

5.1 프로세스 스케줄링: FCFS

1. FCFS(First-Come First-Served) 스케줄링

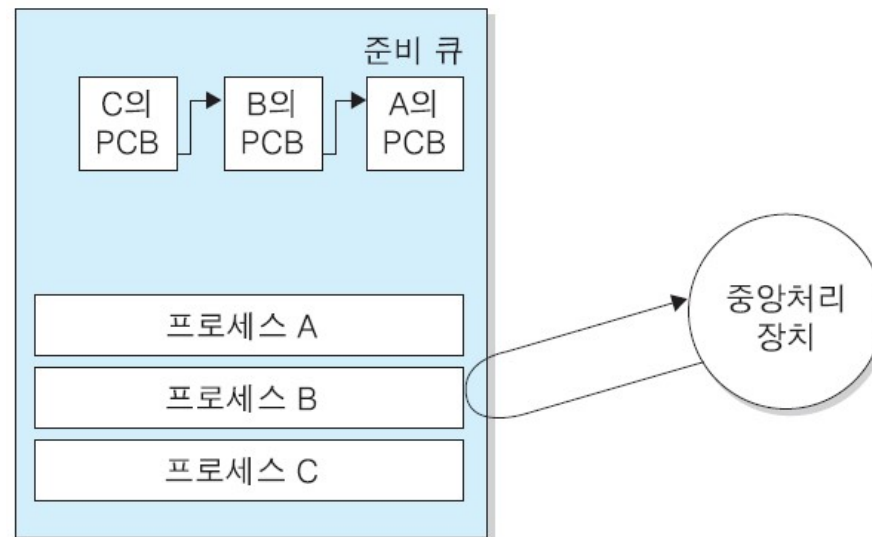
A. 먼저 도착한 프로세스를 먼저 서비스(실행)하는 방법

B. 동작 예:

(1) 프로세스 A는 실행, 프로세스 B와 C는 기다림

(2) 프로세스 A가 종료되면 프로세스 B가 실행된다.

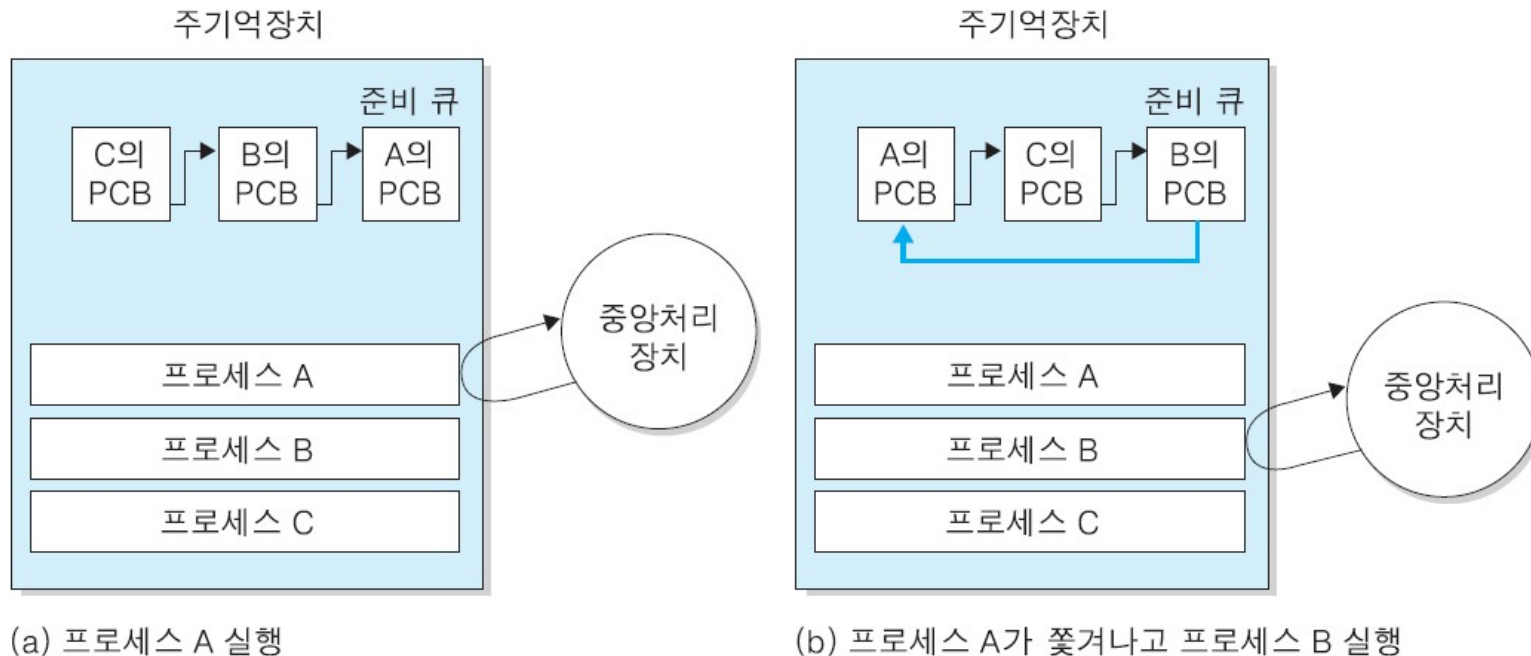
-> 프로세스의 실행 순서에 따라 평균 대기 시간의 차이가 크다.



5.1 프로세스 스케줄링: RR (Round Robin)

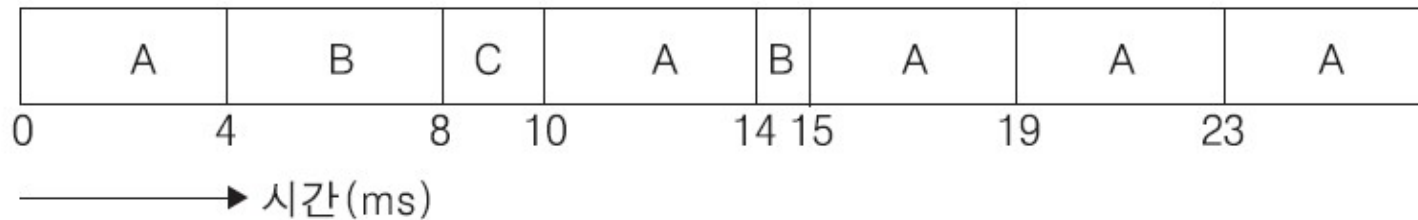
1. 라운드 로빈 스케줄링

- A. 하나의 CPU를 임의의 프로세스가 종료될 때까지 차지하는 것이 아니라 여러 프로세스들이 CPU를 조금씩 돌아가며 할당받아 실행
- B. 프로세스들은 시간 할당량 동안 CPU를 할당 받아 실행되는데, 이 시간 동안 실행을 종료하지 못하면 운영체제에 의해 준비 상태로 쫓겨나고, 준비 큐의 다음 프로세스가 CPU를 할당 받아 실행



5.1 프로세스 스케줄링: RR (Round Robin)

