

Slide 1

Slide 2

The Boot Process What happens between the time you turn the computer on (or initiate reboot) and the point at which you get a login prompt is — complicated, mysterious, and involves both hardware and software. Today's topic is to demystify it as much as possible. Textbook has some useful short information, indexed under "boot" and "BIOS". I'm going by that, from the book *Linux Kernel Internals* (see syllabus), and various online sources.

Booting — Hardware

 When a PC is powered on, hardware starts the BIOS (Basic Input Output System), a program that lives in/on some form of nonvolatile memory. It contains functions to read from the keyboard, write to the screen, and do disk I/O.

Slide 3

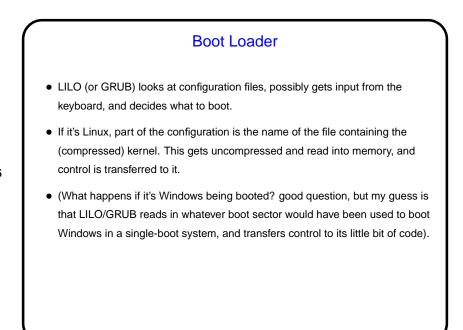
- This BIOS first does a "Power-On Self Test" (POST) check how much memory is installed, whether basic devices are installed and responding.
- It determines which device to try to boot from based on information also stored in non-volatile memory. It then reads the first sector from this device — "boot sector" or "master boot record".

Boot Sector / Master Boot Record

• First sector on device from which we're booting must contain (in a format known to the hardware / BIOS) a little bit of code, enough to get the boot process going.

Slide 4

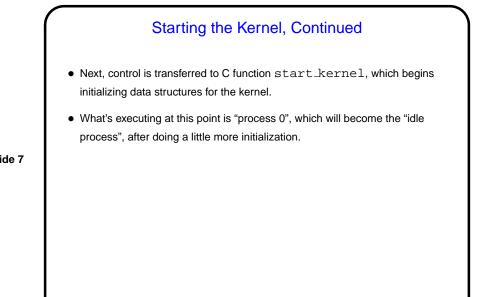
- For partitioned devices, this first sector (MBR) contains a partition table, indicating which partition contains the logical device from which booting is supposed to be done, and where to find that logical device's boot sector.
- Either way, we get a little bit of code, which when executed (presumably with the help of the BIOS) reads in something else from disk to memory, and transfers control to it. The "something else" could be the actual operating system, or a "boot loader" (such as LILO or GRUB, for Linux systems).
- (From here on, the discussion will be somewhat Linux-specific, and alas will be based on Linux as it existed a few years ago.)



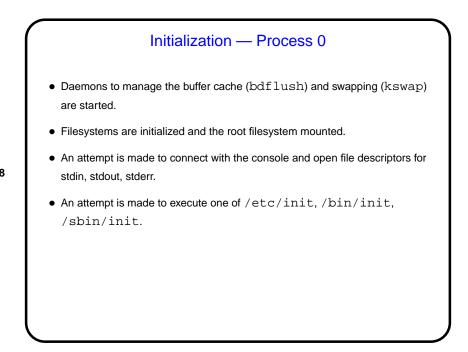
Slide 5

Starting the Kernel First thing executed is assembly code that does hardware initialization, including: Put the processor in protected mode. Do initialization for the MMU (set up page table for kernel). Do initialization for interrupt processing (interrupt table/vector).

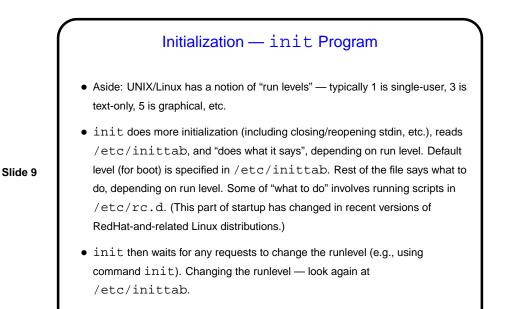
Slide 6



Slide 7



Slide 8



Minute Essay

• We have one more "real" class. I may use it for review. Unless there are other requests?

Slide 10