스케줄링

스케줄링

스케줄링

- 멀티프로그래밍 운영체제에서 cpu스케줄링은 아주 중요함
- 멀티프로그래밍의 목표는 여러가지 프로세스가 동시에 실행되기 하여 cpu 활용성을 최대로 하는 것
- I/O대기할 때, 일 시켜서 CPU 항상 바쁘게 만듬

스케줄링

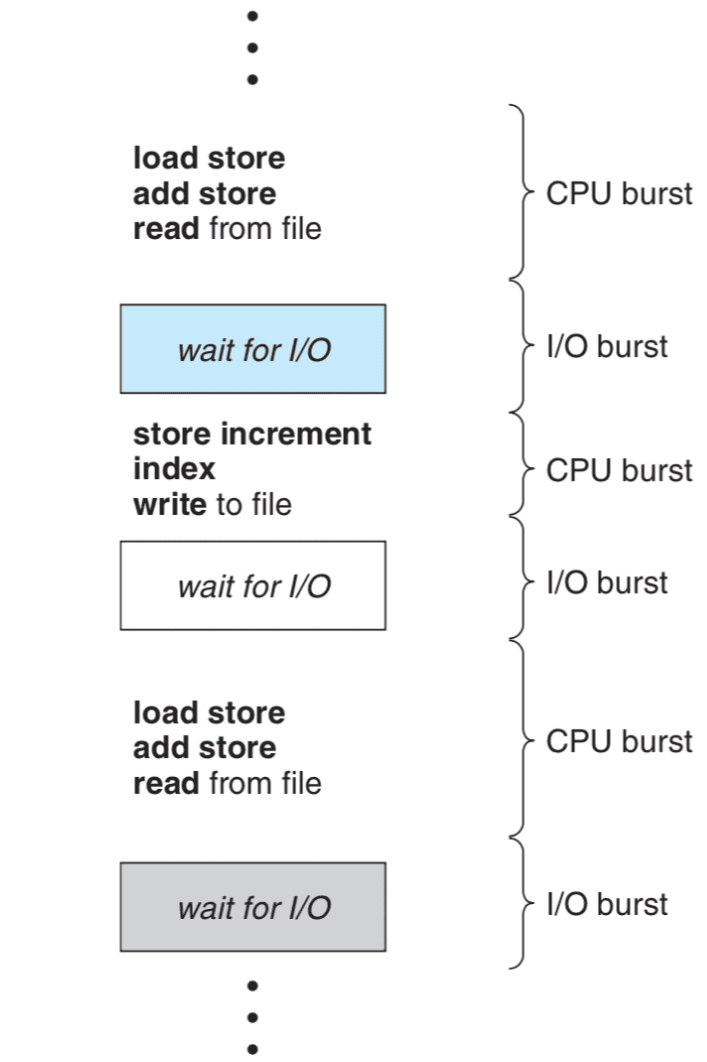


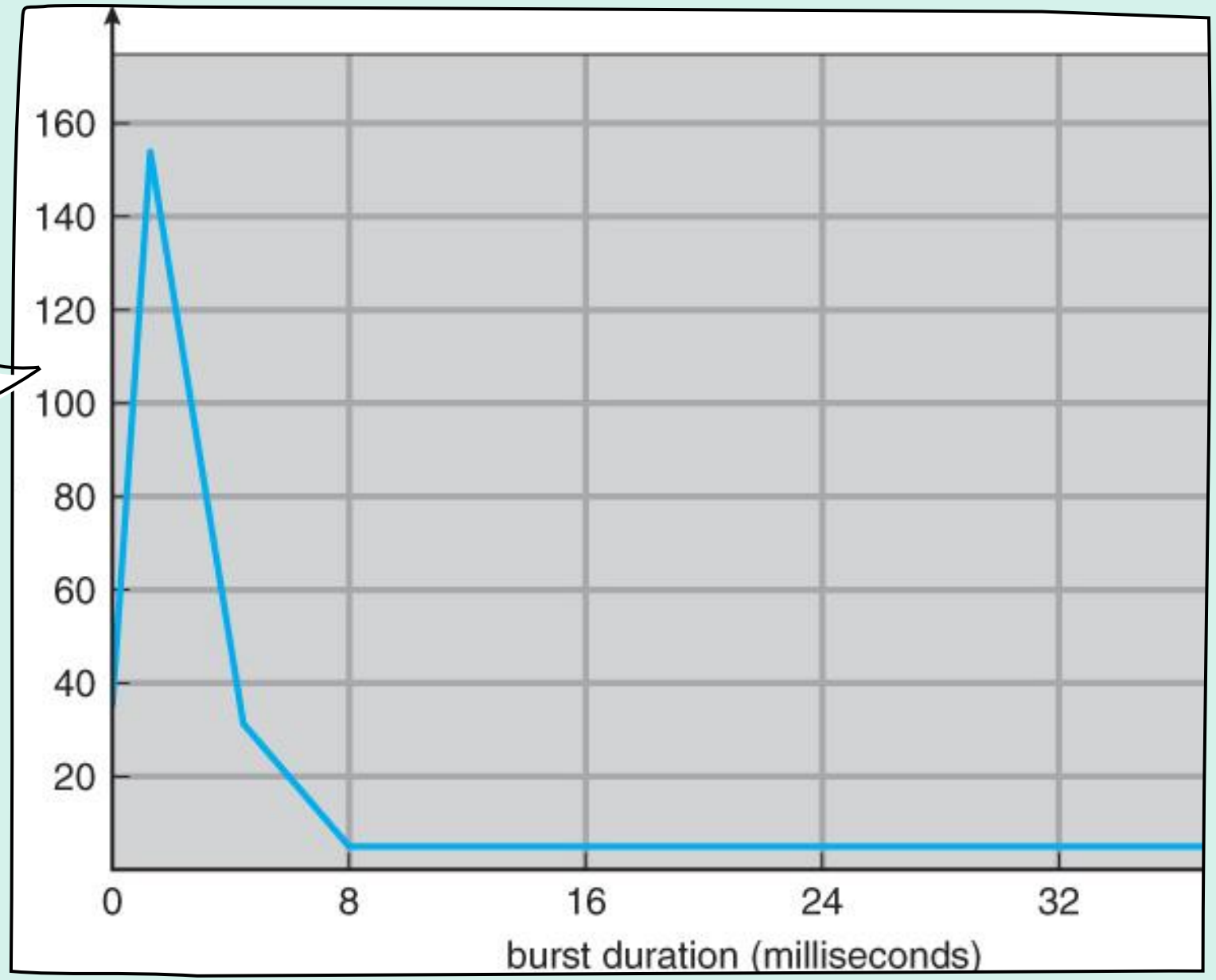
Figure 6.1 Alternating sequence of CPU and I/O bursts.

CPU-I/O Burst Cycle

CPU-I/O Burst Cycle

- 프로세스 실행 시간 = cpu 실행 + I/O 대기
- I/O Burst 프로세스 : I/O작업이 많은 프로세스, 짧은 다수의 cpu burst
- CPU Burst 프로세스 : CPU 작업이 많은 프로세스, 긴 소수의 cpu burst

CPU-I/O Burst Cycle



대부분 cpu burst가 짧음,
cpu burst가 긴 작업은 별로 적음

CPU 스케줄러

CPU 스케줄러

- Cpu가 아이들일 때 수행될 프로세스를 고르는 건 short-time 스케줄러(cpu 스케줄러)
- 스위칭이 일어나는 순간
  1. 프로세스 상태가 실행 -> 대기 될 때(I/O요청), **자발적**으로 cpu 반납
  2. 러닝 -> 준비 상태(인터럽트 발생), **타의적** cpu 반납
  3. 대기 -> 준비 상태(I/O작업 완료), **타의적** cpu 반납
  4. 프로세스가 종료, **자발적** cpu 반납
- 1, 4 번 : nonpreemtive 스케줄링
- 2, 3 번 : preemtive 스케줄링, 타이머 같은 특정H/W 필요, 공유 데이터문제 있음

디스패처

디스패처

- 디스패처의 역할
  1. 컨텍스트 스위칭 수행
  2. 유저모드로 스위칭 수행
  3. 프로그램 재시작 할 때 적절한 위치로 점프
- cpu스케줄러에 의해 선택된 프로세스에게 cpu사용권을 부여
- 디스패처 레이턴시 : 프로세스 정리 후 다른 프로세르 실행까지 걸리는 시간