1. 라운드 로빈 스케줄링에 의한 실행과 시간



프로세스	중앙처리장치 시간
프로세스 A	20ms
프로세스 B	5ms
프로세스 C	2ms

2. 평균 대기 시간 (단, 문맥 전환 시간은 고려하지 않았다.)

$$\frac{7+10+8}{3} = 8.3(\text{ms})$$

5.1 프로세스 스케줄링: 우선순위

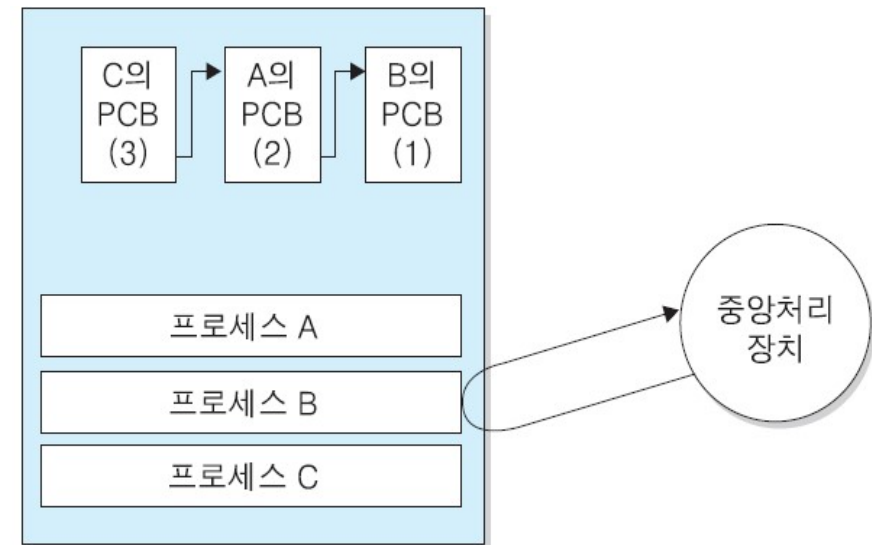
3. 우선순위 스케줄링

A. 가장 높은 우선순위의 프로세스에게 먼저 중앙처리장치를 할당하는 방법, 우선순위가 같은 프로세스들의 경우에는 FCFS 방법을 적용

B. 동작 예

(1) 우선순위를 2, 1, 3로 하는 프로세스 A, B, C가 있다면 운영체제는 우선순위가 가장 높은 프로세스 B에게 중앙처리장치를 배정한다.

(2) 프로세스 B의 실행이 종료되면 다음으로 우선순위가 높은 프로세스 A를 실행.



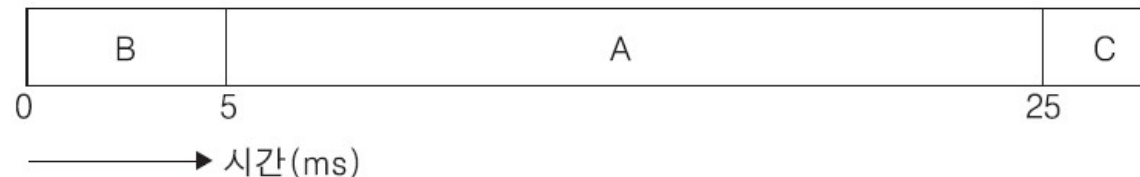
5.1 프로세스 스케줄링: 우선순위

1. 우선순위 스케줄링 시간분석

A. 세 개의 프로세스

프로세스	중앙처리장치 시간	우선순위
프로세스 A	20ms	2
프로세스 B	5ms	1
프로세스 C	2ms	3

B. 우선순위 스케줄링에 의한 실행과 시간



C. 평균 대기 시간 $\frac{0+5+25}{3} = 10(\text{ms})$

D. 문제점? 프로세스의 우선순위가 낮으면 매우 오랜 시간 동안 실행되지 않을 수 있다는 점

5.2 자원에 대한 경쟁의 처리

1. 세마포어 (Semaphore): 제어 플래그
2. 임계구역 (Critical Section, CS):
한 번에 한 프로세스만이 실행시킬 수 있는 명령 그룹
3. 상호배제 (Mutual Exclusive):
한 번에 하나의 프로세스만이 임계구역을 실행하도록 허용하자는 요구사항

예)

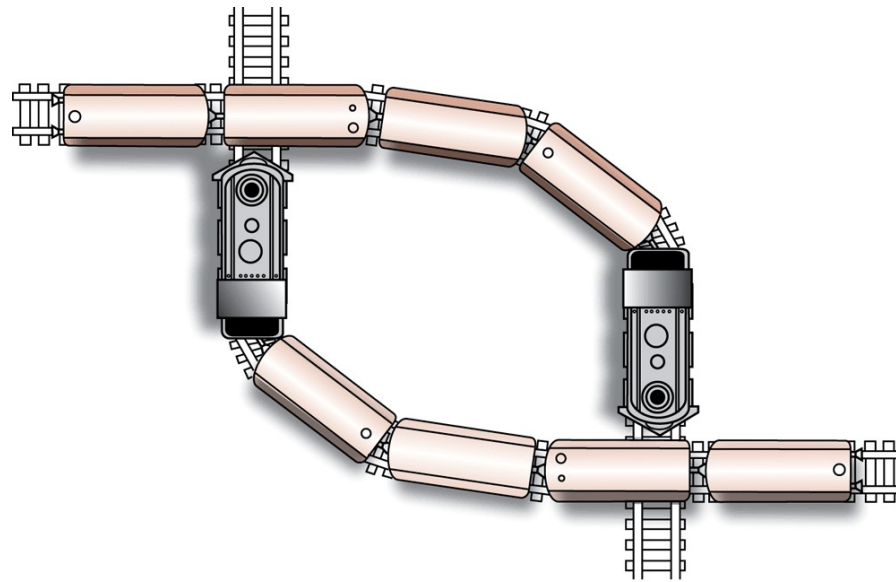
디스크 I/O에서 파일 Writing를 가정

데이터베이스에서 돈을 인출하는 경우를 가정

5.3 교착상태(Deadlock)

1. 프로세스들이 서로의 진행을 가로막는 상태
2. 교착상태 발생 조건
 1. 공유할 수 없는 자원에 대한 경쟁이 존재한다.
 2. 자원에 대한 요청이 단계적으로 이루어진다.
 3. 일단 자원이 할당되면, 강제로 회수할 수 없다.

교착상태 예: 철도 교차점



5.3 교착상태(Deadlock)

1. 교착 상태의 발생 조건

- A. 상호 배제(Mutual exclusion) 조건
- B. 점유와 대기(Hold and wait) 조건
- C. 비선점(Non-preemption) 조건
- D. 환형 대기(Circular wait) 조건

2. 교착 상태의 해결 방법

- A. 교착 상태 예방(Deadlock prevention)
- B. 교착 상태 회피(Deadlock avoidance)
- C. 교착 상태 발견(Deadlock detection)
- D. 교착 상태 회복(Deadlock recovery)

자세한 내용은 운영체제 수업에서 학습

6 운영체제 보안(security)

1. 외부로부터의 공격

A. 문제점

- 취약한 비밀번호
- 스니퍼(Sniffing software)

Sniffing : 코를 킁킁거리다, 냄새를 맡다.

