

CH 01 - 1 [ch1 12(P.1~19)]

Ch 01 : Introduction

- <https://parksb.github.io/article/5.html>

Objectives

- 컴퓨터 시스템의 일반적인 구조, 인터럽트의 역할
- 현대 멀티프로세서 컴퓨터 시스템
- 유저모드와 커널모드의 변환

Basic Organization of Computer Systems

OS란 무엇인가

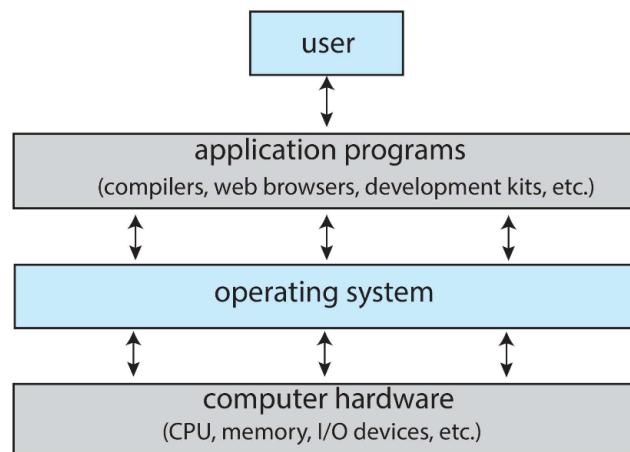
- 물리적인 CPU는 하나인데 이를 이용하여 USER/OS/HW를 모두 처리해야 한다. 운영 체제(Operating System)는 컴퓨터의 하드웨어를 관리하고, 하드웨어와 소프트웨어, 사용자를 매개하는 프로그램이다. 커널(Kernel)은 운영체제의 핵심이며, 실체다. 운영 체제는 커널과 커널 모듈(Kernel module)들로 구성되는데, 커널이 운영체제의 핵심이다보니 일반적으로 운영체제와 커널은 동일시 된다. 커널이 같다면 같은 운영체제로 취급한다.
- OS 시스템의 목적
 - 하드웨어의 복잡성을 알지 못하더라도 유저가 컴퓨터를 사용하는 데 어려움이 없도록 하는 것이 궁극적인 목적이다.
 - 사용자 프로그램을 실행시키고, 사용자의 문제를 더 쉽게 해결하고자 함
 - 컴퓨터 시스템을 더 쉽게 사용 가능하도록 함
 - 컴퓨터 하드웨어를 더 효율적으로 사용하도록 함
- 성능의 요소
 - Throughput(처리량) - 단위시간당 작업 처리량 (Jobs/sec) - 시스템
 - Utilization(활용률) - (% of time busy) - 시스템
 - 하드웨어 자원은 매우 비싸기 때문에 이를 얼마나 잘 활용할 수 있는가가 경제적인 측면에서 매우 중요하다.

- Response time(응답 시간) - (sec/job) - 유저

- 실행시간을 의미하는 것이 X 프로세스의 요청 이후 처리가 시작되는 그! 시점까지 얼마나 걸리는가

✓ utilization과 response time은 반비례 관계이다. 하드웨어를 많은 프로세스가 사용할 수록 응답 시간은 길어지기 때문이다.

- 컴퓨터 시스템은 4가지 컴포넌트로 분류된다.



1. 하드웨어 : 기본적인 컴퓨팅 자원들을 제공

- CPU, Memory, I/O 자원

2. OS : 다양한 어플리케이션과 사용자들 사이에서 하드웨어의 사용을 조절하고 처리

3. Application Programs : 시스템 자원을 사용하여 사용자의 컴퓨팅 문제들을 해결

- 워드 프로세서, 컴파일러, 웹 브라우저, 데이터베이스 등

4. Users : 사람, 기계, 혹은 다른 컴퓨터

- OS의 기능

- 운영체제의 역할은 사용자 관점(User View)과 시스템 관점(System View)으로 나뉘볼 수 있다.
- 사용자 관점 : 일반적으로 사용자는 컴퓨터 앞에 앉아 키보드와 마우스를 조작한다. 이 경우 운영체제는 사용자가 컴퓨터 자원 사용(Resource utilization)을 신경쓰지 않게 도우며, 사용자가 컴퓨터를 쉽게 이용할 수 있도록 만든다. 또 다른 경우, 사용자는 메인프레임(Mainframe)에 연결된 터미널을 사용하거나 미니컴퓨터(Minicomputer)를 사용한다. 이 상황에서는 컴퓨터의 자원을 여러 사용자가 나눠 쓰게 되는데, 운영체제는 사용자들이 자원을 공평하게 사용할 수 있도록 돕는다.