스케줄링 기준

스케줄링 기준

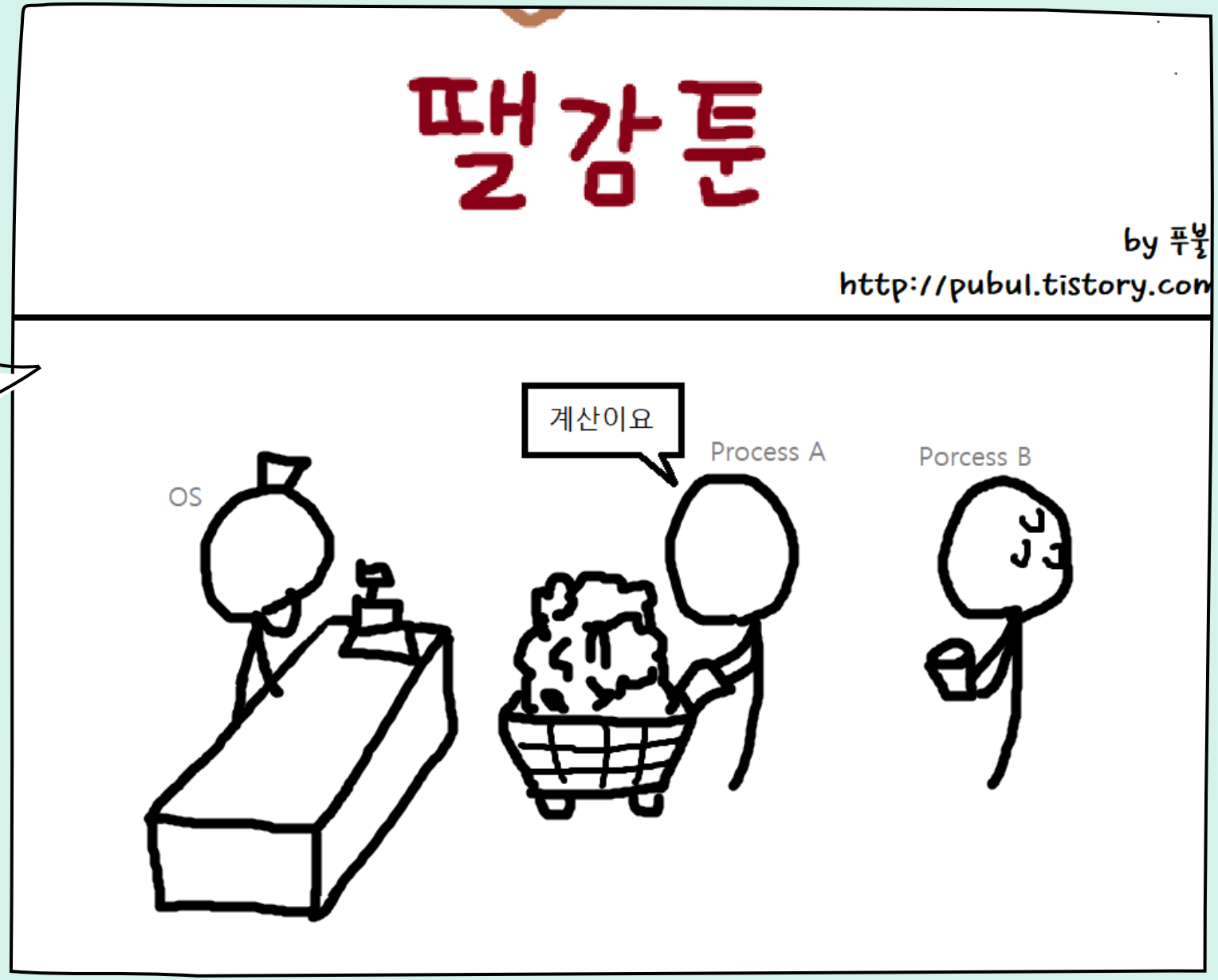
1. Cpu utilization
2. Throughput
3. Turnaround time (프로세스가 시작되어서 종료할 때 까지 걸리는 시간)
4. Waiting time
5. 레디큐에서 기다리는 시간이 클수록 우선
6. Response time

