Booting of a Computer System

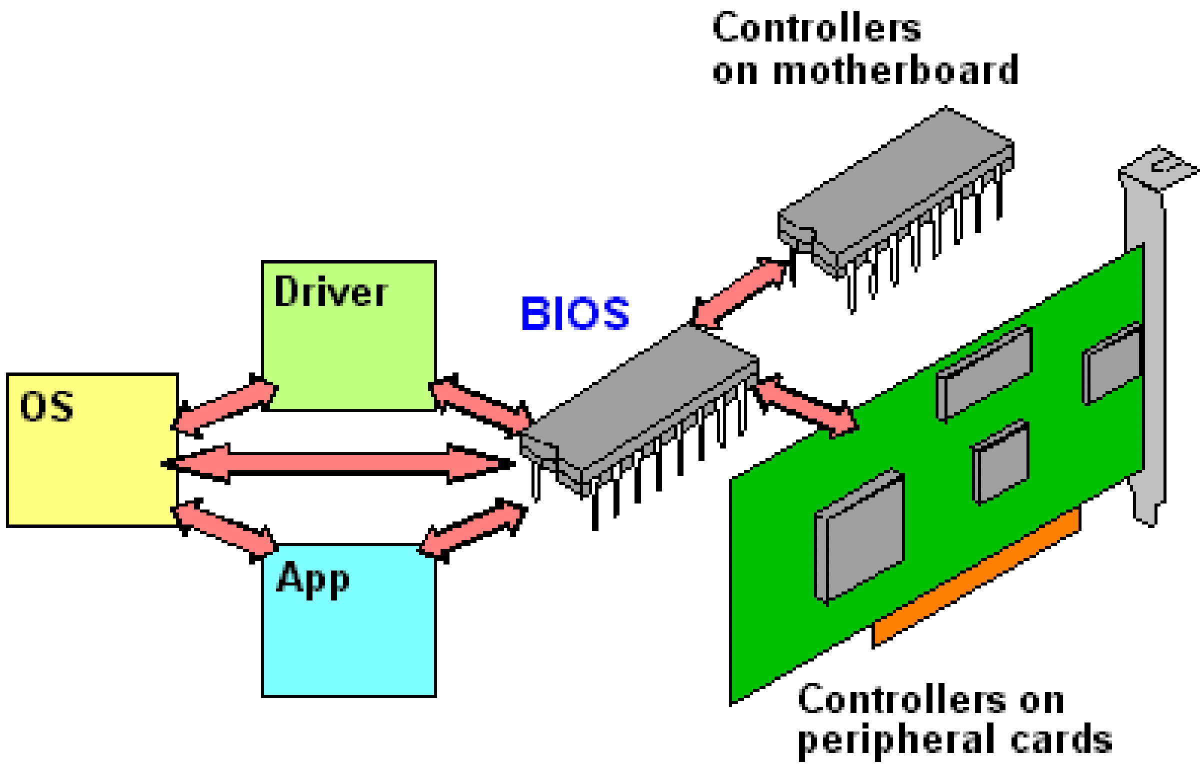
Why is Booting Required ?

- Hardware doesn’t know where the operating system resides and how to load it.
- Need a special program to do this job – Bootstrap loader.
    - E.g. BIOS – Boot Input Output System.
- Bootstrap loader locates the kernel, loads it into main memory and starts its execution.
- In some systems, a simple bootstrap loader fetches a more complex boot program from disk, which in turn loads the kernel.

How Boot process occurs ?

- Reset event on CPU (power up, reboot) causes instruction register to be loaded with a predefined memory location. It contains a jump instruction that transfers execution to the location of Bootstrap program.
- This program is form of ROM, since RAM is in unknown state at system startup. ROM is convenient as it needs no initialization and can’t be affected by virus.

