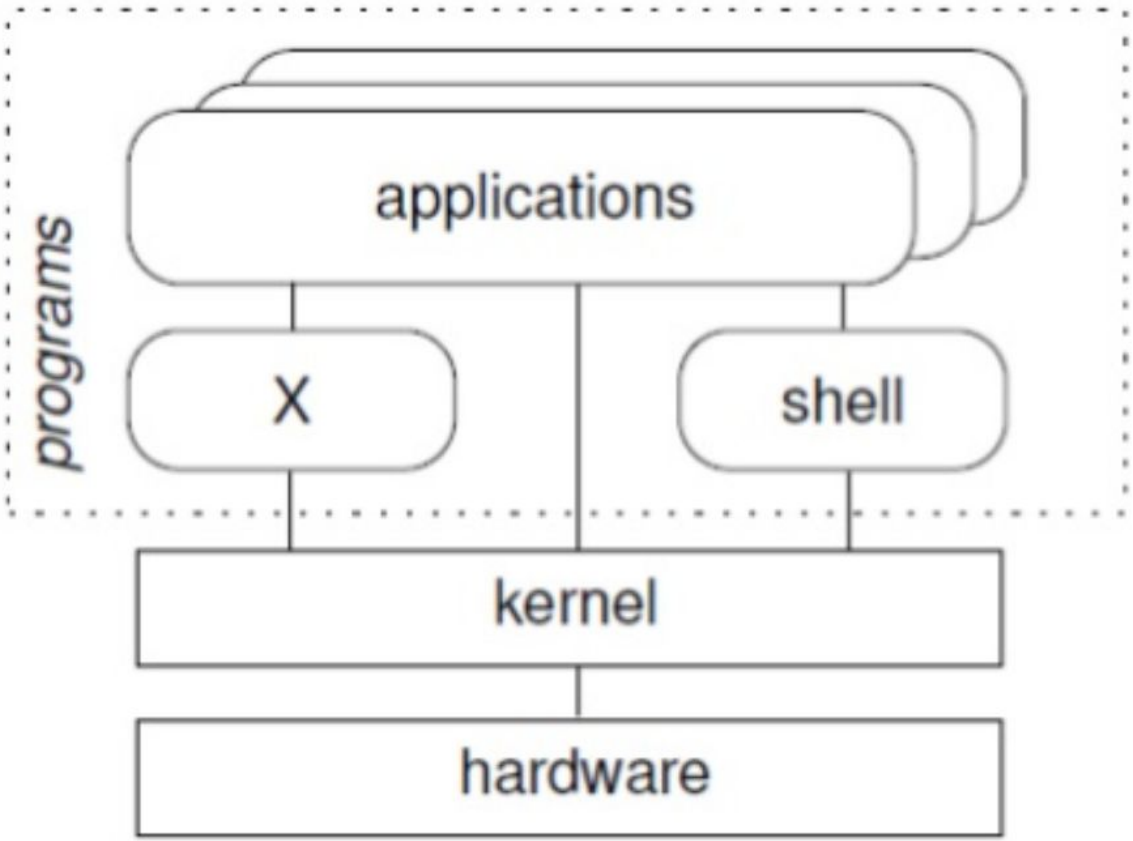
Linux Boot Process

Linux Administration

Ashwini Mathur

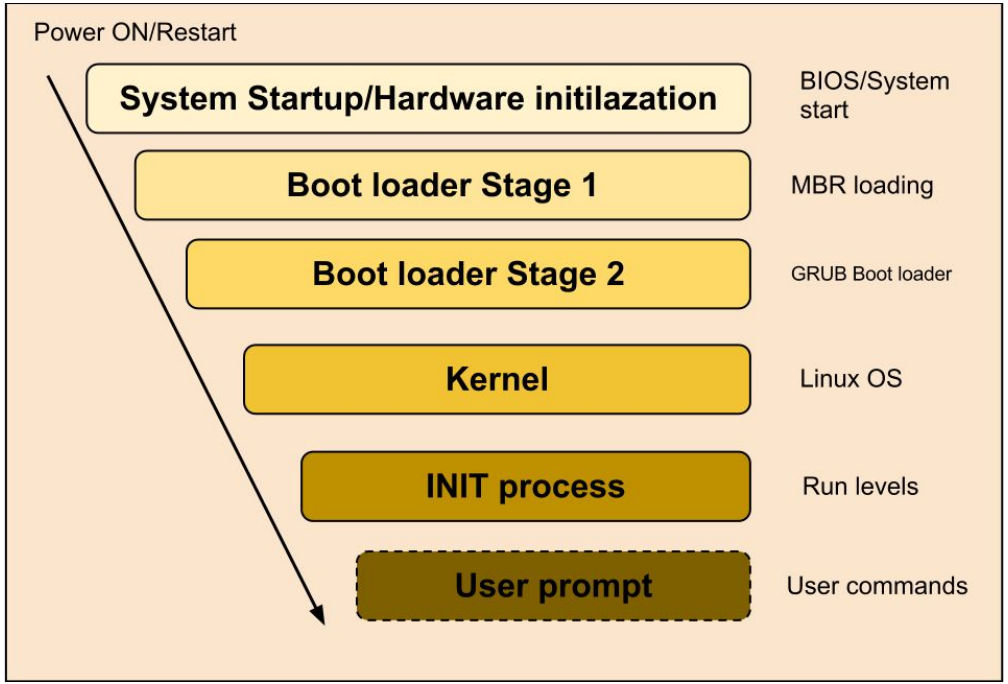

programs
applications
X
shell
kernel
hardware

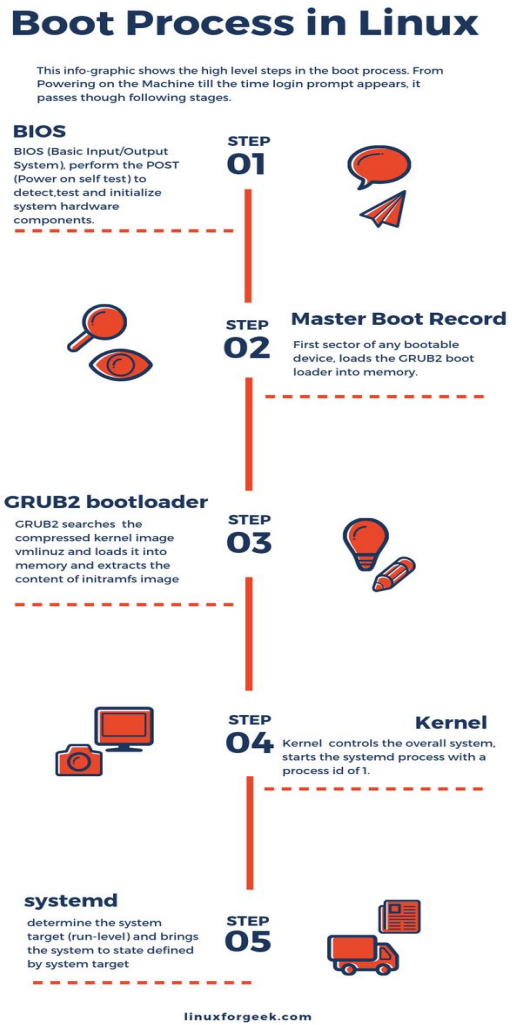

Boostrapping
Code
OS

An operating system (OS) is the low-level software that manages resources, controls peripherals, and provides basic services to other software.

In Linux, there are 6 distinct stages in the typical booting process.