※ 평균값 보다는 최대/최소값 편차를 줄이는게 좋음(응답시간의 다양성 줄이기 > 응답시간 평균 줄이기)

스케줄링 종류

- FCFS
- SJF
- Priority 스케줄링
- 라운드 로빈
- 멀티레벨 큐 스케줄링
- 멀티레벨 피드백 큐 스케줄링

FCFS



- First come, first serve
- Convoy effect 발생 가능

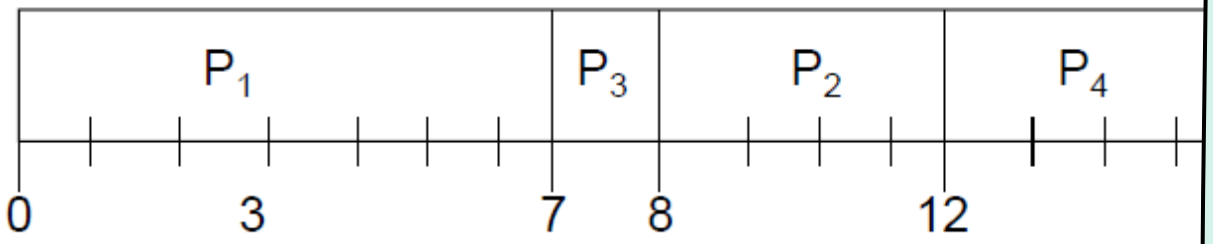
SJF

- 수행시간이 짧은 작업부터 처리, 프로세스의 평균 대기시간 감소
- 프로세스가 얼마나 걸릴지 모르기 때문에 실제로는 구현 불가능
- 사용자가 값을 명시하던가 해야함

Example of Non-Preemptive SJF

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

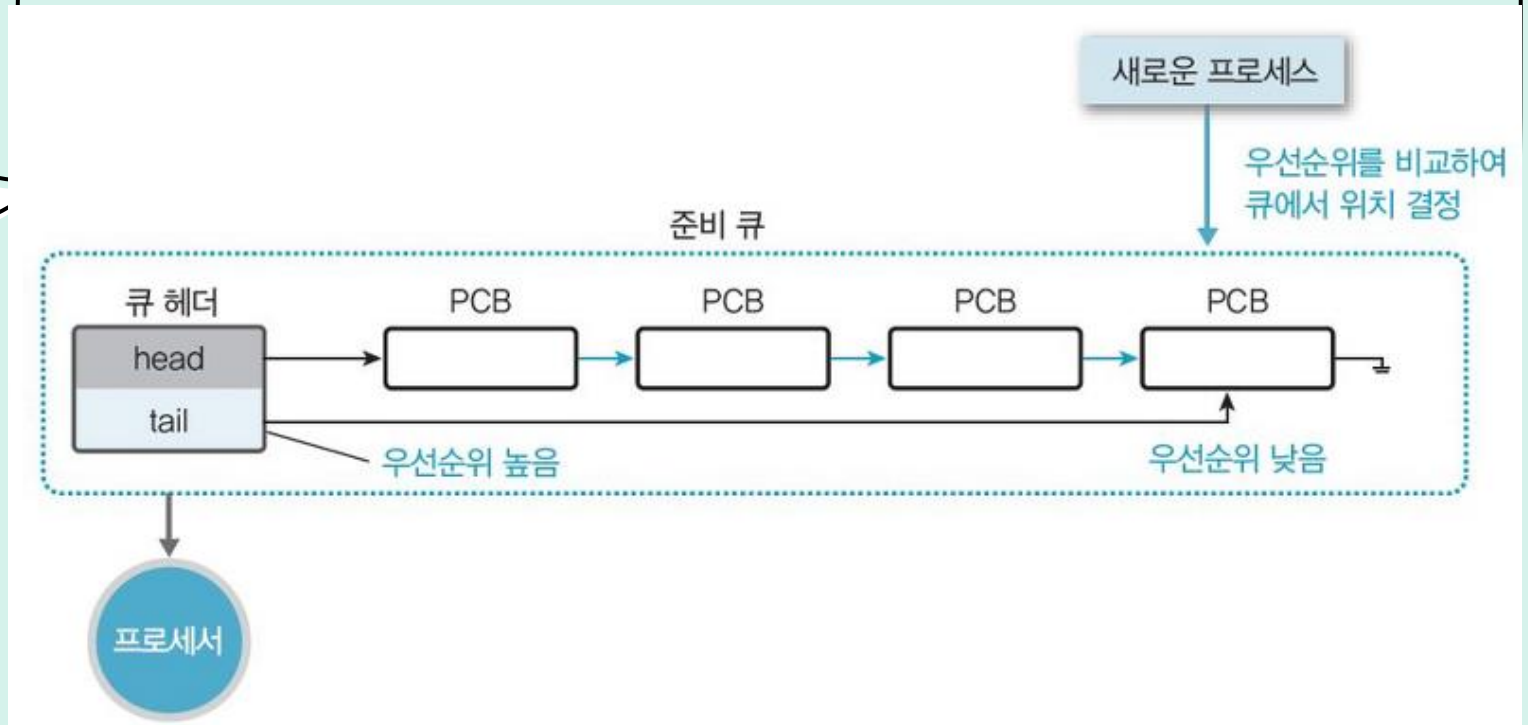
- SJF (non-preemptive)



