

제 1 장 운영체제

- 목 차 -

1.개요

2.운영체제 필요성

3.운영체제 진입점

4.리눅스 커널

학습목표

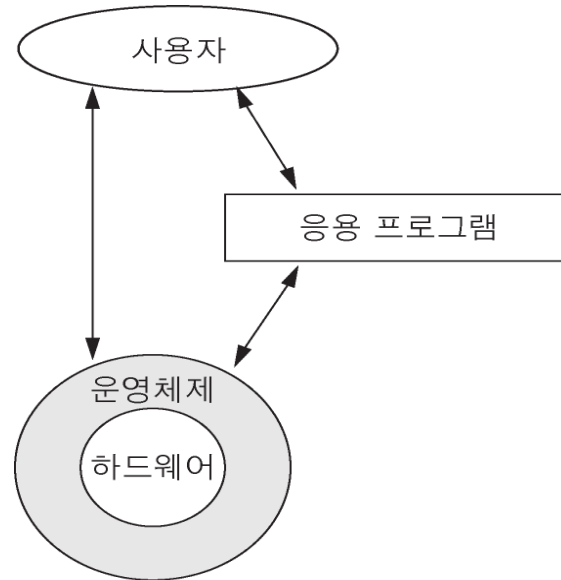
- ✓ 운영체제 정의
- ✓ 컴퓨터 시스템의 동작 원리
- ✓ 부팅 과정
- ✓ 다중 프로그래밍(multi-programming)
- ✓ 운영체제 진입점(entry point)
 - ① 시스템 호출
 - ② 인터럽트
 - ③ 예외현상
- ✓ 리눅스 커널 구조

1.1 개요

운영체제란 무엇인가?

컴퓨터 시스템을 사용해 본 사람은 누구나 운영체제를 접해본 경험이 있다.

그러나 운영체제는 [그림1.1]과 같이 하드웨어와 사용자 혹은 응용 프로그램 사이에 위치하고 있으며 사용자들은 주로 응용 프로그램을 통하여 컴퓨터 시스템을 사용하기 때문에 운영체제의 존재 자체를 인지하지 못할 수 있다.



[그림 1.1] 운영체제 위치

운영체제란?

일반적으로,

응용 프로그램의 실행을 제어하고,
입출력 장치와 사용자 사이의 인터페이스 역할을 하는 프로그램.

사용자 관점으로,

컴퓨터 시스템을 보다 편리하게 사용할 수 있는 환경을 제공해 주는
프로그램

관리자 관점으로,

컴퓨터 시스템의 자원들을 효율적으로 이용할 수 있도록 제어해 주는
프로그램

위키 백과사전에서,

시스템 하드웨어를 관리할 뿐 아니라 응용 소프트웨어를 실행하기 위하여
하드웨어 추상화 플랫폼과 공통 시스템 서비스를 제공하는 시스템 소프트웨어이다.

여러가지 관점에서 다양하게 정의될 수 있다 !!!

- ❖ 우리는 운영체제(Operating System)를 시스템(system) 관점에서 정의해 봅시다!
 - ✓ 시스템(system)이란,
사전적 의미로, “어떤 주어진 목적을 달성하기 위하여
여러 개의 구성요소들이 유기적으로 결합된 집합체”
 - ✓ 그렇다면, 운영체제(Operating System)도 하나의
시스템(system)이기 때문에 운영체제가 추구하는 목적과
구성요소가 있을 것이다.
- ❖ 운영체제의 목적과 구성요소에는 무엇이라고 말할 수 있을까?

