Booting

- Process from computer power on to OS startup.

Bootable device

- Stoarage device (disk, pen drive, CD/DVD, etc.) whose boot block contains bootstrap program, is said to be Bootable device.

Bootstrap program

```
* Bootstrap program can load OS kernel into RAM and start its execution.
* Bootstrap program is different for each OS each version.
* Bootstrap is located in first sector (512 bytes) of a disk/partition.
```

Bootloader program

- Bootloader program can be configured to show multiple boot options to end user.
- Depending on user selection, it runs Bootstrap program of corresponding OS.
- There are many important bootloader programs.
    - ntldr (Before Windows Vista) --> boot.ini
    - "bootmgr" (Windows Vista Onwards) --> bcd (boot config data)
        - (admin command prompt) cmd> bcdedit
    - LiLo --> Linux Loader
    - "GrUB" (Grand Unified Boot Loader) --> menu.lst or grub.cfg
    - BTX (BooT eXtended) --> BSD Unix
    - SILo (Sparc Interactive Loader) --> Solaris
    - Bootcamp --> Mac OS X
    - "uBoot" --> Linux bootloader for Embedded
- The Bootloader has some config file associated with it.

Bootstrap loader

- The set of programs fixed in Base ROM (Motherboard) is called as "Firmware".
    - BIOS (Basic Input Output System) -- Firmware developed for PC by IBM + MS.
    - EFI (Extensible Firmware Interface) -- Firmware developed for PC by Intel + HP.
- Bootstrap loader is a program from Base ROM.
- It find the bootable device as per "boot device priority" in the BIOS setup.
- Once bootable device is found, it starts its bootloader.

Booting steps

- Computer power on.

- Load firmware (BIOS or EFI) programs from base ROM into RAM.

- Run POST/BIST (from firmware).

- Run Bootstrap loader (from firmware) to find bootable device.
- Bootstrap loader loads bootloader program from bootable device.
- Bootloader program shows multiple options to end user and user select one of them.
- Bootloader program run corresponding bootstrap program.
- Bootstrap program load OS kernel into main memory and OS boot
- Startup Process
    - Linux kernel access Root file system.
    - systemd/init (sbin directory) process is executed (pid=1).
    - systemd is designed to start (independent) services parallely. In init services were started sequentially.
    - It starts multiple child processes as per configred run-level.
    - Runlevels are like "checkpoints"/"states".
- Runlevels
    - 1 - Single user mode (Used for failsafe/rescue)
    - 2 - Multi user mode (User login enabled, User files accessible)
    - 3 - Networking (Multiuser with Networking on CLI) -- multi-user.target
    - 4 - Reserved (unused)
    - 5 - Graphical user interface -- graphical.target
    - https://likegeeks.com/linux-runlevels/
- Linux commands
    - terminal> runlevel
    - terminal> init 0 --> shutdown
    - terminal> init 6 --> reboot
    - terminal> init 3 --> go to runlevel 3 (GUI will be stopped if already started)
    - terminal> init 5 --> go to runlevel 5 (GUI will be started)
    - terminal> startx --> start GUI (give this command from runlevel 3)