APPENDIX D

System Boot Flow

When we press the power button on our computing device, we are well aware that the system goes through a bootup process. The boot process culminates with the system being ready for use. But what happens during the boot process is not very well understood widely. In this chapter, we will strive to resolve that.

As shown in Figure D-1, there are four main boot phases on IA devices. The first phase is system hardware bring-up/power-on, which is primarily hardwired to bring up the foundation for software components to get started and take over. Then the BIOS (aka system firmware) phase is responsible for basic initialization and bring-up of system hardware enabling things to pass to the next stage, where the boot loader loads the OS into memory and then begins OS initialization. This last phase takes care of initialization of critical parts of the HW and SW system before making itself available to the user.

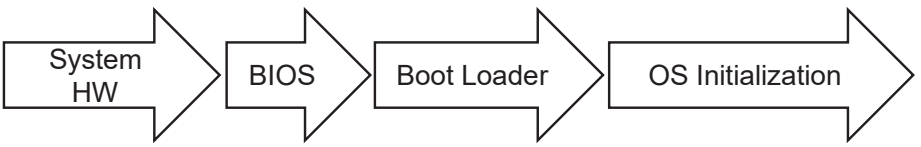

Figure D-1. *High-Level System Boot Flow*

P. D. Crutcher et al., *Essential Computer Science*,
https://doi.org/10.1007/978-1-4842-7107-0

On receiving a "Power Good" signal, CPUs are hardwired to start fetching and executing at a predefined location (address), which is called the "Reset Vector." The Reset Vector points to BIOS code. So, when the CPU is out of reset and starts fetching code from the "Reset Vector," it happens to be BIOS code, which is how BIOS code gets the control and starts executing. Keep in mind that before control comes to CPU and BIOS code, there are a few system hardware- and firmware-related initializations and configurations that happen.

BIOS discovers, enumerates, and initializes the HW devices present. After that it runs power-on self-test (POST). The POST is responsible for validating the sanity of fundamental hardware components. One of the fundamental hardware components in the system happens to be memory. BIOS has a component specialized for memory initialization called the Memory Reference Code (MRC). Another of BIOS's responsibility is to prepare the hardware configuration and memory map and pass those to the OS, in the form of tables. The format and mechanism of information exchange is defined by a standard body, Unified Extensible Firmware Interface (UEFI). Today, most BIOS is UEFI spec compliant. BIOS also adheres to the ACPI specification in passing platform resource(s) information to the OS.

If all goes well, BIOS now identifies a bootable disk and reads the master boot record (MBR) of that disk. The MBR is located in the first sector of the bootable media (could be hard drive, flash, solid-state device, etc.).

The MBR is 512 bytes in size. It has three components: primary boot loader information in the first 446 bytes, partition table in the next 64 bytes, and MBR validation check in the last 2 bytes.

The primary boot loader in the MBR will attempt to locate an active (bootable) partition in the media's partition table. If such a partition is found, the boot sector of that partition is loaded in memory, and then the control jumps to that. Each operating system has its own boot sector

format. The boot sector has a small program that locates the OS loader, reads that into memory, and launches that.

The OS loader loads essential system drivers that are required to read data from the disk and initializes the system to the point where the kernel can begin execution.

After OS loading, the OS initialization phase starts. In the OS initialization phase, first, the kernel initialization and plug-and-play activity happen. After that, relevant services are started, and the user interface (could be a command line shell or a full-blown graphical user interface) is presented and the system is now ready for use.