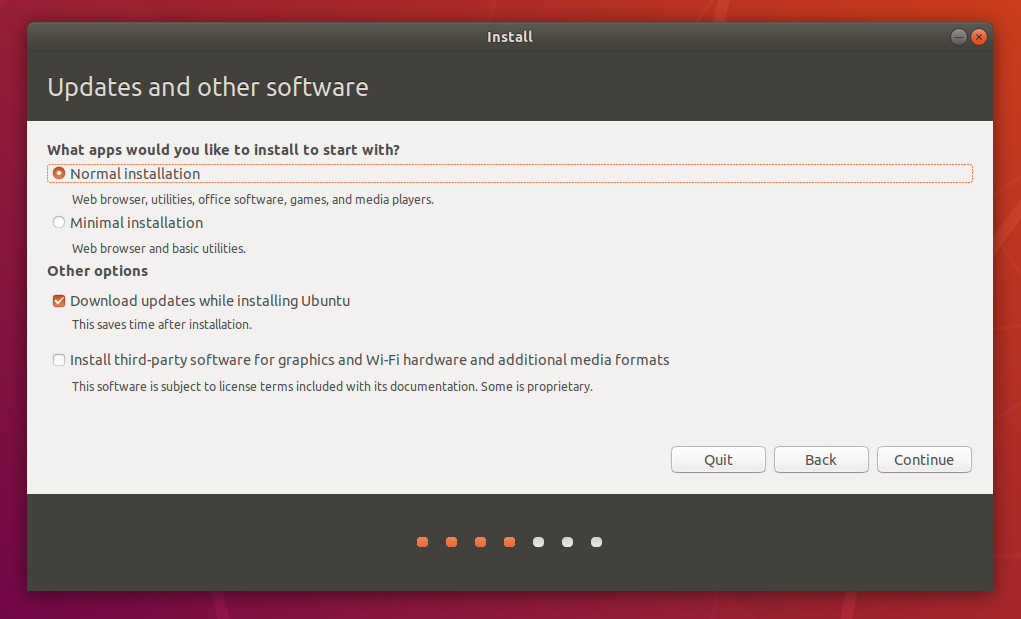
**4. Prepare to install Ubuntu**

You will first be asked to select your keyboard layout. If the installer doesn't guess the default layout correctly, use the ‘Detect Keyboard Layout' button to run through a brief configuration procedure.

After selecting *Continue* you will be asked *what apps would you like to install to start with.* The two options are ‘Normal installation' and ‘Minimal installation'. The first is the equivalent to the old default bundle of utilities, applications, games and media players - a great launch pad for any Linux installation. The second takes considerably less storage space and allows you to install only what you need.

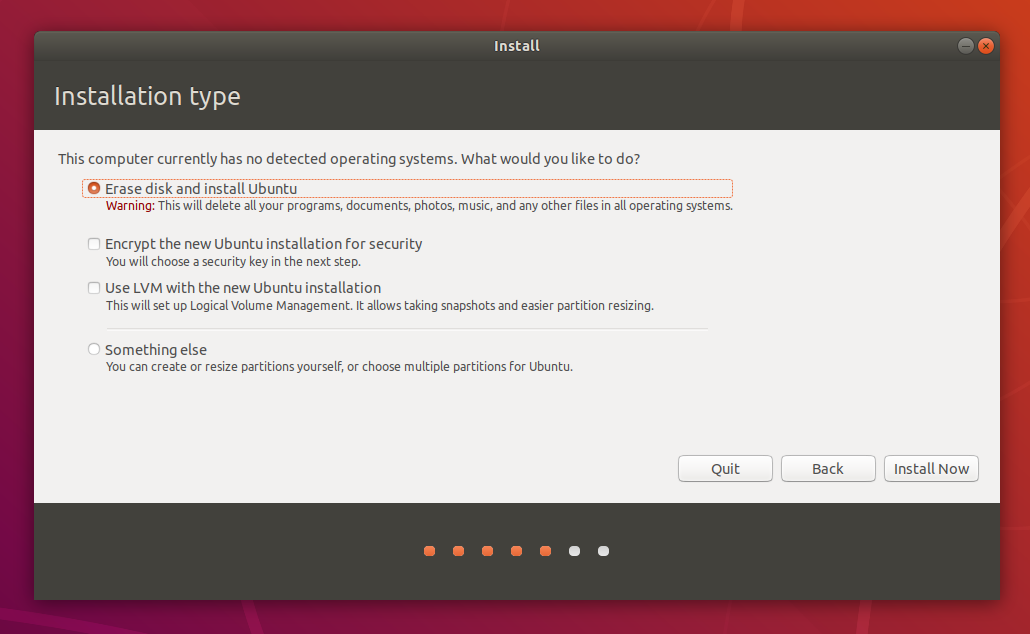
Beneath the installation-type question are two checkboxes; one to enable updates while installing and another to enable third-party software.

