



01_Operating System (OS)

01_운영체제의 개요

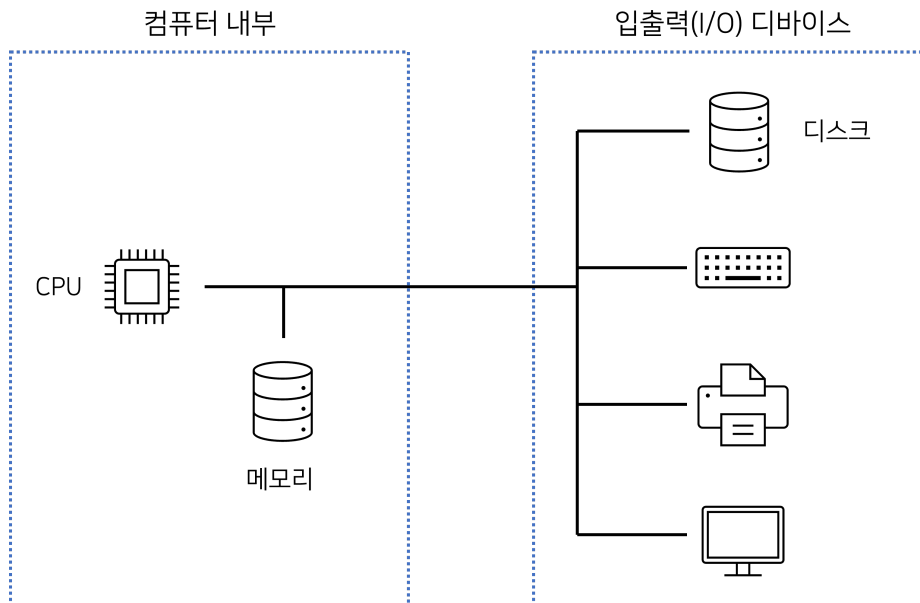
| 운영체제 (Operating System, OS)

- 컴퓨터 하드웨어 바로 위에 설치되는 소프트웨어 계층
- 사용자 및 다른 모든 소프트웨어와 하드웨어를 연결하는 소프트웨어 계층
- 모든 컴퓨터 시스템의 필수적인 부분
- 컴퓨터 하드웨어 - 운영체제 - 각종 소프트웨어 - 사용자



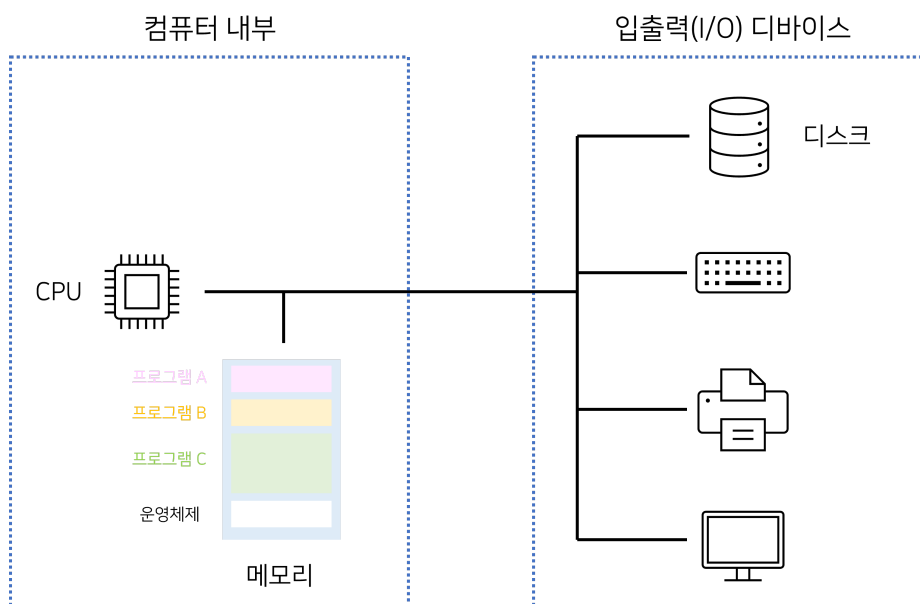
- 컴퓨터 시스템을 편리하게 사용할 수 있는 환경을 제공
 - 운영체제는 동시 사용자/프로그램들이 각각 독자적 컴퓨터에서 수행되는 것 같은 환상(illusion)을 제공
 - 하드웨어를 직접 다루는 복잡한 부분을 운영체제가 대행
- **컴퓨터 시스템의 자원을 효율적으로 관리**
 - CPU, 메모리, I/O 장치 등을 효율적으로 관리
 - 주어진 자원으로 최대한 성능을 내도록 → **효율성**
 - 특정 사용자/프로그램의 지나친 불이익이 발생하지 않도록 → **형평성**
 - 사용자 및 운영체제 자신의 보호