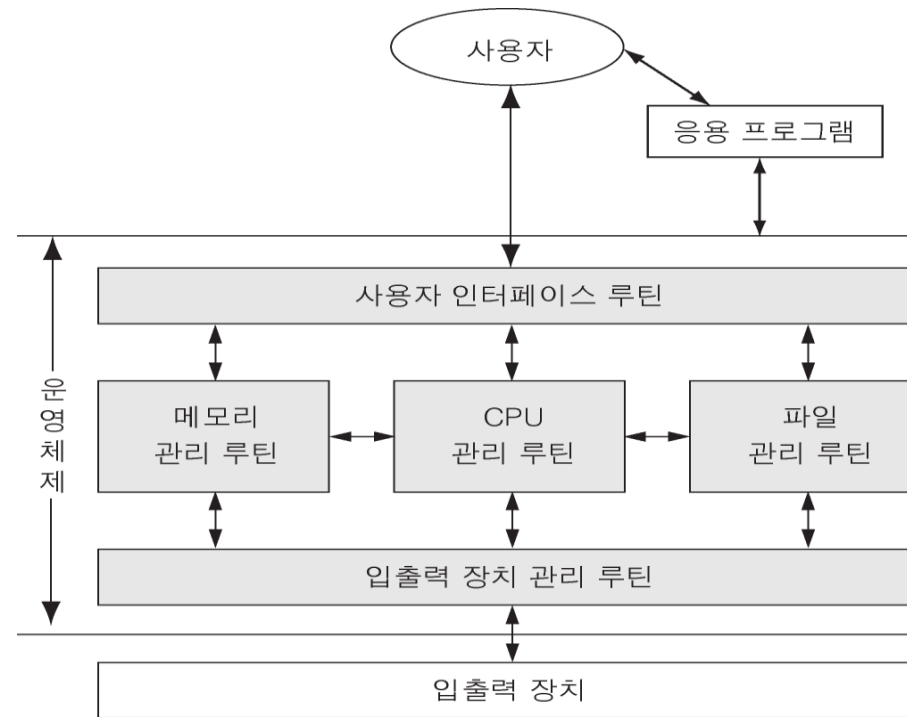
1.1.1 운영체제 목적

기존 운영체제들의 공통적인 주된 목적은 컴퓨터 시스템 자원의 효율적 관리와 사용자의 편의성 제공이라 할 수 있다.

- ① 운영체제가 관리해야 할 컴퓨터 시스템 **자원**에는 CPU, 메모리, 디스크, 프린터, 키보드 등과 같은 **하드웨어**는 물론 파일, 서비스 루틴과 같은 **소프트웨어**가 있다. 이러한 자원들은 컴퓨터 시스템마다 보유하고 있는 종류는 다르지만, 그 보유량은 모두 **제한적**이다. 따라서 컴퓨터 시스템을 보다 경제적으로 이용하기 위해서는 자원의 **효율적인 관리** 기능이 요구된다.
- ② 사용자의 **편의성**이란 사용자들로 하여금 프로그램 작성과 **실행**을 보다 쉽게 할 수 있는 환경을 의미한다.
 - 이 두 가지의 목표는 서로 **이율 배반적**임을 유의하자.
왜냐하면 사용자의 편의성을 위해서는 보다 많은 자원을 가지고 있어야 할 것이고, 보다 많은 자원을 유지하다 보면 자원의 이용률은 떨어지게 될 것이기 때문이다.

1.1.2 운영체제 구성요소

- 운영체제의 구성요소는 컴퓨터 시스템의 용도에 따라 다르다.
- 일반적으로 CPU 관리 루틴, 메모리 관리 루틴, 파일 관리 루틴, 입출력 장치 관리 루틴, 그리고 사용자 인터페이스를 담당하는 루틴들로 구성되어 있다.
- 이 루틴들은 독립적인 것이 아니고 상호 호출 관계를 유지하면서 유기적으로 결합되어 있다



[그림1.2] 운영체제 구성요소

결론적으로(시스템 관점에서 볼 때)