* We advise enabling both Download updates and Install third-party software.
* Stay connected to the internet so you can get the latest updates while you install Ubuntu.
* If you are not connected to the internet, you will be asked to select a wireless network, if available. We advise you to connect during the installation so we can ensure your machine is up to date



**5.  Allocate drive space**

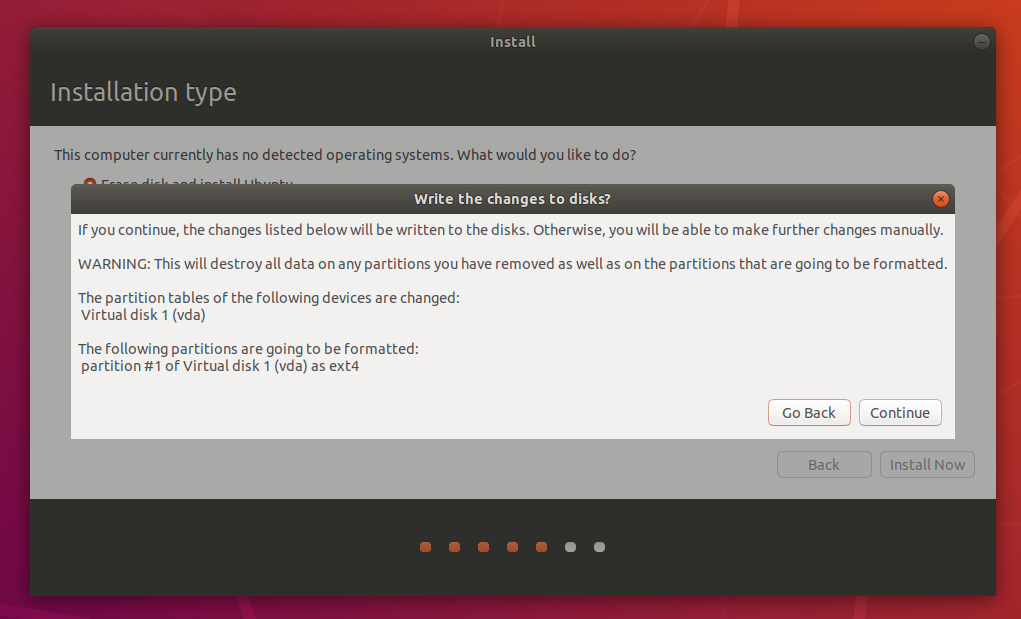
Use the checkboxes to choose whether you'd like to install Ubuntu alongside another operating system, delete your existing operating system and replace it with Ubuntu, or — if you're an advanced user — choose the '**Something else**' option.



**6. Begin installation**

After configuring storage, click on the ‘Install Now' button. A small pane will appear with an overview of the storage options you've chosen, with the chance to go back if the details are incorrect.

