

Understanding Boot Process

BIOS, UEFI, etc.

Abhijit A. M.
abhijit.comp@coep.ac.in

All credits: “Hands on Booting” by Yogesh
Babar

What do we already know about “boot sequence”?

- **When POWERed-ON**

- The CPU's IP (or CS+IR, etc) has manufacturer defined value
 - The Motherboard manufacturer has ensured that a code called “Basic Input Output System(BIOS)” is at this location
- CPU starts doing fetch-decode-execute-repeat
 - So CPU is running BIOS
- BIOS
 - code does a Power On Self Test (POST)
 - Initializes hardware by writing to control ports, etc.
 - Reads list of boot-devices, and finds the first available device

What do we already know about “boot sequence”?

- BIOS (contd)
 - Reads “Sector-0” of the “Boot-device”, and loads it in RAM
 - Sector-0 is “boot-loader” (at least that’s what we said!)
 - Jumps to it. Now boot-loader is running
- Boot-loader
 - Loads the OS kernel (boot-loader somehow knows where kernel is!) in RAM
 - Jumps to kernel code
 - Now kernel is running
- Kernel
 - Creates first process... and then fork-exec will create other processes

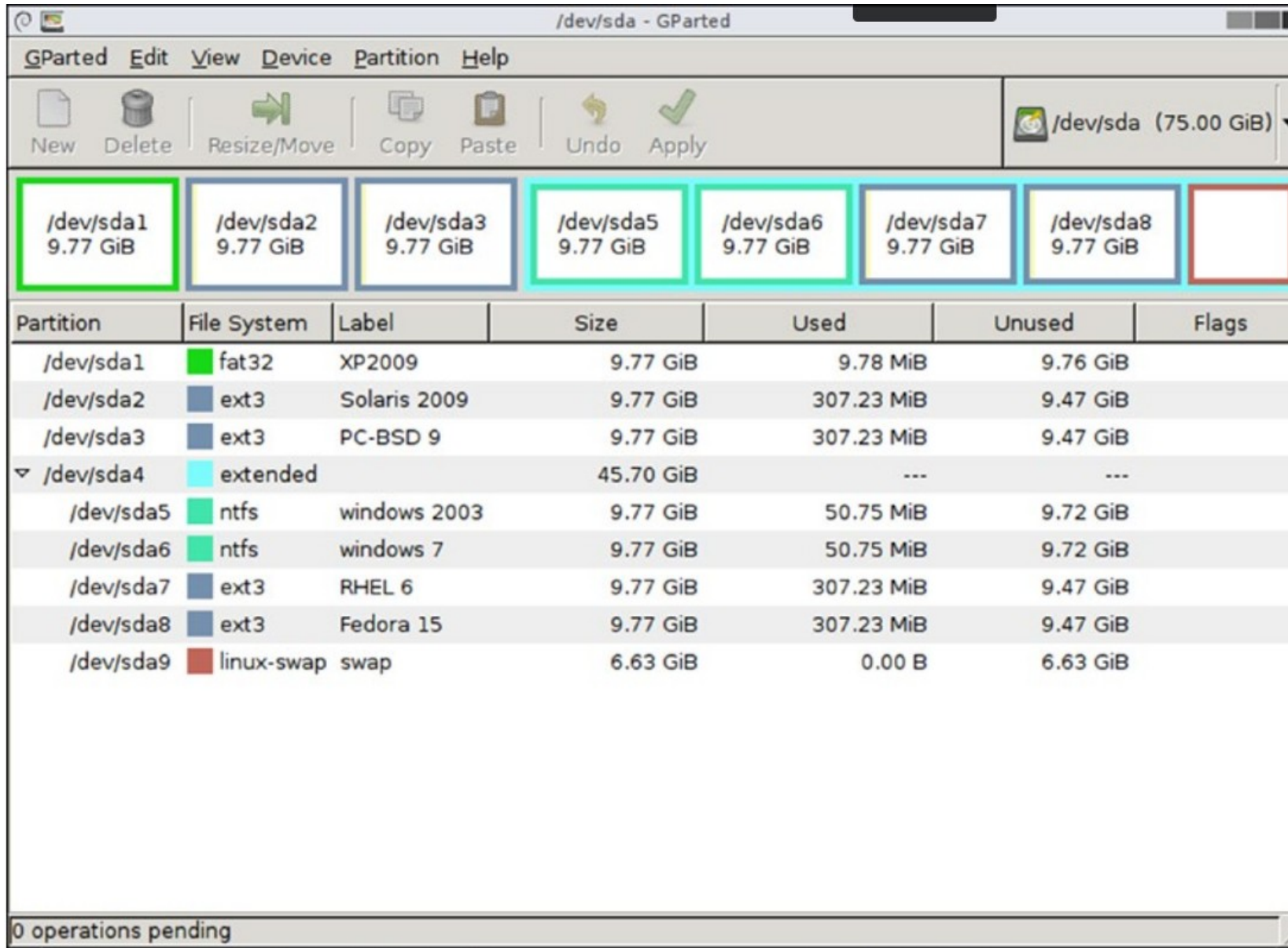
Legacy boot

The world of BIOS + MBR

Primary and Logical Partitions

- **Max 4 primary partitions**
- **Want more?**
 - **2nd or 3rd or 4th partition can be made “extended”**
 - **Within extended partition create logical partitions**

Primary and Logical Partitions



Partition	File System	Label	Size	Used	Unused	Flags
/dev/sda1	fat32	XP2009	9.77 GiB	9.78 MiB	9.76 GiB	
/dev/sda2	ext3	Solaris 2009	9.77 GiB	307.23 MiB	9.47 GiB	
/dev/sda3	ext3	PC-BSD 9	9.77 GiB	307.23 MiB	9.47 GiB	
▼ /dev/sda4	extended		45.70 GiB	---	---	
/dev/sda5	ntfs	windows 2003	9.77 GiB	50.75 MiB	9.72 GiB	
/dev/sda6	ntfs	windows 7	9.77 GiB	50.75 MiB	9.72 GiB	
/dev/sda7	ext3	RHEL 6	9.77 GiB	307.23 MiB	9.47 GiB	
/dev/sda8	ext3	Fedora 15	9.77 GiB	307.23 MiB	9.47 GiB	
/dev/sda9	linux-swap	swap	6.63 GiB	0.00 B	6.63 GiB	