BIOS Interaction

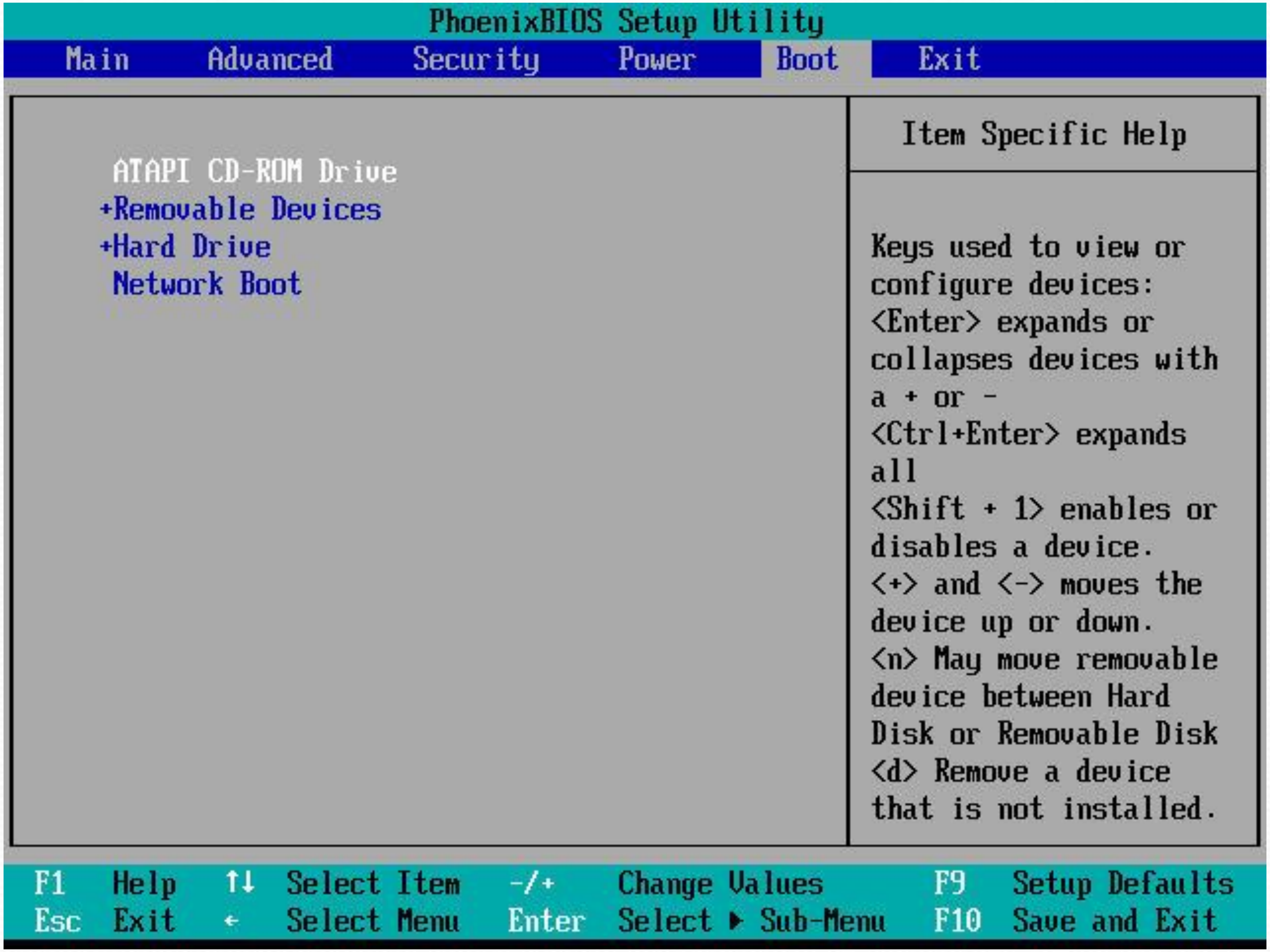


BIOS Interaction

Tasks performed at boot up

- Run diagnostics to determine the state of machine. If diagnostics pass, booting continues.
- Runs a Power-On Self Test (*POST*) to check the devices that the computer will rely on, are functioning.
- BIOS goes through a preconfigured list of devices until it finds one that is bootable. If it finds no such device, an error is given and the boot process stops.
- Initializes CPU registers, device controllers and contents of the main memory. After this, it loads the OS.