컴퓨터 시스템의 구조



- 매순간 CPU가 메모리를 읽는다.
 - **Input (인풋)** : 디바이스를 통해서 데이터가 컴퓨터 내부로 들어가는 것 (키보드, 마우스)
 - **Output (아웃풋)** : 컴퓨터 내부 작업을 통해서 디바이스로 보내는 것 (프린터, 모니터)

⚠ **하드디스크는 보조기억장치, 컴퓨터 외부 디바이스로 본다. (Input / Output 모두 가능)**

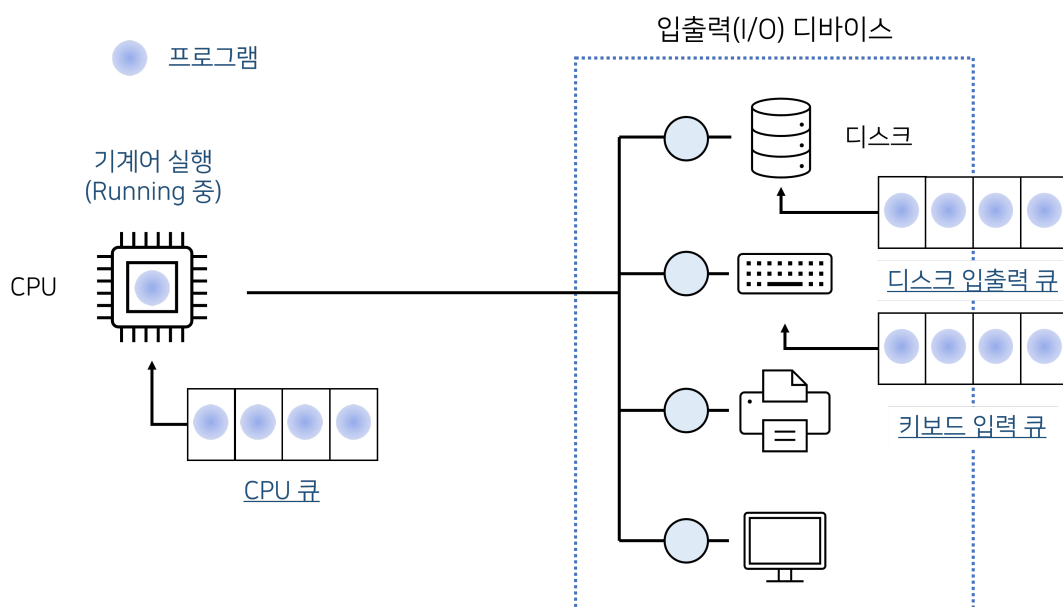


- 부팅하면 운영체제가 메모리에 올라가면서 시작
- 운영체제는 컴퓨터가 꺼질 때 까지 항상 실행되는 프로그램

- **커널**

- OS의 기본 기능을 실행하는 부분
- 응용프로그램이나 주변장치나 조작 감시, 디스크나 메모리 등을 자원 배분
- 응용프로그램의 실행

- CPU의 작업 공간은 메모리, 매 클럭마다 메모리 어딘가 있는 기계어를 읽어서 연산
- CPU가 외부장치에서 직접적으로 데이터 이동을 하지는 않는다. → **외부장치는 컨트롤러를 통해 요청**
- **CPU 스케줄링** : 어떤 프로그램에게 CPU 사용권을 줄 것인가?
- **메모리 관리** : 한정된 메모리를 어떻게 나누어서 쓸 것인가?
