The UNIX Boot Process

May 29, 2019

This document will describe the UNIX boot process in decreasing levels of abstraction.

1 Visible

Power on the computer, the system boots up.

2 Abstract Stages

2.1 Basic Hardware Detection

2.2 Execute Firmware Initialisation

2.3 Locating and Executing Initial Boot Program

2.4 Locating and Starting the Kernel

2.5 Kernel Initialisation and Hardware Checks

2.6 Start INIT process, system processes, subsystems

3 Linux Stages

3.1 BIOS: Basic Input Output System

3.2 MBR: Master Boot Record

3.3 GRUB: Grand Unified Bootloader

3.4 Kernel

3.5 Init

3.6 RunLevel

4 Detailed

For the sake of simplicity this section will describe a cold-boot process from power-up, rather than a warm-boot restart. This means this description is slightly beyond solely the UNIX boot process, as early stages are required before any form of software execution, regardless of the operating system, can be carried out.

4.1 BIOS

All abstractions mention that the Basic Input Output System is the first stage to any boot process. The BIOS is Non-Volatile Firmware built into the hardware of a system in such a location that it can be automatically executed by the default operating location of the processor. Most systems describe the Power On Self Test (POST) in addition to the BIOS, but the exact delineation between the is vague, with many definitions ascribing similar responsibilities to each stage.

4.1.1 Reset Vector

The reset vector is the default location (address/pointer) of a central processing unit, which is used whenever the CPU is reset and able to execute instructions. The reset vector address refers to a section of non-volatile memory that contains instructions to start the operation of the CPU and then the rest of the boot-process. The non-volatile memory referred to is the storage location of the BIOS, which is kept in Read Only Memory, normally on the motherboard. This memory is non-volatile (flash memory in modern systems) so retains the contents while powered-down, but commonly is re-writable to allow BIOS updates or changes. The exact location used is CPU chipset dependent, which is a reason (one of many) for CPU/Motherboard compatibility.

Using the reset vector, the CPU can load instructions from the BIOS into the registers and begin processing instructions from the BIOS.

4.1.2 Hardware Detection and System Integrity

Still outside of the UNIX startup process, the BIOS will carry out a variety of tasks to verify and validate the state of the system hardware, such that it can continue with the boot process. This includes hardware checks; verifying the CPU registers, find size and verify RAM, verify controllers for interrupt (combines and sorts interrupts, the cornerstone of CPU operations, and feeds them to the CPU) and DMA (Direct Memory Access: allows hardware subsystems to access main memory independent of the CPU), verify other sections of the chipset, disks, components, GPUs, keyboard and other hardware. It may verify the BIOS code itself and initialise relevant sections.

4.1.3 Boot Devices

The BIOS will identify and select available hardware devices for booting. Early BIOS' would execute floppy disk then hard disk (in that order) if they could be found. Once

the boot options were selected, the next stage, boot loading, begins. Modern BIOS have many stages of complexity owing to increasing hardware complexity and a wide range of standards beyond this description, and in addition have extra functionality. Most modern BIOS' have configurable settings and allow user-operations like selecting the boot device, rather than running a default.

4.1.4 Modern BIOS

Modern BIOS will perform additional operations: initialising and locating all system buses, showing a UI for user configuration of the BIOS/Boot Process, pre-constructing a system environment for a target OS.

4.1.5 Boot Loading

With the hardware checks complete and the boot device selected, the BIOS will attempt to find 'boot loader' software on the boot device. It will execute the first example of this it finds and hand control of the hardware over to this software.

These boot loaders are located in *boot sectors* of the boot devices (hard disk, floppy disk, CD-ROM, flash drive) that contain machine code instructions to be loaded into memory and executed by the CPU, analogous to the BIOS but an additional level of complexity up. The BIOS checks boot sectors of the boot devices, it may check for a signature in the byte data that marks the sector as a boot sector, before loading the contents and transferring control. The BIOS will not examine the contents of the boot sector, which is the responsibility of the software in that sector. The exact location is normally defined by the hardware, it may be the first disk block or a specific partition.

For LINUX systems, this boot loader is the Master Boot Record (MBR). The BIOS will load the MBR from whatever device and transfer control to it.

4.1.6 Failures

As always, most of these stages could fail which would invariable by fatal to the boot, with varying degrees of BIOS diagnostics between systems allowing for figuring out the issue. The most common is display of BIOS output to a monitor or output device, or emitting a variety of beeps from a speaker built into the motherboard.

4.2 MBR