

운영체제(OS, Operating System)와 컴퓨터

☰ 태그	CS
🕒 생성 일시	@2024년 1월 6일 오후 1:40

▼ 목차

[운영체제의 역할과 구조](#)

[운영체제 역할](#)

[운영체제 구조](#)

[시스템콜](#)

[modebit](#)

[컴퓨터의 요소](#)

[CPU \(Central Processing Unit\)](#)

[CPU 구성요소](#)

[CPU의 연산 처리](#)

[인터럽트](#)

[DMA 컨트롤러](#)

[메모리](#)

[타이머](#)

[디바이스 컨트롤러](#)

운영체제의 역할과 구조

운영체제는 사용자가 컴퓨터를 쉽게 다루게 해주는 인터페이스입니다. 한정된 메모리나 시스템 자원을 효율적으로 분배해주는 역할을 합니다. 참고로 운영체제와 유사하지만 소프트웨어를 추가로 설치할 수 없는 것을 펌웨어(firmware)라고 합니다.

💡 운영체제는 하드웨어와 소프트웨어(유저 프로그램)을 관리하는 일꾼!

운영체제 역할

- 1. CPU 스케줄링과 프로세스 관리
- 2. 메모리 관리
- 3. 디스크 파일 관리
- 4. I/O 디바이스 관리

운영체제 구조



- GUI, 시스템콜, 커널, 드라이버 부분이 운영체제를 지칭합니다
- GUI가 없고 CUI만 있는 리눅스 서버도 있습니다

▼ GUI (Graphical User Interface)

사용자가 전자장치와 상호 작용할 수 있도록 하는 사용자 인터페이스의 한 형태.

단순한 명령어 창이 아닌 아이콘을 마우스로 클릭하는 단순한 동작으로 컴퓨터와 상호 작용할 수 있도록 해준다

▼ CUI (Character User Interface)

그래픽이 아닌 명령어로 처리하는 인터페이스

▼ 커널

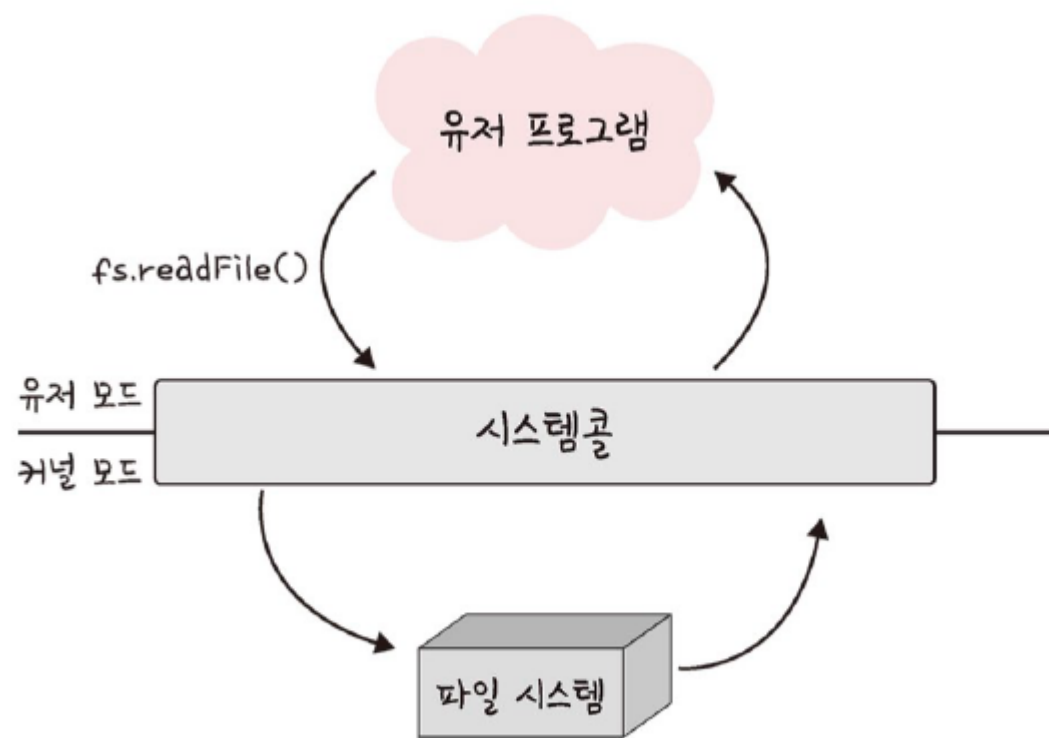
운영체제의 핵심 부분이자 시스템콜 인터페이스를 제공하며 보안, 메모리, 프로세스, 파일 시스템, I/O 요청 관리 등 운영체제의 중추적인 역할을 한다

