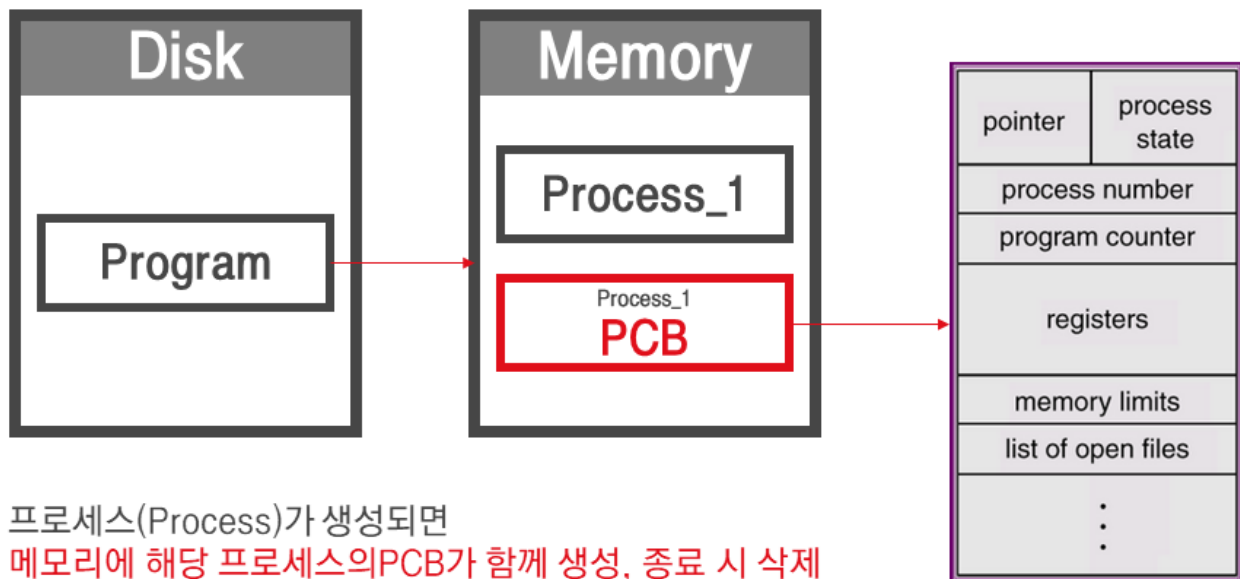


스케줄링

PCB (Process Control Block)

PCB(프로세스 제어 블록)는 운영체제에서 프로세스를 관리하기 위해 해당 프로세스의 상태 정보를 담고 있는 자료구조를 말한다.

프로세스를 컨텍스트 스위칭 할때 기존 프로세스의 상태를 어딘가에 저장해 뒀야 다음에 똑같은 작업을 이어서 할 수 있을 것이고, 새로 해야 할 작업의 상태 또한 알아야 어디서부터 다시 작업을 시작할지 결정할 수 있을 것이다. 즉, PCB는 프로세스 스케줄링을 위해 프로세스에 관한 모든 정보 저장하는 **임시 저장소**인 것이다.



프로세스(Process)가 생성되면
메모리에 해당 프로세스의PCB가 함께 생성, 종료 시 삭제

따라서 운영체제는 PCB에 담긴 프로세스 고유 정보를 통해 프로세스를 관리하며, 프로세스의 실행 상태를 파악하고, 우선순위를 조정하며, 스케줄링을 수행하고, 다른 프로세스와의 동기화를 제어한다.

운영체제에 따라 PCB에 포함되는 항목이 다를 수 있지만 일반적으로 PCB내에는 다음과 같은 정보가 포함되어 있다.

- 포인터 (Pointer) : 프로세스의 현재 위치를 저장하는 포인터 정보
- 프로세스 상태 (Process state) : 프로세스의 각 상태 - 생성(New), 준비(Ready), 실행(Running), 대기(Waiting), 종료(Terminated) 를 저장
- 프로세스 아이디 (Process ID, PID) : 프로세스 식별자를 지정하는 고유한 ID
- 프로그램 카운터 (Program counter) : 프로세스를 위해 실행될 다음 명령어의 주소를 포함하는 카운터를 저장
- 레지스터 (Register) : 누산기, 베이스, 레지스터 및 범용 레지스터를 포함하는 CPU 레지스터에 있는 정보
- 메모리 제한 (Memory Limits) : 운영 체제에서 사용하는 메모리 관리 시스템에 대한 정보
- 열린 파일 목록 (List of open file) : 프로세스를 위해 열린 파일 목록