운영체제란

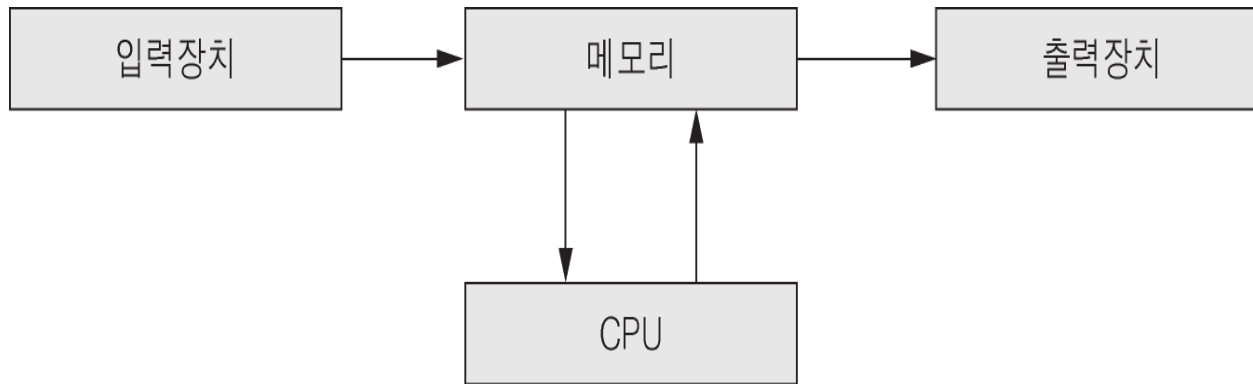
- ✓ 컴퓨터 시스템이 보유하고 있는 **자원들을 효율적으로 관리**하고
사용자에게 편의성을 제공하기 위하여
- ✓ CPU 관리, 메모리 관리, 파일 관리, 입출력 장치 관리, 그리고
사용자 인터페이스를 담당하는 **루틴들이**
- ✓ 유기적으로 결합된 **프로그램**”이라고 정의할 수 있을 것이다.

1.2 운영체제 필요성

- 초기 컴퓨터 시스템에는 운영체제가 없었다.
- 프로그램 실행에 필요한 일련의 작업들을 사용자가 직접 했다.
- ✓ 컴퓨터 사용이 매우 불편하고,
- ✓ 컴퓨터 자원의 이용률이 저조하게 됨.
- 컴퓨터 자원의 이용률을 향상시키기 위하여(**efficient**)
- 사용자의 편의성을 제공하기 위하여(**convenient**)

1.2.1 컴퓨터 시스템 동작원리

컴퓨터 시스템의 하드웨어는 그림 1.3과 같이 CPU, 메모리, 그리고 다양한 종류의 입출력 장치들로 구성되어 있다.



[그림1.3] 컴퓨터 하드웨어 구성

1.2.1 컴퓨터 시스템 동작원리

- ① CPU(중앙처리장치)는 무엇을 처리하는 장치일까?
 - ✓ 명령어(instruction)를 처리한다.
- ② CPU는 처리해야 할 명령어가 있는 메모리 주소를 어떻게 아는가?
 - ✓ 레지스터(register)를 이용.
 - ✓ IP(Instruction Pointer) 혹은 PC(Program Counter) reg.