▼ 드라이버

하드웨어를 제어하기 위한 소프트웨어

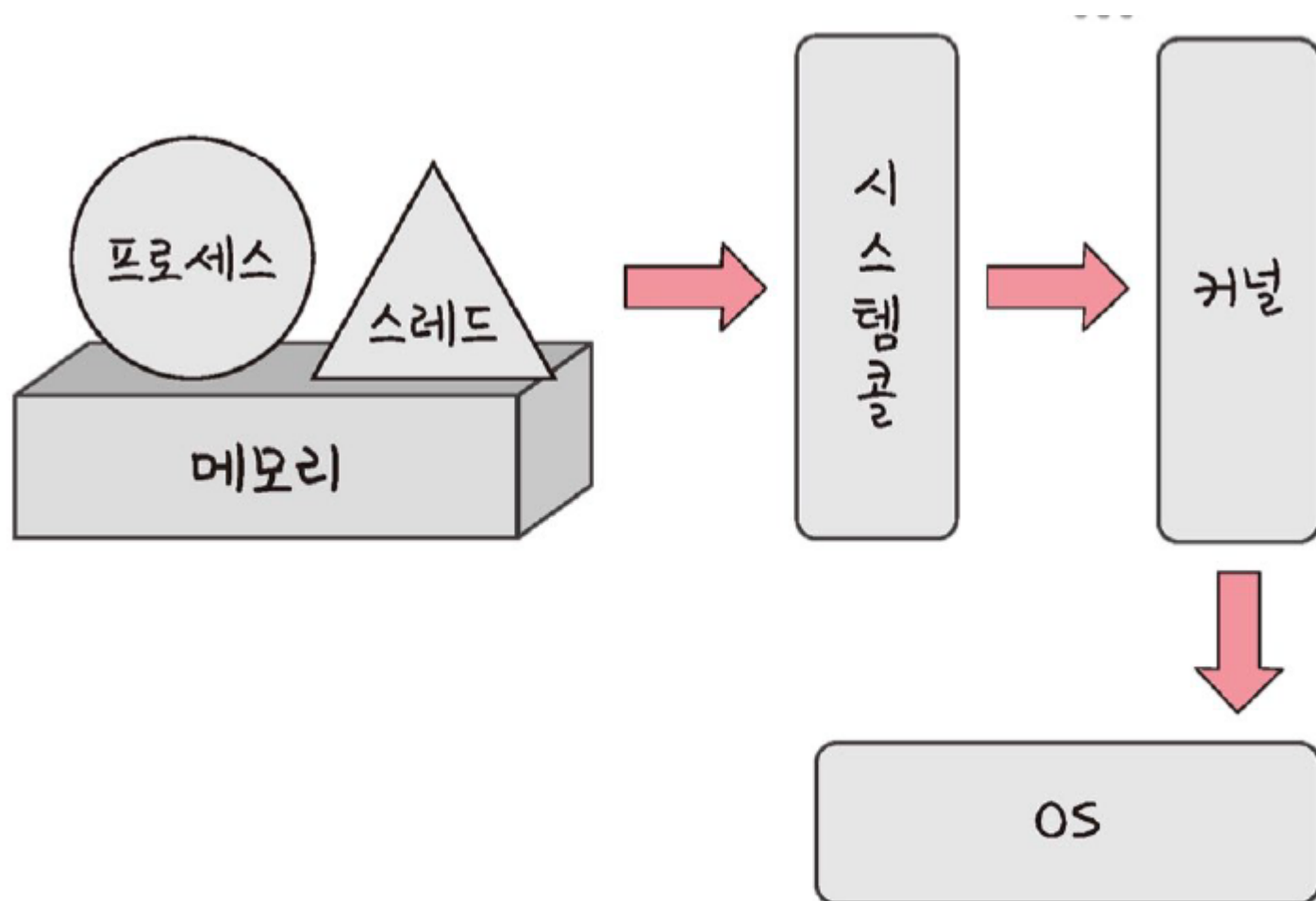
시스템콜

- 운영체제가 커널에 접근하기 위한 인터페이스
- 유저 프로그램이 I/O 요청으로 트랩(trap)을 발동하면 올바른 I/O 요청인지 확인한 후 유저 모드가 시스템콜을 통해 커널모드로 변환되어 실행됩니다.



▲ 그림 3-2 시스템콜

- 이 과정을 통해 컴퓨터 자원에 대한 직접 접근을 차단할 수 있고 프로그램을 다른 프로그램으로부터 보호할 수 있습니다.



