

3. 운영체제

운영체제 (Operating System)는 사용자가 컴퓨터를 쉽게 다루게 해주는 인터페이스
한정된 메모리나 시스템 자원을 효율적으로 분배함
윈도우, 리눅스 등

수많은 프로그램들을 하드웨어들이 잘 수행할 수 있게 해주는 역할

3-1. 운영체제와 컴퓨터

하드웨어와 소프트웨어 (유저 프로그램)를 관리하는 일꾼인
운영체제와 CPU, 메모리 등으로 이루어진 컴퓨터를 알아보자

3-1-1. 운영체제의 역할과 구조

- 우선 OS는 하드웨어인 CPU, 메모리, 디스크 등을 잘 동작하게 해주는 SW

- 디스크에 있는 파일이 사용자에게 의해 실행이 되고
- 그 파일의 일부가 메모리에 올라가고
- 메모리의 일부의 명령어들이 CPU에서 실행된다

운영체제의 역할

운영체제의 역할은 크게 네 가지가 있다

1. CPU 스케줄링과 프로세스 관리

CPU 소유권을 어떤 프로세스에 할당할지, 프로세스의 생성과 삭제, 자원 할당 및 반환을 관리

CPU는 한번에 하나의 프로그램만 실행하는 것이 아니라 윈도우의 수많은 프로그램들이 동시에 (시분할) 실행되고 있다

CPU는 계산만 하기 때문에, CPU가 처리할 일을 가져다 주는 것이 운영체제의 역할

2. 메모리 관리

한정된 메모리를 어떤 프로세스에 얼마만큼 할당해야 하는지 관리

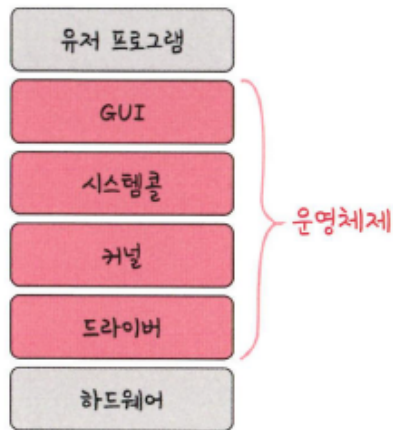
3. 디스크 파일 관리

디스크 파일을 어떤 방법으로 보관할지 관리

4. I/O 디바이스 관리

I/O 디바이스인 마우스, 키보드와 컴퓨터 간에 데이터를 주고 받는 것을 관리

운영체제의 구조



유저 프로그램이 맨 위, 하드웨어가 가장 밑에 있는 구조

GUI, 시스템콜, 커널, 드라이버 부분이 운영체제를 지칭한다

<GUI>

사용자가 전자 장치와 상호 작용할 수 있도록 하는 사용자 인터페이스의 한 형태, 단순 명령어 창이 아닌 아이콘을 마우스로 클릭하는 단순한 동작으로 컴퓨터와 상호 작용할 수 있도록 해줌