#Reference : Free Code Camp

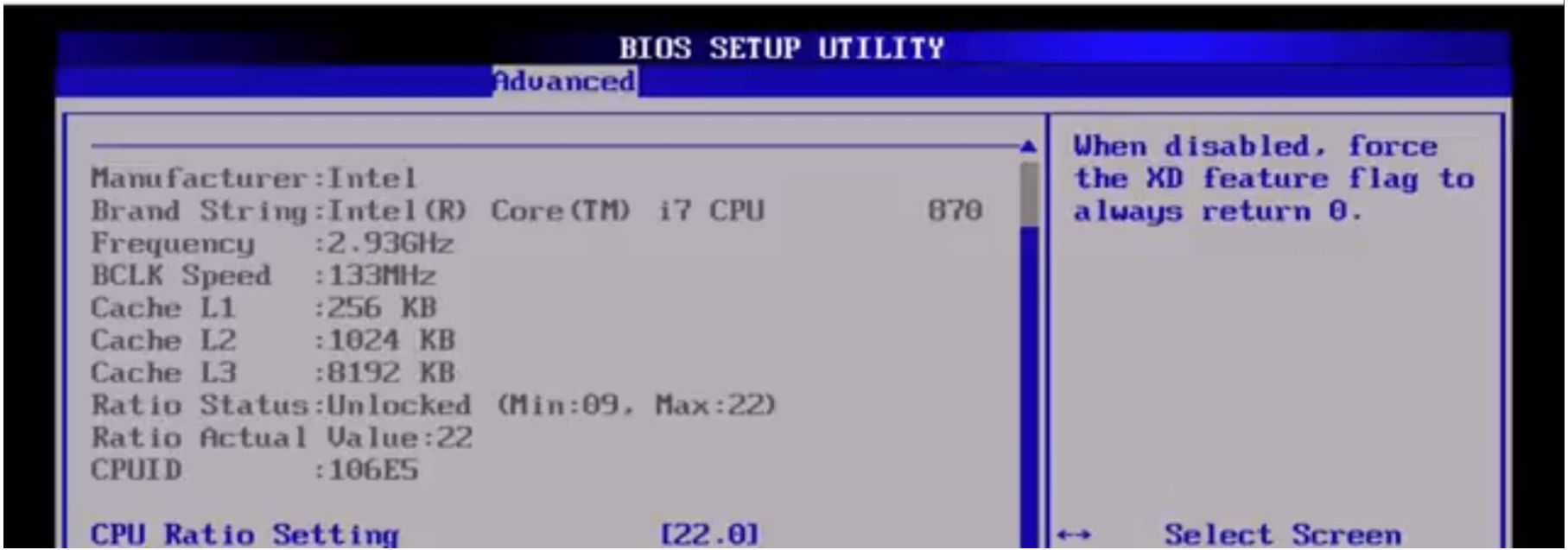

Power ON/Restart
System Startup/Hardware initilazation
BIOS/System start
Boot loader Stage 1
MBR loading
Boot loader Stage 2
GRUB Boot loader
Kernel
Linux OS
INIT process
Run levels
User prompt
User commands


Boot Process in Linux
This info-graphic shows the high level steps in the boot process. From Powering on the Machine till the time login prompt appears, it passes though following stages.
BIOS
BIOS (Basic Input/Output System), perform the POST (Power on self test) to detect,test and initialize system hardware components.
STEP 01
STEP 02
Master Boot Record
First sector of any bootable device, loads the GRUB2 boot loader into memory.
GRUB2 bootloader
GRUB2 searches the compressed kernel image vmlinuz and loads it into memory and extracts the content of initramfs image
STEP 03
STEP 04
Kernel
Kernel controls the overall system, starts the systemd process with a process id of 1.
systemd
determine the system target (run-level) and brings the system to state defined by system target
STEP 05
linuxforgeek.com

BIOS - Basic Input/Output System

Linux Boot Process

BIOS = Basic Input/Output System



BIOS stands for Basic Input/Output System. In simple terms, the BIOS loads and executes the Master Boot Record (MBR) boot loader.

When you first turn on your computer, the BIOS first performs some integrity checks [POST TEST - [Power on Self Test]] of the HDD or SSD.

Then, the BIOS searches for, loads, and executes the boot loader program, which can be found in the Master Boot Record (MBR). The MBR is sometimes on a USB stick or CD-ROM such as with a live installation of Linux.

Once the boot loader program is detected, it's then loaded into memory and the BIOS gives control of the system to it.