▲ 그림 3-3 시스템콜과 커널, 운영체제

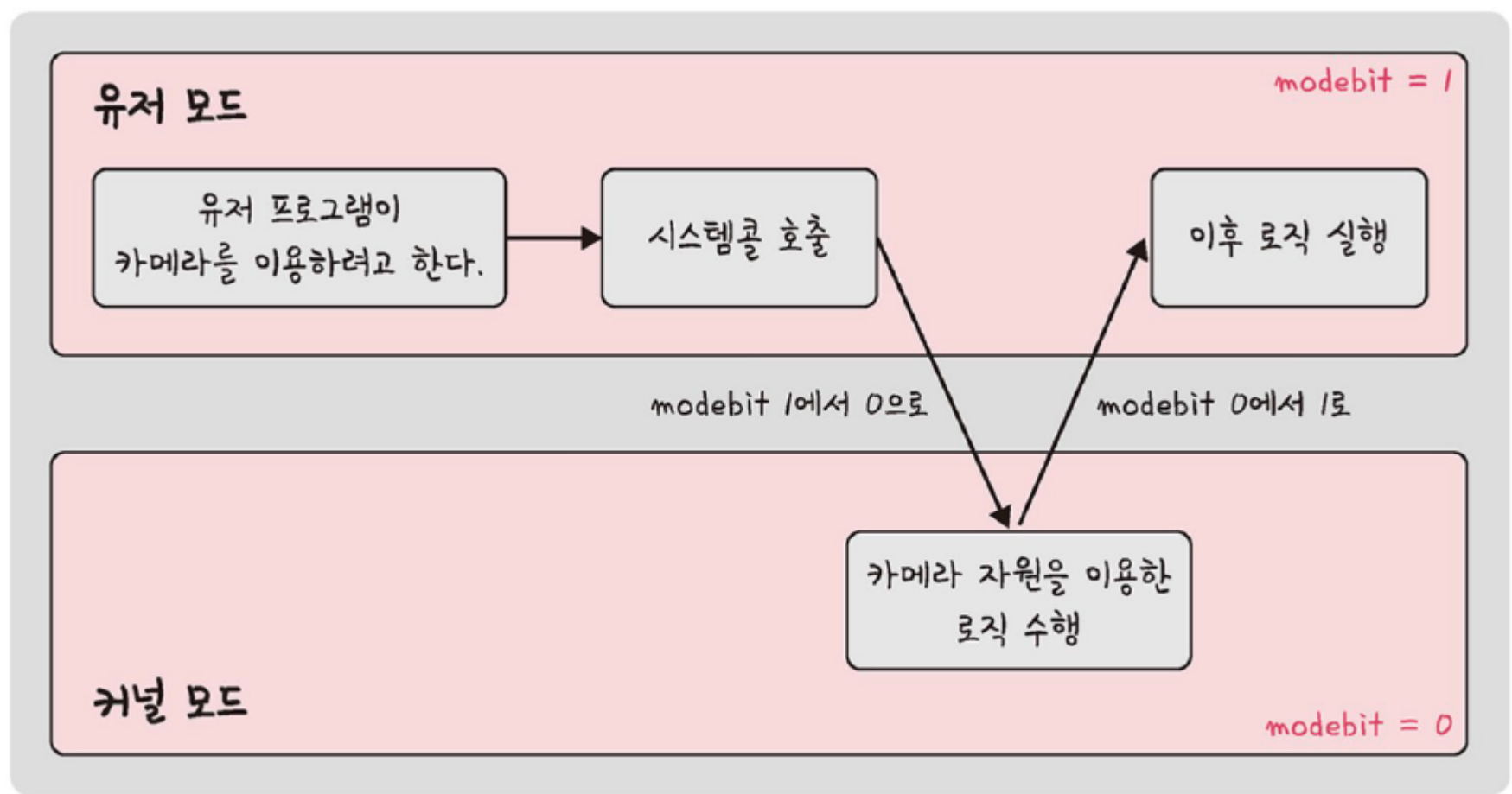
- 위 그림처럼 프로세스나 스레드에서 운영체제로 어떠한 요청을 할 때 시스템콜이라는 인터페이스에 전달됩니다.
- 이 시스템콜은 하나의 추상화 계층입니다. 그렇기 때문에 이를 통해 네트워크 통신이나 데이터베이스와 같은 낮은 단계의 영역 처리에 대한 부분을 많이 신경쓰지 않고 프로그램을 구현할 수 있는 장점이 있습니다.

modebit

modebit은 1또는 0의 값을 가지는 플래그 변수로, 시스템콜이 작동될 때 modebit을 참고해서 유저 모드와 커널 모드를 구분합니다.