B. 대처 방법

- 감사(Auditing) 소프트웨어

2. 내부로부터의 공격

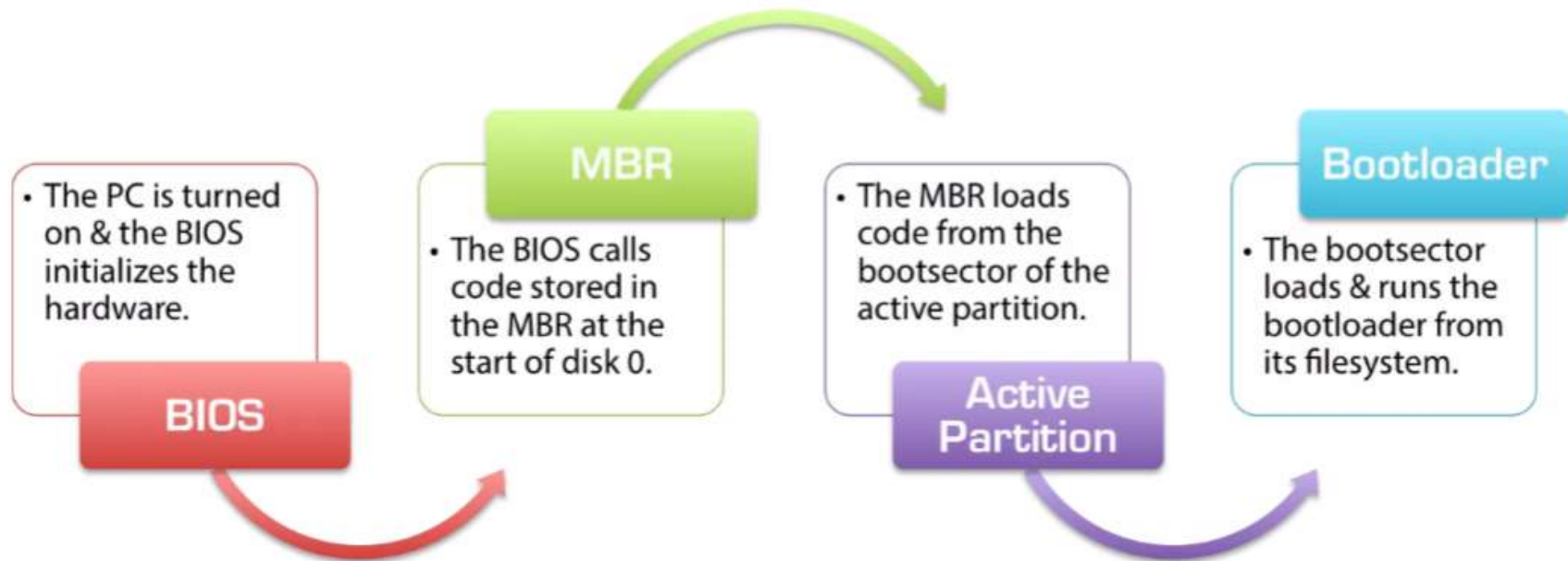
A. 문제점: 악의적 프로세스

B. 대처 방법: 특권 모드와 특권 명령을 통한 프로세스 활동의 통제

7 Understanding Boot Process - 0

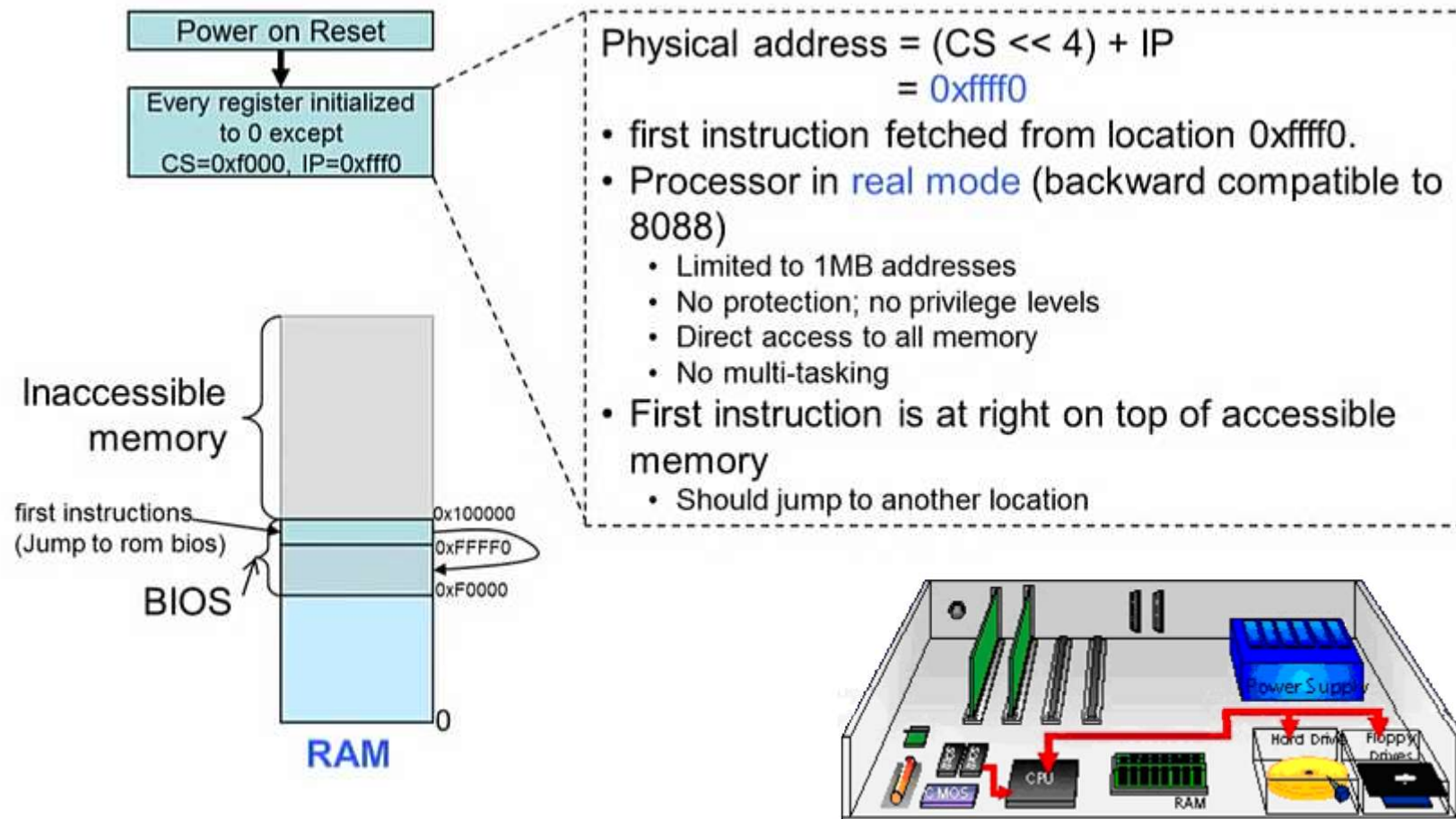
Steps of the Boot Sequence

1. Power Up
2. Power-On Self Test
3. Find a Boot Device
4. Load the Operating System
5. Transfer control to the OS



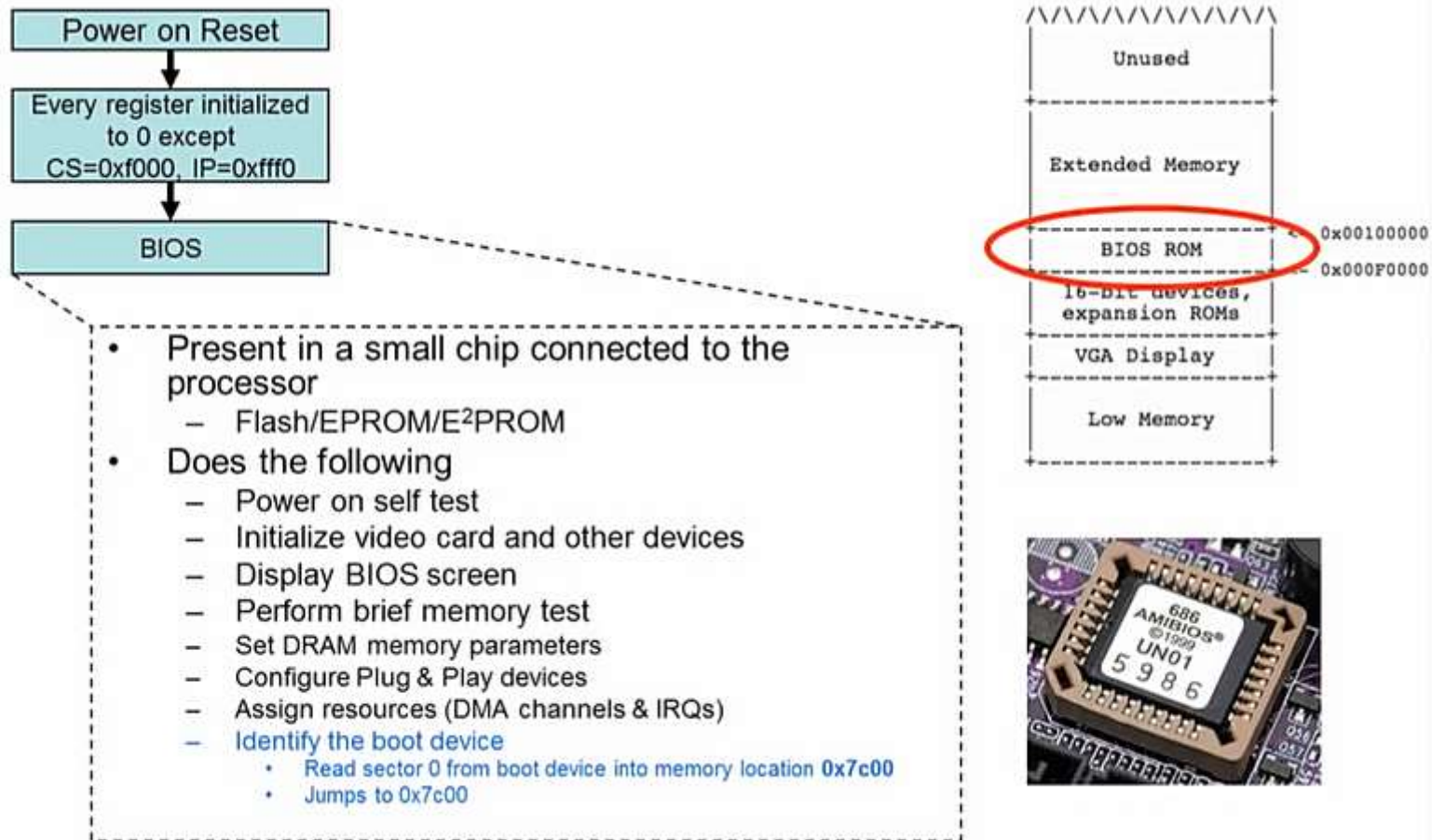
7 Understanding Boot Process - 1

The Boot Sequence 1. Power up (Reset)



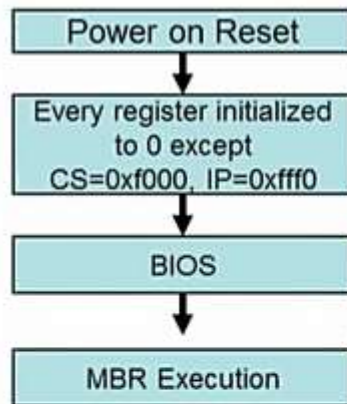
7 Understanding Boot Process - 2

The Boot Sequence 2. BIOS (POST)



7 Understanding Boot Process - 3