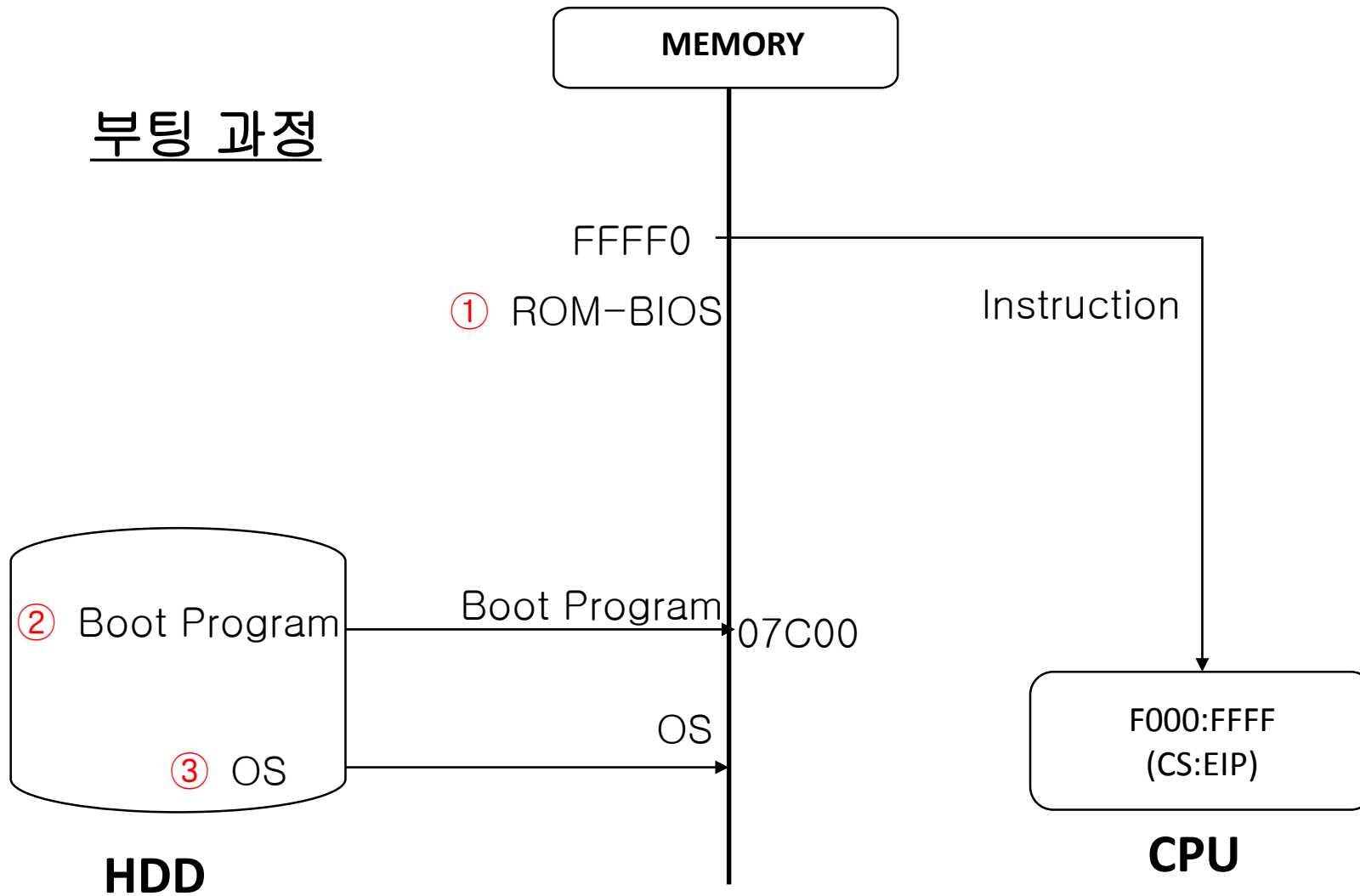
※ CPU는 전원이 가해지는 순간부터 커질 때까지,
IP 레지스터가 가리키는 곳의 메모리에 저장된 명령어
처리하기를 반복한다 !!!

1.2.1 컴퓨터 시스템 동작원리

❖ IBM PC에 적용해 보면,

- ✓ CPU의 PC 레지스터 이름은? CS : EIP
- ✓ PC 레지스터의 초기 값은? F000:FFF0 (=> FFFF0)
- ✓ 프로그램 실행 순서(부팅과정):
 - ① ROM-BIOS(Basic Input Output System)
 - ② Boot Program (운영체제를 메모리로 적재)
 - ③ OS(Windows10)
 - ④ 바탕화면(무한 루프를 돌면서 사용자 입력 대기)

부팅 과정



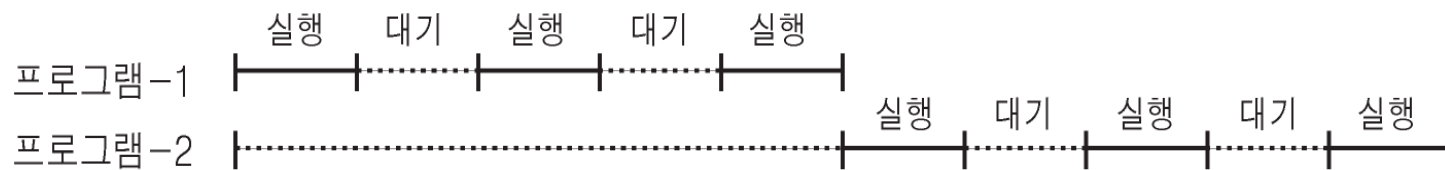
※ 부팅 후, 사용자 프로그램이 실행되는 과정
(마우스로 프로그램 실행을 위해 아이콘 더블 클릭)

- ① 마우스 인터럽트를 처리(인터럽트 처리 루틴)
- ② 아이콘에 해당된 파일 찾기(파일 시스템)
- ③ 메모리 빈 공간 할당(메모리 관리)
- ④ 프로그램 적재(입출력 장치관리)
- ⑤ CS:EIP 값 조정(CPU 관리)