유저모드를 기반으로 프로그램이 실행된다면 사용자가 의도하지 않았는데 공격자가 프로그램을 조종하기 쉽습니다.

물론 커널 모드를 사용한다고 해서 100% 막을 순 없지만 운영체제를 통해 작동하게 해야 막기 쉽습니다. 이를 위한 장치가 modebit입니다. modebit의 0은 커널 모드, 1은 유저 모드라고 설정됩니다.



▲ 그림 3-4 modebit의 역할

▼ 유저 모드

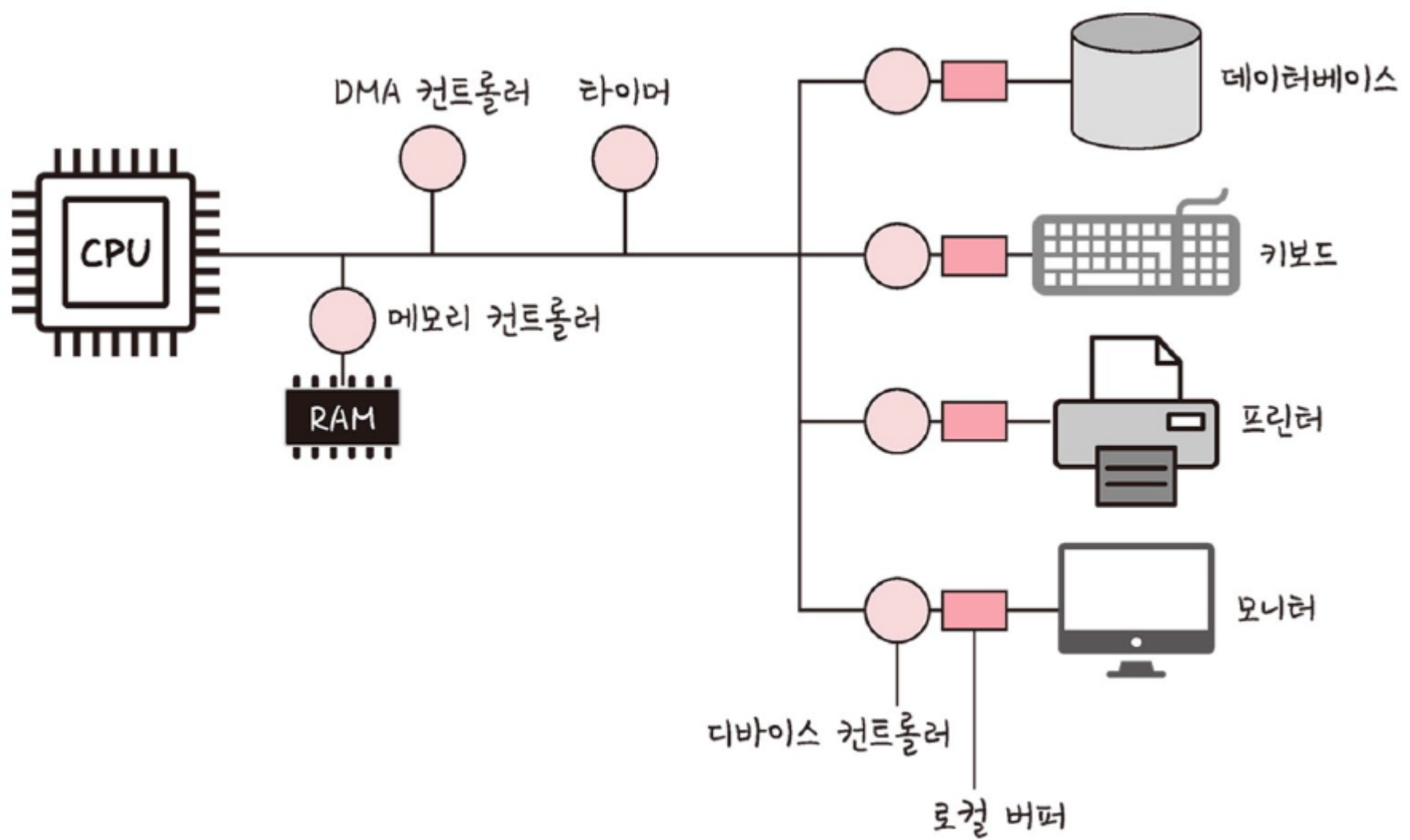
유저가 접근할 수 있는 영역을 제한적으로 두며 컴퓨터 자원에 함부로 침범하지 못하는 모드

▼ 커널 모드

모든 컴퓨터 자원에 접근할 수 있는 모드

컴퓨터의 요소

컴퓨터는 CPU, DMA, 컨트롤러, 메모리, 타이머, 디바이스 컨트롤러 등으로 이루어져 있습니다.