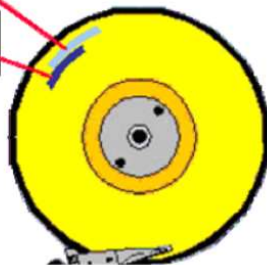
The Boot Sequence 3. Find boot device(MBR, Master Boot Record)



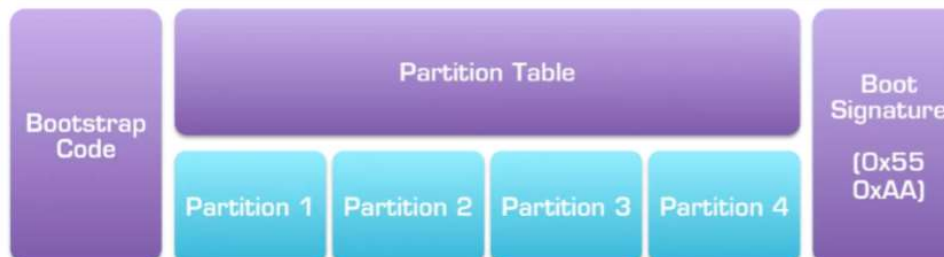
- Sector 0 in the disk called **Master Boot Record (MBR)**
- Contains code that boots the OS or another boot loader
- Copied from disk to RAM (@0x7c00) by BIOS and then begins to execute
- Size 512 bytes
 - 446 bytes bootable code
 - 64 bytes disk partition information (16 bytes per partition)
 - 2 bytes signature
- Typically, MBR code looks through partition table and loads the bootloader (such as Linux or Windows)
- or, it may directly load the OS