<드라이버>

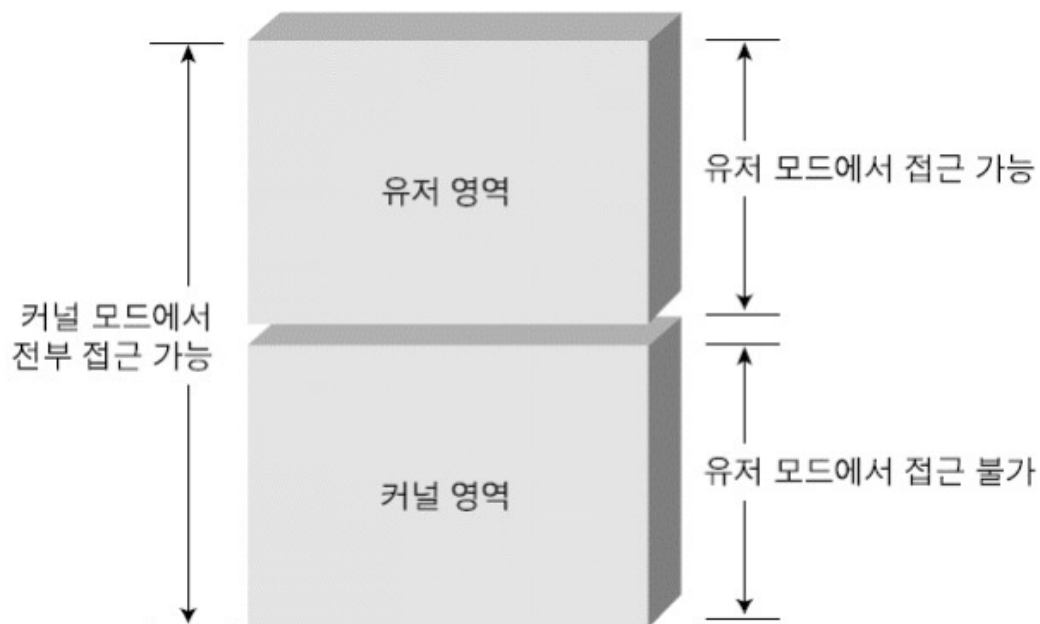
하드웨어를 제어하기 위한 소프트웨어

<CUI>

그래픽이 아닌 명령어로 처리하는 인터페이스

유저 모드와 커널 모드

<시스템의 메모리 영역 분리>



프로그램이 실행되면서 유저모드에서 커널 모드로 전환되기도 하고 커널모드에서 다시 유저 모드로도 전환되기도 한다

우리가 개발하는 프로그램은 일반적으로 유저 모드에서 실행된다

프로그램 실행 중에 인터럽트가 발생하거나 시스템콜을 호출하게 되면 커널 모드로 전환된다

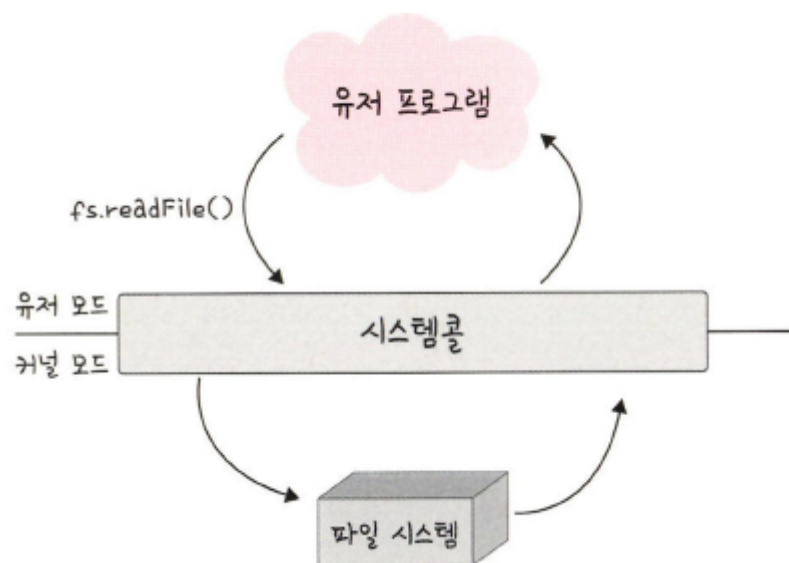
보통 운영체제에서는 커널모드와 유저모드 두가지 프로세서 접근모드를 지원한다. 그 이유는 유저 어플리케이션이 함부로 운영체제의 치명적인 데이터를 수정하거나 삭제하지 못하게 하기 위해서 이다. 커널모드는 모든 시스템과 메모리에 접근이 허가된 프로세스 실행 모드이다. 유저모드보다 커널모드에 더 높은 권한을 줌으로써 유저모드에서 에러가 발생했을 때 시스템 전체의 안전성을 보장해 준다.

시스템콜

운영체제가 커널에 접근하기 위한 인터페이스이며 유저 프로그램이 운영체제의 서비스를 받기 위해 커널 함수를 호출할 때 쓴다

- 유저 프로그램이 I/O 요청으로 트랩(trap)을 발동하면
- 올바른 I/O 요청인지 확인한 후
- 유저 모드가 시스템콜을 통해 커널 모드로 변환되어 실행된다.