0 operations pending

Primary and Logical Partitions

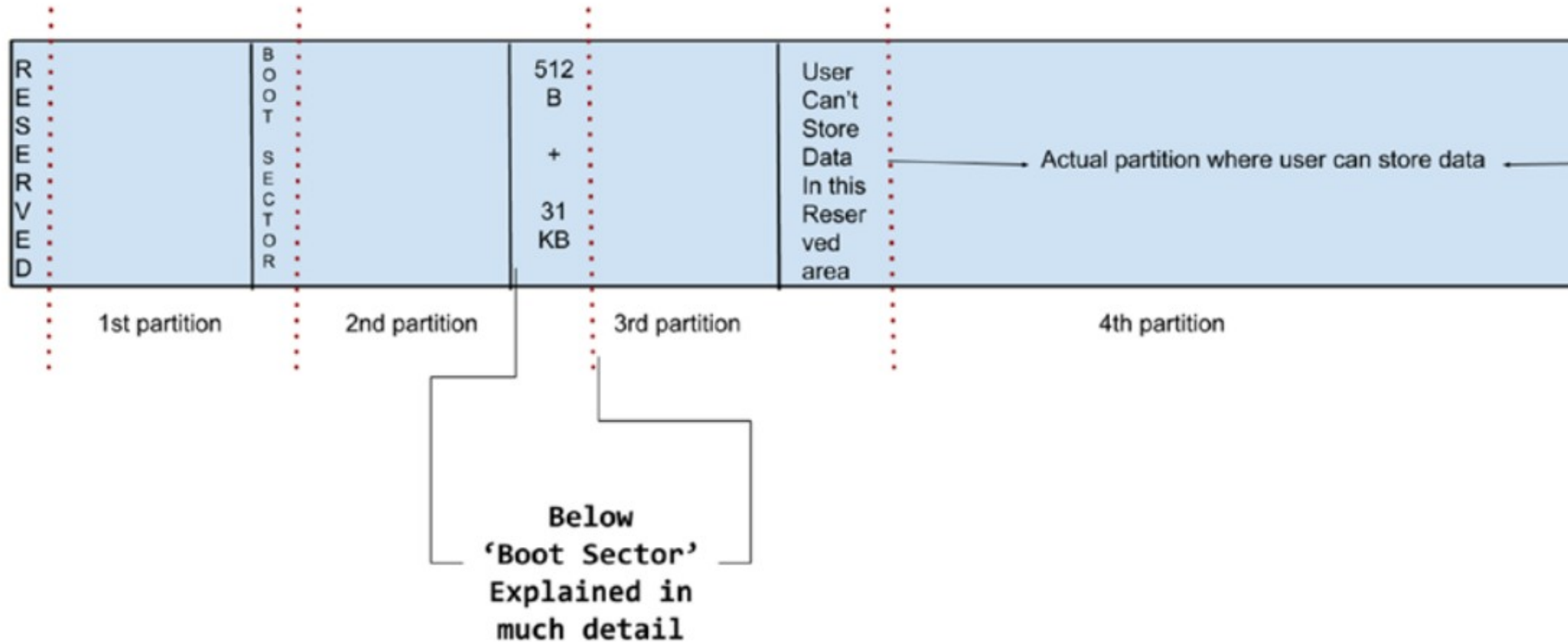


Figure 2-6. *The disk layout on a BIOS-based system*

OS	Rules
Unix	Unix operating systems (OpenSolaris and BSD) have to be installed on a primary partition only.
Linux	Linux does not have any installation rules. It can be installed on any primary or logical partition.
Windows	The Windows operating system can be installed on any partition (primary or logical), but the predecessor of the Windows family has to be present on the first primary. That means you can install Windows 7 on a logical partition, but its predecessor, which is XP or win2k3, has to be present on the first primary partition. Also, you cannot break the Windows operating system sequence of installation. For example, one cannot install Windows 7 first and then the older win2k3 or XP. It has to be in this sequence: 98, then 2000, and then XP.

Windows XP installer

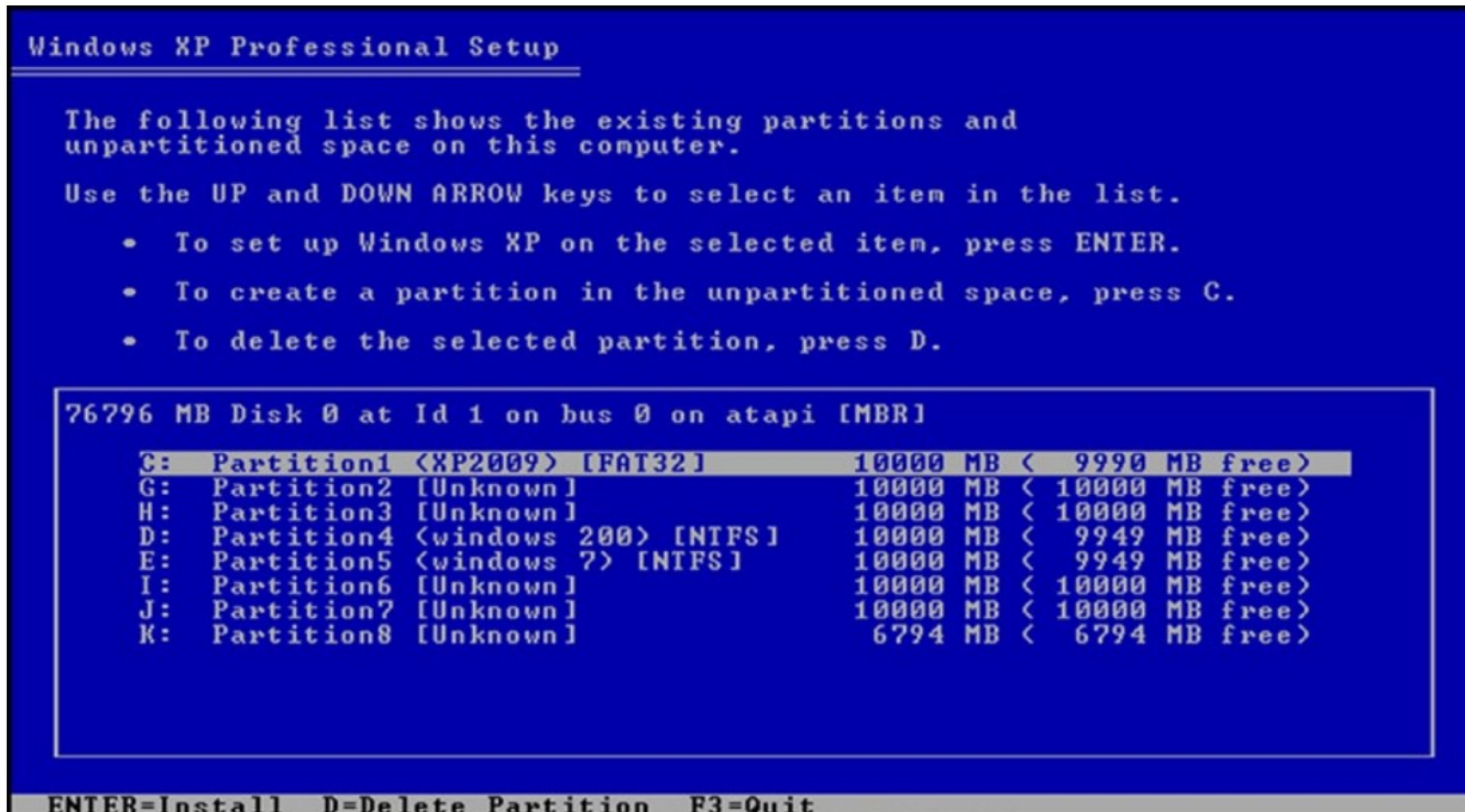


Figure 2-4. Partition layout shown by XP's installer

“Boot sector” is a misnomer here!

it's more than a sector!