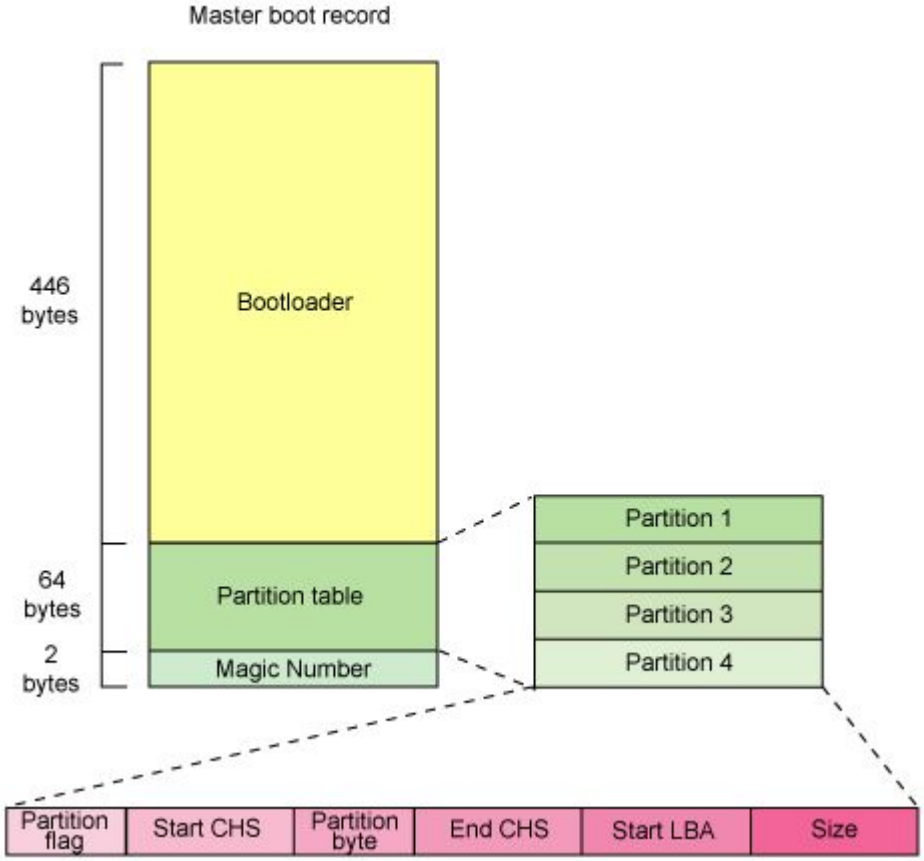
MBR

MBR stands for Master Boot Record, and is responsible for loading and executing the GRUB boot loader.

The MBR is located in the 1st sector of the bootable disk, which is typically `/dev/hda`, or `/dev/sda`, depending on your hardware. The MBR also contains information about GRUB, or LILO in very old systems.

MBR- Master Boot Record

Grub - GRand Unified Bootloader, LILO - Linux Loader


Master boot record
446 bytes
Bootloader
64 bytes
Partition table
2 bytes
Magic Number
Partition 1
Partition 2
Partition 3
Partition 4
Partition flag
Start CHS
Partition byte
End CHS
Start LBA
Size

Linux Boot Process

Boot Loaders: LILO and GRUB

GRUB

Sometimes called GNU GRUB, which is short for GNU GRand Unified Bootloader, is the typical boot loader for most modern Linux systems.

The GRUB splash screen is often the first thing you see when you boot your computer. It has a simple menu where you can select some options. If you have multiple kernel images installed, you can use your keyboard to select the one you want your system to boot with. By default, the latest kernel image is selected.

The splash screen will wait a few seconds for you to select and option. If you don't, it will load the default kernel image.

Linux Boot Process

- The `/boot` directory contains:
  - initrd
  - Linux Kernel
    - vmlinux = Uncompressed format
    - vmlinuz = Compress format
  - Other files required to boot Linux.