예를 들어 I/O 요청인 `fs.readFile()` 이라는 파일 시스템의 파일을 읽는 함수가 발동했다고 가정한다면



이때 유저 모드에서 파일을 읽지 않고 커널 모드로 들어가 파일을 읽고 다시 유저 모드로 돌아가 그 뒤에 있는 유저 프로그램의 로직을 수행한다

이 과정을 통해 컴퓨터 자원에 대한 직접 접근을 차단할 수 있고, 프로그램을 다른 프로그램으로부터 보호할 수 있다

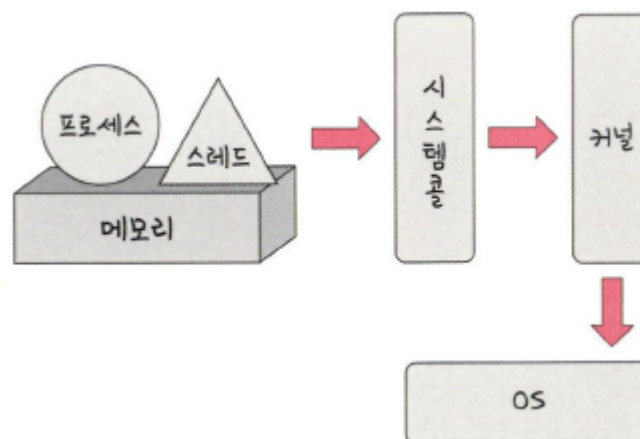
시스템콜이 필요한 이유

우리가 일반적으로 사용하는 프로그램은 '응용 프로그램' 이다. 유저레벨의 프로그램은 유저레벨의 함수들 만으로는 많은 기능을 구현하기 힘들기 때문에, **커널(kernel)**의 도움을 반드시 받아야 한다. 이러한 작업은 응용프로그램으로 대표되는 유저 프로세스(User Process)

에서 유저모드에서는 수행할 수 없다. 반드시 kernel에 관련된 것은 커널모드로 전환한 후에야, 해당 작업을 수행할 권한이 생긴다.

그렇다면 권한은 왜 필요한 것일까?

그 이유는 만약 권한이 없을 때, 해커가 피해를 입히기 위해 악의적으로 시스템 콜을 사용하는 경우나 초보 사용자가 하드웨어 명령어를 잘 몰라서 아무렇게 함수를 호출했을 경우에 시스템 전체를 망가뜨릴 수도 있기 때문이다. 따라서 이러한 명령어들은 특별하게 커널 모드에서만 실행할 수 있도록 설계되었고, 만약 유저 모드에서 시스템 콜을 호출할 경우에는 운영체제에서 불법적인 접근이라 여기고 trap을 발생시킨다.



프로세스나 스레드에서 운영체제로 어떠한 요청을 할 때 시스템콜이라는 인터페이스와 커널을 거쳐 운영체제에 전달된다

이 시스템콜은 하나의 추상화 계층이다. 그렇기 때문에 이를 통해 네트워크 통신이나 데이터베이스와 같은 낮은 단계의 영역 처리에 대한 부분을 많이 신경 쓰지 않고 프로그램을 구현할 수 있는 장점이 있다