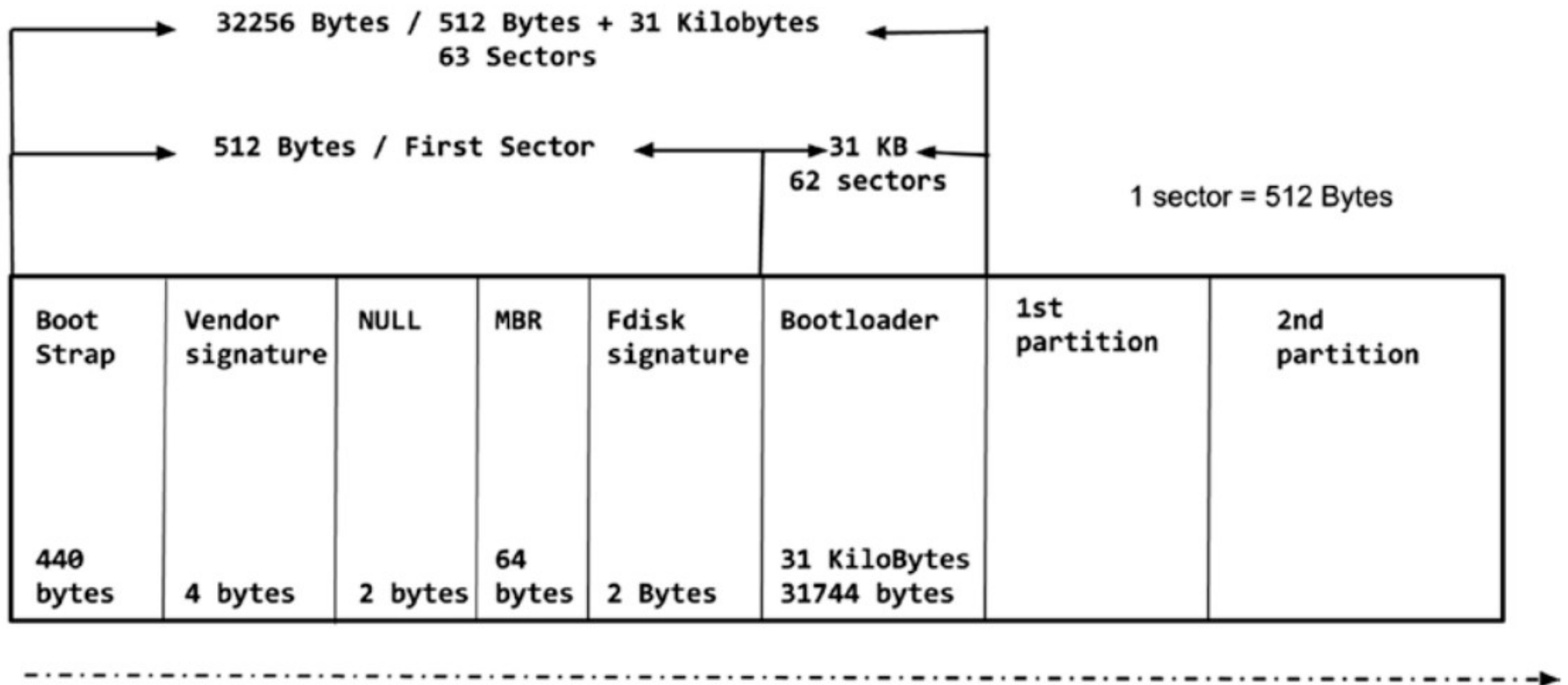


Figure 2-7. The boot sector

Bootloader code is in three parts!

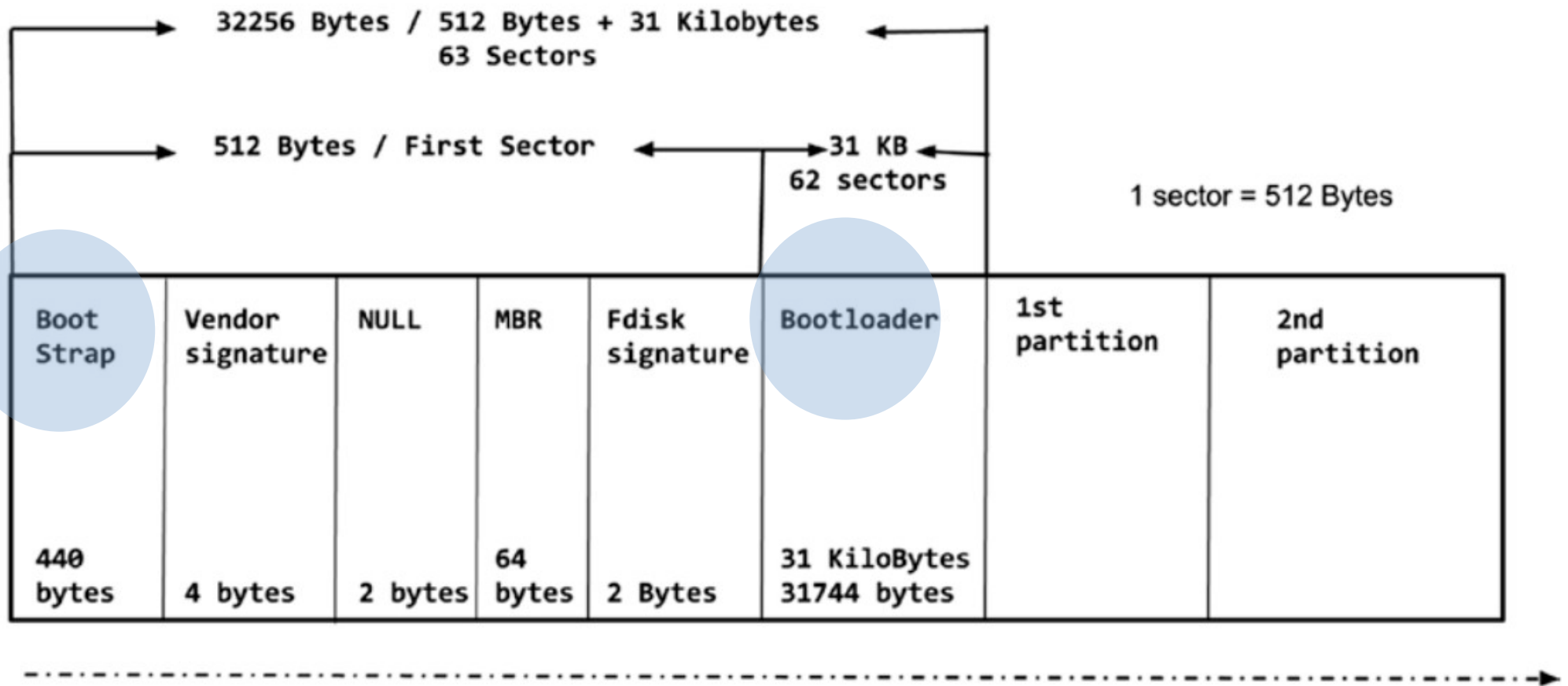


Figure 2-7. The boot sector

Bootloader code is in three parts!

