

LINUX BASICS

SAMANTI SARKAR
SC-E, ISSD

- linux at a glance

The linux operating system is a set of programs that acts as a link between the computer and the user.

- The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the **operating system** or the **kernel**.
- Users communicate with the kernel through a program known as the **shell**. The shell is a command line interpreter; it translates commands entered by the user and converts them into a language that is understood by the kernel.
- Several people can use a linux computer at the same time; hence linux is called a multiuser system.
- A user can also run multiple programs at the same time; hence linux is a multitasking environment.

Basic Features

- Following are some of the important features of Linux Operating System.
- **Portable** – Portability means software can work on different types of hardware in same way. Linux kernel and application programs supports their installation on any kind of hardware platform.
- **Open Source** – Linux source code is freely available and it is community based development project. Multiple teams work in collaboration to enhance the capability of Linux operating system and it is continuously evolving.
- **Multi-User** – Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at the same time.
- **Multiprogramming** – Linux is a multiprogramming system means multiple applications can run at same time.
- **Hierarchical File System** – Linux provides a standard file structure in which system files/ user files are arranged.
- **Shell** – Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs. etc.
- **Security** – Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

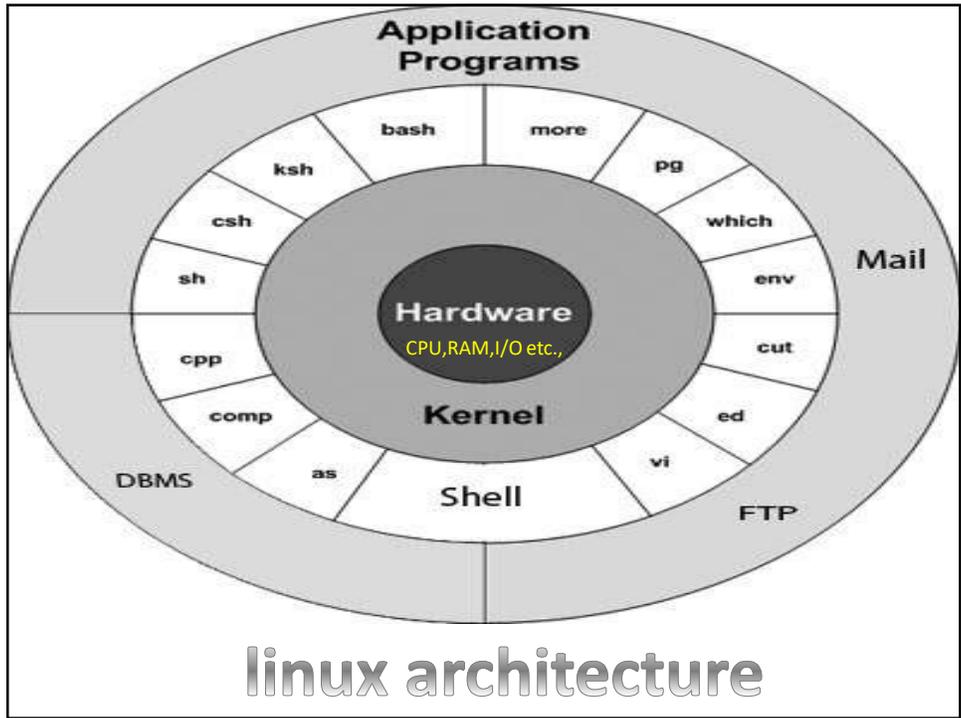
The main concept that unites all the versions of linux are the following four basics –

Kernel – The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.

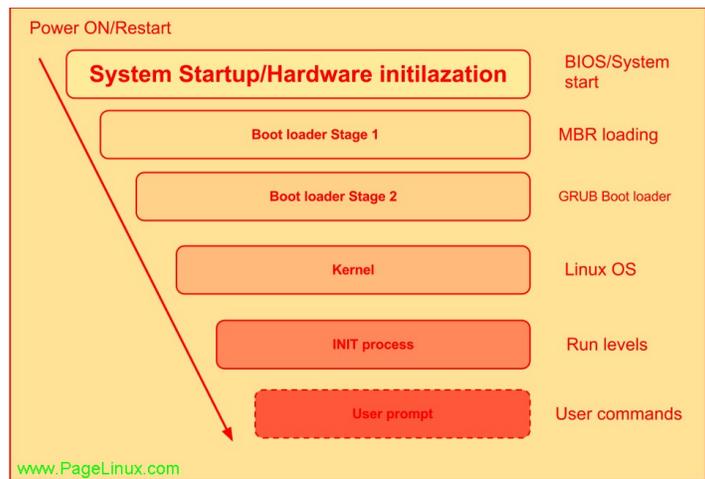
Shell – The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the linux variants.