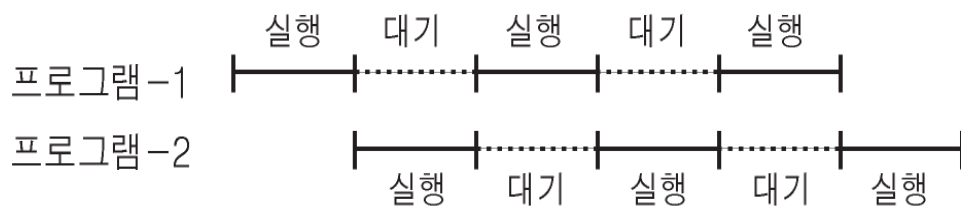
결국, 운영체제에 의해 실행되고 있음을 알 수 있다!!!

1.2.2 다중 프로그래밍 (multi-programming)

- ❖ CPU의 이용률을 극대화하기 위하여 고안된 기법으로써 운영체제 발전과정의 커다란 전환점이 되었다.
- ❖ 하나의 프로그램이 실행되는 과정에서 입출력 요구가 발생하게 되면 입출력 처리가 완료될 때까지 CPU는 유휴 상태에서 기다리게 된다. 이 시간을 “CPU의 유휴 시간(idle time)”이라고 한다.
- ❖ CPU의 유휴 시간을 활용하기 위하여 여러 개의 프로그램 들을 메모리에 적재시켜 놓고 하나의 프로그램이 실행되는 과정에서 입출력 요구가 발생하면 운영체제로 하여금 다른 프로그램을 실행시키도록 CPU 사용을 전환함으로써 CPU의 이용률을 향상하는 것이다.
- ❖ 컴퓨터 시스템에서 단위 시간당 처리되는 프로그램의 수 즉 시스템의 처리율(throughput)이 향상될 수 있다.



(a) 다중 프로그래밍 기법을 이용하지 않을 경우



(b) 다중 프로그래밍 기법을 이용 할 경우

(a) CPU 이용률 : $6/10(60\%)$, 처리율 : $2/10(\text{프로그램/초})$

(b) CPU 이용률 : $6/6(100\%)$, 처리율 : $2/6(\text{프로그램/초})$

[그림 1.4] 다중 프로그래밍 기법의 개념

1.2.3 운영체제 기능

- ❖ 운영체제는 기본적으로 다음과 같은 기능들을 제공한다.
- ① CPU 관리 : 프로그램 실행 및 종료, 프로그램들 간의 CPU 공유 등의 기능.
- ② 메모리 관리 : 메모리의 사용 영역과 빈 영역을 유지 및 관리하여 새로운 영역의 요구와 사용 영역의 반납 등의 기능.
- ③ 파일 관리 : 사용자가 입력한 프로그램이나 데이터를 파일 형태로 보조 기억장치에 저장하고 사용자의 요청에 따라 해당 파일을 보조기억 장치로부터 읽어 들이는 등의 기능.
- ④ 입출력 장치 관리 : 실행 중인 프로그램에서 필요로 하는 입출력 장치들을 사용할 수 있도록 관리하는 기능.
- ⑤ 네트워킹 : 서로 다른 컴퓨터 시스템 간의 통신을 위한 네트워킹 기능.
- ⑥ 보호 : 효율적인 자원관리 위해 공유하는 자원에 대한 보호기능.
- ⑦ 오류 처리와 복구 : 컴퓨터 시스템의 오류를 탐지하며, 이를복구하거나 사용자에게 경고하는 기능.