Booting the OS from the MBR

- 1] Master Boot Record
Cyl = 0, Head = 0, Sector = 1
- 2] Boot Record

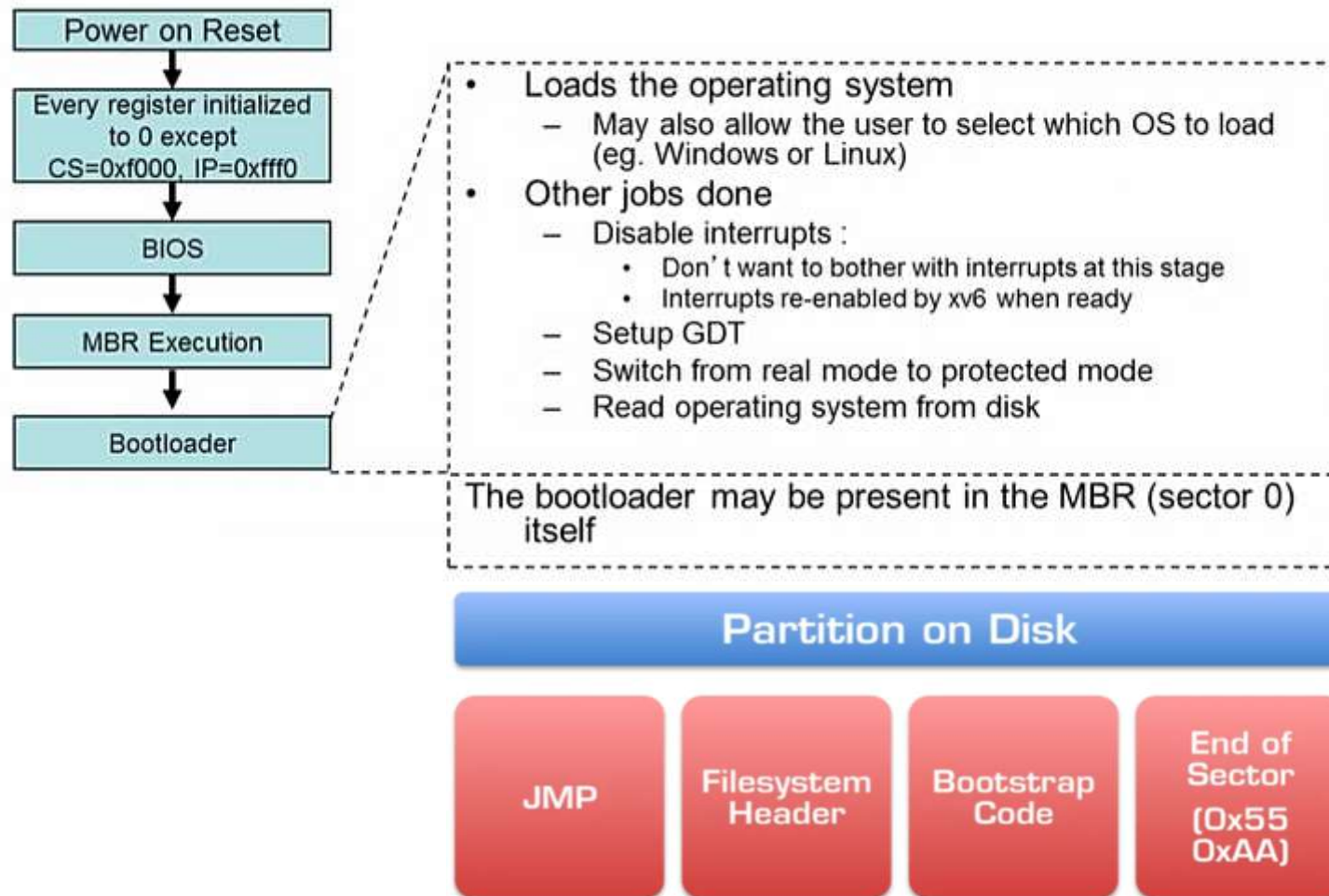


Master Boot Record



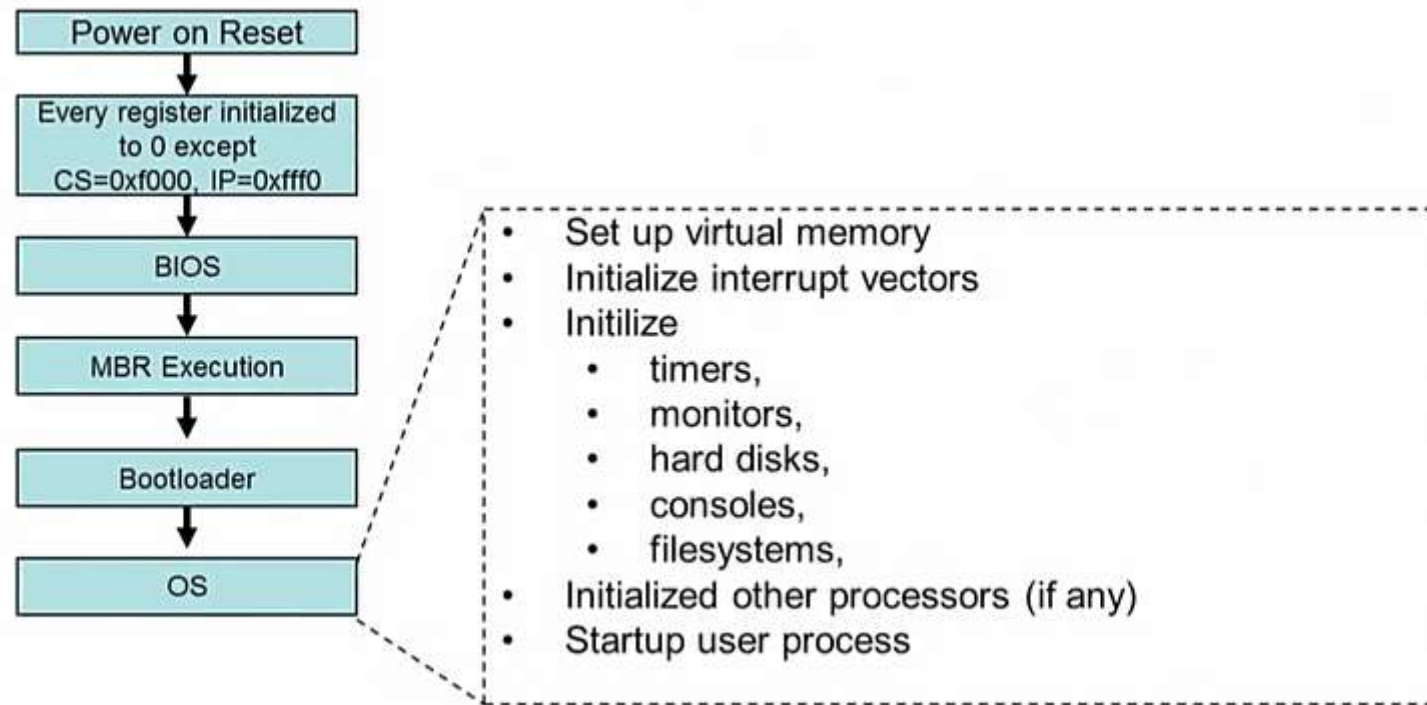
7 Understanding Boot Process - 4

The Boot Sequence 4. Bootloader (Load Operating System)