Location	Size	Part	Information
Bootstrap	440 bytes	NTLDR part-1	The tiniest part
Bootloader	31 KB	NTLDR part-2	Bigger compared to part-1
Inside an actual OS partition	No size limitation	NTLDR part-3	The biggest part

See **C:\NTLDR** (NTLDR-3 file) file on Windows XP
Try deleting this file and booting

C:\Windows\winload.exe is Windows kernel

Boot sequence of Win-Xp

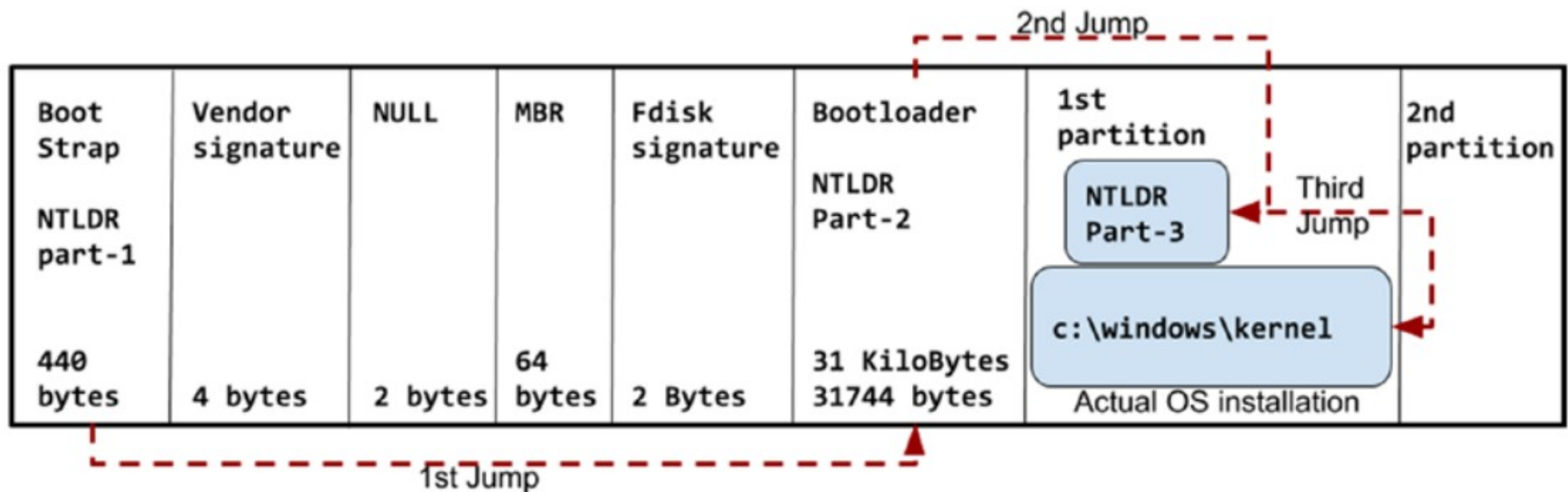
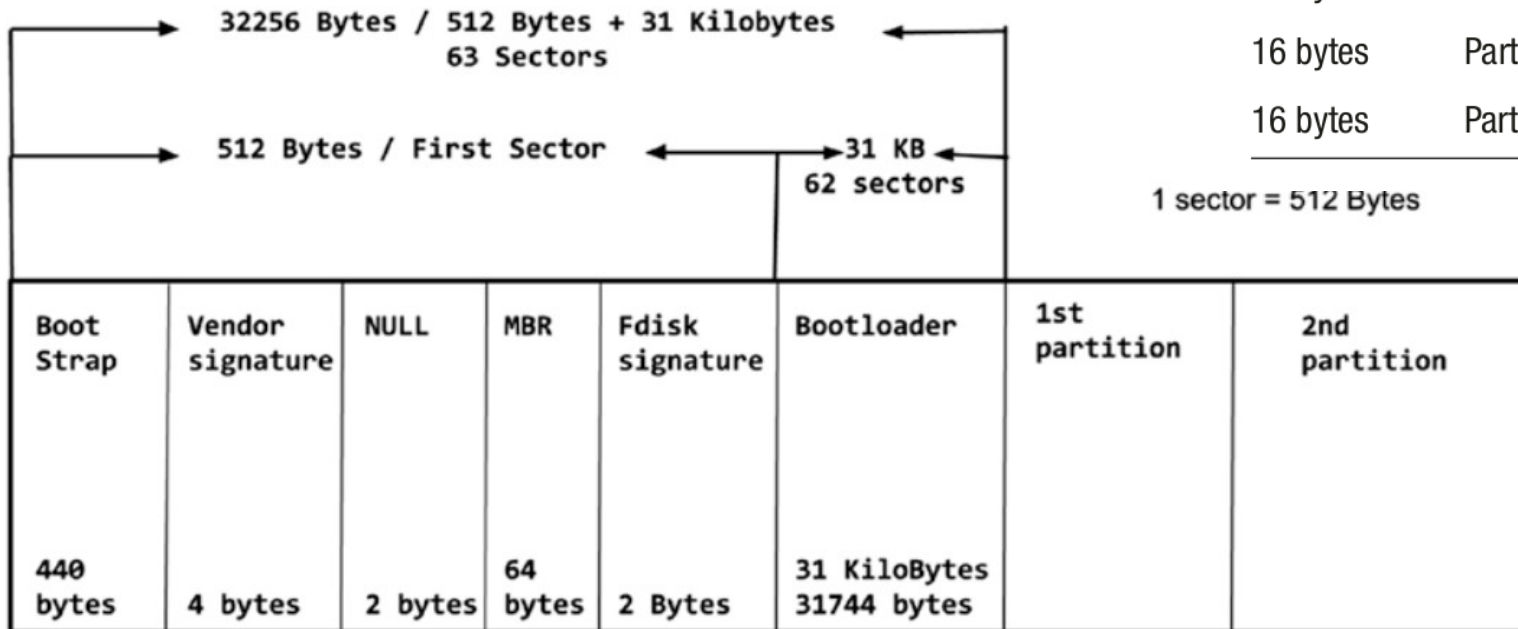


Figure 2-9. *The boot sequence of Windows XP*

Note: The diagram is “conceptual”. In reality, each stage loads the next stage in RAM and “jumps” (that is calls “jump” instruction) to the location of the next stage in RAM. NO jumps happen on disk!