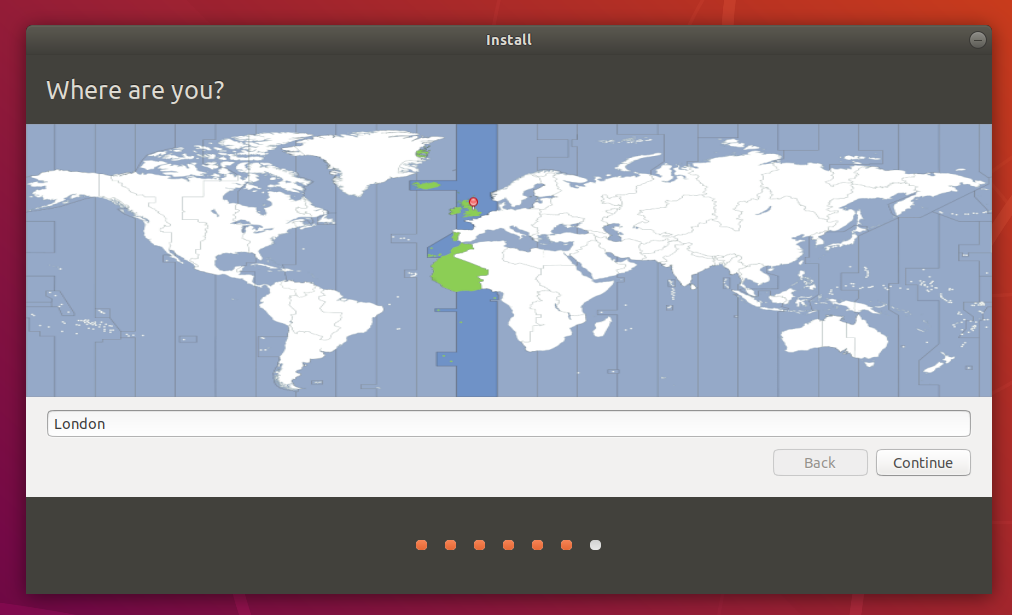
Click Continue to fix those changes in place and start the installation process.



**7. Select your location**

If you are connected to the internet, your location will be detected automatically. Check your location is correct and clicks 'Forward' to proceed.

If you're unsure of your time zone, type the name of a local town or city or use the map to select your location.

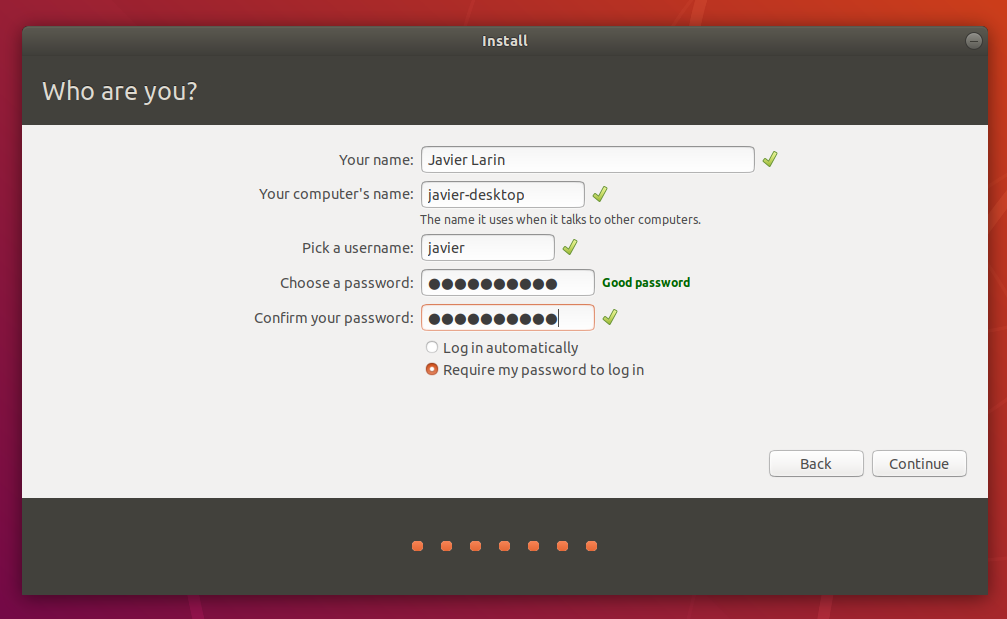


**8. Login details**

Enter your name and the installer will automatically suggest a computer name and username. These can easily be changed if you prefer. The computer name is how your computer will appear on the network, while your username will be your login and account name.