Commands and Utilities – There are various commands and utilities which you can make use of in your day to day activities. **cp**, **mv**, **cat** and **grep**, etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various options.

Files and Directories – All the data of linux is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the **filesystem**.



Linux Booting Process



Booting process contd.

- **BIOS (Basic Input Output System):**
- When Computer is booted; the processor looks at the end of the system memory for the BIOS (Basic Input/output System) and runs it. The BIOS program is written into permanent read-only memory and is always available for use. The **BIOS provides the lowest level interface to peripheral devices and controls the first step of the boot process.**
- The BIOS tests the system and checks peripherals and system integrity. It checks for the available bootable device, depending on the setting, it checks for CD, USB stick and hard drive. The BIOS looks for a Master Boot Record (MBR) starting at the first sector on the first hard drive, loads its contents into memory, then passes control to it.
- **MBR (Master Boot Record):**
- MBR is located in the first sector of bootable device. MBR is less than 512 bytes in size. It has three components, 1st contains primary boot loader info in 446 bytes, 2nd has partition table information in next 64 bytes and 3rd has info of mbr validation check in last 2 bytes. MBR contains instructions on how to load the GRUB (or LILO) boot-loader, using a pre-selected operating system. The MBR then loads the boot loader, which takes over the process (if the boot-loader is installed in the MBR).

Booting process contd.

- **GRUB: (Grand Unified Boot Loader):**
- This boot method is called direct loading because instructions are used to directly load the operating system, with no intermediary code between the boot-loaders and the operating system's main files (such as the kernel). Microsoft's DOS and Windows operating systems completely overwrite anything on the MBR when they are installed without incorporating any of the current MBR's configurations. This destroys any other information stored in the MBR by other operating systems, such as Linux. Microsoft operating systems are loaded using a chain loading boot method. With this method, the MBR points to the first sector of the partition holding the operating system, where it finds the special files necessary to actually boot that operating system.
- GRUB supports both boot methods, allowing you to use it with almost any operating system, most popular file systems, and almost any hard disk your BIOS can recognize.
- **Kernel:**
- Kernel mounts the root file system as specified in the "root=" in grub.conf. Kernel executes the /sbin/init program. init is the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. initrd stands for Initial RAM Disk. Kernel use initrd as a temporary basis to mount root file system until kernel is booted and the real root file system is mounted.
- **INIT:**
- After the kernel is booted and initialized, the kernel starts the first user-space application.
- In a Linux system, the first application started is commonly /sbin/init .
- First, it runs the /etc/rc.d/rc.nsysinit script, which sets the environment path, starts swap, checks the file systems, and executes all other steps required for system initialization. For example, most systems use a clock, so on them rc.sysinit reads the /etc/sysconfig/clock configuration file to initialize the hardware clock.

Booting process contd.

- **Runlevel:**
- The init command then runs the /etc/inittab script to decide which run level OS has to come up.
- Default run levels are as follows :
- # 0 – halt (Do NOT set init default to this)
- # 1 – Single user mode
- # 2 – Multiuser, without NFS
- # (The same as 3, if you do not have networking)
- # 3 – Full multiuser mode
- # 4 – unused
- # 5 – X11
- # 6 – reboot (Do NOT set init default to this)
- The init program starts all of the background processes by looking in the appropriate rc directory for the runlevel specified as default in /etc/inittab. The rc directories are numbered to corresponds to the runlevel they represent. For instance, /etc/rc.d/rc5.d/ is the directory for runlevel 5.
- In /etc/rc.d/rc5.d/ directory what are all files starting with “K” those responsible for Kill the process when system is rebooting or shouting down. And Starting “S” those responsible for Start the process.
- After this the system finished Booting and you can login to system.

Type of Linux Installation

- Following are the different types of installing Linux:



- **Fresh Installation:**
- In this method you can format the computer’s hard drive and install Linux from Bootable CD/DVD or Bootable USB drive. In this method Linux is the only Operating system in your computer.
- **Dual Boot System:**
- One of the things that make Linux special is that it can play nice with other operating systems. You can run Linux alongside of other operating systems quite easily. Running Linux along with other operating system like windows is called “Dual Boot” system. It means that you have a system with two different operating systems, and during the boot process you will need to decide which one you would like to boot into.

Linux Installation types contd.

- **Live CD/DVD:**
- Many Linux distributors provide the option to try it without installing. In that Linux runs as a completely bootable operating system from the CD/DVD. The files are loaded into your computer's memory, rather than being run from a hard disk drive. So you can run Linux from a CD/DVD, and when you reboot your PC, and remove the CD/DVD, it will boot back into its old operating system without any difference to your PC. This gives you an easy way to try out several distributions of Linux.
- **Virtual Installation:**
- You can install Linux inside Virtual Machine also. This method is preferred by many peoples in these days. There are a number of ways to do this, the simplest one is to download and install a Virtual Server application, and then install your Linux distribution under that host software.