- 시스템 관점 : 시스템에게 운영체제는 **자원 할당자(Resource allocator)**다. 컴퓨터 시스템은 CPU 시간, 메모리 공간, 파일 저장소 공간, 입출력 장치 등 다양한 문제를 해결해야 한다. 운영체제는 이러한 컴퓨터 자원들을 관리하는 **제어 프로그램(Control program)**으로서 동작한다.
- 스마트폰과 테블릿과 같은 모바일 디바이스는 유용성과 배터리 성능을 위해 자원이 적고, 최적화 되어있다.
- 임베디드 컴퓨터나 오토모바일과 같은 소수의 컴퓨터들은 유저 인터페이스가 없다. 주로 사용자의 개입 없이 사용된다.

• OS의 정의

- OS는 자원 할당자이다.
 - CPU 시간, 메모리 공간, 파일 저장 공간, 입출력 장치와 같은 모든 자원들을 관리한다. (자원에 대한 다양한 요청과 충돌을 관리)
 - **효과적이고 공정한 자원의 사용**을 위해 수많은 요청 순서를 결정한다.
 - 컴퓨터는 "우선순위"에 의해 동작하는 것이 매우 중요하다.
- OS는 제어 프로그램이다.
 - 오류를 막고 부적절한 컴퓨터의 사용을 방지하기 위해 프로그램의 실행을 제어한다. (보안 및 예외 처리)

• OS의 정의 [2]

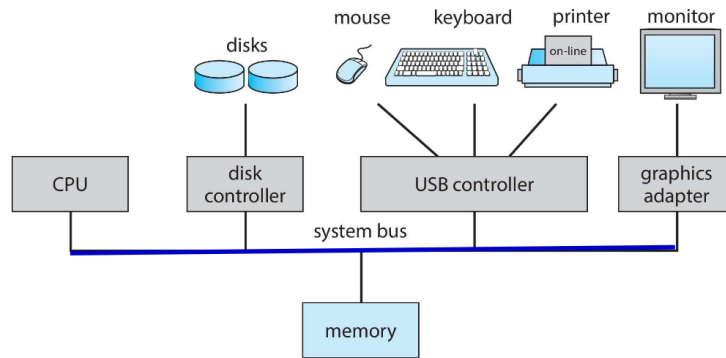
- 보편적으로 인정되는 정의는 없다. (OS의 기능은 시간이 지나며 꾸준히 변화하기 때문이다.)
- 운영체제의 커널 : 컴퓨터에서 항상 돌아가고 있는 한 개의 프로그램 (항상 메모리에 적재되어 있다.)
- 모든 프로그램은 두 가지로 분류된다.
 - system program : OS와 항상 함께 실행
 - application program : OS와 함께하지 않는 모든 프로그램
- 오늘날의 일반적인 목적을 위한 OS는 미들웨어를 포함한다.
 - middleware : 시스템의 일부가 데이터를 통신하고 관리할 수 있도록 하는 모든 소프트웨어