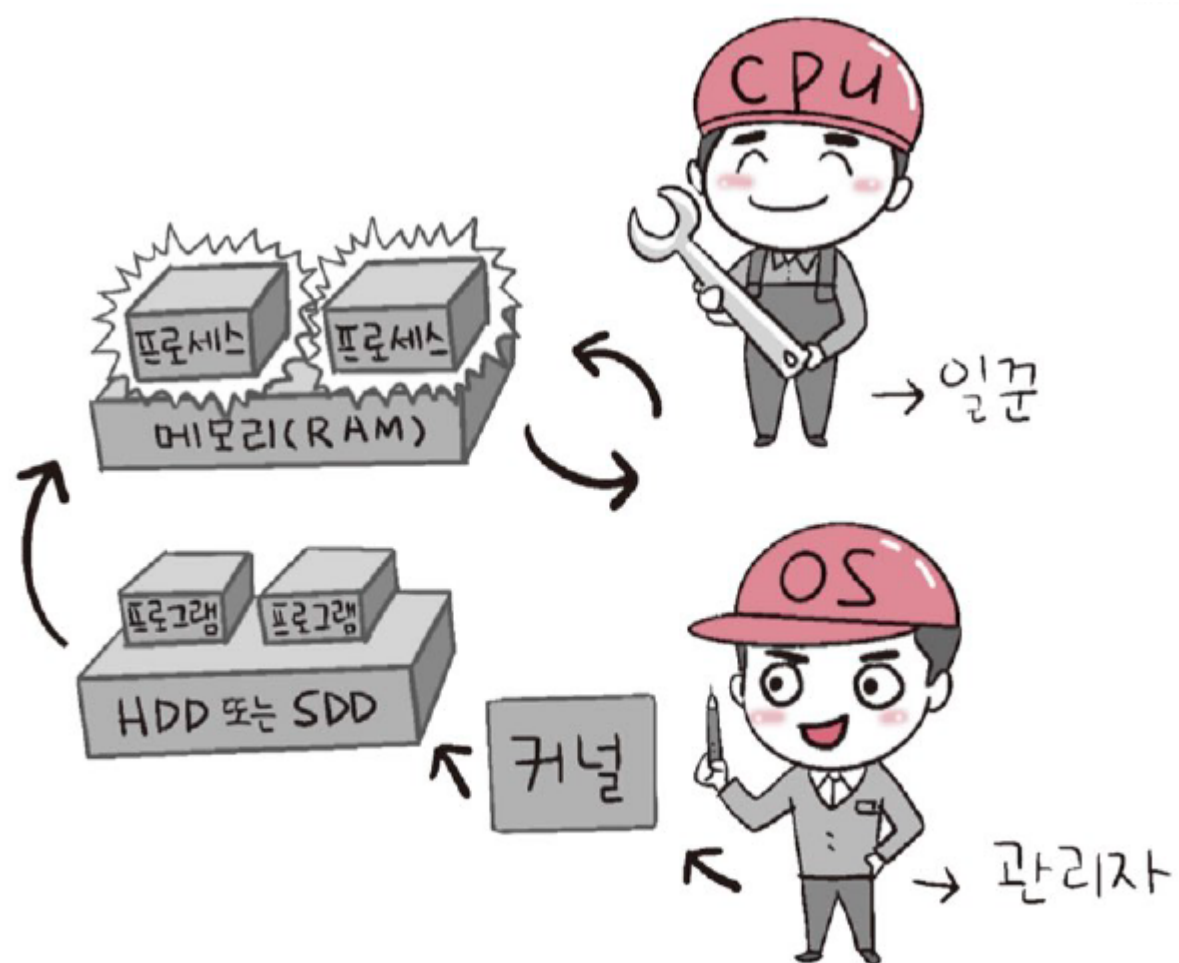


▲ 그림 3-5 컴퓨터의 요소

CPU (Central Processing Unit)

CPU는 주 기억장치인 메모리에서 명령어를 읽어 들이고 이를 해석하여 수행하는 일꾼입니다.

CPU의 주요 구성 요소로는 산술논리연산장치(ALU), 제어장치, 레지스터가 있습니다.