스케줄링

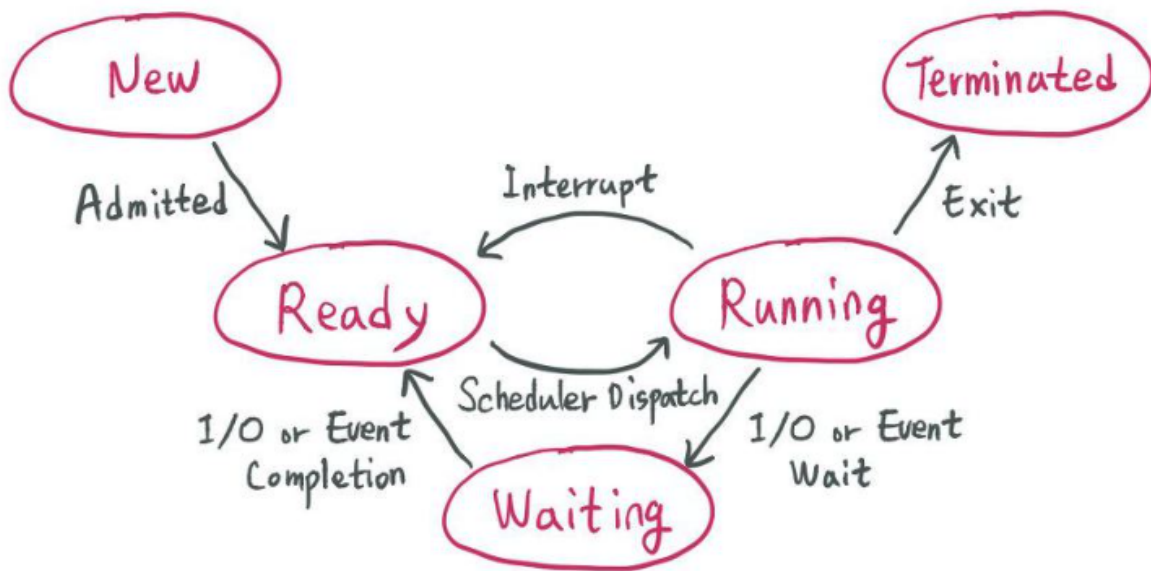
| CPU 를 잘 사용하기 위해 프로세스를 잘 배정하기

- 조건 : 오버헤드 ↓ / 사용률 ↑ / 기아 현상 ↓
- 목표
 1. **Batch System** : 가능하면 많은 일을 수행. 시간(time) 보단 처리량(throughout)이 중요
 2. **Interactive System** : 빠른 응답 시간. 적은 대기 시간.
 3. **Real-time System** : 기한(deadline) 맞추기.

2. 선점 / 비선점 스케줄링

- 선점 (preemptive) : OS가 CPU의 사용권을 선점할 수 있는 경우, 강제 회수하는 경우 (처리시간 예측 어려움)
- 비선점 (nonpreemptive) : 프로세스 종료 or I/O 등의 이벤트가 있을 때까지 실행 보장 (처리시간 예측 용이함)

3. 프로세스 상태



- 선점 스케줄링 : Interrupt , I/O or Event Completion , I/O or Event Wait , Exit
- 비선점 스케줄링 : I/O or Event Wait , Exit

프로세스의 상태 전이

- ✓ **승인 (Admitted)** : 프로세스 생성이 가능하여 승인됨.
- ✓ **스케줄러 디스패치 (Scheduler Dispatch)** : 준비 상태에 있는 프로세스 중 하나를 선택하여 실행시키는 것.
- ✓ **인터럽트 (Interrupt)** : 예외, 입출력, 이벤트 등이 발생하여 현재 실행 중인 프로세스를 준비 상태로 바꾸고, 해당 작업을 먼저 처리하는 것.
- ✓ **입출력 또는 이벤트 대기 (I/O or Event wait)** : 실행 중인 프로세스가 입출력이나 이벤트를 처리해야 하는 경우, 입출력/이벤트가 모두 끝날 때까지 대기 상태로 만드는 것.
- ✓ **입출력 또는 이벤트 완료 (I/O or Event Completion)** : 입출력/이벤트가 끝난 프로세스를 준비 상태로 전환하여 스케줄러에 의해 선택될 수 있도록 만드는 것.