- **디스크 스케줄링** : 디스크에 들어온 요청을 어떤 순서로 처리할 것인가?
- **인터럽트, 캐싱** : 빠른 CPU와 느린 I/O 장치간의 속도차를 어떻게 극복할 것인가?
- **프로세서** : 중앙 처리 장치를 의미하며 CPU라고 불리는 부품이 프로세서인 하드웨어를 의미한다.
- **프로세스** : 메모리에 적재되어 실행 중이거나 대기 중인 프로그램을 프로세스 라고 한다.



- **Interactive Application** : 입출력(I/O) 디바이스를 통해 사람과 상호 작용하는 프로그램

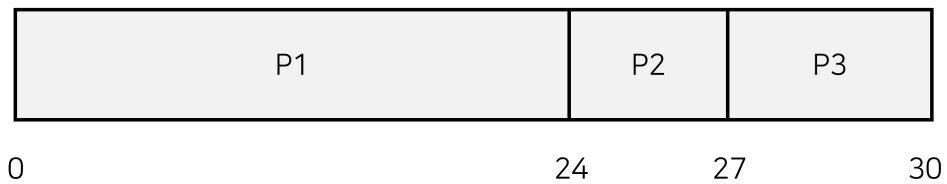
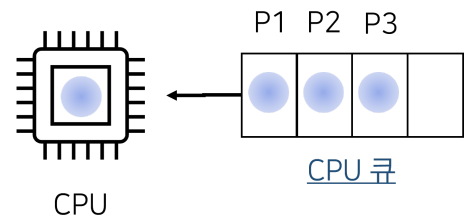
- **Scientific Application** : CPU만 오래 사용하는 프로그램

CPU 스케줄링

1. FCFS (First-Come First-Served)

- 선착순으로 처리, 효율적이지는 않다.

Process	금번 CPU 사용 시간
P1	24
P2	3
P3	3

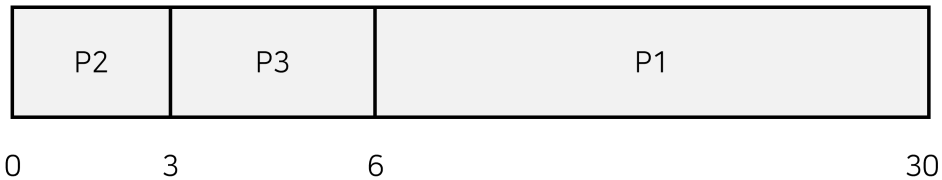
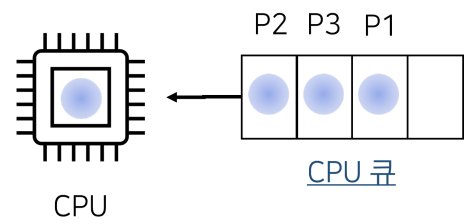


