Intro - 프로그램 실행 시 일어나는 일

1. 프로세서가 메모리로부터 명령을 Fetch 해 옴
2. Decode : 받아온 명령이 무슨 명령인지 알아냄
3. Execute : 명령 실행
4. 프로세서가 다음 명령어로 이동

OS(운영체제)

- 프로그램 실행을 쉽게 함
- 프로그램들이 메모리를 공유하도록 해줌
- 프로그램들이 장치들과 상호작용하도록 해줌
- 시스템이 바르게 그리고 효율적으로 작동하도록 확인하는 것

Virtualization

- 운영체제가 물리적 자원(프로세서, 메모리, 디스크)을 가상의 형태로 바꾸는 것
- 가상의 형태가 일반적이고, 강력하며, 사용하기 쉽다
- OS를 '가상머신(VM)'이라고 부르기도 함

System call

- 사용자가 OS에게 명령을 내릴 수 있도록 해줌
- 대표적인 OS는 100개정도의 System Call을 내보냄 ex) 프로그램 실행, 메모리 접근, 장치 접근

OS as a resource manager

- OS는 CPU, 메모리, 디스크와 같은 자원을 관리한다
- 많은 프로그램을 동시에 실행 -> **CPU를** 공유하도록 함
- 다양한 프로그램이 동시다발적으로 그들의 명령과 데이터에 접근하도록 해줌 -> **메모리** 공유
- 다양한 프로그램이 장치에 접근하도록 해줌 -> **디스크** 공유

CPU Virtualization

- 시스템은 수많은 가상 CPU를 가지고 있음
- 하나의 CPU를 무한대로 보이는 숫자의 CPU로 바꿈
- 많은 프로그램들이 동시에 실행되는것처럼 보이게 하는 것 = Virtualing the CPU

Memory Virtualization

- 물리적 메모리는 바이트의 배열이다
- 프로그램은 자신의 모든 자료구조를 메모리에 저장한다 -> 읽기(load) : 주소를 명시해 데이터에 접근 -> 쓰기(store) : 데이터를 명시해 해당 주소에 쓰기
- 각각의 프로세스는 각각의 Virtual Address Space에 접근함 OS는 이 가상 주소를 물리적 주소에 매핑 한 프로그램의 메모리 레퍼런스가 다른 프로세스의 Address Space에 영향을 끼치지 않음 물리적 주소는 공유되는 자원(OS가 관리)

Concurrency Problem

- OS는 많은 것을 동시에 관리
- 현대의 Multi-thread 프로그램도 Concurrency 문제 존재
- Thread : 같은 메모리 공간에서 실행되는 Function, worker()에서 시작
- 명령이 Atomic 하지 않으면, Concurrency Problem 발생(명령 실행 도중에 값이 변할 수 도 있음)

Persistence

- DRAM같은 장치들은 휘발성
- 하드웨어/소프트웨어는 데이터를 영구적으로 저장해야함 Hardware : HDD, SDD Software : 파일시스템이 디스크를 관리, 파일시스템이 사용자가 만드는 파일을 저장하는 일을 함
- 디스크에 쓰기작업을 할때 OS는 : 이 데이터가 디스크 어디에 위치할지 알아내고, 저장장치에게 입출력 요청을 보냄
- 파일 시스템이 쓰기 작업에서 발생 할 수 있는 시스템 오류를 Handle

Design Goals

- 추상화 : 시스템을 편하고 쓰기 쉽게 만들기
- 높은 성능 제공 : 운영체제의 간접비(Overhead)를 최소화 -> OS는 과부하 없이 Virtualization을 할 수 있도록 노력해야 함
- 프로그램간의 보호 : 한 프로그램의 오작동이 운영체제의 손상으로 이어지면 안됨
- 높은 신뢰성 : 운영체제가 멈추지 않고 동작해야함
- 에너지 효율성, 보안, 유동성