modebit

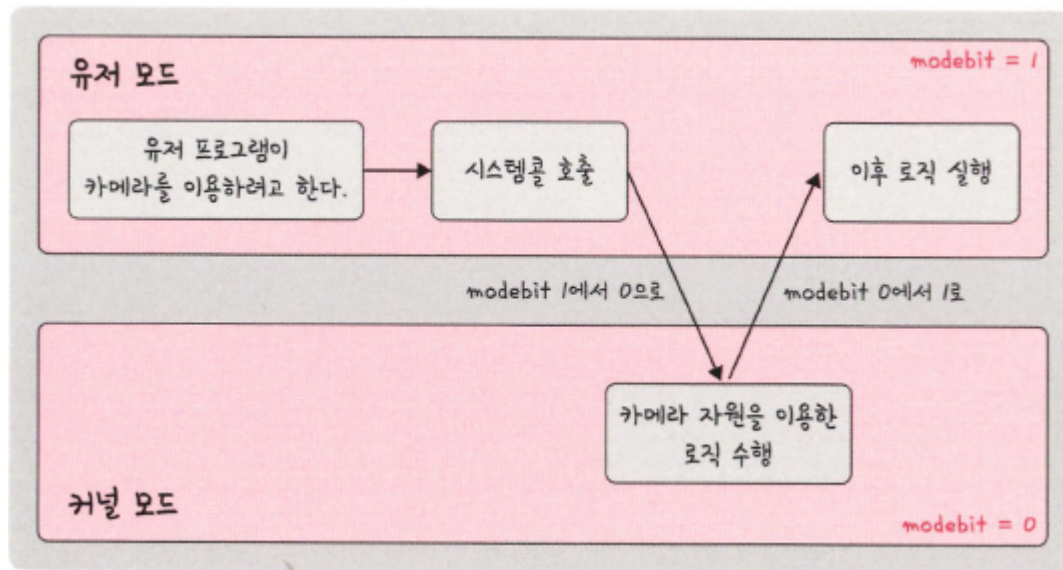
시스템콜이 작동될 때 modebit을 참고해서 유저 모드와 커널 모드를 구분한다

modebit은 1또는 0 값을 가지는 플래그 변수이다. 카메라, 키보드 등 I/O 디바이스는 운영체제를 통해서만 작동해야 한다.

카메라를 켜는 프로그램이 있다고 가정해보자. 만약 유저 모드를 기반으로 카메라가 켜진다면, 사용자가 의도하지 않았는데 공격자가 카메라를 갑자기 켤 수 있다.

물론 커널 모드를 거쳐 운영체제를 통해 작동한다고 해도 100% 막을 수는 없지만, 운영체제를 통해 작동하게 해야 막기가 쉽다. 이를 위한 장치가 바로 modebit이다.

modebit의 0은 커널모드, 1은 유저 모드라고 설정되며 유저 모드일 경우에는 시스템콜을 못하게 막아서 한정된 일만 가능하게 한다.



- 유저 프로그램이 카메라를 이용하고 할 때 시스템콜을 호출하고
- modebit을 1에서 0으로 바꾸며 커널 모드로 변경한 후 카메라 자원을 이용한 로직을 수행
- 이후에 modebit을 0에서 1로 바꿔서 유저 모드로 변경하고 이후 로직을 수행