프로세스의 도착 순서 : $P1 P2 P3$

Waiting time : $P1 = 0, P2 = 24, P3 = 27$

Average waiting time : $(0 + 24 + 27) / 3 = 17$

Process	금번 CPU 사용 시간
P1	24
P2	3
P3	3



프로세스의 도착 순서 : $P2\ P1\ P3$

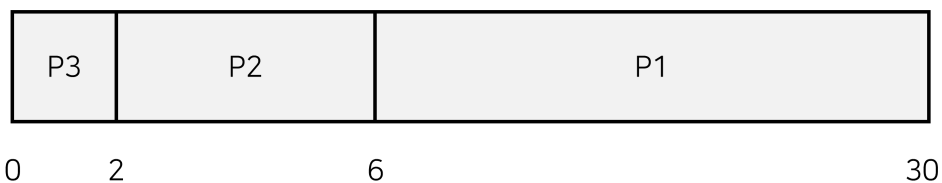
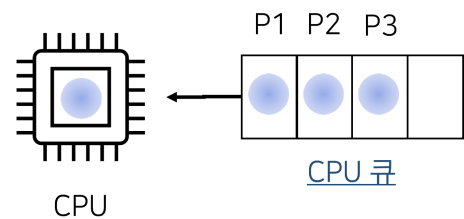
Waiting time : $P1 = 6, P2 = 0, P3 = 3$

Average waiting time : $(6 + 0 + 3) / 3 = 3$

2. SJF (Shortedst-Job-First)

- 금번 CPU 사용시간이 가장 짧은 프로세스를 제일 먼저 스케줄링
- SJF는 Minimum Average Waiting Time을 보장

Process	금번 CPU 사용 시간
P1	24
P2	4
P3	2



프로세스의 도착 순서 : $P1\ P2\ P3$

Waiting time : $P1 = 6, P2 = 2, P3 = 0$

Average waiting time : $(6 + 2 + 0) / 3 = 2.67$

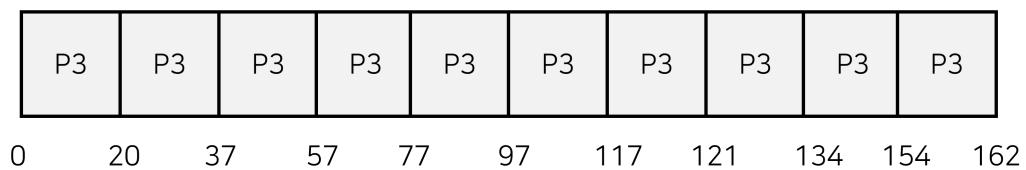
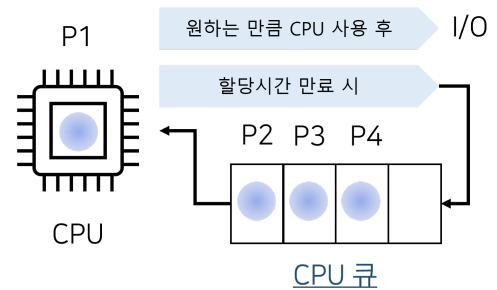
- Starvation (기아 현상) 발생 가능

- P1이 무한정 기다려야 할 수도 있음
- CPU 큐가 계속 추가될 수 있음

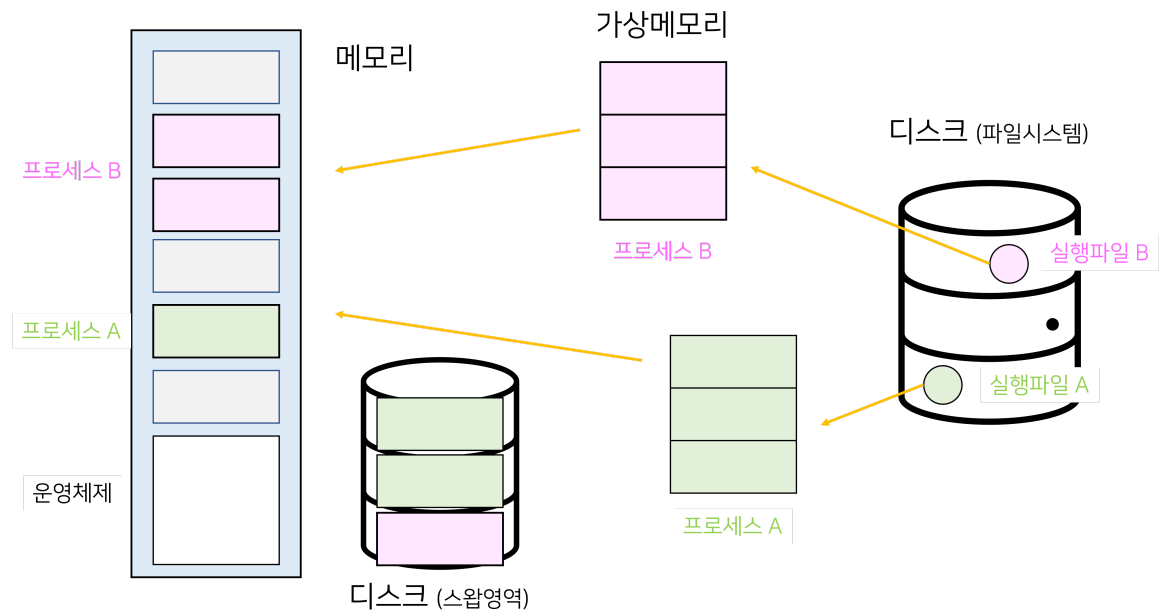
3. Round Robin (RR)

