

# 1강

요즘의 컴퓨터? 이런거 다 운영체제다 이 말이야

## 운영체제란 무엇인가?

**하드웨어를 손쉽게 그리고 효율적으로 사용할 수 있는 abstraction을 제공한다.**

- CPU - process (하드웨어 - 그 하드웨어를 효율적으로 사용하기 위한 abstraction)

CPU를 사용하기 위해서는 process라는 abstraction이 필요하다.

- Memory - address space

주로 process, address space를 이번 수업에서 다룬다.

- Disk - file
- Network - port

**자원 공유 및 분배를 위한 policy를 결정한다.**

윈도우 창을 하나 띄우는 것에 대해서 자원이 할당되는 것이다.

- Policy의 예 : FIFO - First IN First Out, LRU - Least Recently Use
- 설계설정 (design decisions)이 필요
  - 대형컴퓨터, 휴대전화에 사용되는 policy가 다르다
  - 이 자원의 운용은 design decisions에 의해서 결정되는 경우가 많다.

최소한 수업시간에 다 이해하고 가라

---

## 운영체제 공부는

**Abstraction을 이해하는 것이다**

- 코드에 대한 이해가 필수적
- 외우는 것이 큰 도움이 안된다.
- "연관"을 지어 사고하는 것이 매우 중요
  - abstraction과 abstraction을 연관된 것으로 보라
  - process와 memory address과 떨어져있는 것이 아니다.

운영체제를 왜 공부해야 하는가? 이게 왜 필수적인가? 왜 실제로 도움이 되는가?

검색엔진이 3초가 걸리더라 이것 결국 3ms로 1000배를 개선시켰다. 그런데 어떻게 줄였는가?

메모리 시스템과 네트워킹 시스템을 개선했더니 이렇게 되더라. 운영체제 짱이다.

## Abstraction: process

### Program

- 컴퓨터를 실행 시키기 위한 일련의 순차적으로 작성된 명령어의 모음
- 컴퓨터 시스템의 disk와 같은 secondary storage에 바이너리 형태로 저장되어 있다.

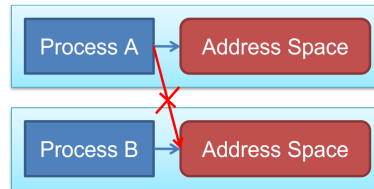
### Abstraction : Process

- 실행되고 있는 프로그램의 추상화(abstraction)
- 프로그램이 실행되고 있는 상태
- 프로그램에 대해서 프로세스는 여러 개일 수 있다.
- Program counter, stack, data section등을 포함하고 있다.
- 프로세스는 물리적이지 않지만, 그 동작을 볼 수 있다.

### Abstraction : address space

프로세스에게 할당된 메모리 공간

프로세스는 다른 프로세스의 메모리 공간을 할당할 수 없다. → Protection Domain  
내 프로세스에서 다른 프로세스의 메모리 공간을 읽거나 쓸 수 없다.  
이 원칙을 깨는 것이 해킹이다.



- 프로세스가 차지하는 메모리 공간
- 왜 필요할까:
  - Protection Domain - 서로의 주소 공간을 침범 할 수 없음

Protection Domain은 일단은 모든 OS에 적용되는 경우이다.  
다만 수업이 진행되면서 예외가 있을 수 있다.

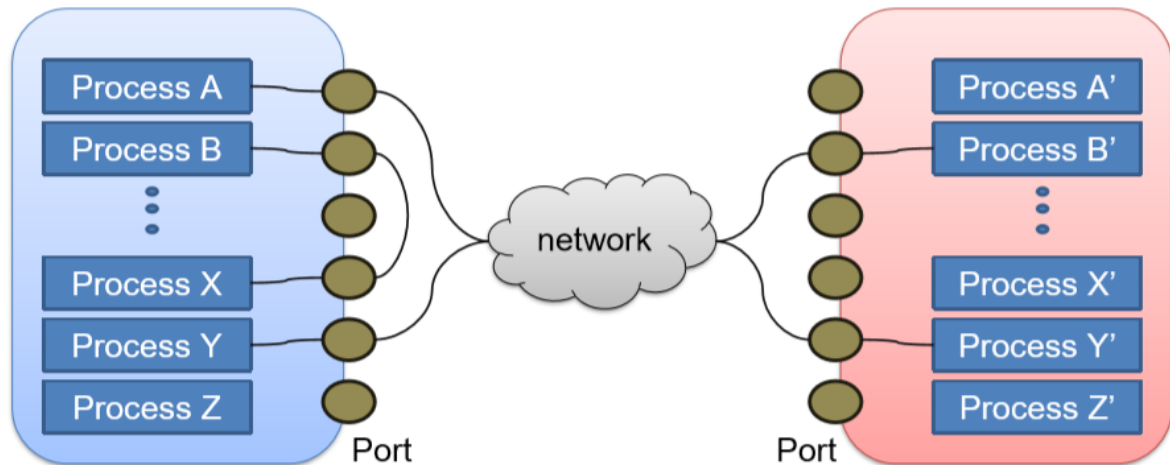
## Abstraction : File

파일의 property는 Persistent storage이다.  
프로세스와 address space는 전원을 끄면 사라진다.  
그러나 파일은 전원과 관계없이 늘 남아있다.

File은 실제로는 여러군데에 나뉘어 존재하나 하나로 묶어서 우리가 생각하곤한다.

## Abstraction : Port

컴퓨터 간에 메시지를 주고 받는 communication endpoint



카카오톡으로 통신할때 친구가 어디에 있는지 알 필요가 없다. 그 역할을 포트가 다 해준다.

- 포트가 데이터를 잘 받고 잘 수신해서 전달하는 것이다.
- 그리고 프로세스간의 목적지를 헛갈리지 않는다.

Process 간에 주고 받는 메시지도 port를 사용한다. Loopback이라는 특수한 포트를 말이다!

## Software의 구분

### System Software

- 목적 : 컴퓨터 시스템을 구동시키는 SW
  - OS 이외의 system software는?
  - Compiler/Assembler

### Application Software

- 목적 : 특정 용도로 사용됨
  - AI 인식기, 빅데이터 처리기, Word, Internet Explorer
  - 매우 매우 다양한 응용프로그램이 존재

# 운영 체제의 특징 (Application과 비교)

- OS는 항상 동작 한다.

멈추면 컴퓨터가 멈추는 것이다.

- 제어 기능으로서 항상 자원에 대한 감시 활동을 한다.

Supervisor mode라는 별도의 측별한 모드에서 활동한다. 그리고 이 모드에서만 하드웨어에 접근 가능하다.

superviser mode ↔ User mode

- 하드웨어에 대한 제어 기능

Device driver이다.

마우스 키보드 스피커 다 제어한다.

---

## OS and Kernel

### OS와 kernel에 대한 두가지 관점

1. os = kernel
2. os = kernel + windows system(GUI) + library

### Kernel

운영체제의 핵심 부분으로, 자원할당, 하드웨어 인터페이스, 보안등을 담당

슈퍼바이저 모드에서 작동

### Windows system

윈도우 형태의 그래픽 사용자 인터페이스

예) OpenGL

### Library

서브루틴과 자주 사용되는 함수들의 집합

예) libc, win32.dll

---

## Relation of Hardware, O/S and Application

Application이 Hardware에 접근하기 위해서는 OS를 반드시 거쳐야 한다.

그리고 그러한 경우에는 많은 경우에 라이브러리를 거쳐야 한다.