<I/O 요청>

입출력 함수, 데이터베이스, 네트워크, 파일 접근 등에 관한 일

<드라이버>

하드웨어를 제어하기 위한 소프트웨어

<유저 모드>

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

<커널 모드>

모든 컴퓨터 자원에 접근할 수 있는 모드

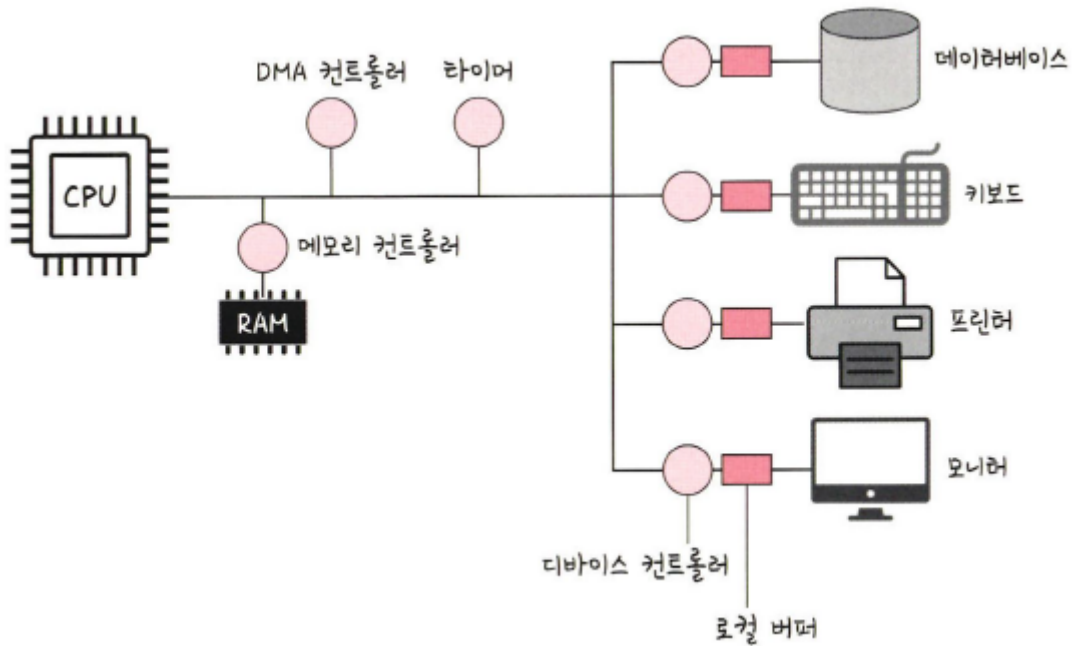
<커널>

운영체제의 핵심 부분이자 시스템콜 인터페이스를 제공

보안, 메모리, 프로세스, 파일 시스템, I/O 디바이스, I/O 요청 관리 등 운영체제의 중추적인 역할

3-1-2. 컴퓨터의 요소

컴퓨터는 CPU, DMA 컨트롤러, 메모리, 타이머, 디바이스 컨트롤러 등으로 이루어짐



CPU

CPU(Central Processing Unit)는 **산술논리연산장치**, 제어장치, 레지스터로 구성되어 있는 장치

인터럽트에 의해 단순히 메모리에 존재하는 명령어를 해석해서 실행하는 일꾼

관리자 역할을 하는 운영체제의 커널이 프로그램을 메모리에 올려 프로세스로 만들면, 일꾼인 CPU가 이를 처리한다

제어장치

제어장치 (CU, Control Unit)는 프로세스 조작을 지시하는 CPU의 한 부품이다

입출력 장치간 통신을 제어하고 명령어들을 읽고 해석하며 데이터 처리를 위한 순서를 결정

레지스터

CPU 안에 있는 매우 빠른 **임시기억장치**

CPU와 직접 연결되어 있으므로 연산 속도가 메모리보다 수십 배에서 수백 배까지 빠르다

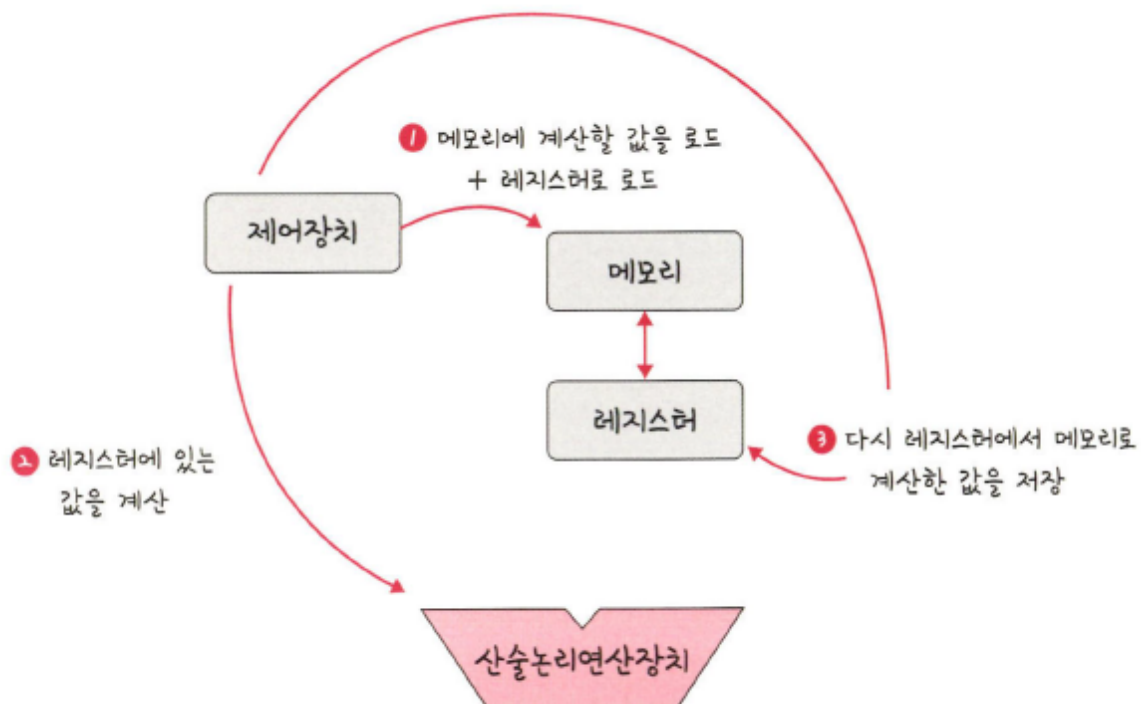
CPU는 자체적으로 데이터를 저장할 방법이 없기 때문에 레지스터를 거쳐 데이터를 전달