4. CPU 스케줄링의 종류

- 비선점 스케줄링
 1. FCFS (First Come First Served)

- 큐에 도착한 순서대로 CPU 할당
- 실행 시간이 짧은 게 뒤로 가면 평균 대기 시간이 길어짐

2. SJF (Shortest Job First)

- 수행시간이 가장 짧다고 판단되는 작업을 먼저 수행
- FCFS 보다 평균 대기 시간 감소, 짧은 작업에 유리

3. HRN (Hightest Response-ratio Next)

- 우선순위를 계산하여 점유 불평등을 보완한 방법(SJF의 단점 보완)
- 우선순위 = (대기시간 + 실행시간) / (실행시간)

• 선점 스케줄링

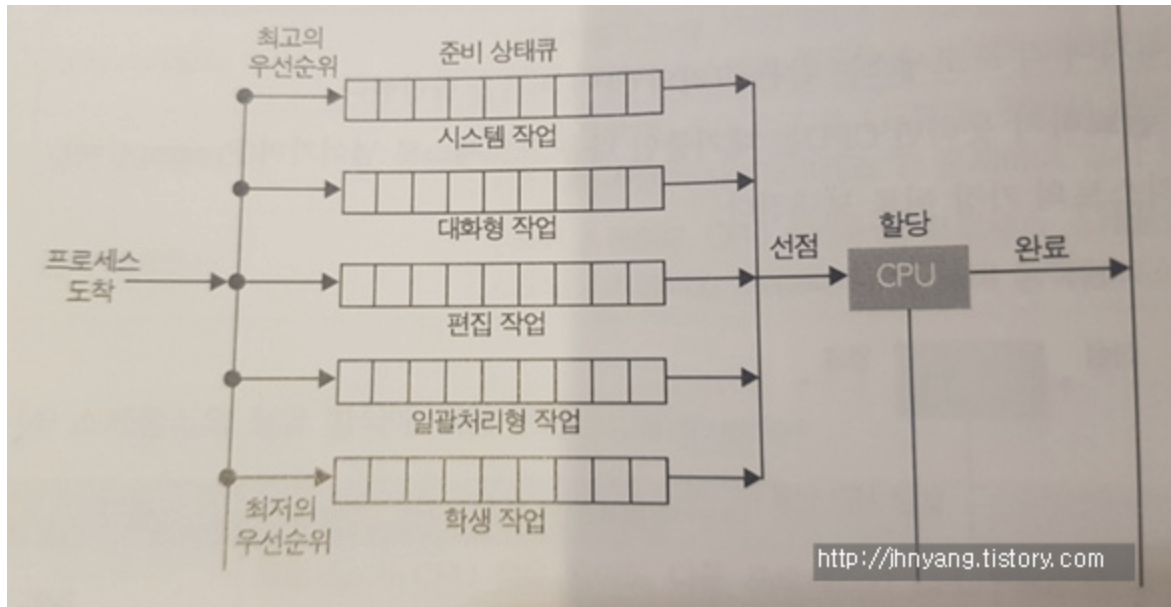
1. Priority Scheduling

- 정적/동적으로 우선순위를 부여하여 우선순위가 높은 순서대로 처리
- 우선 순위가 낮은 프로세스가 무한정 기다리는 Starvation 이 생길 수 있음
- Aging 방법으로 Starvation 문제 해결 가능