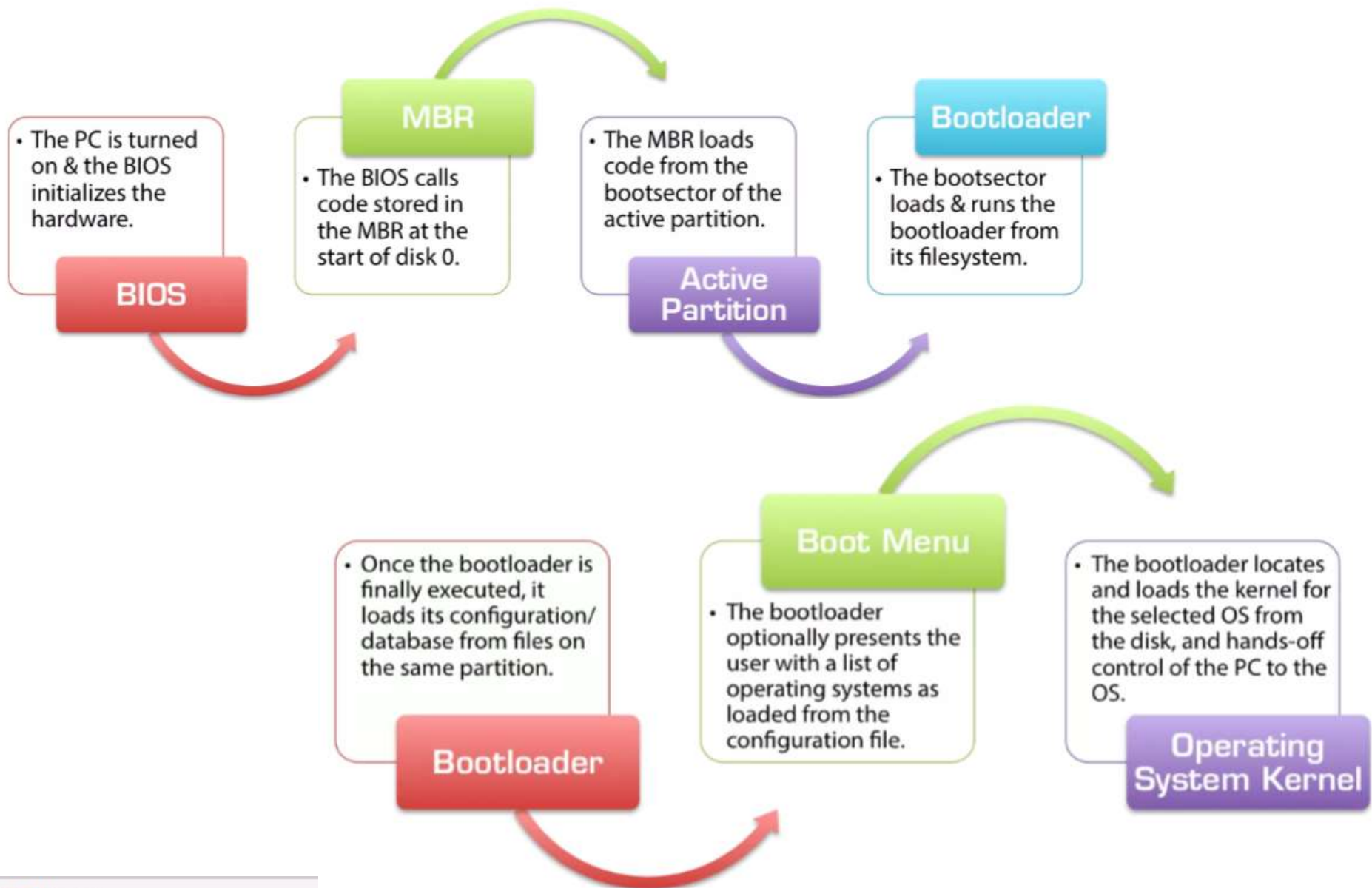


7 Understanding Boot Process - 5

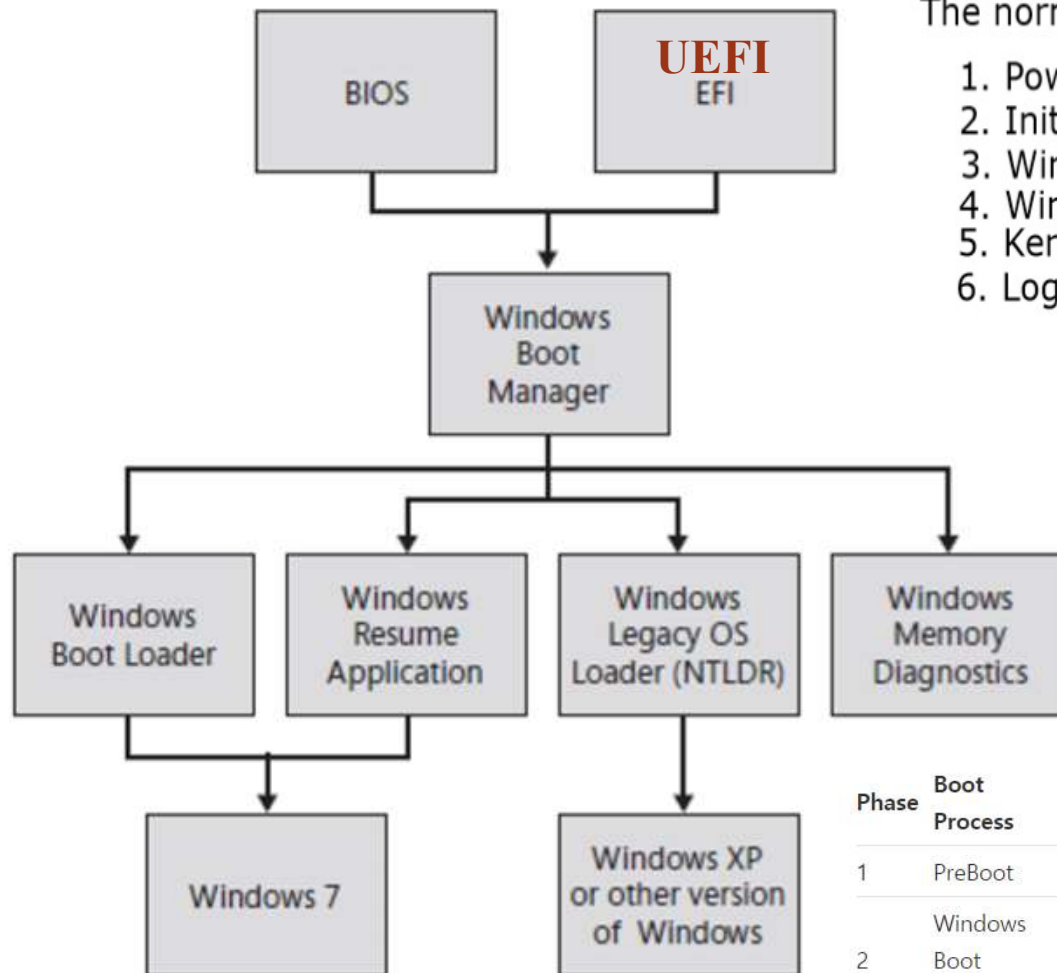
The Boot Sequence 5. OS (Transfer control to the OS)



7 Understanding Boot Process - 6



7 Understanding Boot Process - 7



The normal startup sequence for Windows 7 is:

1. Power-on self test (POST) phase.
2. Initial startup phase.
3. Windows Boot Manager phase.
4. Windows Boot Loader phase.
5. Kernel loading phase.
6. Logon phase.