- Average waiting time = (0 + 6 + 3 + 7)/4 = 4

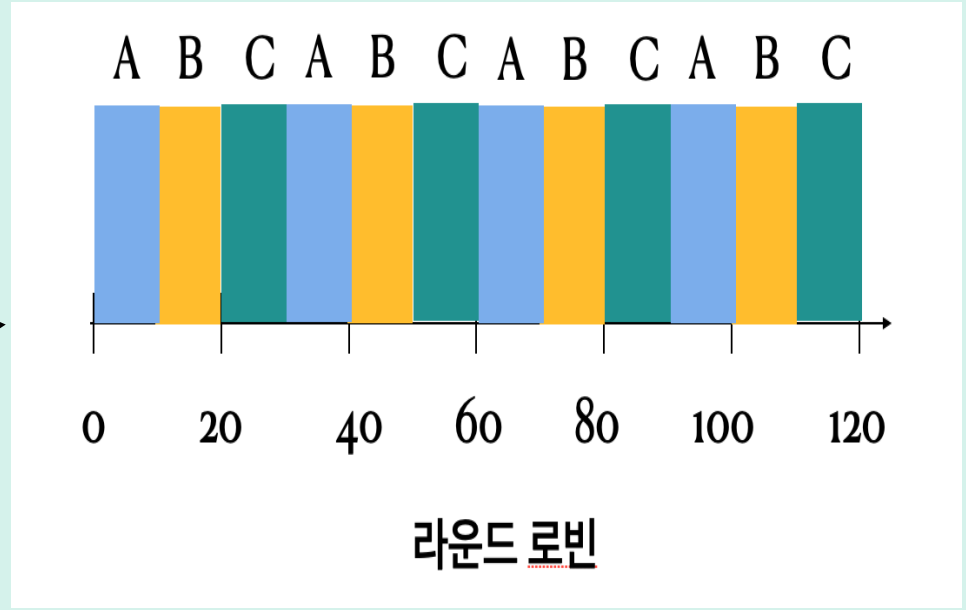
Priority 스케줄링

- 우선순위에 따라 스케줄링, 우선순위가 같으면 FCFS
- Starvation 문제 있음, 우선순위 높은 애들이 계속 오면 우선순위 낮은 애들은 실행이 안됨
- Aging으로 starvation 해결, 시간이 지날수록 우선순위 낮은 프로세스의 우선순위를 높여줌