• 컴퓨터의 시동

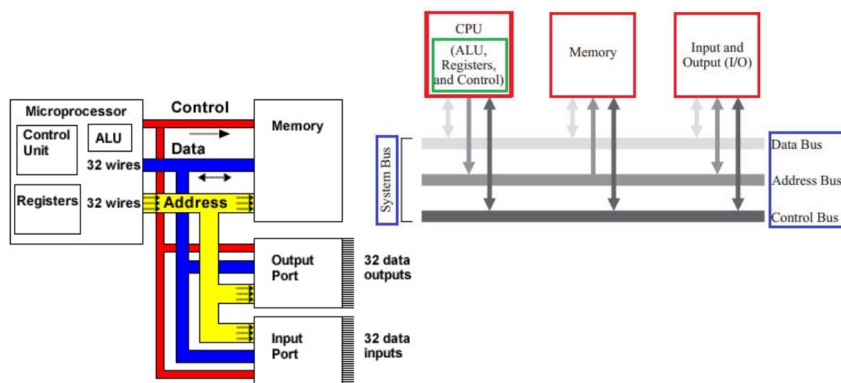
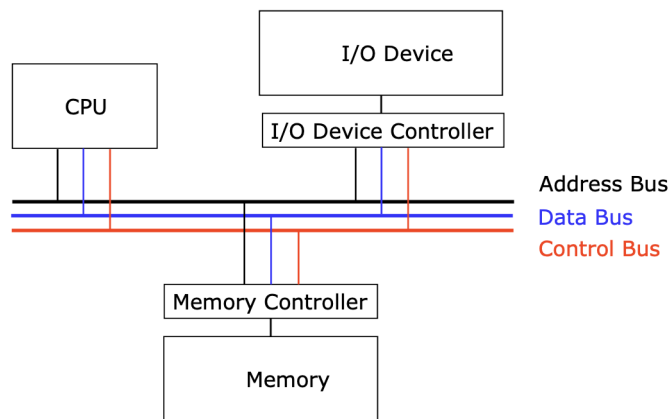
- bootstrap 프로그램 혹은 bootstrap loader
 - 컴퓨터를 켜면 **부트스트랩 프로그램**(Bootstrap program)이라는 초기화 프로그램이 실행된다.
 - 일반적으로 ROM(비휘발 메모리)나 EPROM에 저장되고, 펌웨어라고 알려져 있다. (전기가 나가더라도 지워지지 않는다.)
 - 부트스트랩 프로그램은 시스템을 초기화하고, 부트로더(Boot loader)를 실행한다. (멀티부팅 컴퓨터의 경우 부트로더가 여러 운영체제를 가리키고 있는데, 이 경우엔 어떤 운영체제를 실행할지 선택해야 한다.) 그리고 부트로더는 최종적으로 운영체제를 실행하게 된다.
 - Loading ...
 - 디스크에서 OS 커널을 찾는다.
 - 메모리에 OS 커널을 로드한다.
 - OS를 실행한다.
- 커널이 로드, 실행되면 시스템과 사용자에게 서비스를 제공해야 한다. 이때 일부 서비스는 커널 외부에서 제공되는데, 이들은 부팅할 때 메모리에 로드되는 시스템 프로세스(System processes)나 시스템 데몬(System daemons)이다. UNIX의 경우 첫 시스템 프로세스는 `init` 이며, 이 프로세스는 또 다른 데몬들을 실행시킨다. 데몬은 프로세스로 백그라운드에서 돌면서 시스템 로그는 남기는 등의 여러 작업을 한다. 이러한 과정이 끝나면 시스템이 완전히 부팅되고, 이벤트가 발생하기를 기다리게 된다.

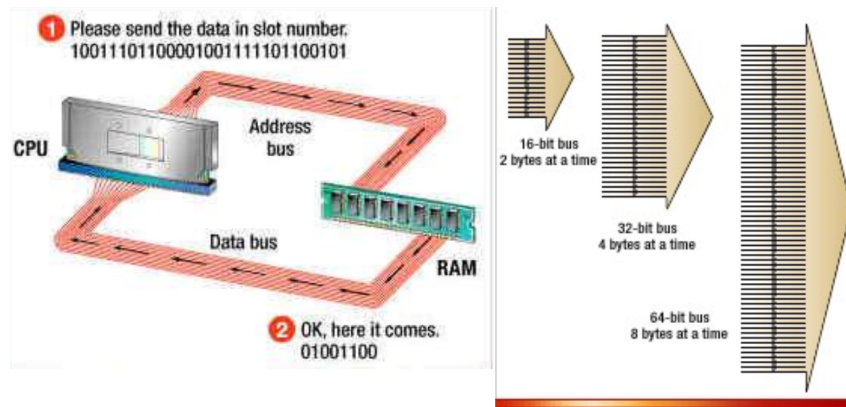
-
- 시스템 부트
 - 시스템에 전원이 들어올 때 고정된 메모리 위치에서 실행된다. 초기 부트 코드를 유지하는 데 사용되는 펌웨어 ROM
 - OS는 하드웨어를 사용 가능하도록 만들어야 한다.
 - 부트 스트랩 로더는 ROM이나 EEPROM에 저장되어 있으며, 커널을 위치시키고 메모리로 로드시킨 후 실행한다.
 - 디스크의 부트 부분을 boot disk 혹은 system disk라고 부른다.