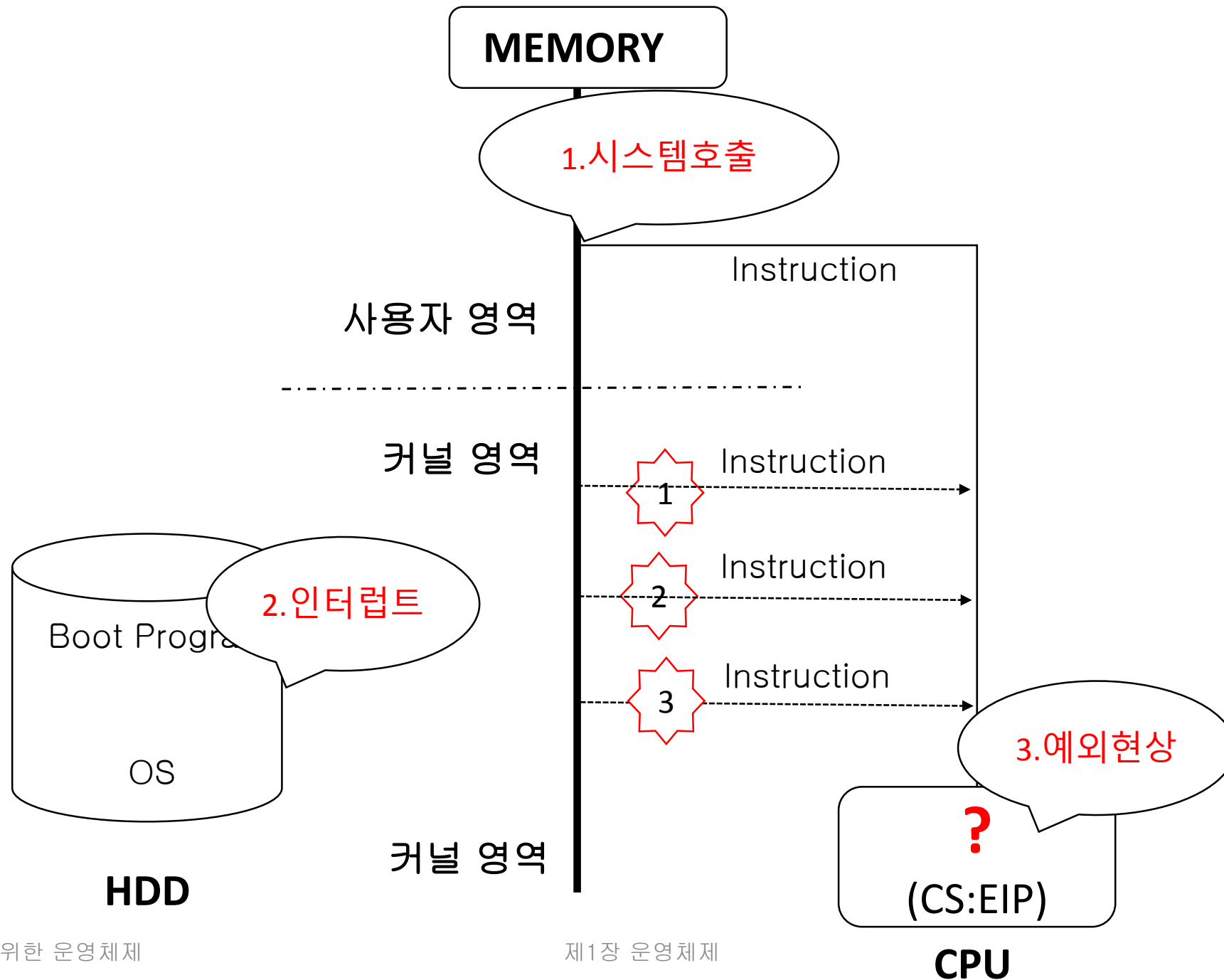
1.3 운영체제 진입점(Entry Point)

❖ 이중모드 : 운영체제 보호를 위하여

- ① 사용자 모드 : 사용자 프로그램 실행
- ② 커널 모드 : 커널 프로그램 실행

❖ 운영체제 접근을 위한 모드 변환

- ① 시스템 호출(System Call)
- ② 인터럽트(Interrupt)
- ③ 예외현상(Exception)



1.3.1 시스템 호출(System Call)

- 사용자 프로그램과 운영체제 사이의 인터페이스 채널

예) `open()`, `read()`, `write()`, ...

cf) `fopen()`, `scanf()`, `printf()`, ...

1.3.2 인터럽트(Interrupt)

- CPU와 입출력 장치 사이의 인터페이스 채널
 - ① 마우스, 키보드, 하드 디스크 등의 입출력 장치에서 CPU로 보내는 신호.
 - ② CPU는 어떤 장치에서 발생한 인터럽트 신호인지를 식별하여 해당 서비스 루틴을 실행한다.

1.3.3 예외현상(Exception)

- CPU 내부에서 발생하는 시스템 오류

예) Divide by Zero, Page Fault ...