More on boot region

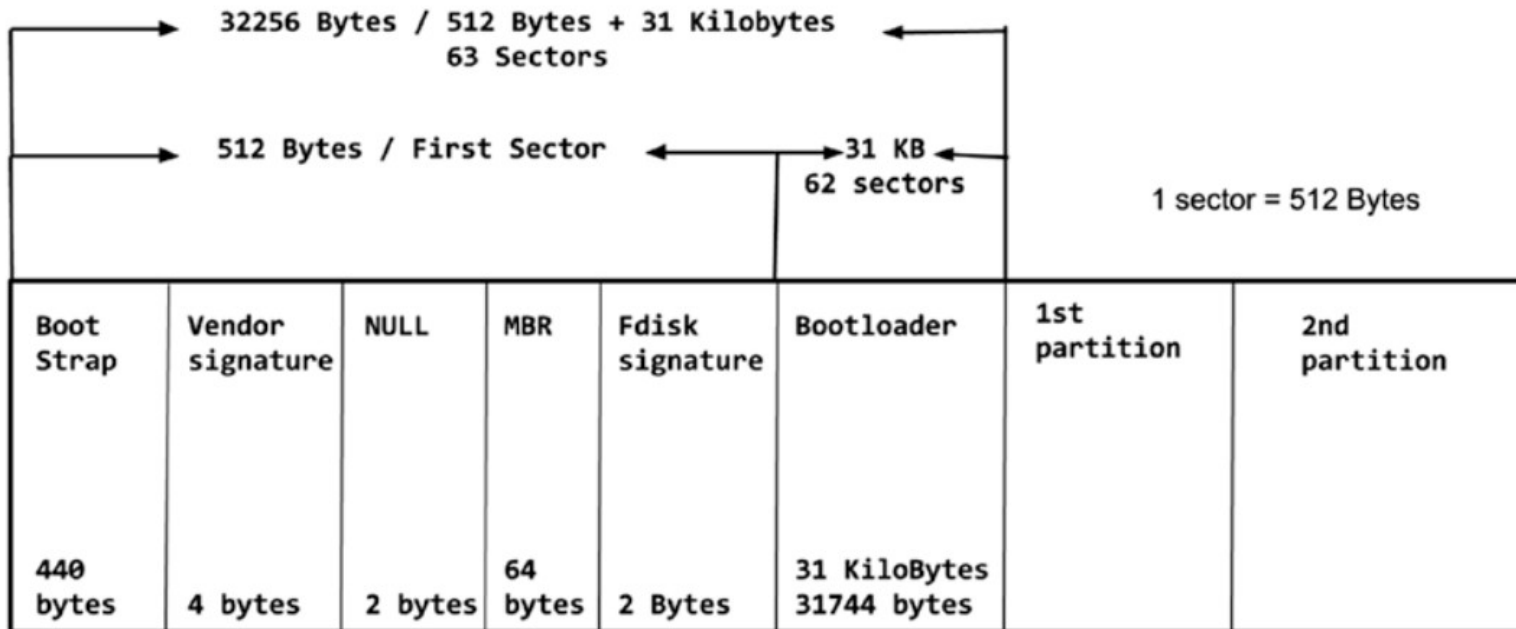
- Vendor signature: Seagate, WD, etc.
- NULL (is 0, else disk is bad)
- MBR: reason why only 4 partitions!



Size	Parts	Stores
16 bytes	Part-1	First partition's information
16 bytes	Part-2	Second partition's information
16 bytes	Part-3	Third partition's information
16 bytes	Part-4	Fourth partition's information

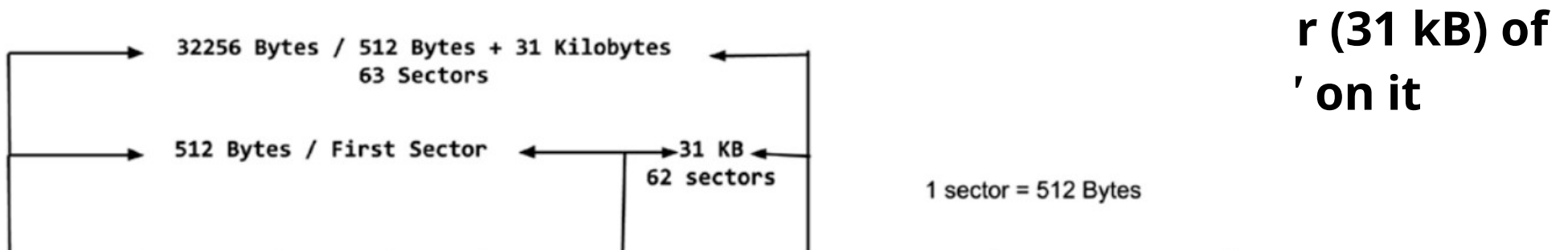
More on boot region

- **Fdisk signature**
 - Can be "*" or something else
 - When *, called boot flag, or active/inactive flag



More on boot region

- **Fdisk signature**
 - Can be "*" or something else
 - When , called boot flag, or active/inactive flag, OS kernel is supposed to be loaded from partition with "*" flag
 - **Makes sense only when you have multi-boot system**



Boot Strap	Vendor signature	NULL	MBR	Fdisk signature	Bootloader	1st partition	2nd partition
440 bytes	4 bytes	2 bytes	64 bytes	2 Bytes	31 KiloBytes 31744 bytes		

Flowchart

- **Run: BIOS**
- **Run: Partition-1, Bootloader part-1: Sector-0 , 440 bytes**
- **Run: Partition-1, Bootlaoder part-2: 31k**
 - Checks where is the “*” mark, that is the active flag
 - **If on partition-1**
 - **Run Partition-1, Bootlaoder part-3**
 - **Else**
 - **Run, Partition-x, Bootloader, part-0**
- **The Bootloader-3 part of whichever partition in the end**
 - should be able to recognize the other operating systems
 - Linux recognizes all, windows recognizes only itself, for other OS: depends

UEFI

Unified Extended Firmware Interface

BIOS Limitations

- BIOS will only be able to jump to the first sector, which is 512 bytes.
- BIOS cannot generate good graphics/GUIs.
- You cannot use a mouse in the BIOS.
- The maximum partition size is 2.2 TB.
- The BIOS is dumb because it does not understand the bootloader or the OS.
- It struggles to initialize the new-generation hardware devices.

UEFI advantages

- UEFI can use the full CPU. Unlike the BIOS (which is stuck with 16 bits of processor), UEFI can access up to 64 bits.
- UEFI can use a full RAM module. Unlike 1 MB of address space of the BIOS, UEFI can support and use terabytes of RAM.
- Instead of 64 bytes of a tiny MBR, UEFI uses the GPT (GUID) partition table, which will provide an infinite number of partitions, and all will be primary partitions. In fact, there is no concept of primary and logical partitions.
- A maximum partition size is 8 zettabytes.
- UEFI has enterprise management tools.
 - Boot remotely
 - Browse internet
 - Change firmware remotely

UEFI advantages

- **It's a small OS**
 - a) You will have full access to audio and video devices.
 - b) You will be able to connect to WiFi.
 - c) You will be able to use the mouse.
 - d) In terms of the GUI, UEFI will provide a rich graphics interface.
 - e) UEFI will have its own app store like we have for Android and Apple phones.
 - f) You will be able to download and use the applications from the UEFI app store, just like with Android and Apple phones. Hundreds of apps are available such as calendars, email clients, browser, games, shells, etc.
 - g) UEFI is able to run any binary that has an EFI executable format.
 - h) It boots operating systems securely with the help of the Secure Boot feature. We will discuss the Secure Boot feature in depth later in this book.
 - i) UEFI is backward compatible, meaning it will support the “BIOS way” of booting. In other words, operating systems that do not have UEFI support will also be able to boot with UEFI.