-
- 컴퓨터 시스템 조직



- 현대의 일반적인 컴퓨터 시스템은 여러개의 CPU와 장치 컨트롤러(Device controllers)로 구성되어 있다. 그리고 이들은 공통버스(Common bus - address, data, control)로 이어져 메모리를 공유한다.

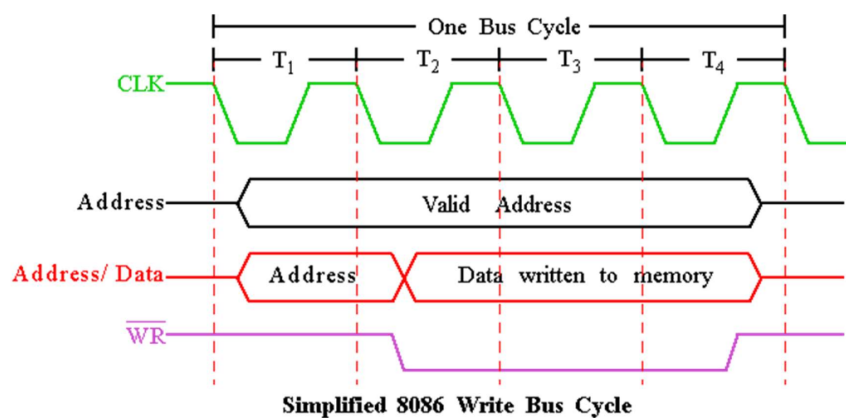




- 읽기/쓰기 bus cycle

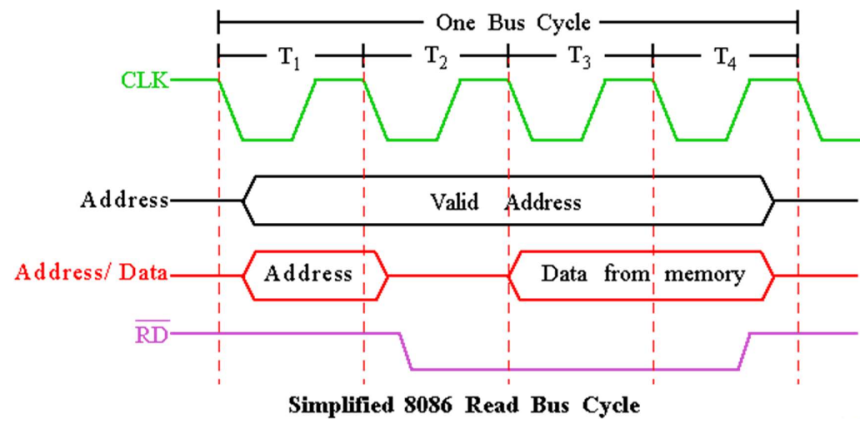
☀ 그림 어떻게 그리는 지 알아두면 좋을것이라고 교수님이 언급

[쓰기]



1. T1 상태 동안 주소 버스에 주소를 넣는다.
2. 데이터는 T2 상태 동안 **CPU에 의해 데이터 버스에 저장**되며 T3, T4 시간 동안 유지된다.
3. 메모리 또는 입출력 장치에 기록된다. (쓰여진다.)

[읽기]



1. T₁의 시간 동안 마이크로프로세서가 주소 버스에 주소를 입력한다.
2. 읽기에서는 **메모리에 의해 데이터가 데이터 버스에 저장**된다. 이후 데이터가 읽혀진다.