산술논리연산장치

산술논리연산장치(ALU, Arithmetic Logic Unit) 는 덧셈 뺄셈 같은 두 숫자의 산술 연산과 배타적 논리합, 논리곱 같은 연산을 계산하는 디지털 회로

CPU의 연산 처리

CPU에서 제어장치, 레지스터, 산술논리연산장치를 통해 연산하는 예는 다음과 같다



1. 제어장치가 메모리에 계산할 값을 로드, 레지스터에도 로드
2. 제어장치가 레지스터에 있는 값을 계산하라고 산술논리연산장치에 명령
3. 제어장치가 계산된 값을 다시 레지스터에서 메모리로 계산한 값을 저장

인터럽트

인터럽트는 어떤 신호가 들어왔을 때 CPU를 잠깐 정지시키는 것

키보드, 마우스 등 I/O 디바이스로 인한 인터럽트, 0으로 숫자를 나누는 산술 연산에서의 인터럽트, 프로세스 오류 등으로 발생

인터럽트가 발생되면 인터럽트 핸들러 함수가 모여 있는 인터럽트 백터로 가서 인터럽트 핸들러 함수가 실행된다

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

하드웨어 인터럽트

키보드를 연결한다거나 마우스를 연결하는 일 등의 IO 디바이스에서 발생하는 인터럽트를 말함

이때 인터럽트 라인이 설계된 이후 순차적인 인터럽트 실행을 중지하고 운영체제에 시스템 콜을 요청해서 원하는 디바이스로 향해 디바이스에 있는 작은 로컬 버퍼에 접근하여 일을 수

행

소프트웨어 인터럽트

소프트웨어 인터럽트는 **트랩(trap)** 이라고도 한다

프로세스 오류 등으로 프로세스가 시스템콜을 호출할 때 발동

DMA 컨트롤러

DMA 컨트롤러는 IO 디바이스가 메모리에 직접 접근할 수 있도록 하는 하드웨어 장치를 뜻함

CPU 에만 너무 많은 인터럽트 요청이 들어오기 때문에 CPU 부하를 막아주며 CPU의 일을 부담하는 보조 일꾼이라고 보면 된다

하나의 작업을 CPU와 DMA 컨트롤러가 동시에 하는 것을 방지

메모리

메모리는 전자회로에서 데이터나 상태, 명령어 등을 기록하는 장치를 말한다

보통 RAM을 일컬어 메모리라고 한다. CPU는 계산을 담당하고 메모리는 기억을 담당

CPU - 일꾼

메모리 - 공장의 크기

메모리가 크면 클수록 많은 일을 동시에 진행할 수 있다

타이머

타이머는 몇 초 안에는 작업이 끝나야 한다는 것을 정하고 특정 프로그램에 시간 제한을 다는 역할

시간이 많이 걸리는 프로그램이 작동할 때 제한을 걸기 위해 존재

디바이스 컨트롤러

컴퓨터와 연결되어 있는 IO 디바이스들의 작은 CPU를 말한다
