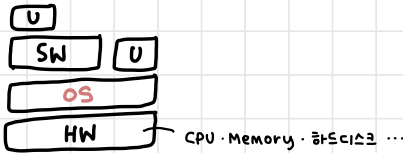


OS - 운영체제 개요

공개 소프트웨어
 서버용 (Linux) · PC 용 (Windows) · 스마트디바이스용 (Android) 운영체제

운영체제 : 컴퓨터 하드웨어 바로 위에 설치되어 사용자 및 다른 모든 소프트웨어와 하드웨어를 연결하는 소프트웨어 계층

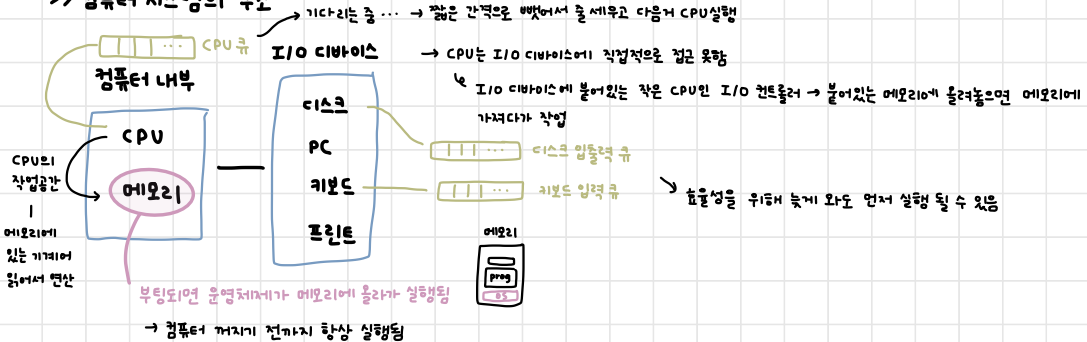


→ 컴퓨터 시스템을 편리하게 사용할 수 있는 환경 제공 + 컴퓨터 시스템의 **자원을 효율적으로 관리**

동시에 여러개의 프로그램을 수행할 수 있음

↳ CPU, 메모리, I/O 장치의 효율적 관리 → OS가 자원 sharing
 실행중인 프로그램에 짧은 시간의 CPU 분할을 할당
 + 메모리 공간 적절히 분배
 ↓
 주어진 자원으로 최대한의 성능을 내도록 → **효율성**
 특정 사용자 / 프로그램의 지나친 불이익이 발생하지 않도록 → **형평성**

>> 컴퓨터 시스템의 구조



CPU 스케줄링 - 어떤 프로그램에게 CPU 사용권을 줄까

↳ 어떤 프로그램? CPU 얼마큼?

메모리 관리 - 한정된 메모리 어떻게 잘게서 쓸까

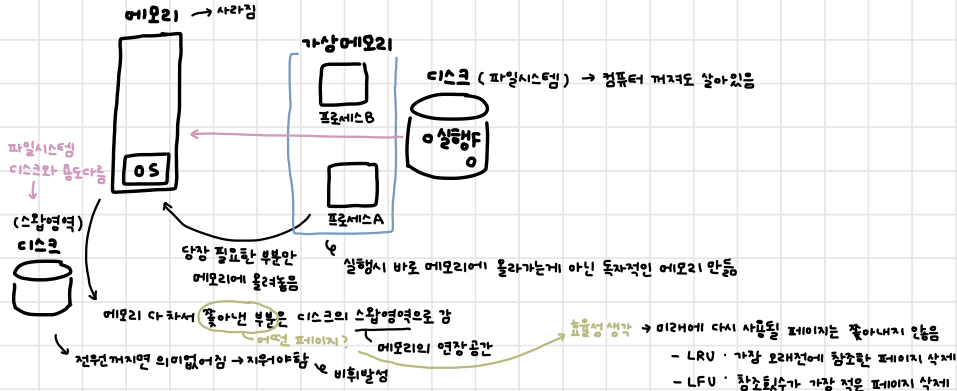
디스크 스케줄링 - 디스크에 들어온 요청 어떤 순서로 처리할까

인터럽트 캐싱 - 빠른 CPU와 느린 I/O 장치간 속도 차이 어떻게 극복할까

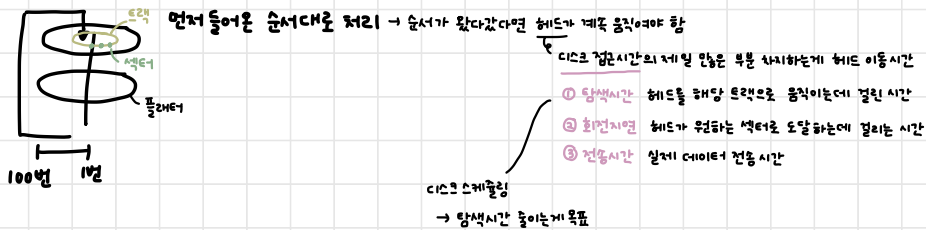
↳ 캐싱 → 중간 단계를 두는 것 (ex. 같은 데이터 요청할 때) · 인터럽트 → I/O 컨트롤러에 일시키고 당장 일할 것 줌
 ↓
 요청한 것 다 되었다 전달작업을 인터럽트하함
 항상 인터럽트 체크해서 완료된게 있는지 확인

