# Linux Boot Process:

<https://markdown-it.github.io/>

The booting process takes the following 4 steps that we will discuss in greater detail:

* **BIOS** Integrity check (**POST**)
* Loading of the Boot loader (**GRUB2**)
* Kernel initialization
* Starting **systemd**, the parent of all processes

**BIOS** Integrity check (**POST**)

When the Linux system powers up, the **BIOS** (**Basic Input Output System**) kicks in and performs a **Power On Self Test** (**POST**). This is an integrity check that performs a plethora of diagnostic checks.

The **POST** probes the hardware operability of components such as the **HDD** or **SSD**, **Keyboard**, **RAM**, **USB** ports, and any other piece of hardware. If some hardware device is not detected, or if there’s a malfunction in any of the devices such as a corrupt HDD or SSD, an error message is splashed on the screen prompting your intervention.

Boot loader (**GRUB2**)

Once the **POST** is complete and the coast is clear, the **BIOS** probes the **MBR** (**Master Boot Record**) for the bootloader and disk partitioning information.

There are 3 main types of bootloaders in Linux: **LILO**, **GRUB,** and **GRUB2**. The **GRUB2** bootloader is the latest and primary bootloader in modern Linux distributions and informs our decision to leave out the other two which have become antiquated with the passage of time

**GRUB2** stands for **GRand Unified Bootloader** version 2. Once the **BIOS** locates the grub2 bootloader, it executes and loads it onto the main memory (**RAM**).

The **grub2** menu allows you to do a couple of things. It allows you to select the Linux kernel version that you’d want to use. Additionally, it gives you the ability to edit some kernel parameters by pressing a combination of keyboard keys.

Kernel initialization

The kernel is the core of any Linux system. It interfaces the PC’s hardware with the underlying processes. The kernel controls all the processes on your Linux system. Once the selected Linux kernel is loaded by the bootloader, it must self extract from its compressed version before undertaking any task. Upon self-extracting, the selected kernel mounts the root file system and initializes the **/sbin/init** program commonly referred to as **init**.

**Init** is always the first program to be executed and is assigned the process ID or PID of 1. It’s the init process that spawns various daemons & mounts all partitions that are specified in the **/etc/fstab** file.

The kernel then mounts the initial RAM disk (**initrd**) which is a temporary root filesystem until the real root filesystem is mounted. All kernels are located in the /boot directory together with the initial RAM disk image.

Starting **system**

**Systemd** is the mother of all Linux processes and manages among other things mounting of file systems, starting and stopping services to mention just a few.

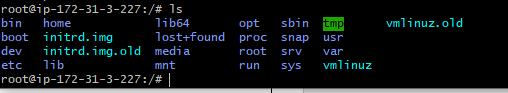
Systemd uses the **/etc/systemd/system/default.target** file to determine the state or target that the Linux system should boot into.

* For a desktop workstation (with a GUI) the default target value is 5 which is the equivalent of run level 5 for the old SystemV init.
* For a server, the default target is **multi-user.target** which corresponds to run level 3 in SysV init.

To check the current target on your system, run the command:

$ systemctl get-default

# Directory Structure



|  |  |
| --- | --- |
| **Directory** | **Description** |
| /bin/ | Essential user command binaries; all executable files for basic commands cat, du etc. |
| /boot/ | Static files of boot loader |
| /dev/ | Device files |
| /etc/ | Host specific system configuration; start stop startup shutdown scripts for every program |
| /home/ | User home directories; contains desktop, documents, downloads etc. |
| /lib/ | Shared libraries and kernel modules |
| /media/ | Mount point for removable media |
| /mnt/ | Mount point for temporary mounted file system |
| /opt/ | Software packages |
| /proc/ | Virtual file system as text file; A virtual and pseudo file-system which contains information about **running process** with a particular **Process-id** aka **pid**. |
| /root/ | Home directory for root user |
| /sbin/ | System binaries; Contains binary executable programs, required by **System Administrator**, for **Maintenance**. Viz., [iptables](https://www.tecmint.com/basic-guide-on-iptables-linux-firewall-tips-commands/), [fdisk](https://www.tecmint.com/fdisk-commands-to-manage-linux-disk-partitions/" \t "_blank), [ifconfig](https://www.tecmint.com/ifconfig-command-examples/), swapon, reboot, etc |
| /srv/ | Data for services |
| /sys/ |  |
| /usr/ | User utilities and applications |
| /var/ | Variable files; This directory contains **log**, **lock**, **spool**, **mail** and **temp** files. |

# Commands

|  |  |
| --- | --- |
| **Ls** | **Lists files and directories** |
| Ls -l | shows file or directory, size, modified date and time, file or folder name and owner of the file, and its permission. |
| Ls -a | List all files including hidden files |
| Ls -r | display files and directories in reverse order |
| Ls -R | Display files and directories artifacts recursively |
| Ls -ltr | show the latest modification file or directory date as last. |
| Ls -l /var | list files under directory **/var** |

|  |  |
| --- | --- |
| **CD** | **Change directory** |
| cd /usr/local | Change from current directory to /usr/local |
| cd - | Switch back to previous directory where you working earlier |
| Cd .. | Change Current directory to parent directory. |
| Cd ../ ../ | Move two directory up from where you are now |
| Cd ~ | Move to users home directory from anywhere |

|  |  |
| --- | --- |
| **DIR** | **List directory** |
| Dir -a | List all files in a directory including hidden |
| Dir -al |  |

|  |  |
| --- | --- |
| **Touch** | **create, change and modify timestamps of a file** |
| Touch filename1 file2 file3 … | Creates empty files |
| Touch -c filename | Will create file if it doesnot exist else will not |

|  |  |
| --- | --- |
| cp <source> <destination> | **Copy from source to destination** |
| Cp -r source dest | Copy recursively |
| mv <source> <destination> | Move |
| Mv -I source dest | Throw warning for overwriting |
|  |  |