BIOS Setup



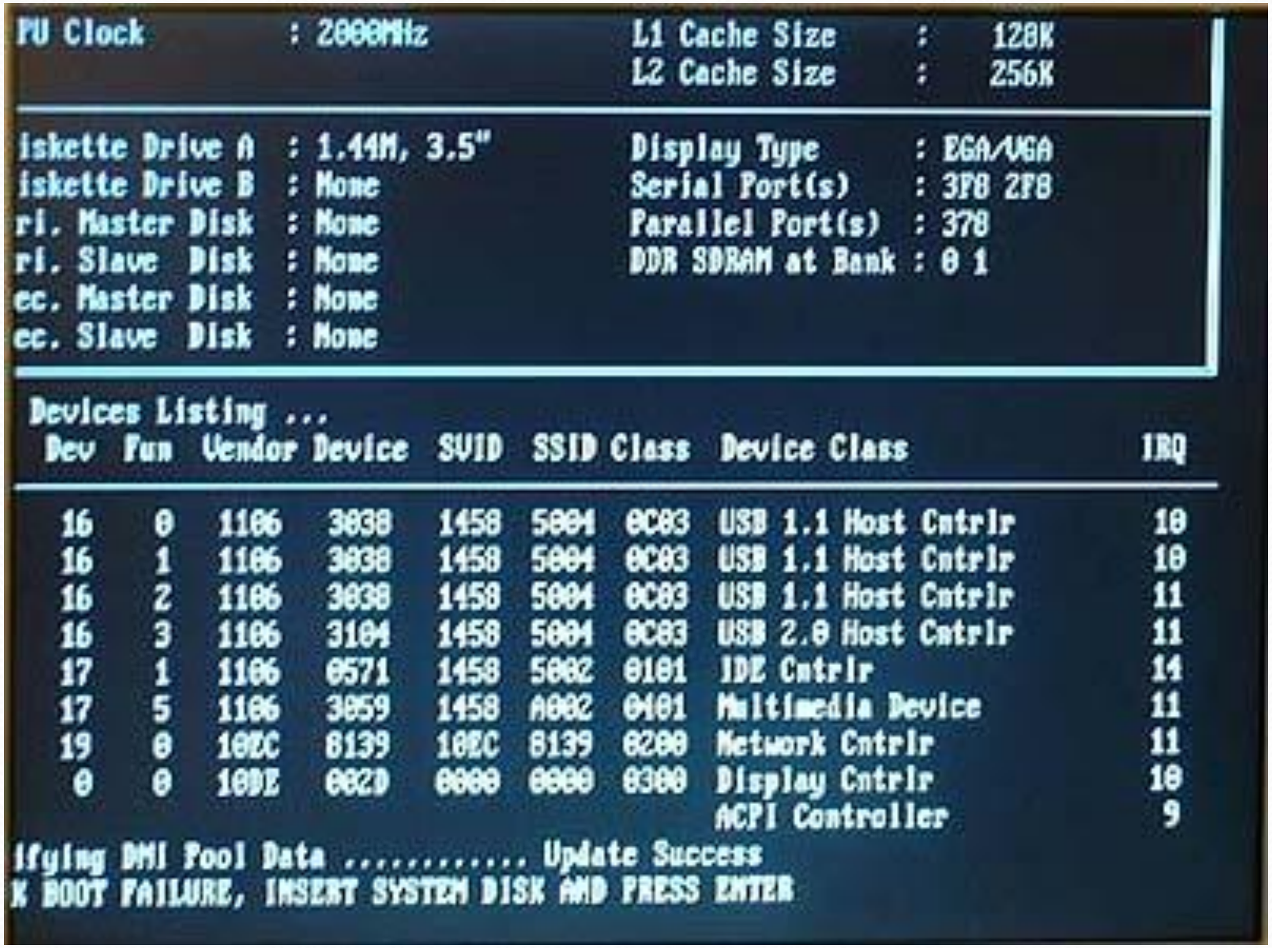
PhoenixBIOS Setup Utility

ATAPI CD-ROM Drive
+Removable Devices
+Hard Drive
Network Boot

Item Specific Help

Keys used to view or configure devices:
<Enter> expands or collapses devices with a + or -
<Ctrl+Enter> expands all
<Shift + 1> enables or disables a device.
<+> and <-> moves the device up or down.
<n> May move removable device between Hard Disk or Removable Disk
<d> Remove a device that is not installed.

Boot Procedure



```
PU Clock          : 2000MHz              L1 Cache Size      :    128K
                                         L2 Cache Size      :    256K

iskette Drive A   : 1.44M, 3.5"          Display Type       : EGA/VGA
iskette Drive B   : None                 Serial Port(s)     : 3F8 2F8
ri. Master Disk   : None                 Parallel Port(s)   : 378
ri. Slave  Disk   : None                 DDR SDRAM at Bank  : 0 1
ec. Master Disk   : None
ec. Slave  Disk   : None

 Devices Listing ...
  Dev  Fun  Vendor Device  SVID  SSID Class  Device Class              IRQ

   16    0   1106   3038   1458  5004  0C03  USB 1.1 Host Cntrlr        10
   16    1   1106   3038   1458  5004  0C03  USB 1.1 Host Cntrlr        10
   16    2   1106   3038   1458  5004  0C03  USB 1.1 Host Cntrlr        11
   16    3   1106   3104   1458  5004  0C03  USB 2.0 Host Cntrlr        11
   17    1   1106   0571   1458  5002  0101  IDE Cntrlr                 14
   17    5   1106   3059   1458  A002  0401  Multimedia Device          11
   19    0   10EC   8139   10EC  8139  0200  Network Cntrlr             11
    0    0   10DE   002D   0000  0000  0300  Display Cntrlr             10
                                             ACPI Controller             9
ifying DMI Pool Data ............ Update Success
K BOOT FAILURE, INSERT SYSTEM DISK AND PRESS ENTER
```