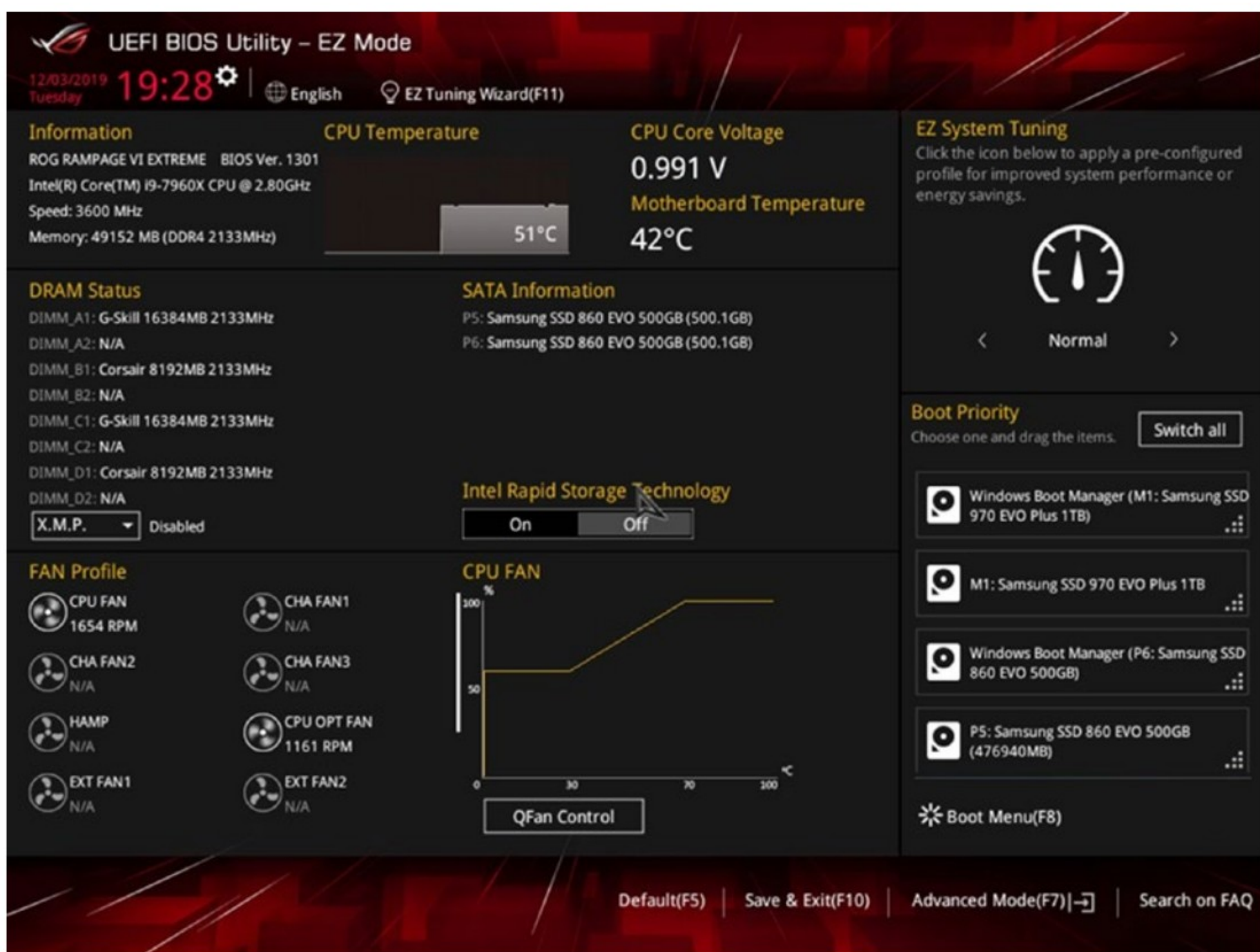


Figure 2-105. ASUS UEFI implementation

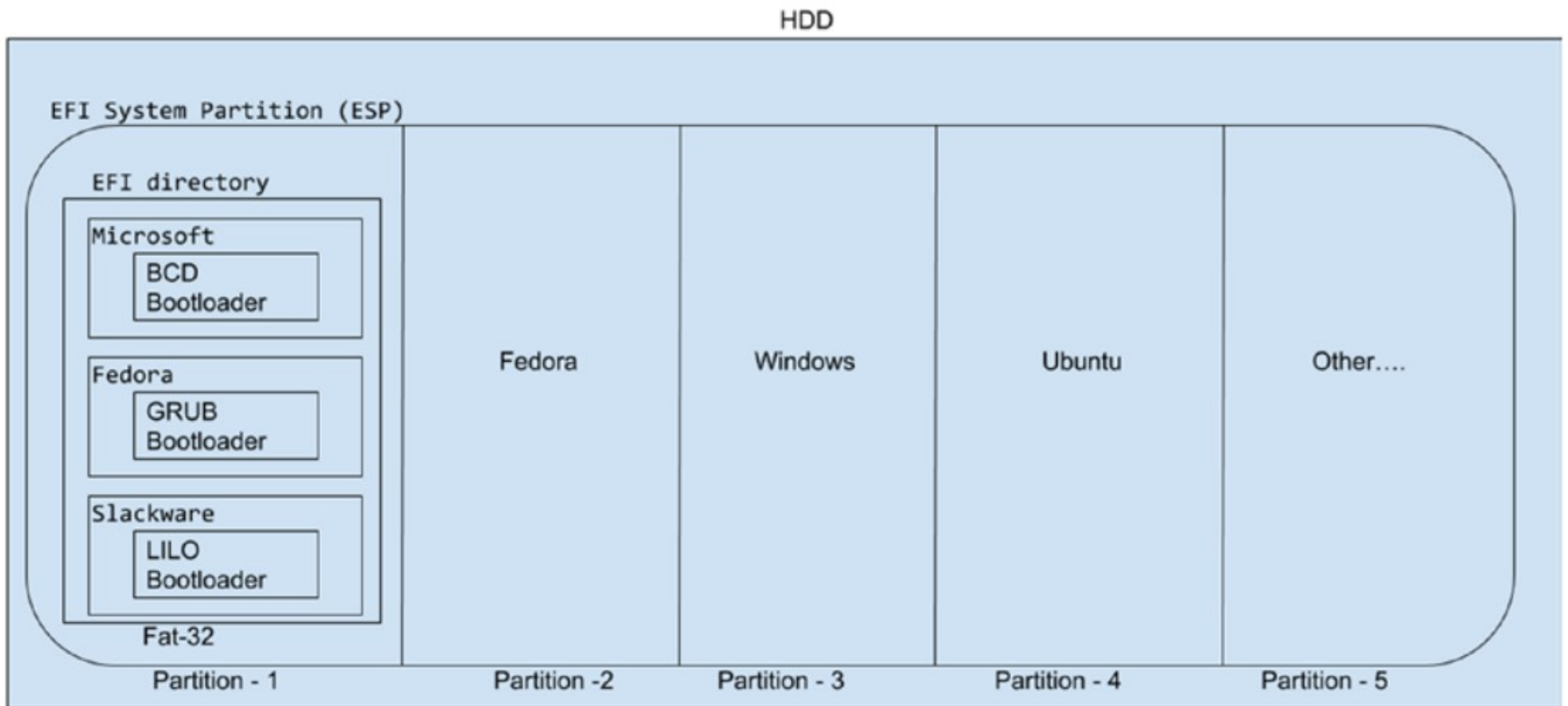


Figure 2-106. *The UEFI structure*

```
root@yogesh-virtual-machine:/home/yogesh# ls -l /boot/efi/EFI/ubuntu/
total 3724
-rwx----- 1 root root    108 Dec  2 17:47 BOOTX64.CSV
drwx----- 2 root root   4096 Dec  2 17:47 fw
-rwx----- 1 root root  75992 Dec  2 17:47 fwupx64.efi
-rwx----- 1 root root    126 Dec  2 17:47 grub.cfg
-rwx----- 1 root root 1116024 Dec  2 17:47 grubx64.efi
-rwx----- 1 root root 1269496 Dec  2 17:47 mmx64.efi
-rwx----- 1 root root 1334816 Dec  2 17:47 shimx64.efi
root@yogesh-virtual-machine:/home/yogesh#
```