- 각 프로세스는 동일 크기의 CPU 할당시간을 가짐
- 할당시간이 끝나면 인터럽트가 발생하여 프로세스는 CPU를 빼앗기고 CPU 큐의 제일 뒤에 줄음 섬
- n 개의 프로세스가 CPU 큐에 있는 경우
 - 어떤 프로세스와 $(n-1) * \text{할당시간}$ 이상 기다리지 않음
 - 대기시간이 프로세스의 CPU 사용시간에 비례

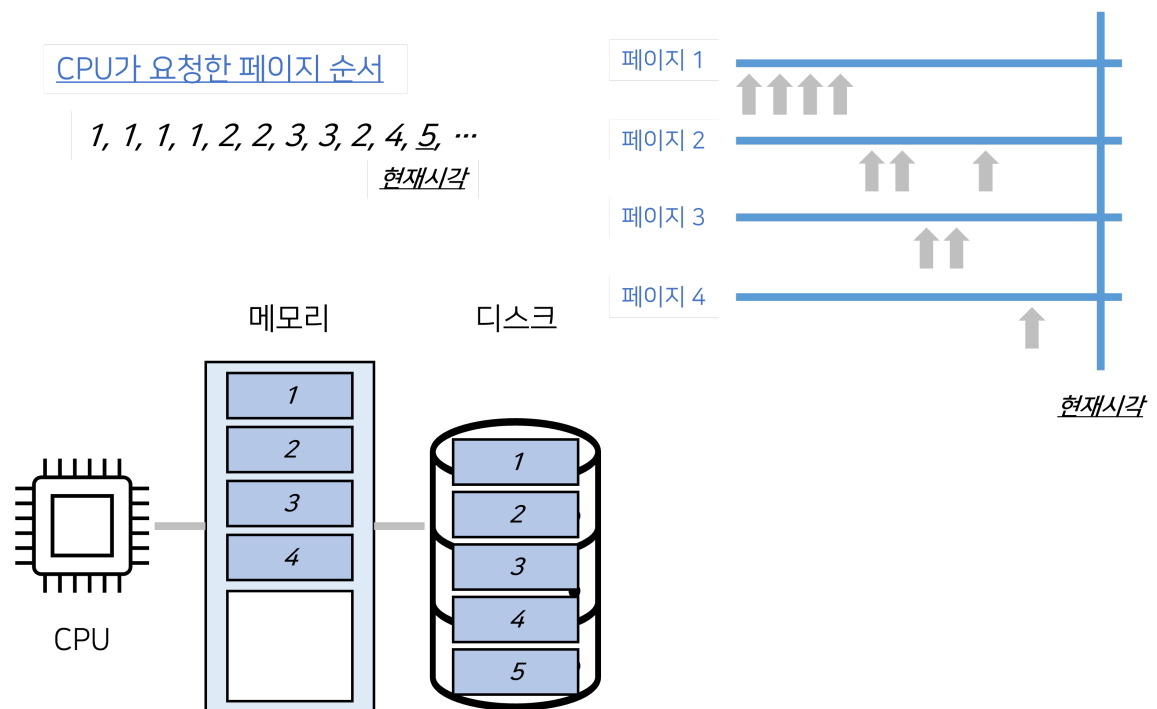
Process	금번 CPU 사용 시간
P1	53
P2	17
P3	68
P4	24