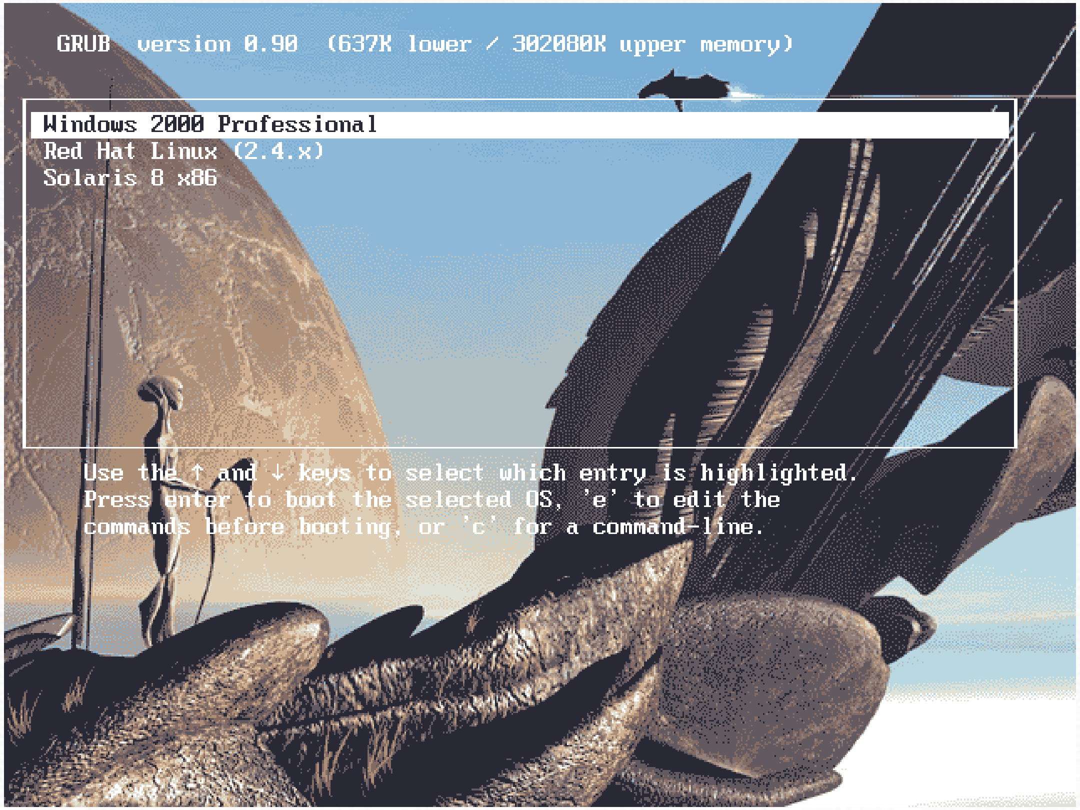
Tasks performed at boot up (Contd)

- On finding a bootable device, the BIOS loads and executes its boot sector. In the case of a hard drive, this is referred to as the master boot record (*MBR*) and is often not OS specific.
- The MBR code checks the partition table for an active partition. If one is found, the MBR code loads that partition's boot sector and executes it.
- The boot sector is often operating system specific, however in most operating systems its main function is to load and execute a kernel, which continues startup.

Secondary Boot Loaders

- If there is no active partition or the active partition's boot sector is invalid, the MBR may load a secondary boot loader and pass control to it and this secondary boot loader will select a partition (often via user input) and load its boot sector.
- Examples of secondary boot loaders
  - GRUB – GRand Unified Bootloader
  - LILO – LInux LOader
  - NTLDR – NT Loader

GRUB Loader



```
GRUB  version 0.90  (637K lower / 302080K upper memory)

 Windows 2000 Professional
 Red Hat Linux (2.4.x)
 Solaris 8 x86

    Use the ↑ and ↓ keys to select which entry is highlighted.
    Press enter to boot the selected OS, 'e' to edit the
    commands before booting, or 'c' for a command-line.
```

Booting and ROM

- System such as cellular phones, PDAs and game consoles stores entire OS on ROM. Done only for small OS, simple supporting hardware, and rugged operation.
- Changing bootstrap code would require changing ROM chips.
  - EPROM – Erasable Programmable ROM.
- Code execution in ROM is slower. Copied to RAM for faster execution.

Example : DOS

- After identifying the location of boot files, BIOS looks at the first sector (512 bytes) and copies information to specific location in RAM (7C00H) - Boot Record.
- Control passes from BIOS to a program residing in the boot record.
- Boot record loads the initial system file into RAM. For DOS, it is IO.SYS .
- The initial file, IO.SYS includes a file called SYSINIT which loads the remaining OS into the RAM.
- SYSINIT loads a system file MSDOS.SYS that knows how to work with BIOS.
- One of the first OS files that is loaded is the system configuration file, CONFIG.SYS in case of DOS. Information in the configuration file tells loading program which OS files need to be loaded (e.g. drivers)
- Another special file that is loaded is one which tells what specific applications or commands user wants to be performed as part of booting process. In DOS, it is AUTOEXEC.BAT. In Windows, it’s WIN.INI .