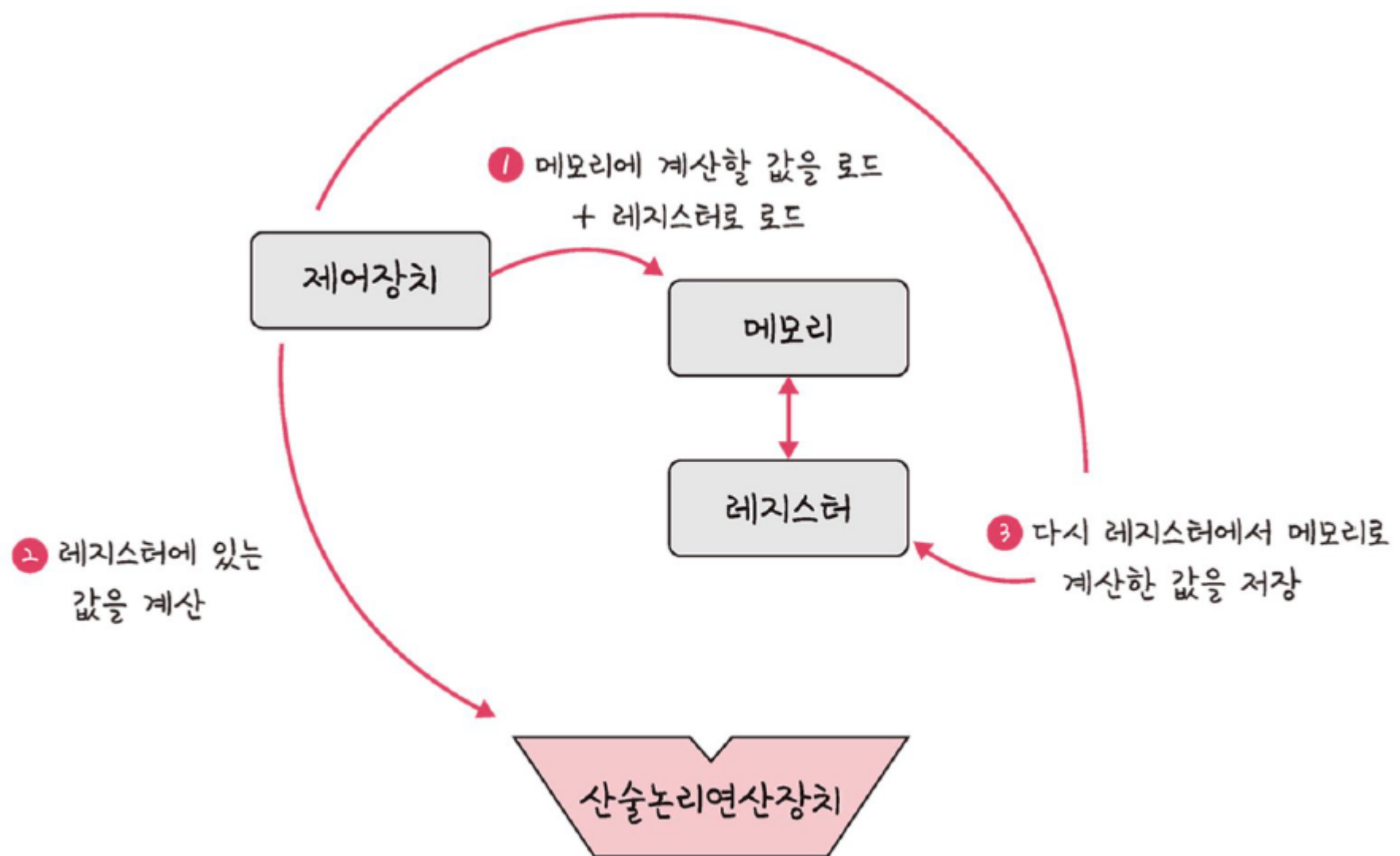
▲ 그림 3-6 CPU라는 일꾼

관리자 역할을 하는 운영체제의 커널이 프로그램을 메모리에 올려 프로세스로 만들면 일꾼인 CPU가 이를 처리합니다.

