

영상: [운영체제](#)

## 1 운영체제 개요

---

운영 체제란?

- 

운영 체제의 목적

- 

운영 체제의 분류

- 

몇 가지 용어

- 

운영 체제의 예

- 

운영 체제의 구조

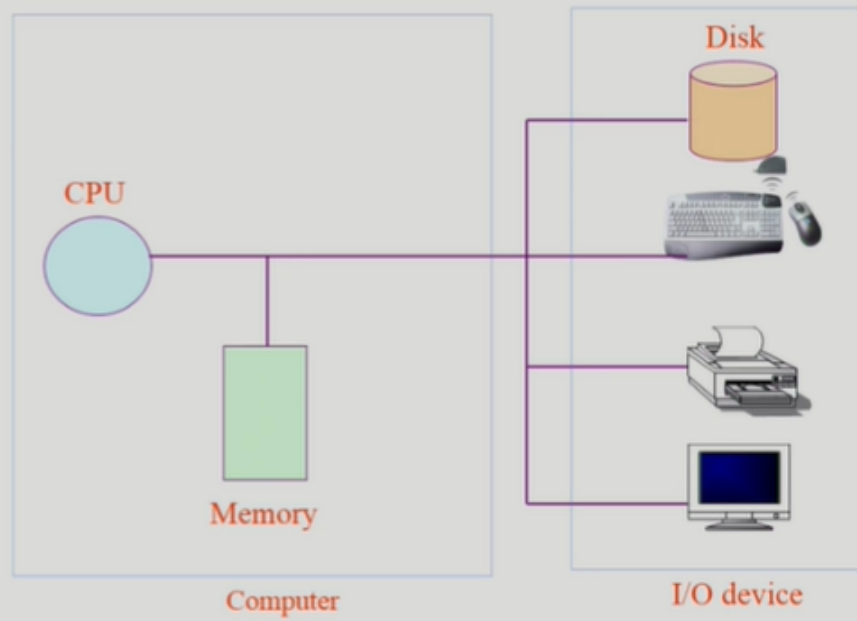
- 

## 2 시스템 구조 및 프로그램의 실행 Part-1

---

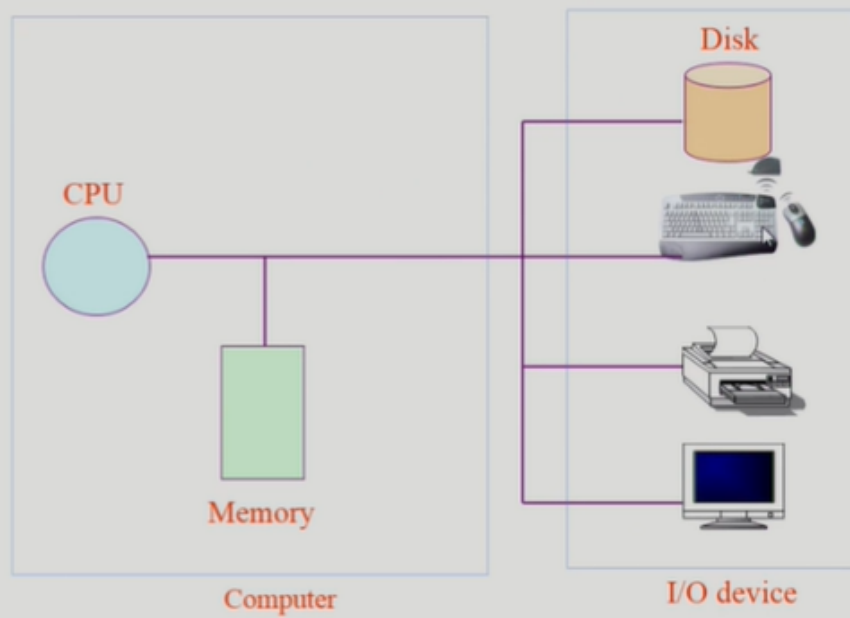
컴퓨터 시스템 구조

## 컴퓨터 시스템 구조

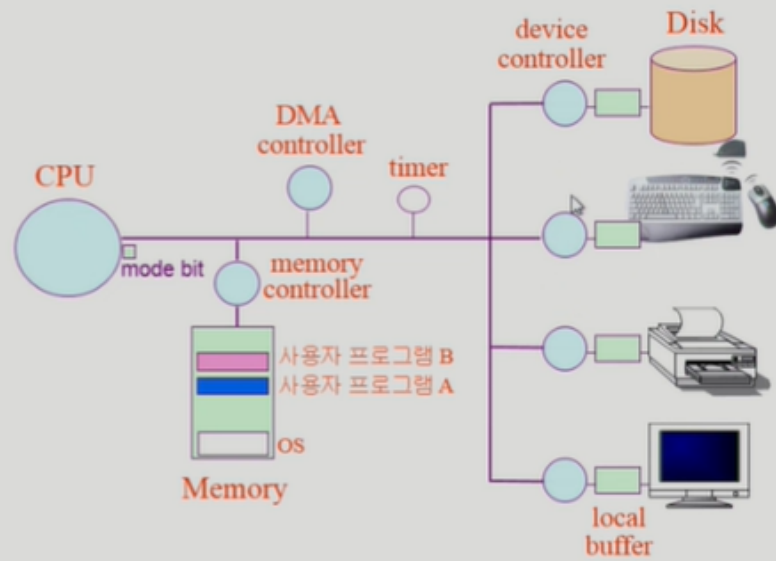


- 컴퓨터의 구조는 CPU & Memory

## 컴퓨터 시스템 구조

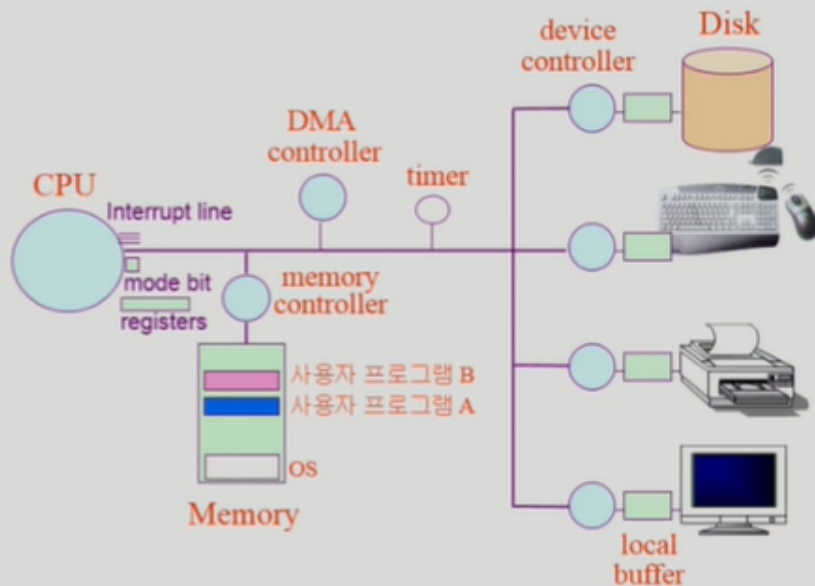


## 컴퓨터 시스템 구조



- 자세히 본 구조
- device 컨트롤러는 작은 cpu라고 보면 된다.
- 디스크와 cpu사이 처리속도가 매우 많이 차이난다.

## 컴퓨터 시스템 구조



- 레지스터와 모드빌이랑게 있다.
- I/O같은 작업이나 다른 선결작업이 있을때 메모리에서 대기한다.
- 만약

## Mode bit

- 운영체제가 실행하는 프로세스인지 아니면 프로그램에서 실행하는 프로세스인지 구분할 때 사용하는 비트
- mode bit이 0일때는 메모리접근, io device접근 등 운영체제가 cpu제어권한을 가진 상태
- mode bit = 1 사용자 프로그램이 cpu를 가짐,,, 제한적인 일만 가능
- 사용자프로그램은 잘못된 수행으로 다른 프로그램 및 운영체제에 피해가 가지 않도록 보호장치가 필요