CPU 스케줄링

- FCFS (First come First Served) → 앞에 온게 CPU 사용시간이 길어지면 기다리는 시간 길어짐
- SJF (Shortest Job First) → 금번 CPU 사용시간이 가장 짧은 프로세스 먼저 → minimum average waiting time 보장
 ↳ 사용시간이 긴 프로세스는 계속 기다려야 함 → starvation 문제 발생
- RR (Round Robin) → 각 프로세스 동일한 CPU 사용시간을 가져서 할당시간 끝나면 CPU 뺏기고 뒤에서 줄을 섬
 ↳ 어떤 프로세스도 $(n-1) \times \text{할당시간}$ 이상 기다리지 않음 · 대기시간이 프로세스의 CPU 사용시간에 비례



디스크 스케줄링



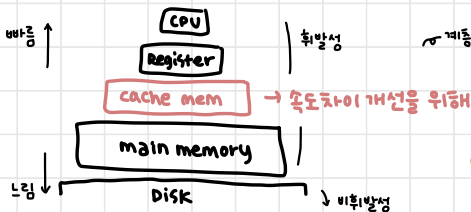
SSTF (Shortest Seek Time First)

탐색 시간 짧은 것 먼저 → starvation 문제

SCAN

헤드가 디스크 한쪽 끝에서 다른쪽 끝으로 이동하며 가는 길에 있는 모든 요청 처리

>> 저장장치 계층구조



누를 저장할지가 제일 큰 알고리즘 문제

caching → 동일한 데이터 요청시 아래로 내려갈 필요 없음

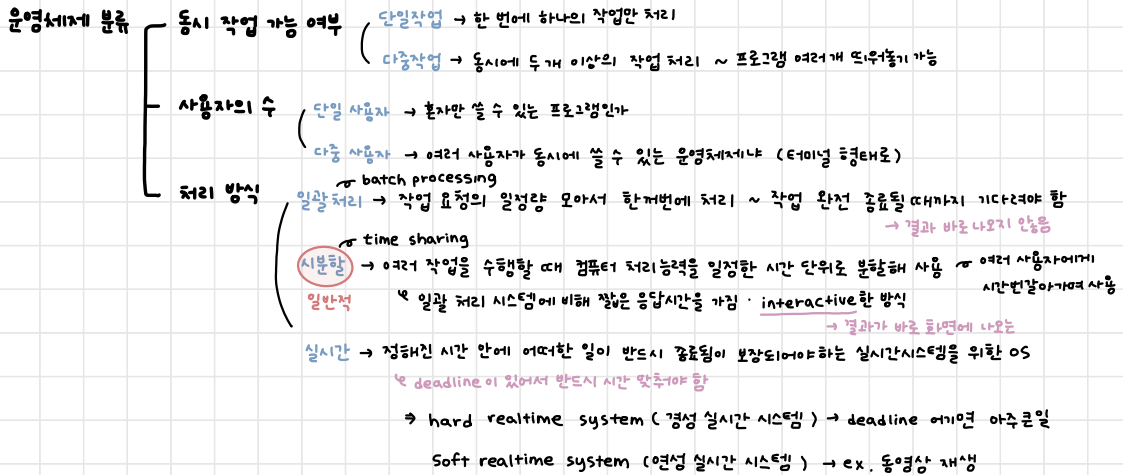
↳ 복사본 저장

플래시메모리 (SSD, USB ...)

↳ 반도체장치

- ⊕ 비휘발성 · 전력소모 적음 · 물리적 충격에 강함 · 가볍다 → 모바일 장치로 좋음
- but 쓰기 횟수 제한이 있음 · 데이터 변질 가능성 있음

운영체제 → 항상 메모리에 상주하는 부분인 커널 (핵심 부분) ⑤ 각종 주변 시스템 유틸리티 포함한 개념



Multitasking : 컴퓨터 안에 프로그램 여러개 동시에 돌리는 것

Multiprogramming : 메모리 측면을 강조 → 메모리에 여러 프로그램이 동시에 올라가 있는 상황

Timesharing : 시간을 공유해서 씀 → CPU가 시간 분할해서 사용

셋 다 컴퓨터에서 프로그램 여러개 동시에 돌릴 수 있다

Multiprocessor : 하나의 컴퓨터에 CPU가 여러 개 붙어있음 → 동시에 여러 프로그램이 CPU에 붙어 실행할 수 있음
↳ 중독 생길 수 있음

UNIX → 서버를 위함 만들어진 운영체제 · 대부분 C 언어로 코드 작성 · 최소한의 커널 구조로 효율적
↳ OS 예시

DOS → 단일 프로그램만 실행 가능

MS Windows → 다중 작업용 OS