Desktop Environments for Linux

- **Desktop Environment:**
- In graphical computing, a desktop environment (DE) is an implementation of a desktop metaphor graphical user interface (GUI). The desktop environment was seen on most personal computers until the rise of mobile computing. Desktop GUIs help the user in easily accessing, configuring and modifying many important and frequently accessed specific operating system (OS) features. The GUI usually does not afford access to all the many features found in an OS. Instead, the traditional command-line interface (CLI) is still used when full control over the OS is required in such cases. A desktop environment typically consists of icons, windows, toolbars, folders, wallpapers and desktop widgets. (From Wikipedia)
- **GNOME:**
- GNOME is one of the most popular Linux desktop environment. The GNOME 2 series was used as default desktop environment by Ubuntu, Fedora, Debian and other Linux distributions. GNOME 2 was lightweight desktop environment. The GNOME 3 comes with GNOME Shell user interface and there is a drastic change from GNOME 2.

**GNOME 3**

Desktop Environments for Linux contd.

- **Xfce:**
 - Xfce is famous for its lightweight Desktop environment. Xfce embodies the traditional UNIX philosophy of modularity and re-usability. It consists of a number of components that provide the full functionality one can expect of a modern desktop environment, while remaining relatively light. Xfce is a great choice for older computers and it is still a full-fledged desktop environment that offers a great deal to the user. Xfce provides a nice balance between functionality and conservation of system resources, while still having a beautiful desktop. It was once very similar to GNOME, but with GNOME 3 striking out in a different direction, Xfce now has its own identity as a more traditional Linux desktop environment that's quite similar to GNOME 2.

Desktop Environments for Linux contd.

- **LXDE:**
 - LXDE is more lightweight than Xfce. LXDE is focused on being as lightweight as possible and is especially designed for older computers, netbooks, and other systems with low hardware resources. It comes with a beautiful interface, multi-language support, standard keyboard short cuts and additional features like tabbed file browsing. Fundamentally designed to be lightweight, LXDE uses less CPU and RAM than other environments. It is especially beneficial for cloud computers with low hardware specifications, such as netbooks, mobile devices or older computers. While it's a lightweight desktop, it includes all the standard desktop features — some lightweight desktops omit the taskbar entirely, but LXDE doesn't.



Desktop Environments for Linux contd.

- **Cinnamon:**
- **Cinnamon** was developed for **Linux Mint**. Cinnamon is based on GNOME 3, so it uses up-to-date libraries and other software, but it takes that software and tries to create a more traditional-looking desktop with it. Cinnamon is a fork of GNOME 3. Cinnamon strives to provide a traditional user experience, similar to GNOME 2.

Desktop Environments for Linux contd.

- **MATE:**
- **MATE** is a fork of the original GNOME 2 that aims to preserve GNOME 2, continually updating it so it will continue to work on modern Linux distributions. MATE provides an intuitive and attractive desktop to Linux users using traditional metaphors. MATE has also seen some new features, but the main purpose of MATE is to give people who desperately miss GNOME 2 the opportunity to install it on new Linux distributions. This desktop environment is ideal for people who really miss GNOME 2.



Desktop Environments for Linux contd.

- Enlightenment:
- Enlightenment is a window manager, it can also be considered a desktop environment. One very nice feature of Enlightenment is its flexibility, which among other things allows it to run on a wide variety of devices that includes mobile phones, game systems, laptops, and powerful desktop computers. Enlightenment requires less system resources than the GNOME, KDE. The Enlightenment desktop is somewhat unique in its appearance, and users can simply click anywhere on it to access the menu.

- Pantheon:
- Pantheon is the default desktop environment originally created for the elementary OS distribution. It is written from scratch using Vala and the GTK3 toolkit. With regards to usability and appearance, the desktop has some similarities with GNOME Shell and Mac OS X.

- Razor-qt:
- Razor-qt is an advanced, easy-to-use, and fast desktop environment based on Qt technologies. It is tailored for users who value simplicity, speed, and an intuitive interface. A nice feature of Razor-qt is its ability to run with a variety of window managers such as Openbox, Metacity, or KWin. Razor-qt does not yet have its own file manager or other applications, but perhaps it will have more applications in the future. It has been tailored for users who value simplicity, speed, and an intuitive interface. While still a new project, Razor-qt already contains all the key DE components.

- ROX:
- ROX is a fast, user friendly desktop which makes extensive use of drag-and-drop. The interface revolves around the file manager, following the traditional UNIX view that 'everything is a file' rather than trying to hide the filesystem beneath start menus, wizards, or druids. The aim is to make a system that is well designed and clearly presented.