CPU 구성요소

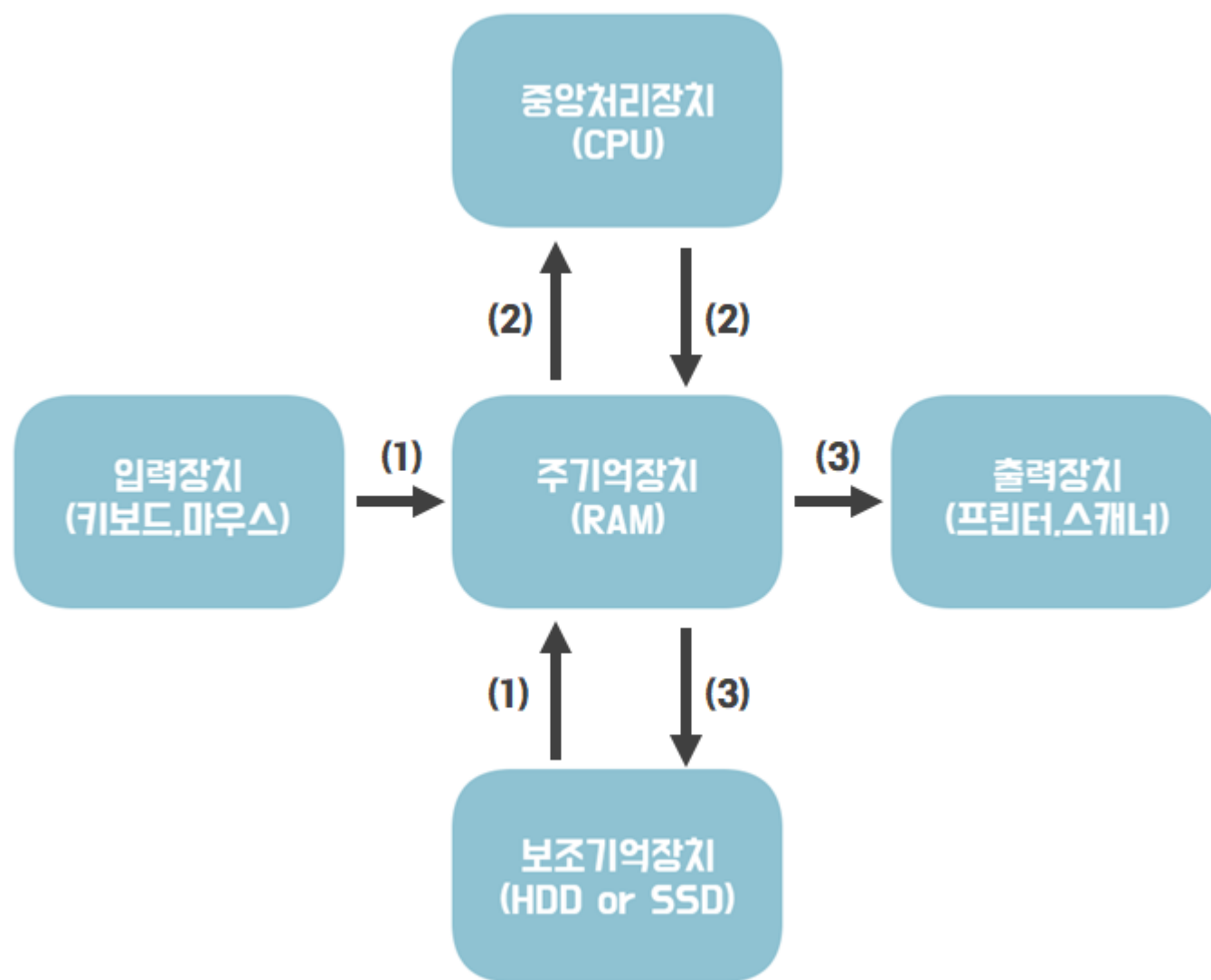
- 제어장치 (CU, Control Unit)
 - 프로세스 조작을 지시하는 CPU의 한 부품.
입출력장치 간 통신을 제어하고 명령어들을 읽고 해석하며 데이터 처리를 위한 순서 결정
- 레지스터
 - CPU 안에 있는 매우 빠른 임시기억장치.
CPU와 직접 연결되어 있으므로 연산 속도가 메모리보다 수십배에서 수백배까지 빠름.
CPU는 자체적으로 데이터를 저장할 방법이 없기 때문에 레지스터를 거쳐 데이터 전달
- 산술논리연산장치 (ALU, Arithmetic Logic Unit)
 - 덧셈, 뺄셈 같은 두 숫자의 산술 연산과 배타적 논리합, 논리곱 같은 논리 연산을 계산하는 디지털 회로입니다.