Figure 2-111. The EFI directory of Ubuntu

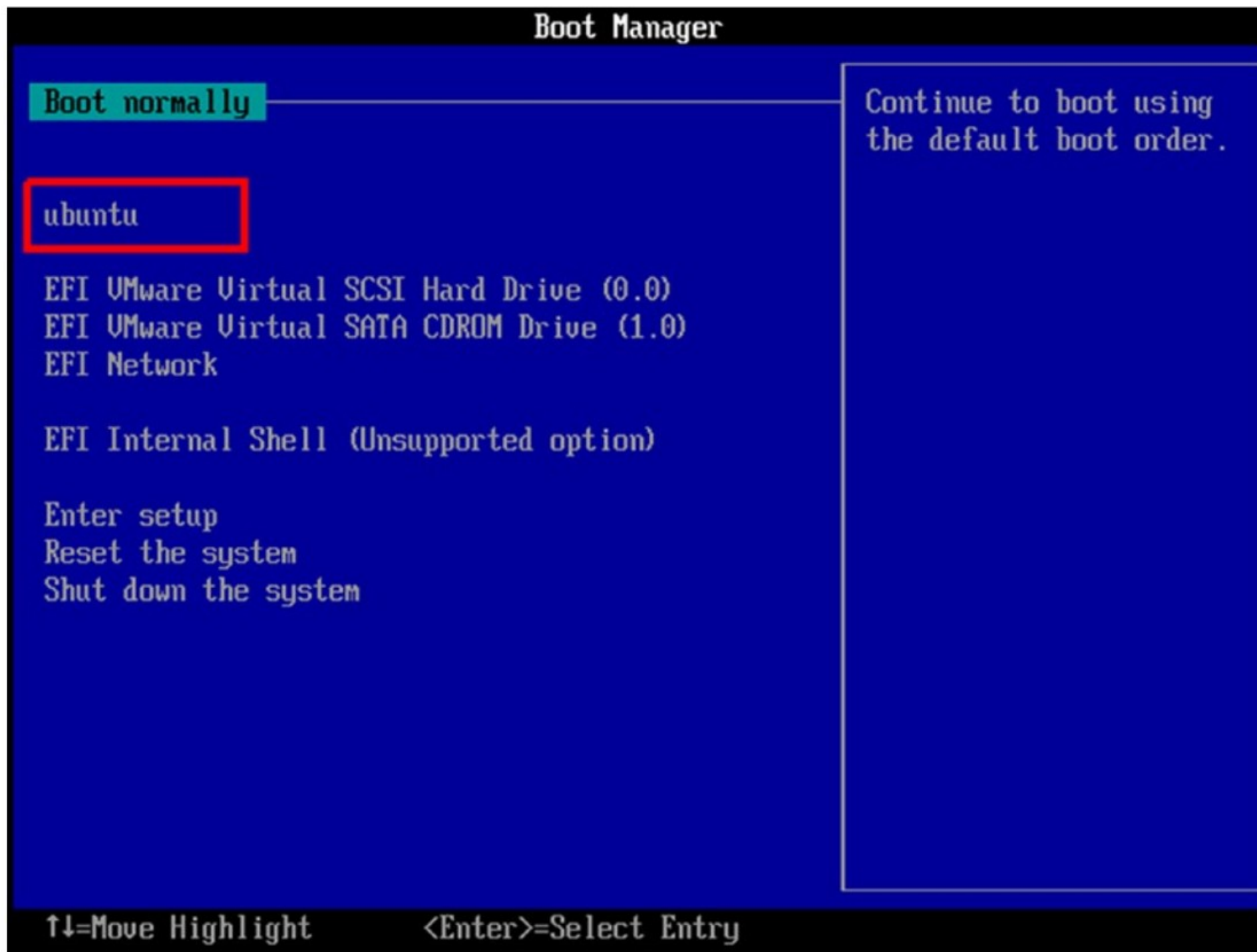


Figure 2-112. The boot priority window of UEFI

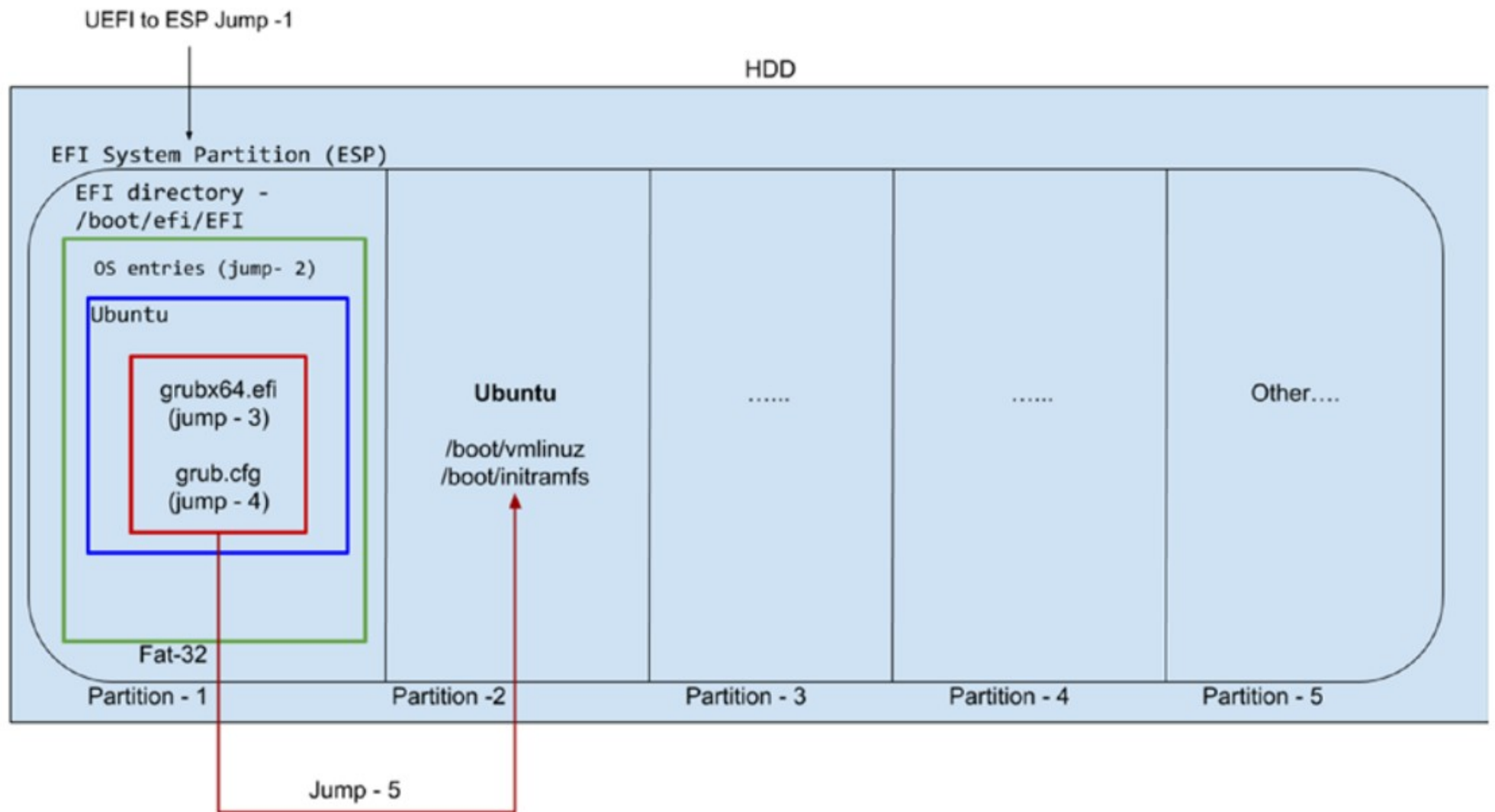


Figure 2-114. *Ubuntu's boot sequence*

Boot Manager

Boot normally

Continue to boot using
the default boot order.

ubuntu

Windows Boot Manager