메모리 관리



- 가상메모리에서 당장 필요한 부분이 메모리로 옮겨간다.
- 나머지는 디스크(스왑영역)으로 간다. → 디스크(스왑영역)은 메모리의 연장공간
- 메모리 : 휘발성
- 디스크 (파일시스템) : 비휘발성
- 디스크 (스왑영역) : 비휘발성 (의미가 없음)



- CPU가 디스크에서 요청하고 메모리에 저장
- 중복 요청은 메모리에서 읽어 오기만 하면됨

Q. 하지만, 메모리가 가득 찼다면?

A. 미래를 모르는 상황에서 다시 사용될 가능성이 높은 것은 유지하는 것

- **LRU** (가장 오래전에 참조 페이지를 삭제) : 1번
- **LFU** (참조 횟수가 가장 적은 페이지 삭제) : 4번

| 디스크 스케줄링

- 사용할 데이터가 디스크상의 여러 곳에 저장되어 있을 경우 데이터를 액세스 하기 위해서 디스크 헤드가 움직이는 경로를 결정하는 기법

- **디스크 접근 시간 Access time의 구성**

- **탐색 시간 (Seek time)**

- 헤드를 해당 트랙(실린더)으로 움직이는 데 걸리는 시간
- 가장 많은 시간 소요

- **회전 지연 (Rotational latency)**

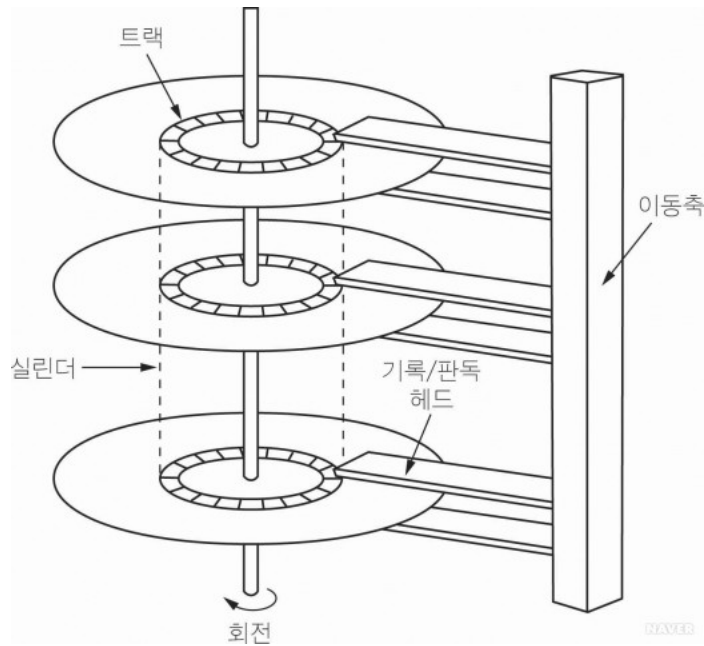
- 헤드가 원하는 섹터에 도달하기 까지 걸리는 시간

- **전송시간 (Transfer time)**

- 실제 데이터 전송 시간

- Seek time을 최소화 하는 것이 목표

- $\text{Seek time} = \text{Seek distance}$



1. FCFS (First-Come First-Served)

- 먼저 온 것부터 처리
- 총 헤드의 이동 : 640 cylinders
- Queue = 98, 183, 37, 122, 14, 124, 65, 67
- Head starts at 53

53 → 98 → 183 → 37 → 122 → 14 → 124 → 65 → 67

2. SSTF (Shortest Seek Time First)

- 제일 가까운 것을 찾아감
- Starvation 문제
- 총 헤드의 이동 : 236 cylinders
- Queue = 98, 183, 37, 122, 14, 124, 65, 67
- Head starts at 53