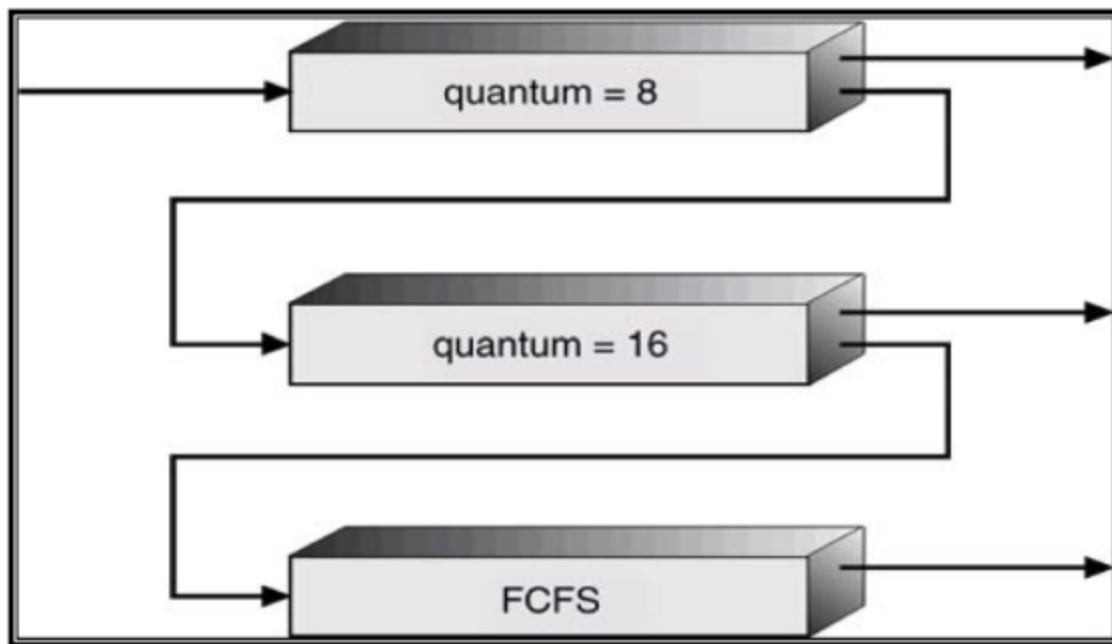
2. Round Robin

- FCFS에 의해 프로세스들이 보내지면 각 프로세스는 동일한 시간의 **Time Quantum** 만큼 CPU를 할당 받음
 - **Time Quantum** or **Time Slice** : 실행의 최소 단위 시간
- 할당 시간(**Time Quantum**)이 크면 FCFS와 같게 되고, 작으면 문맥 교환 (Context Switching) 잦아져서 오버헤드 증가

3. Multilevel-Queue (다단계 큐)

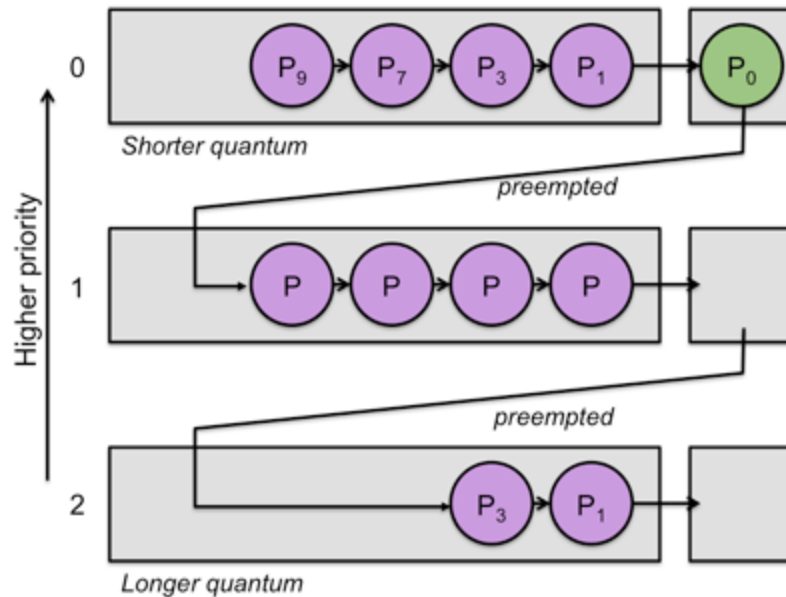


작업들을 여러 종류의 그룹으로 나누어 여러 개의 큐를 이용하는 기법



1. 우선순위가 낮은 큐들이 실행 못하는 걸 방지하고자 각 큐마다 다른 **Time Quantum** 을 설정 해주는 방식 사용우선순위가 높은 큐는 작은 **Time Quantum** 할당. 우선순위가 낮은 큐는 큰 **Time Quantum** 할당.

4. Multilevel-Feedback-Queue (다단계 피드백 큐)



- 다단계 큐에서 자신의 **Time Quantum** 을 다 채운 프로세스는 밑으로 내려가고 자신의 **Time Quantum** 을 다 채우지 못한 프로세스는 원래 큐 그대로 Time Quantum 을 다 채운 프로세스는 CPU burst 프로세스로 판단하기 때문 짧은 작업에 유리, 입출력 위주 (Interrupt가 잦은) 작업에 우선권을 줌 처리 시간이 짧은 프로세스를 먼저 처리하기 때문에 Turnaround 평균 시간을 줄여줌

5. CPU 스케줄링 척도

1. Response Time

- 작업이 처음 실행되기까지 걸린 시간

2. Turnaround Time

- 실행 시간과 대기 시간을 모두 합한 시간으로 작업이 완료될 때 까지 걸린 시간