- 1 : 사용자모드 0 : 모니터모드(운체 코드 수행)

## Timer

- cpu를 한 프로세스가 계속 독점하게 할 수 없으니까 시분할에서 timer가 필요하다.
- 정해진 시간이 흐른 뒤 운영체제에게 제어권이 넘어가도록 인터럽트 발생
- 타이머는 매 클럭 틱마다 1씩 감소
- 타이머값이 0이되면 타이머 인터럽트 발생
- cpu를 특정 프로그램이 독점하는것으로부터 보호
- device driver
  - OS가 장치별 처리루틴 -> software (File System으로 작동한다. (i/o))
- device controller
  - 각 장치를 통제하는 일종의 작은 CPU

## Device Controller

- device 컨트롤러는 작은 cpu라고 보면 된다.

## 입출력과 인터럽트 (Input-Output and Interrupt)

### 시스템 콜 (System Call)

- 사용자 프로그램은 운영체제 커널 함수 호출하는것
- 일반 함수호출과 다르다.
- trap을 사용하여 인터럽트 벡터의 특정위치로 이동

## 2 시스템 구조 및 프로그램의 실행 Part-2

---

### 동기식 입출력과 비동기식 입출력

- 

### DMA (Direct Memory Access)

-

서로 다른 입출력 명령어

- 

저장장치 계층 구조

- [12장 메모리 구조](#)

프로그램의 실행 (메모리 load)

- 

커널 주소 공간의 내용

- 

사용자 프로그램이 사용하는 함수

-