CPU의 연산 처리



▲ 그림 3-7 CPU 연산 처리

1. 제어장치가 메모리에 계산할 값을 로드합니다. 또한 레지스터에도 로드합니다.
2. 제어장치가 레지스터에 있는 값을 계산하라고 산술논리연산장치에 명령합니다.
3. 제어장치가 계산된 값을 다시 '레지스터에서 메모리로' 계산한 값을 저장합니다



1. 보조기억장치에서 저장된 프로그램을 읽거나, 입력장치에서 입력받은 데이터를 주기억장치에서 읽는다.
2. 주기억장치에서 읽어온 데이터를 중앙처리장치(CPU)가 읽고 처리한 후 다시 주기억장치로 보낸 후 저장한다.
3. 주기억장치는 연산된 데이터를 출력장치에 보내거나 보조기억장치에 저장한다.
4. 제어장치는 1~3 과정에서 명령어가 순서대로 잘 실행되도록 제어하는 역할을 수행한다.

인터럽트

마이크로프로세서(CPU)가 프로그램을 실행하고있을 때, 예외 상황이 발생하여 처리가 필요할 경우에 마이크로프로세서에게 알려 처리할 수 있도록 하는 것입니다.

즉, 어떤 신호가 들어왔을 때 CPU를 잠깐 정지시키는 것을 말합니다.

인터럽트가 발생하면 CPU는 현재 실행 중인 작업을 일시 중단하고, 해당 인터럽트에 대한 처리를 수행합니다.

키보드, 마우스 등 IO 디바이스로 인한 인터럽트, 0으로 숫자를 나누는 산술 연산에서의 인터럽트, 프로세스 오류 등으로 발생합니다.

인터럽트가 발생되면 인터럽트 핸들러 함수가 모여있는 인터럽트 벡터로 가서 인터럽트 핸들러 함수가 실행됩니다.

- 인터럽트 발생 → 인터럽트 벡터로 감 → 인터럽트 핸들러 함수 실행

인터럽트 간에는 우선순위가 있고 우선순위에 따라 실행되며 인터럽트는 하드웨어 인터럽트, 소프트웨어 인터럽트 두 가지로 나뉩니다.

- 하드웨어 인터럽트
 - 키보드를 연결한다거나 마우스를 연결하는 일 등의 IO 디바이스에서 발생하는 인터럽트를 말합니다.
 - 이때 인터럽트 라인이 설계된 이후 순차적인 인터럽트 실행을 중지하고 운영체제에 시스템콜을 요청해서 원하는 디바이스로 향해 디바이스에 있는 작은 로컬 버퍼에 접근하여 일을 수행합니다.
- 소프트웨어 인터럽트

- 프로그램 실행 중에 명시적으로 발생시킬 수 있는 인터럽트입니다.
- 트랩(trap)이라고도 하며, 프로세스 오류 등으로 프로세스가 시스템 콜을 호출할 때 발동합니다.

💡 정리하면

CPU는 명령어를 해석하고 실행하는 작업을 수행하다가 외부에서 발생한 이벤트나 예외 상황에 대응하기 위해 인터럽트를 처리하고, 그 후에 이전 작업을 계속합니다. 인터럽트는 컴퓨터 시스템의 유연성과 효율성을 높이는 데 기여합니다.

이외의 컴퓨터 요소

DMA 컨트롤러

DMA 컨트롤러는 I/O 디바이스가 메모리에 직접 접근할 수 있도록 하는 하드웨어 장치를 뜻합니다.

CPU에만 너무 많은 인터럽트 요청이 들어오기 때문에 CPU 부하를 막아주며 CPU의 일을 부담하는 보조 일꾼이라고 보면 됩니다.

또한, 하나의 작업을 CPU와 DMA 컨트롤러가 동시에 하는 것을 방지합니다.

메모리

메모리(memory)는 전자회로에서 데이터나 상태, 명령어 등을 기록하는 장치를 말하며, 보통 RAM(Random Access Memory)을 일컬어 메모리라고도 합니다.

CPU는 계산을 담당하고, 메모리는 기억을 담당합니다.

공장에 비유하자면 CPU는 일꾼이고, 메모리는 작업장이며, 작업장의 크기가 곧 메모리의 크기입니다. 작업장이 클수록 창고에서 물건을 많이 가져다놓고 많은 일을 할 수 있듯이 메모리가 크면 클수록 많은 일을 동시에 할 수 있습니다.

타이머

타이머(timer)는 몇 초 안에는 작업이 끝나야 한다는 것을 정하고 특정 프로그램에 시간 제한을 다는 역할을 합니다. 시간이 많이 걸리는 프로그램이 작동할 때 제한을 걸기 위해 존재합니다.

디바이스 컨트롤러

디바이스 컨트롤러(device controller)는 컴퓨터와 연결되어 있는 IO 디바이스들의 작은 CPU를 말합니다.

참고

도서 - 면접을 위한 CS 전공 지식 노트

<https://maloveforme.tistory.com/152>

<https://hongcoding.tistory.com/139>

<https://whatisthenext.tistory.com/147>