53 → 65 → 67 → 37 → 14 → 98 → 122 → 124 → 183

3. SCAN

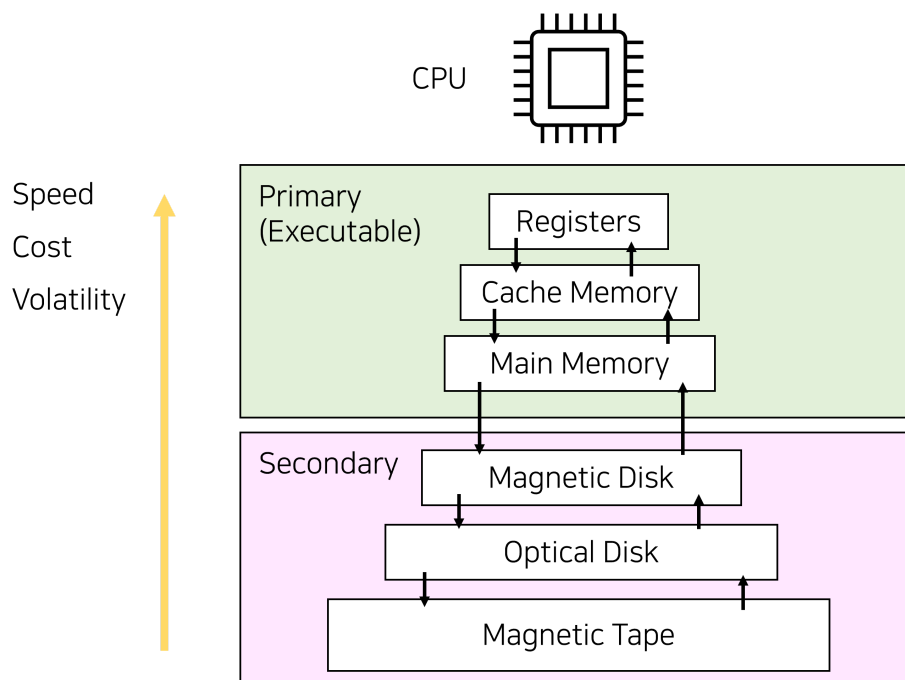
- 헤드가 디스크의 한쪽 끝에서 다른쪽 끝까지 이동하며 가는 길목에 있는 모든 요청을 처리
- 다른 한쪽 끝에 도달하면 역방향으로 이동하며 오는 길목에 있는 모든 요청을 처리하며 다시 반대쪽 끝으로 이동한다.
- 총 헤드의 이동 : 208 cylinders
- Queue = 98, 183, 37, 122, 14, 124, 65, 67
- Head starts at 53

53 → 37 → 14 → 0 → 65 → 67 → 98 → 122 → 124 → 183

- **C-SCAN**

- 항상 한쪽 방향으로만 요청된 서비스를 지원해주는 기법

저장장치 계층 구조와 캐싱(caching)



- **Caching** : copying information into faster storage system

플래시 메모리

- 반도체장치 (하드디스크 : 마그네틱)
- NAND형(스토리지), NOR형(임베디드 코드 저장용)
- **특징**
 - Nonvolatile
 - Low Power consumption
 - Shock resistance
 - Small size
 - Lightweight
 - 쓰기 횟수 제약
- **사용형태**
 - 휴대폰, PDA등 임베디드 시스템 구성용
 - USB용 메모리 스틱
 - 디지털카메라 등의 SD카드, CompactFlash, Smart Media Card
 - 모바일 장치 뿐 아니라 대용량 시스템에서 SSD (Solid state Drive)란 이름으로 하드디스크 대체 시도
- **장단점**
 - 크기가 작고 가벼움
 - 물리적인 충격에 강함
 - 데이터가 시간 흐름에 따라 변질될 위험이 있음 → 전하가 조금씩 빠져나감
- 하드웨어적인 부분은 소프트웨어가 보완해야 한다.