Before Installation

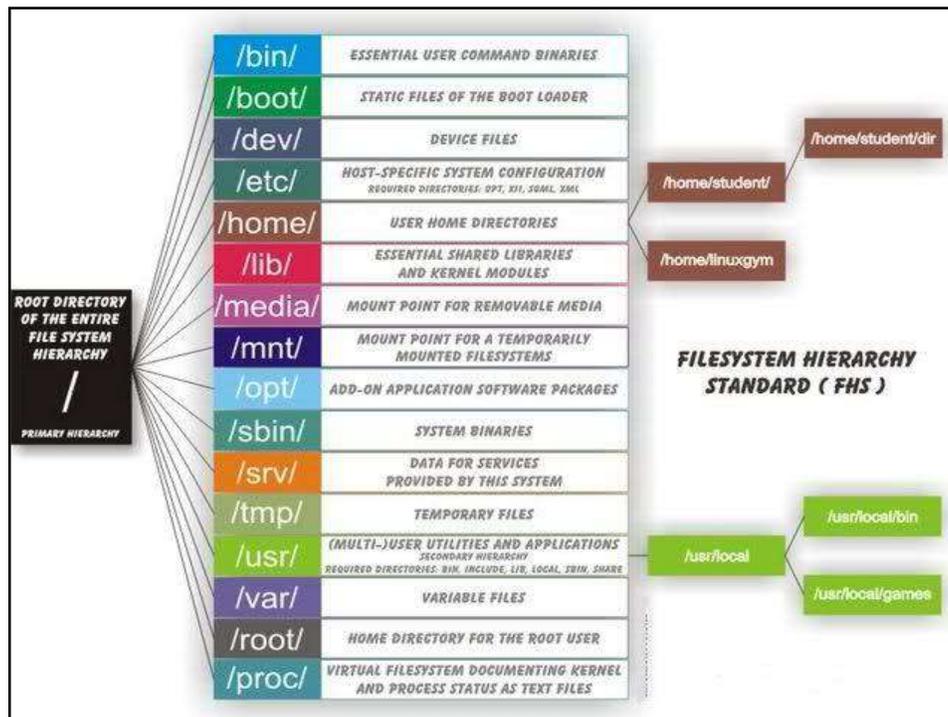
- **Platform:**
- Which Platform or flavor (Distribution) you want to install. As I have already discussed there are many flavours of Linux are available in market. So first choose you Flavour. If you are a beginner and you don't know which is best for you then you can try Ubuntu. It is very user-friendly and good for beginners. For servers CentOS is best and if you can invest then you can choose Red Hat.

- **Version:**
- After choosing the Platform or Flavor you have to decide which version you want to install. There are many versions are available like Desktop version or for Server. If your system support 64 bit then you can choose 64 bit installation.

- **Desktop Environment:**
- There are lots of desktop environment available for *Linux*. Linux users have a choice of many different desktop environments, all with their own styles and strengths. You can install one of these desktop environments after installing your Linux distribution and switch between desktop environments from the login screen. You can also choose to *install* a *Linux* distribution that comes with the desktop environment. Some popular Desktop Environment are Unity, GNOME, KDE, XFCE, LXDE, Cinnamon, MATE and many more.

- **Medium of Installation:**
- Choose the right medium of Installation. You can install via CD, DVD, Bootable USB drive, or through networking.

- **Type of Installation:**
- Are you *installing* it as personal use or for servers? And do you want to install it as a single Operating system (Linux is only Operating system in that computer) or along with Windows or other Operating System. So Choose type of Installation Dual Boot, Live CD, Fresh Install or Virtual Install.
- **Hardware Support:**
- Finally check the compatibility of your hardware. you can check with the respective *Linux* Distributors website for hardware compatibility.



Each of the above directory (which is a file, at the first place) contains important information, required for booting to device drivers, configuration files, etc.

Let's start hierarchically for **describing briefly** the purpose of each directory:

/bin : All the executable binary programs (file) required during booting, repairing, files required to run into single-user-mode, and other important, basic commands viz., cat, du, df, tar, rpm, wc, history, etc.

/boot : Holds important files during boot-up process, including Linux Kernel.

/dev : Contains device files for all the hardware devices on the machine e.g., cdrom, cpu, etc

/etc : Contains Application's configuration files, startup, shutdown, start, stop script for every individual program.

/home : Home directory of the users. Every time a new user is created, a directory in the name of user is created within home directory which contains other directories like Desktop, Downloads, Documents, etc.

/lib : The Lib directory contains kernel modules and shared library images required to boot the system and run commands in root file system.

/lost+found : This Directory is installed during installation of Linux, useful for recovering files which may be broken due to unexpected shut-down.

/media : Temporary mount directory is created for removable devices viz., media/cdrom.

/mnt : Temporary mount directory for mounting file system.

/opt : Optional is abbreviated as opt. Contains third party application software. Viz., Java, etc.