1.4 리눅스 커널

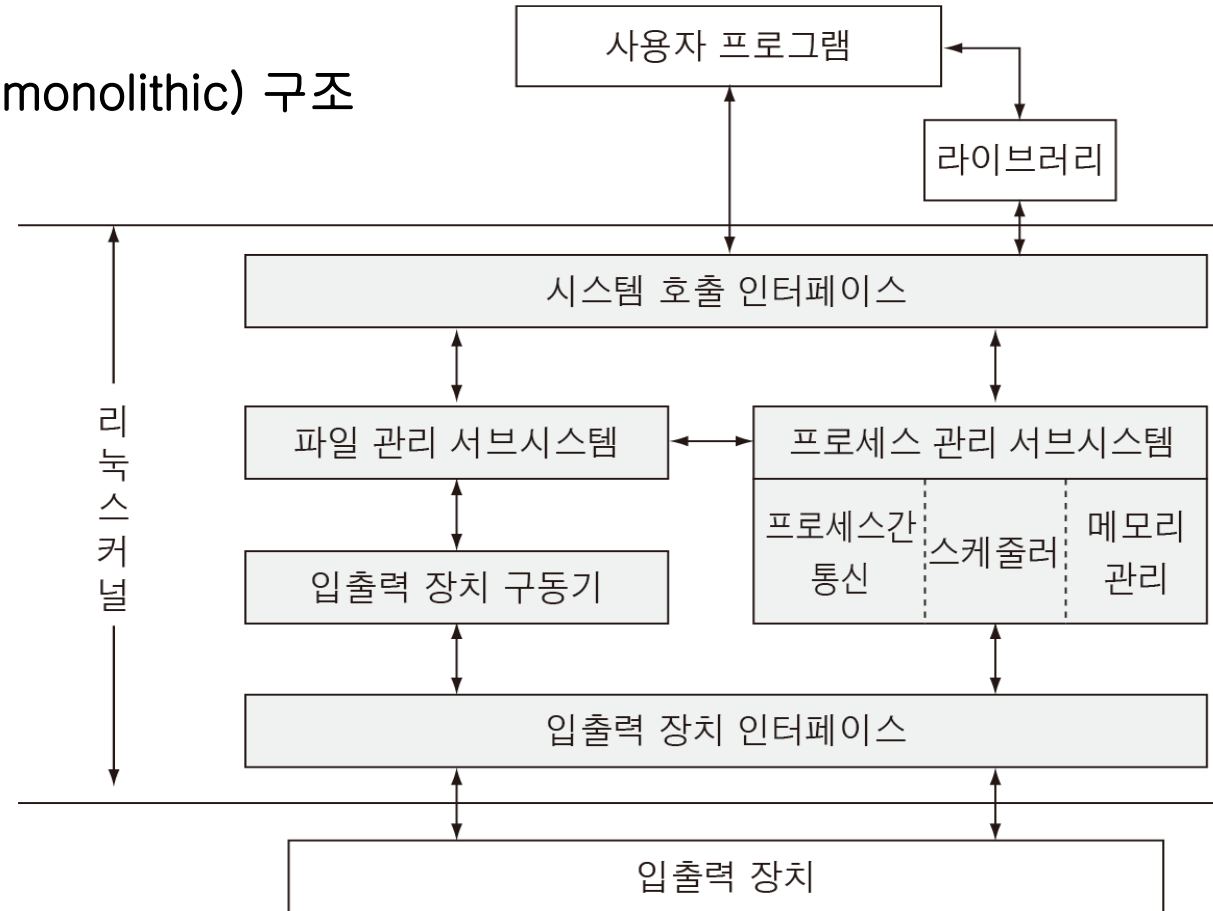
❖ 커널(kernel) : 운영체제의 핵심 기능들.

- ✓ 메모리 상주
- ✓ 리눅스(배포판) = 커널 + a

❖ 리눅스 커널 구조 :

- ✓ 파일관리 서브 시스템
- ✓ 프로세스 관리 서브 시스템
- ✓ 시스템 호출 인터페이스
- ✓ 입출력 장치 인터페이스

모노리딕(monolithic) 구조



[그림 1.5] 리눅스 커널 구조