The boot process

Pressing the power button

1. The button wires send a signal to the motherboard
2. The motherboard reroutes the signal to the power supply (PSU)
3. The signal contains a single bit (not byte) of data
   1. 0 == no power, 1 == power is supplied
4. Upon receiving the signal, the PSU begins supplying power
5. The PSU sends the power\_good signal to the BIOS

BIOS POST

1. BIOS receives the power\_good signal
2. BIOS starts POST
3. POST loads BIOS at the end of memory, possibly 0xFFFFF0, and puts a jump instruction at the first byte of memory
4. The processor's instruction pointer (IP) is set to 0, and the processor takes control
5. The processor starts executing instructions at the IP location, which is 0x0. Recall that POST placed a jump instruction at this address. The instruction jumps to where the BIOS is loaded (again, possibly 0xFFFFF0)
6. The processor jumps to the BIOS and starts executing it

The BIOS

1. The BIOS creates an Interrupt Vector Table (IVT), and provides some basic interrupt services
2. The BIOS performs some more hardware tests, and supplies a setup utility
3. The BIOS executes interrupt 0x19 to try to find a bootable device. Two possibilities:
   1. 0x19 returns. In this case no bootable device is found. The BIOS tries the next device in the boot order.
      1. If there is no next device, the BIOS prints a "No OS found" error message, and halts the system.

Interrupts and the IVT

1. An interrupt is a subroutine that can be executed from many different programs.
2. These interrupts are stored at address 0x0, in a table called the IVT.

BIOS interrupt 0x19

Int 0x19 is the interruption for the bootloader. It reboots the system through a warm reboot, without clearing memory or restoring the IVT.

The BIOS executes int 0x19.

Int 0x19 reads the first sector (sector 1, head 0, track 0) of the first hard disk.

1. Sector: a group of 512 bytes. Sector 1 is the first 512 bytes of the disk
2. Head: side of the disk. Head 0 == front side, head 1 == back side. Most disks only have one head: head 1.
3. Track: a collection of sectors
   1. On floppy disks there are 18 sectors per track

Back to the BIOS attempting to execute 0x19. Suppose 0x19 succeeds:

1. The bootsector is loaded at 0x7C00. 0x19 jumps to this address, thereby giving control to the bootloader.
2. **Repeat: the bootloader is loaded at 0x7C00**

At this point the bootloader is in control

Bootloader theory

Bootloaders:

1. Are stored with the Master Boot Record (MBR)
2. Are in the first sector of the disk
3. Fit within a single sector (512 bytes)
4. Are loaded by the BIOS INT 0x19 at address 0x7C00.

Hardware exceptions

These are like software exceptions, but it's the processor that executes hardware exceptions.

CLI and STI instructions

1. CLI: clear interrupts. Disables all interrupts
2. STI: enables interrupts

Double fault hardware exception

If the processor finds a problem during execution, it executes a Second Fault Exception Handler (aka double fault), which is interrupt 0x8.

If the processor still can't continue after a double fault, it will execute a triple fault.

Triple fault

A CPU that triple faults == the system hard reboots.

In early stages, such as the bootloader, the system triple faults upon any bug.

Developing a simple bootloader