/proc : A virtual and pseudo file-system which contains information about running process with a particular Process-id aka pid.

/root : This is the home directory of root user and should never be confused with ‘/’

/run : This directory is the only clean solution for early-runtime-dir problem.

/sbin : Contains binary executable programs, required by System Administrator, for Maintenance. Viz., iptables, fdisk, ifconfig, swapon, reboot, etc.

/srv : Service is abbreviated as ‘srv’. This directory contains server specific and service related files.

/sys : Modern Linux distributions include a /sys directory as a virtual filesystem, which stores and allows modification of the devices connected to the system.

/tmp : System’s Temporary Directory, Accessible by users and root. Stores temporary files for user and system, till next boot.

/usr : Contains executable binaries, documentation, source code, libraries for second level program.

/var : Stands for variable. The contents of this file is expected to grow. This directory contains log, lock, spool, mail and temp files

Exploring Important file, their location and their Usability

Linux is a complex system which requires a more complex and efficient way to start, stop, maintain and reboot a system unlike Windows. There is a well defined configuration files, binaries, man pages, info files, etc. for every process in Linux.

/boot/vmlinuz : The Linux Kernel file.

/dev/hda : Device file for the first IDE HDD (Hard Disk Drive)

/dev/hdc : Device file for the IDE Cdrom, commonly

/dev/null : A pseudo device, that don’t exist. Sometime garbage output is redirected to /dev/null, so that it gets lost, forever.

/etc/bashrc : Contains system defaults and aliases used by bash shell.

/etc/crontab : A shell script to run specified commands on a predefined time Interval.

/etc/exports : Information of the file system available on network.

/etc/skel : Script that populates new user home directory.

/etc/termcap : An ASCII file that defines the behaviour of Terminal, console and printers.

/etc/X11 : Configuration files of X-window System. (**X11** is a network protocol designed for linux and similar operating systems to enable remote graphical access to applications).

/usr/bin : Normal user executable commands.
/usr/bin/X11 : Binaries of X windows System.
/usr/include : Contains include files used by 'c' program.
/usr/share : Shared directories of man files, info files, etc.
/usr/lib : Library files which are required during program compilation.
/usr/sbin : Commands for Super User, for System Administration.
/proc/cpuinfo : CPU Information
/proc/filesystems : File-system Information being used currently.
/proc/interrupts : Information about the current interrupts being utilised currently.
/proc/ioports : Contains all the Input/Output addresses used by devices on the server.
/proc/meminfo : Memory Usages Information.
/proc/modules : Currently using kernel module.
/proc/mount : Mounted File-system Information.
/proc/stat : Detailed Statistics of the current System.
/proc/swaps : Swap File Information.
/version : Linux Version Information.
/var/log/lastlog : log of last boot process.
/var/log/messages : log of messages produced by syslog daemon at boot.
/var/log/wtmp : list login time and duration of each user on the system currently.
/etc/fstab : Information of Disk Drive and their mount point.
/etc/group : Information of Security Group.

/etc/grub.conf : grub bootloader configuration file.
/etc/init.d : Service startup Script.
/etc/lilo.conf : lilo bootloader configuration file.
/etc/hosts : Information of Ip addresses and corresponding host names.
/etc/hosts.allow : List of hosts allowed to access services on the local machine.
/etc/host.deny : List of hosts denied to access services on the local machine.
/etc/inittab : INIT process and their interaction at various run level.
/etc/issue : Allows to edit the pre-login message.
/etc/modules.conf : Configuration files for system modules.
/etc/motd : motd stands for Message Of The Day, The Message users gets upon login.
/etc/mtab : Currently mounted blocks information.
/etc/passwd : Contains password of system users in a shadow file, a security implementation.
/etc/printcap : Printer Information
/etc/profile : Bash shell defaults
/etc/profile.d : Application script, executed after login.
/etc/rc.d : Information about run level specific script.
/etc/rc.d/init.d : Run Level Initialisation Script.
/etc/resolv.conf : Domain Name Servers (DNS) being used by System.
/etc/securetty : Terminal List, where root login is possible.

Installing Linux 7.1 Dual Boot With Windows 8.1 on UEFI Firmware Systems

We shall discuss the installation of Linux in dual-boot with Windows on UEFI Firmware machines that comes pre-installed with Windows Operating System. However, if any machine has no Operating System installed by default and still required to use the dual-boot, Windows alongside Linux, it's recommended that Windows OS, to be installed first to create the necessary partitions during the Windows installation process and, then, install Linux Operating System.

Other solutions:

Install Virtualbox on Linux system. It will allow to run Windows XP operating system under it as a Guest OS or run Linux OS on Windows system and so on. This way, anyone can install and run as many as guest operating systems as anyone like, the only limit is disk space and memory.