Windows 10 boot process and the list of stages it goes through:

1. PreBoot
2. Windows Boot Manager
3. Windows OS Loader.
4. Windows NT OS Kernel.

During every process, a program is loaded.

Depending on whether it uses Legacy BIOS or UEFI, the file paths and files change.

Phase	Boot Process	BIOS	UEFI
1	PreBoot	MBR/PBR (Bootstrap Code)	UEFI Firmware
2	Windows Boot Manager	%SystemDrive%\bootmgr	UEFI\Microsoft\Boot\bootmgfw.efi
3	Windows OS Loader	%SystemRoot%\system32\winload.exe	%SystemRoot%\system32\winload.efi
4	Windows NT OS Kernel	%SystemRoot%\system32\ntoskrnl.exe	

7 Understanding Boot Process - 8

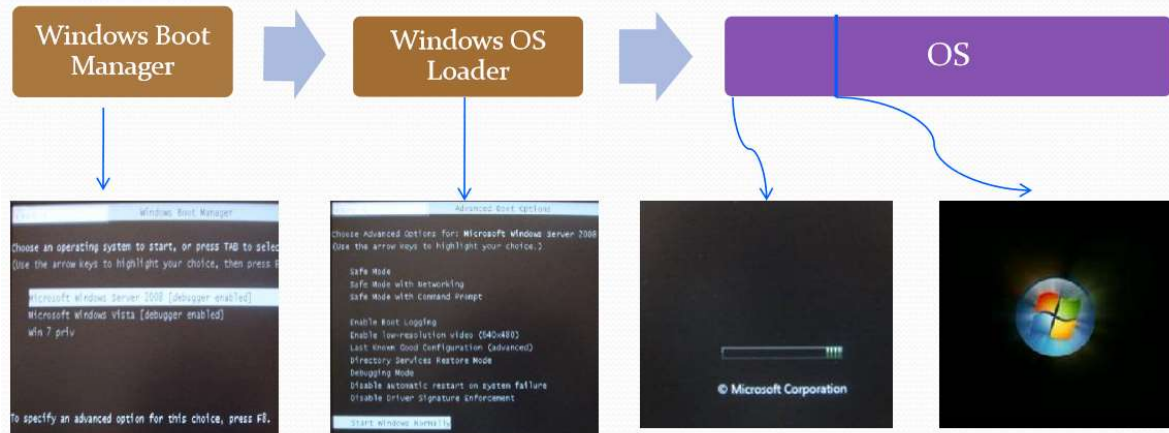
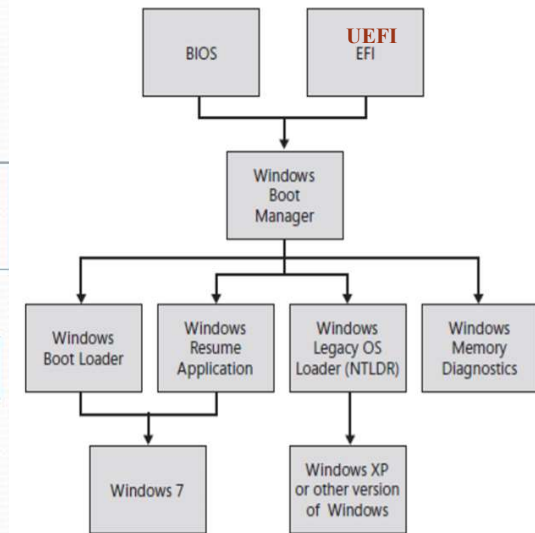


Performs CPU and Chipset initialization, load drivers etc.

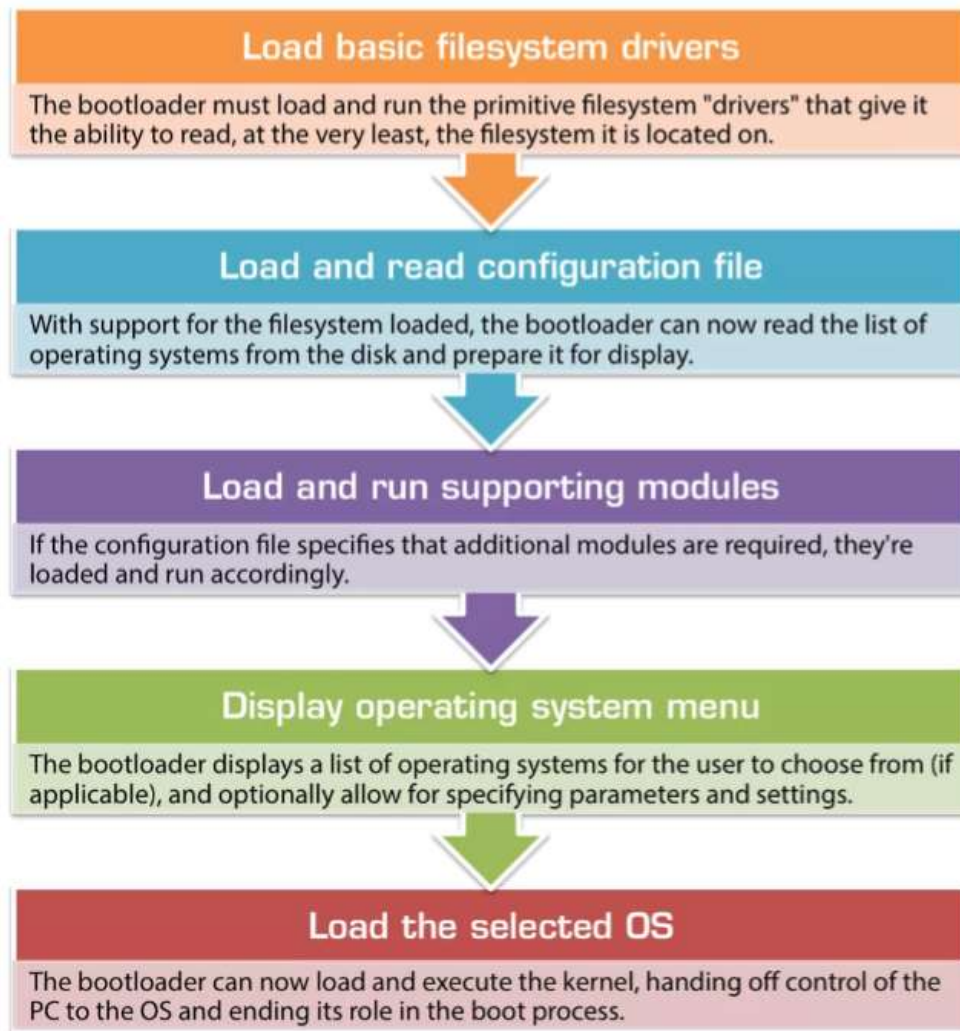
Loads UEFI device drivers (based on NVRAM DriverXxx variable), loads boot application

Loads Windows OS loader selected by user

Loads the Windows OS, calls ExitBootServices()



7 Understanding Boot Process - 9



NTLDR

- NTLDR is the default bootloader for Windows NT, 2000, and XP.
- BOOT.INI on the active partition contains the list of operating systems and their locations.
- NTDETECT.COM is a helper program that runs to detect hardware and identify devices.