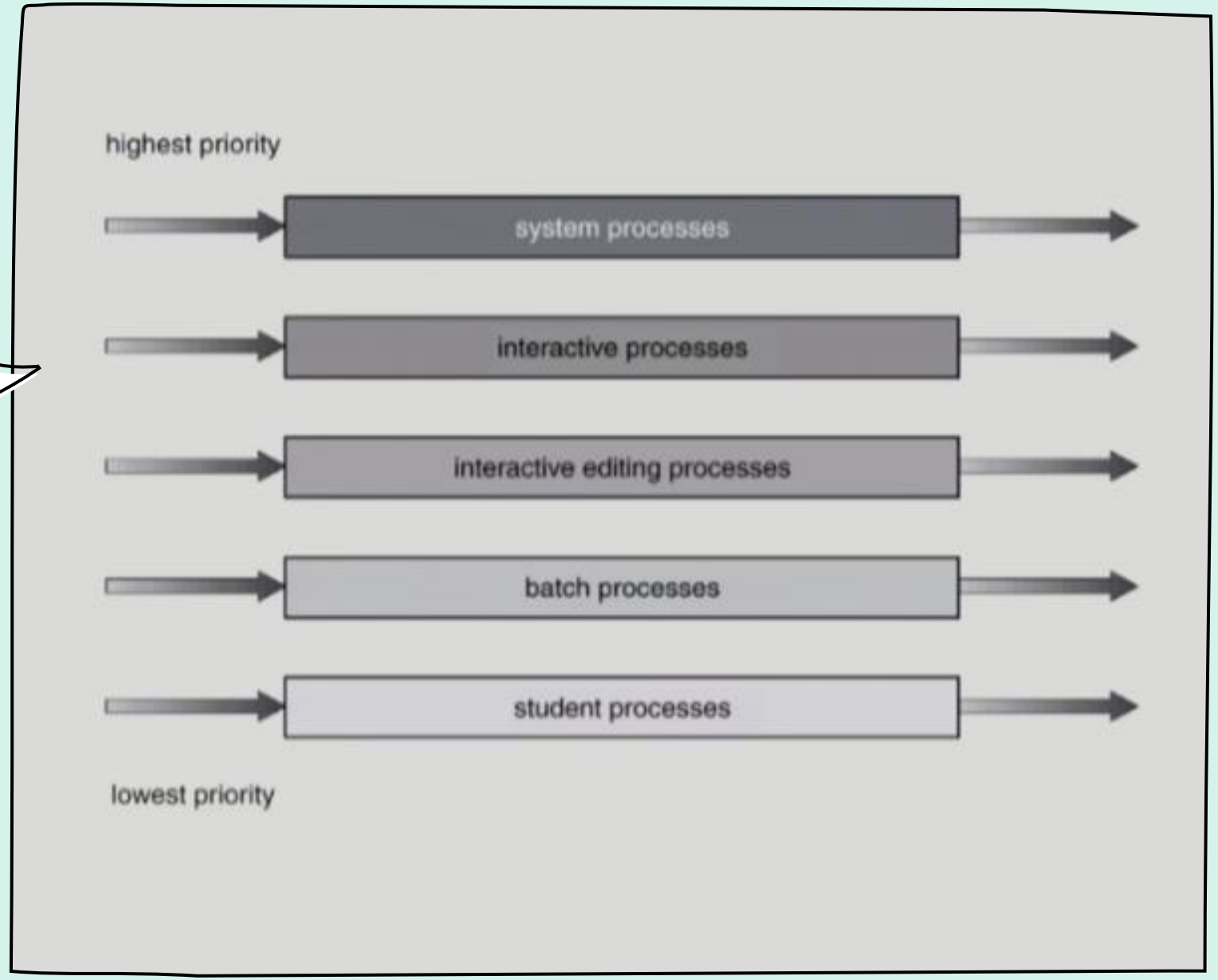
라운드 로빈

- 각 프로세스마다 동일한 시간할당을 가짐
- 문맥전화 오버헤드 증가, 응답시간 감소
- 할당시간 증가 -> FCFS 처럼 동작
- 할당시간 감소 -> 문맥전환만 하다가 시간 다 보냄
- 최적화된 할당시간을 찾는 것이 매우매우매우 중요
- 보통 cpu burst time의 80%보다 길게 설정



멀티레벨 큐 스케줄링

- 프로세스 특징에 따라 큐가 정해짐
- 큐마다 우선순위 또는 time slice 존재
- 큐마다 각각 다른 스케줄링 알고리즘



멀티레벨 피드백 큐 스케줄링

- 큐마다 우선순위 존재
- 프로세스는 큐 종류에 상관없이 들어갈 수 있음
- 만약 cpu를 너무 많이 쓰는 프로세스가 있다면 우선순위가 낮은 큐로 보내서 조금만 실행되게 할 필요가 있음

ilevel Feedback Queues