EFI VMware Virtual SCSI Hard Drive (0.0)

EFI VMware Virtual SATA CDROM Drive (1.0)

EFI Network

EFI Internal Shell (Unsupported option)

Enter setup

Reset the system

Shut down the system

↑↓=Move Highlight

<Enter>=Select Entry

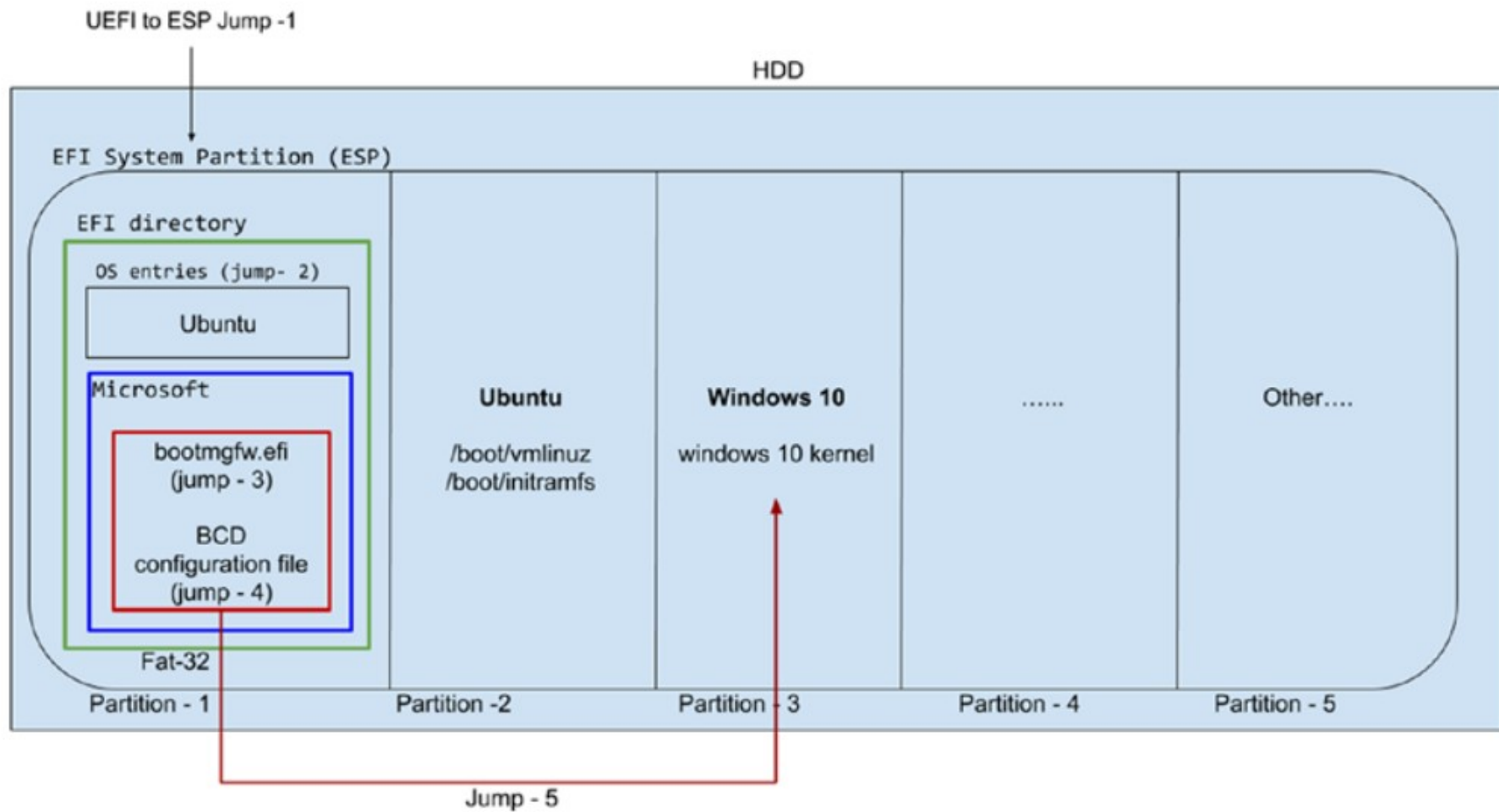


Figure 2-121. *The boot sequence of Windows 10*