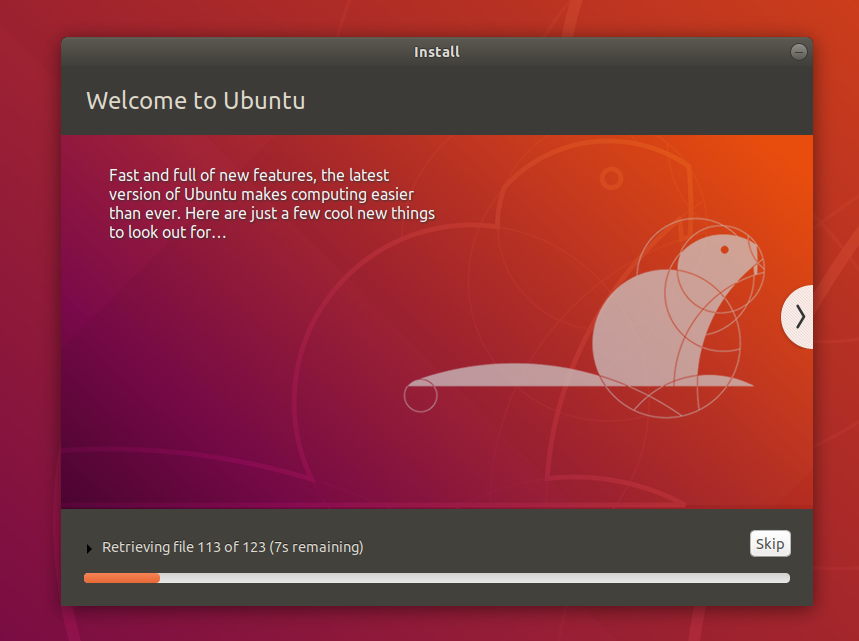
Next, enter a strong password. The installer will let you know if it's too weak.

You can also choose to enable automatic login and home folder encryption. If your machine is portable, we recommend keeping automatic login disabled and enabling encryption. This should stop people accessing your personal files if the machine is lost or stolen.



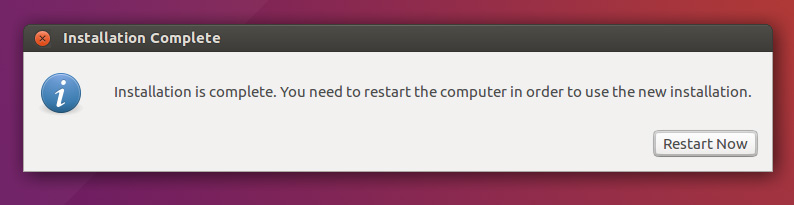
**9. Background installation**

The installer will now complete in the background while the installation window teaches you a little about how awesome Ubuntu is. Depending on the speed of your machine and network connection, installation should only take a few minutes.



**10. Installation complete**

After everything has been installed and configured, a small window will appear asking you to restart your machine. Click on Restart Now and remove either the DVD or USB flash drive when prompted. If you initiated the installation while testing the desktop, you also get the option to continue testing.



**fdisk command in Linux with examples**

fdisk also known as format disk is a dialog-driven command in Linux used for creating and manipulating disk partition table. It is used for the view, create, delete, change, resize, copy and move partitions on a hard drive using the dialog-driven interface.  
fdisk allows you to create a maximum of four primary partitions and the number of logical partition depends on the size of the hard disk you are using. It allows the user:

* To Create space for new partitions.
* Organizing space for new drives.
* Re-organizing old drives.
* Copying or Moving data to new disks(partitions).

**Synopsis:**

**fdsik [options] device**

**or**

**fdisk -l [device...]**

**1. View All Disk Partitions:**

The first thing to do before doing anything with the disks and partition is to view basic details about all available partition in the system. The below command is used to list the partitions on your system and see their */dev* names**. For example** /dev/sda, /dev/sdb or /dev/sdc.

**$ sudo fdisk -l**

**2. View Partition on a Specific Disk:**

Below command is used to view all disk partitions on device /dev/sda.

**$ sudo fdisk -l /dev/sda**

**3. View all fdisk Commands:**

To see all the command which are available under fdisk command you can use /dev/sda partition with fdisk command.

**$ sudo fdisk /dev/sda**

Note: This will prompt for a command. Type m for seeing all the operations which can perform on /dev/sda. After pressing m you will get:

**4. Create a Hard Disk Partition:** For this go inside the hard drive partition that is **the /dev/sdapartition, and use the following command:**

**$ sudo fdisk /dev/sda**

Now you have to type n to create new partition and then type p for making a primary partition and e for making an extended or logical partition depending on the type of partition to make.