Kernel

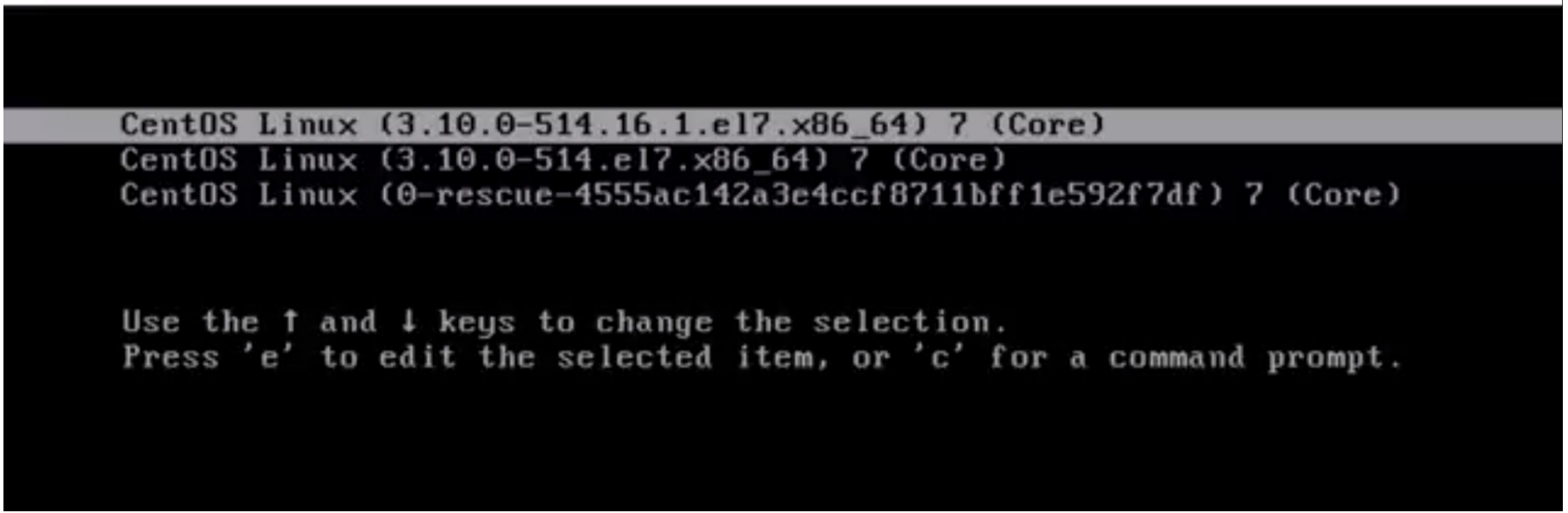
The kernel is often referred to as the core of any operating system, Linux included. It has complete control over everything in your system.

In this stage of the boot process, the kernel that was selected by GRUB first mounts the root file system that's specified in the `grub.conf` file. Then it executes the `/sbin/init` program, which is always the first program to be executed. You can confirm this with its process id (PID), which should always be 1.

The kernel then establishes a temporary root file system using Initial RAM Disk (initrd) until the real file system is mounted.

Linux Boot Process

initrd = Initial RAM Disk



Viewing Boot Messages

dmesg

/var/log/dmesg

```
[    0.387820] usbcore: registered new interface driver usbfs
[    0.388446] usbcore: registered new interface driver hub
[    0.389104] usbcore: registered new device driver usb
[    0.390253] PCI: Using ACPI for IRQ routing
[    0.391273] NetLabel: Initializing
[    0.391814] NetLabel:  domain hash size = 128
[    0.392411] NetLabel:  protocols = UNLABELED CIPSOv4
[    0.393218] NetLabel:  unlabeled traffic allowed by default
[    0.394498] Switched to clocksource kvm-clock
[    0.402719] pnp: PnP ACPI init
[    0.403285] ACPI: bus type PNP registered
[    0.404560] pnp: PnP ACPI: found 2 devices
[    0.405194] ACPI: bus type PNP unregistered
[    0.415060] NET: Registered protocol family 2
```

Init

At this point, your system executes runlevel programs. At one point it would look for an init file, usually found at `/etc/inittab` to decide the Linux run level.

Run Level

Depending on which Linux distribution you have installed, you may be able to see different services getting started. For example, you might catch `starting sendmail …. OK`.

These are known as runlevel programs, and are executed from different directories depending on your run level. Each of the 6 runlevels described above has its own directory:

- Run level 0 – `/etc/rc0.d/`
- Run level 1 – `/etc/rc1.d/`
- Run level 2 – `/etc/rc2.d/`
- Run level 3 – `/etc/rc3.d/`
- Run level 4 – `/etc/rc4.d/`
- Run level 5 – `/etc/rc5.d/`
- Run level 6 – `/etc/rc6.d/`

- Typical boot sequence in PC (x86_64)