Recently Oracle has released latest stable version of Virtualbox 6.0.0, the newest version of Virtual box comes with so many major changes and new features added to it.

VirtualBox is an open source cross-platform virtualization software, it can be installed on any operating system and enable you to install and run multiple guest operating systems on same computer.

Unified Extensible Firmware Interface (UEFI) is a specification for a software program that connects a computer's firmware to its operating system (OS). UEFI is expected to eventually replace BIOS. Like BIOS, UEFI is installed at the time of manufacturing and is the first program that runs when a computer is turned on. The UEFI settings screen allows you to disable Secure Boot, a useful security feature that prevents malware from hijacking Windows or another installed operating system.

How to Use Grub Rescue to Fix Linux Boot Failure

- The GRUB (Grand Unified Bootloader) is a tool for booting and loading operating system kernels and the default bootloader for systems based on the [Linux kernel](#). Although it runs first when a machine is turned on, regular users rarely see GRUB in action. It functions automatically and requires no user input.
- However, when attempting to boot another operating system alongside Linux on the same machine, the other system's bootloader may overwrite GRUB, resulting in the inability of the Linux system to boot up.
- **Prerequisites**
 - An account with sudo privileges.
 - Access to the command line.
- **GRUB Boot Issues**
 - The most common reason for GRUB not booting into the operating system is another OS's bootloader overwriting GRUB boot configuration. The problem occurs during an attempt a dual boot with an existing Linux installation. Another reason is the accidental removal of GRUB configuration files.
 - When GRUB is not able to boot the system, the GRUB Rescue prompt appears.

```
error: no such partition.
Entering rescue mode..
grub rescue> _
```

GRUB contd.

- The example above shows GRUB displaying the "no such partition" error before displaying the grub rescue prompt. Another common GRUB error is "unknown filesystem", followed by the same prompt.

```
error: unknown filesystem.
Entering rescue mode..
grub rescue> _
```

- Sometimes, the screen may show the grub prompt only.

```
grub> _
```

GRUB Rescue Commands

Below is the list of the commonly used GRUB Rescue commands.
Use the commands in the prompts mentioned in the previous section.

Command	Description	Example
boot	Start booting (shortcuts: F10 , CTRL + x).	The command is issued without arguments.
cat	Write the contents of a file to standard output.	cat (hd0,1)/boot/grub/grub.cfg
configfile	Load a configuration file.	configfile (hd0,1)/boot/grub/grub.cfg
initrd	Load the initrd.img file.	initrd (hd0,1)/initrd.img
insmod	Load a module.	insmod (hd0,1)/boot/grub/normal.mod

GRUB contd.

Command	Description	Example
insmod	Load a module.	insmod (hd0,1)/boot/grub/normal.mod
loopback	Mount an image file as a device.	loopback loop0 (hd0,1)/iso/image.iso
ls	Display the contents of a directory or partition.	ls (hd0,1)
lsmod	Display a list of loaded modules.	The command is issued without arguments.
normal	Activate the normal module.	The command is issued without arguments.
search	Search for devices. Option --file searches for files, --label searches for labels, --fs-uuid searches for filesystem UUID.	search -file [filename]
set	Set an environment variable. If issued with no arguments, the command prints the list of all environment variables and their values.	set [variable-name]=[value]

Fixing Boot Failure

- There are two ways to resolve GRUB boot issues, using the **GRUB Rescue prompt**, and the **Boot Repair tool**.
- **Via Grub Terminal**
- 1. Use the [set command](#) with no arguments to view the [environment variables](#):
- `$set`
- The example output shows that GRUB is set up to boot from (hd0,msdos3) partition:

```
grub> set
?=0
color_highlight=black/white
color_normal=white/black
default=0
gfxmode=800x600
lang=en_US
locale_dir=(hd0,msdos3)/boot/grub/locale
pager=
prefix=(hd0,msdos3)/boot/grub
root=hd0,msdos3
grub> _
```

Boot Failure contd.

- 2. The [ls command](#) lists the available partitions on the disk.

```
grub> ls
(hd0) (hd0,msdos3) (hd0,msdos2) (hd0,msdos1)
grub> _
```

Use the ls command to find the partition containing the boot directory.

`$ls [partition-name]`

The example shows the boot directory in the (hd0,msdos1) partition

```
grub> ls (hd0,msdos1)
lost+found var/ dev/ run/ etc/ tmp/ sys/ proc/ usr/ bin boot/ home/ lib lib64
mnt/ opt/ root/ sbin srv/
grub> _
```

- 3. Set the boot partition as the value of the root variable. The example uses the partition named (hd0,msdos1).

`$set root=(hd0,msdos1)`

- 4. Load the normal boot mode.

`$insmod normal`

- 5. Start the normal boot mode.

`$normal`

The normal mode enables you to issue more complex commands.