**Run w command to write the changes and reboot your system.**

**5. Delete a Hard Disk Partition:** To delete a partition for the hard disk and free up space occupied by that partition for example /dev/sdb. Go to the command menu using follwing:

**$ sudo fdisk /dev/sda**

and then type d to go to the delete partition menu. It will prompt the partition number you want to delete(type the number).

Run w command to write the changes and reboot the system.

**6. How to view the size of your Partition:**

**$ sudo fdisk -s /dev/sda**

Note:

* To check for the manual page of fdisk command, use the following command:
* $ man fdisk
* To see the help message and listing of all options, use the following command option:
* $ sudo fdisk –h

**Linux File System**

**File system**, is the embodiment of a method of storing and organizing arbitrary collections of data in a human-usable form.

**Different Types of File systems Supported by Linux:**

* Conventional disk file systems: ext2, ext3, ext4, XFS, Btrfs, JFS, NTFS, etc.
* Flash storage file systems: ubifs, JFFS2, YAFFS, etc.
* Database file systems
* Special purpose files systems: procfs, sysfs, tmpfs, debugfs, etc.

**Partitions and File systems**

A partition is a logical part of the disk, whereas a filesystem is a method of storing/finding files on a hard disk (usually in a partition). By way of analogy, you can think of filesystems as being like family trees that show descendants and their relationships, while the partitions are like different families (each of which has its own tree).

A comparison between filesystems in Windows and Linux is given in the following table:

|  |  |  |
| --- | --- | --- |
| **Features** | **Windows** | **Linux** |
| Partition | Disk1 | /dev/sda1 |
| File System type | NTFS/FAT32 | EXT3/EXT4/XFS... |
| Mounting Parameters | Drive Letter | Mount Point |
| Base Folder where OS is stored | C drive | / |

**The File System Hierarchy Standard**

Linux systems store their important files according to a standard layout called the File system Hierarchy Standard, or FHS.

This standard ensures that users can move between distributions without having to re-learn how the system is organized. Linux uses the ‘/’ character to separate paths (unlike Windows, which uses ‘\’), and does not have drive letters. New drives are mounted as directories in the single file system, often under /media.

All Linux file system names are case-sensitive, so /boot, /Boot, and /BOOT represent three different directories (or folders). Many distributions distinguish between core utilities needed for proper system operation and other programs, and place the latter in directories under /usr (think "user").

**The Boot Process**

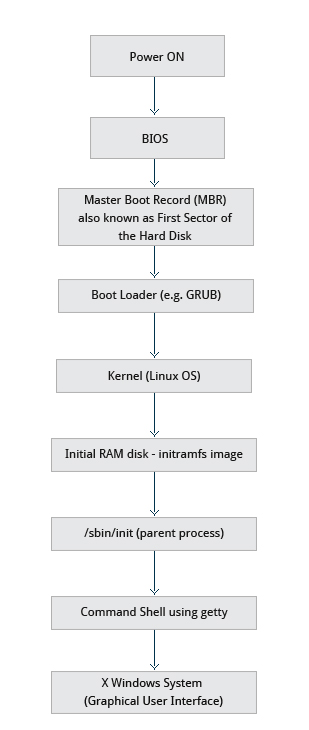
The Linux**boot process** is the procedure for initializing the system. It consists of everything that happens from when the computer power is first switched on until the user interface is fully operational.

Once you start using Linux, you will find that having a good understanding of the steps in the boot process will help you with troubleshooting problems as well as with tailoring the computer's performance to your needs.

**BIOS**

Starting an **x86-**based Linux system involves a number of steps. When the computer is powered on, the **Basic Input/Output System** (**BIOS)** initializes the hardware, including the screen and keyboard, and tests the main memory. This process is also called **POST** (**Power On Self Test**).

The BIOS software is stored on a ROM chip on the motherboard. After this, the remainder of the boot process is completely controlled by the operating system.



Press the power button on your system, and after few moments you see the Linux login prompt.

Have you ever wondered what happens behind the scenes from the time you press the power button until the Linux login prompt appears?

The following are the 6 high level stages of a typical Linux boot process.  
  