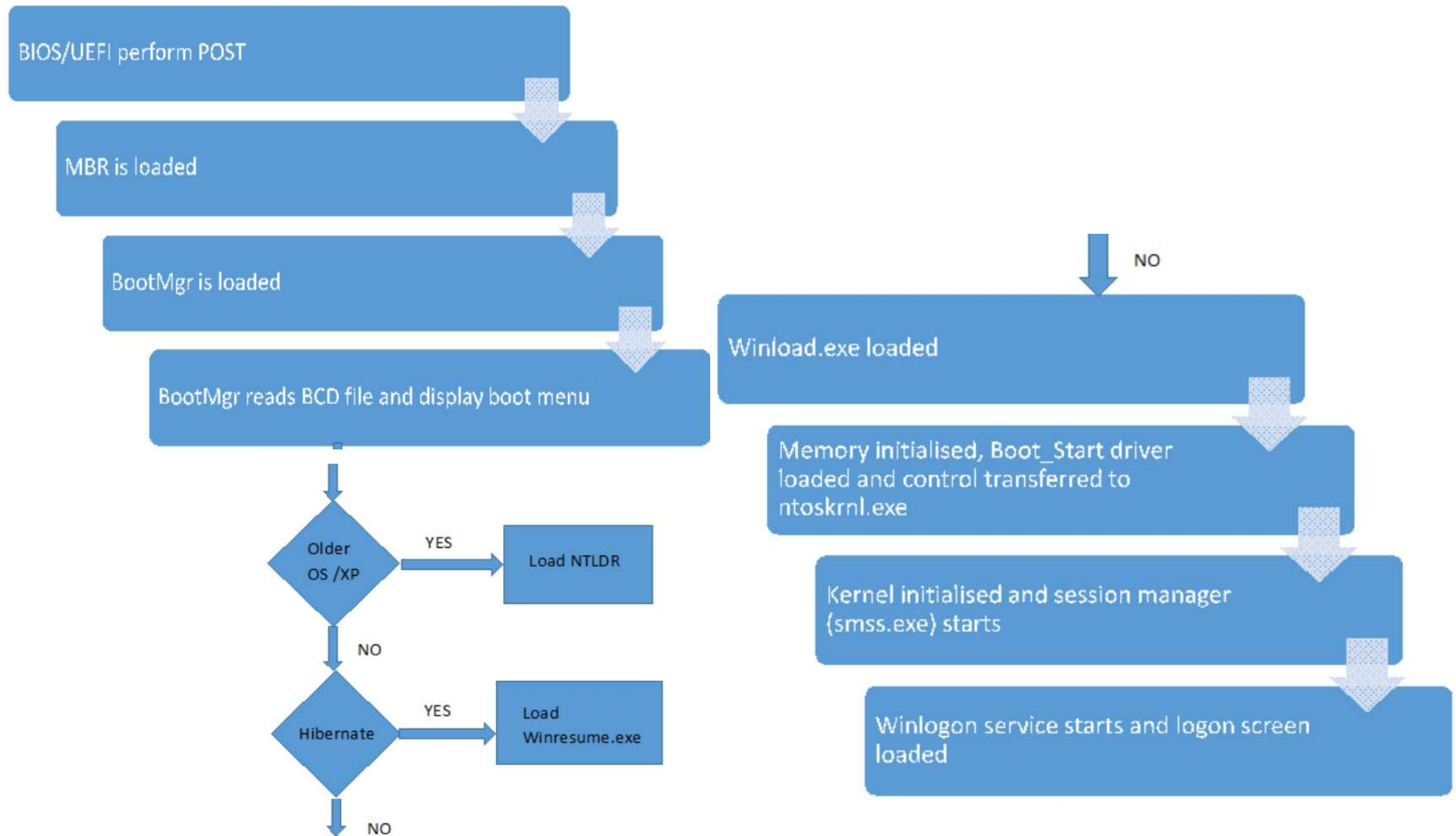
BOOTMGR

- BOOTMGR is the new Windows and is used on Windows Vista, 7, 8, and 10.
- The list of operating systems is now read from the BCD file in the BOOT directory on the active partition.
- BOOTMGR is self-contained, and does not need any helper programs or routines.

GRUB(2)

- GRUB is the most-popular bootloader for Linux, though it can boot numerous other OSes as well.
- Its boot settings are stored in a file usually called grub.cfg (GRUB2) or menu.lst (GRUB).
- GRUB is a modular bootloader, that can load additional modules from disk.

7 Understanding Boot Process - 10



7 Understanding Boot Process - 11



BIOS vs UEFI

What is BIOS?

The **Basic Input Output System** is the older standard and dates back to old IBM-compatible computers. The BIOS is a special software called firmware that is stored in a special chip soldered on the motherboard called ROM (usually EEPROM these days). When you press the power button, the BIOS is the first software that is run on the machine.

7 Understanding Boot Process - 12

What the BIOS can and can't do

BIOS can be updated to support newer hardware/features and can be programmed to perform specific tasks such as:

- * Turning on/off USB ports, Serial ports or IDE/SATA ports;
- * Over/Underclocking CPUs/RAMs frequencies;
- * Regulate motherboard fan controllers;

Although BIOS can perform these task well, they still operate in the 16-bit real mode. The most prominent limitation can be observed when using 2TB+ disks. Most BIOS can only boot from an MBR-partitioned disks, but MBR itself supports up to 2TB partitions meaning it won't recognize the disk past that. Well *there's GPT that solves the problem of disks bigger than 2TB*, most BIOS can't boot from GPT.

7 Understanding Boot Process - 13

UEFI the BIOS successor

The **Unified Extensible Firmware Interface** aims to resolve what BIOS could not. UEFI itself is the second version (2.*), the former being EFI (1.*), *BIOS and UEFI do the same thing, but they are pretty different in how and what they do.*

A **UEFI can** (in addition to what a BIOS can):

- * Boot from disks larger than 2TB using GPT.
- * Provide the user with a graphical user interface.
- * Provide support for mouse devices (BIOS can rarely do this).
- * Boot securely using a chain-of-trust.
- * Network boot (although most BIOS can do that, that's not a given).
- * Provide a modular interface which is independent from the CPU architecture.
- * Provide a modular interface for applications and devices based on EFI drivers. (commonly called EBCs, EFI Byte-Code).