- 6. Load the Linux kernel using the linux command.

`$linux /boot/vmlinuz-4.2.0-16-generic root=/dev/sda1 ro`

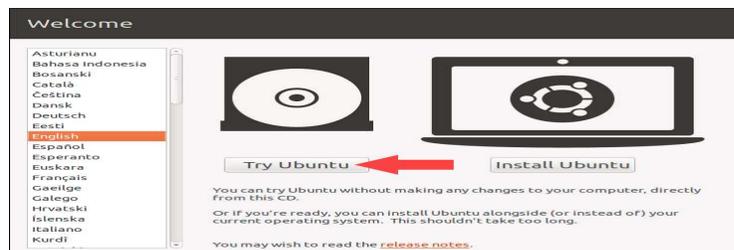
- 7. Issue the boot command.

`$boot`

The system now boots properly.

Boot Failure contd.

- **Via Live image**
- Another way to fix your GRUB boot issues is to use a Linux live image to boot from an external device.
- 1. Download a live Linux installer. This example uses the Ubuntu 20.04 ISO image.
- 2. Use a tool such as [Etcher](#) to write the Linux image to an SD card or a USB flash drive.
- 3. Insert the bootable device and start the computer.
- 4. Select **Try Ubuntu** on the welcome screen

**Boot Failure contd.**

5. When the live system boots up, connect to the internet.
6. Open the terminal and type the following command to add the repository for the Boot Repair tool.


```
$sudo add-apt-repository ppa:yannubuntu/boot-repair
```

```
marko@test-main:~$ sudo add-apt-repository ppa:yannubuntu/boot-repair
Simple tool to repair frequent boot problems.

Website: https://sourceforge.net/p/boot-repair/home
More info: https://launchpad.net/~yannubuntu/+archive/ubuntu/boot-repair
Press [ENTER] to continue or Ctrl-c to cancel adding it.
```

Press Enter and wait for the repository to be added.

7. Update the repositories.

```
$sudo apt update
```

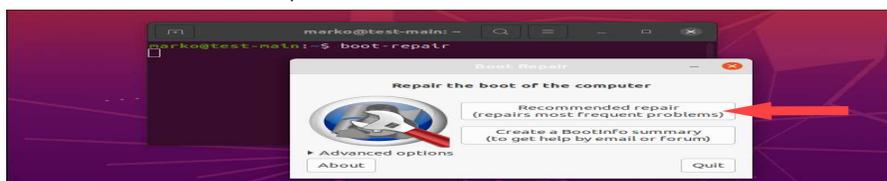
8. Install the Boot Repair tool.

```
$sudo apt install boot-repair
```

9. Start the Boot Repair tool via the terminal.

```
$boot-repair
```

10. Select Recommended repair.



Updating GRUB config file

- When the system successfully boots up, make sure the GRUB configuration is up to date.
- Run this command: **\$sudo update-grub**

```
marko@test-main:~$ sudo update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.11.0-27-generic
Found initrd image: /boot/initrd.img-5.11.0-27-generic
Found linux image: /boot/vmlinuz-5.8.0-59-generic
Found initrd image: /boot/initrd.img-5.8.0-59-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
marko@test-main:~$
```

- 1. Mount the partition containing the OS installation. The example mounts the /dev/sda1 partition to the /mnt directory.
- **\$sudo mount /dev/sda1 /mnt**
- 2. Bind the /dev, /dev/pts, /proc, and /sys directories to the corresponding directories in the /mnt folder.
- **sudo mount --bind /dev /mnt/dev &&**
- **sudo mount --bind /dev/pts /mnt/dev/pts &&**
- **sudo mount --bind /proc /mnt/proc &&**
- **sudo mount --bind /sys /mnt/sys**

Updating GRUB contd.

- 3. Install GRUB.

Basic 'ls' Command Examples in Linux

[ls command](#) is one of the most frequently used command in Linux

1 List Files using ls with no option

ls with no option list files and directories in bare format where we won't be able to view details like file types, size, modified date and time, permission and links etc.

ls

```
0001.pcap Desktop Downloads index.html install.log.syslog Pictures Templates
```

2 List Files With option -l

total 176

```
-rw-r--r--. 1 root root 683 Aug 19 09:59 0001.pcap
-rw-----. 1 root root 1586 Jul 31 02:17 anaconda-ks.cfg
drwxr-xr-x. 2 root root 4096 Jul 31 02:48 Desktop
drwxr-xr-x. 2 root root 4096 Jul 31 02:48 Documents
```

3. View Hidden Files

List all files including hidden file starting with '.'.

ls -a

```
. .bashrc Documents .gconfd install.log .nautilus .pulse-cookie
.. .cache Downloads .gnome2 install.log.syslog .netstat.swp .recently-used.xbel
0001.pcap .config .elinks .gnome2_private .kde .opera
```

• First	second	third	fourth	fifth	sixth	seventh
-rw-r--r--	1	root	root	683	Aug 19 09:59	0001.pcap

- Here is the information about all the listed columns –
- First Column – Represents the file type and the permission given on the file. Below is the description of all type of files.
- Second Column – Represents the number of memory blocks taken by the file or directory.
- Third Column – Represents the owner of the file. This is the linux user who created this file.
- Fourth Column – Represents the group of the owner. Every linux user will have an associated group.
- Fifth Column – Represents the file size in bytes.
- Sixth Column – Represents the date and the time when this file was created or modified for the last time.
- Seventh Column – Represents the file or the directory name.