 

**1. BIOS**

* BIOS stands for Basic Input/Output System
* Performs some system integrity checks
* Searches, loads, and executes the boot loader program.
* It looks for boot loader in floppy, cd-rom, or hard drive. You can press a key (typically F12 of F2, but it depends on your system) during the BIOS startup to change the boot sequence.
* Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
* So, in simple terms BIOS loads and executes the MBR boot loader.

**2. MBR**

* MBR stands for Master Boot Record.
* It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda
* MBR is less than 512 bytes in size. This has three components 1) primary boot loader info in 1st 446 bytes 2) partition table info in next 64 bytes 3) mbr validation check in last 2 bytes.
* It contains information about GRUB (or LILO in old systems).
* So, in simple terms MBR loads and executes the GRUB boot loader.

3. GRUB

* GRUB stands for Grand Unified Bootloader.
* If you have multiple kernel images installed on your system, you can choose which one to be executed.
* GRUB displays a splash screen, waits for few seconds, if you don’t enter anything, it loads the default kernel image as specified in the grub configuration file.
* GRUB has the knowledge of the filesystem (the older Linux loader LILO didn’t understand filesystem).
* Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this). The following is sample grub.conf of CentOS.

#boot=/dev/sda

default=0

timeout=5

splashimage=(hd0,0)/boot/grub/splash.xpm.gz

hiddenmenu

title CentOS (2.6.18-194.el5PAE)

root (hd0,0)

kernel /boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/

initrd /boot/initrd-2.6.18-194.el5PAE.img

* As you notice from the above info, it contains kernel and initrd image.
* So, in simple terms GRUB just loads and executes Kernel and initrd images.

4. Kernel

* Mounts the root file system as specified in the “root=” in grub.conf
* Kernel executes the /sbin/init program
* Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a ‘ps -ef | grep init’ and check the pid.
* initrd stands for Initial RAM Disk.
* initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

**5. Init**

* Looks at the /etc/inittab file to decide the Linux run level.
* Following are the available run levels
  + 0 – halt
  + 1 – Single user mode
  + 2 – Multiuser, without NFS
  + 3 – Full multiuser mode
  + 4 – unused
  + 5 – X11
  + 6 – reboot
* Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.
* Execute ‘grep initdefault /etc/inittab’ on your system to identify the default run level
* If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
* Typically you would set the default run level to either 3 or 5.

**6. Runlevel programs**

* When the Linux system is booting up, you might see various services getting started. For example, it might say “starting send mail …. OK”. Those are the run level programs, executed from the run level directory as defined by your run level.
* Depending on your default init level setting, the system will execute the programs from one of the following directories.
  + Run level 0 – /etc/rc.d/rc0.d/
  + Run level 1 – /etc/rc.d/rc1.d/
  + Run level 2 – /etc/rc.d/rc2.d/
  + Run level 3 – /etc/rc.d/rc3.d/
  + Run level 4 – /etc/rc.d/rc4.d/
  + Run level 5 – /etc/rc.d/rc5.d/
  + Run level 6 – /etc/rc.d/rc6.d/
* Please note that there are also symbolic links available for these directory under /etc directly. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d.
* Under the /etc/rc.d/rc\*.d/ directories, you would see programs that start with S and K.
* Programs starts with S are used during startup. S for startup.
* Programs starts with K are used during shutdown. K for kill.
* There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed.
* For example, S12syslog is to start the syslog daemon, which has the sequence number of 12. S80sendmail is to start the send mail daemon, which has the sequence number of 80. So, syslog program will be started before send mail.

**Viva Questions:**

* Define BIOS
* What do you mean by POST?
* What is the first sector of Hard disk known as?
* Why is boot program always stored on a firmware?
* Identify Linux file systems.
* Identify the differences between partitions and file systems.
* Describe the boot process.
* Know how to install Linux on a computer.
* Name different flash storage file systems?
* Name different database file systems?
* Name conventional file systems?
* What is the command gunzip used for?
* What is intramfs?
* What is GRUB2 boot Loader?
* What is the command chmod used for?
* What does UID stand for?
* What do you mean by root?
* What do you mean by a superuser?
* What is boot loader?
* What do you mean by boot program?
* What is the first process initialized by the boot loader?

**Assignment:**

* Install Ubuntu on your personal laptops.
* Identify Linux file systems.
* Identify the differences between partitions and file systems.