4. List Files with Human Readable Format with option -lh**# ls -lh**

total 176K

```
-rw-r--r--. 1 root root 683 Aug 19 09:59 0001.pcap
-rw-----. 1 root root 1.6K Jul 31 02:17 anaconda-ks.cfg
drwxr-xr-x. 2 root root 4.0K Jul 31 02:48 Desktop
drwxr-xr-x. 2 root root 4.0K Jul 31 02:48 Documents
```

5. List Files and Directories with '/' Character at the end

Using -F option with ls command, will add the '/' Character at the end each directory.

ls -F

```
0001.pcap Desktop/ Downloads/ index.html install.log.syslog Pictures/ Templates/
anaconda-ks.cfg Documents/ fbcmd_update.php install.log Music/ Public/ Videos/
```

6. List Files in Reverse Order

The following command with ls -r option display files and directories in reverse order.

```
Videos Public Music install.log fbcmd_update.php Documents anaconda-ks.cfg
Templates Pictures install.log.syslog index.html Downloads Desktop 0001.pcap
```

7. Recursively list Sub-Directories

ls -R option will list very long listing directory trees. See an example of output of the command.

ls -R

total 1384

```
-rw-----. 1 root root 33408 Aug 8 17:25 anaconda.log
-rw-----. 1 root root 30508 Aug 8 17:25 anaconda.program.log
```

./httpd:

total 132

```
-rw-r--r-- 1 root root 0 Aug 19 03:14 access_log
-rw-r--r--. 1 root root 61916 Aug 10 17:55 access_log-20120812
```

./lighttpd:

total 68

```
-rw-r--r-- 1 lighttpd lighttpd 7858 Aug 21 15:26 access.log
-rw-r--r--. 1 lighttpd lighttpd 37531 Aug 17 18:21 access_log-20120819
```

9. Sort Files by File Size

With combination of -lS displays file size in order, will display big in size first.

ls -lS

```
total 176
-rw-r--r--. 1 root root 48867 Jul 31 02:17 install.log
-rw-r--r--. 1 root root 46701 Jul 31 09:58 index.html
-rw-r--r--. 1 root root 21262 Aug 12 12:42 fbcmd_update.php
-rw-r--r--. 1 root root 11439 Jul 31 02:13 install.log.syslog
drwxr-xr-x. 2 root root 4096 Jul 31 02:48 Desktop
```

10. Display Inode number of File or Directory

We can see some number printed before file / directory name. With -li options list file / directory with inode number.

ls -li

```
20112 0001.pcap 23610 Documents 23793 index.html 23611 Music 23597
Templates 23564 anaconda-ks.cfg 23595 Downloads 22 install.log 23612 Pictures
236
```

11. Shows version of ls command

Check version of ls command.

```
# ls --version
```

```
ls (GNU coreutils) 8.4
```

```
Copyright (C) 2010 Free Software Foundation, Inc.
```

```
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
```

```
This is free software: you are free to change and redistribute it.
```

12. Show Help Page

List help page of ls command with their option.

ls --help

```
Usage: ls [OPTION]... [FILE]...
```

13. List Directory Information

With ls -ld command list files under directory /tmp. Wherein with -ld parameters displays information of /tmp directory.

ls -ld /tmp

```
total 408
drwx-----. 2 narad narad 4096 Aug 2 02:00 CRX_75DAF8CB7768
-r-----. 1 root root 384683 Aug 4 12:28 htop-1.0.1.tar.gz
drwx-----. 2 root root 4096 Aug 4 11:20 keyring-6Mfjnk
drwx-----. 2 root root 4096 Aug 16 01:33 keyring-pioZlr
```

Metacharacters

Metacharacters have a special meaning in linux. For example, * and ? are metacharacters. We use * to match 0 or more characters, a question mark (?) matches with a single character.

For Example –

ls ch*.doc

Displays all the files, the names of which start with ch and end with .doc –

```
ch01-1.doc ch010.doc ch02.doc ch03-2.doc  
ch04-1.doc ch040.doc ch05.doc ch06-2.doc  
ch01-2.doc ch02-1.doc
```