[Pre-lecture Assignment#4]

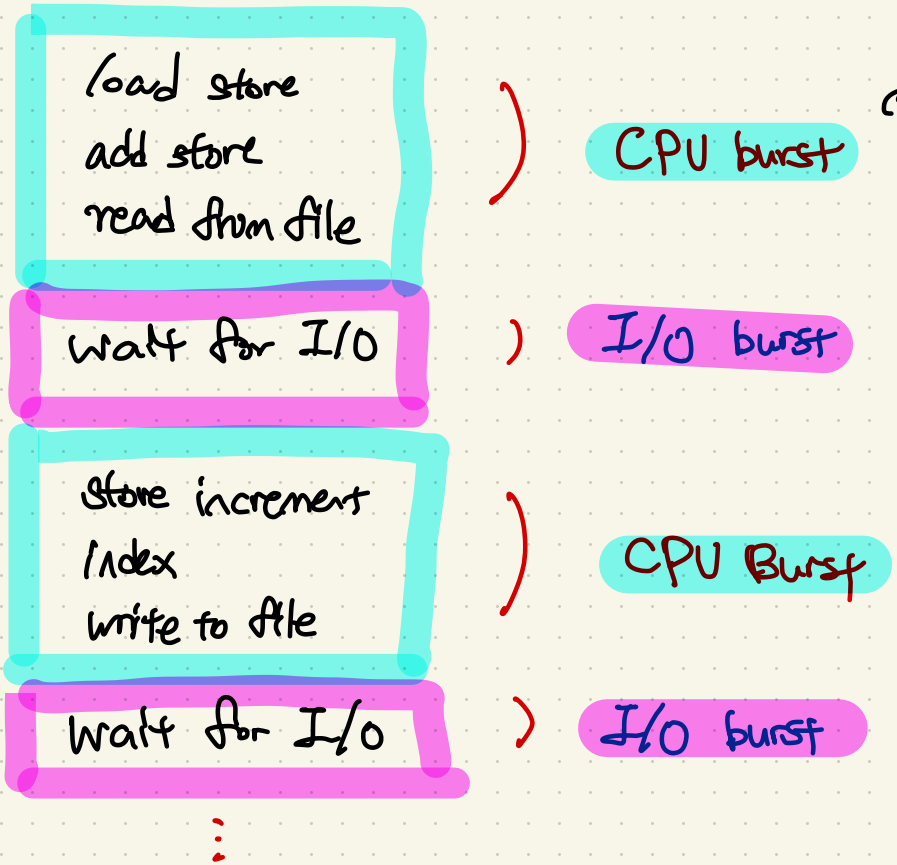
- Reading 1: Read Sections 5.1.2 CPU Scheduler, 5.1.3 Preemptive and Nonpreemptive Scheduling, 5.1.4 Dispatcher (page 201-204). Note that reading assignments may appear in next week discussion or quiz.

- **Video Lecture 1** (Around 30 minutes): CPU Scheduling

https://core.ewha.ac.kr/publicview/C0101020170327151556728127

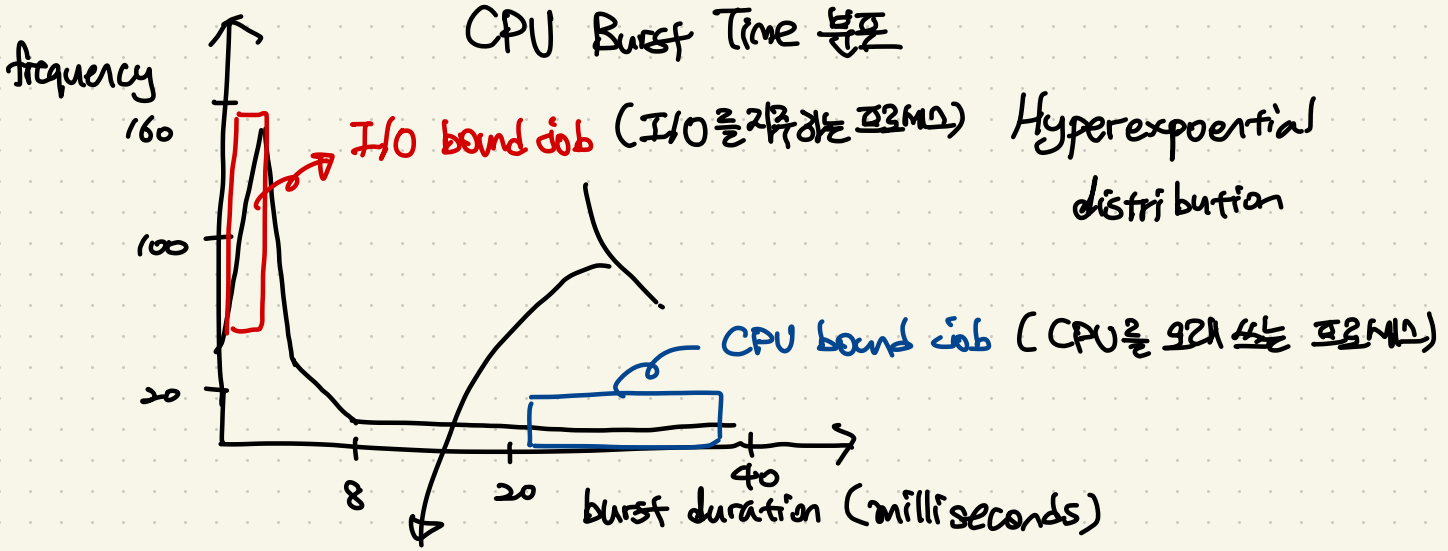
- **No** submissions. However, pre-lecture assignments, as well as slides, class discussions, programming assignments, and textbook contents covering last week's lecture may appear in your next quiz.

CPU and I/O Bursts in Program Execution



↗ 연산과정

1. CPU를 오래 쓰고 I/O를 하나요.
2. I/O를 중간중간 넣어서 CPU를 연속적으로 사용하는 시간이 짧아지나요?



두 job이 동시에 CPU를 요청했다면?? 무엇을 먼저 줘야할까?

↳ I/O bound job

왜? 잠깐만 주면 되니까! 그리고 CPU bound job는 CPU를 좀 늦게 줘도 별로 크게 달라지지 않는다. 그리고 CPU bound job을 우선시하면 I/O device도 같이 놀게 된다

- I/O-bound process

  many short CPU bursts.

- CPU-bound process

  few, very long CPU bursts

CPU burst

짧게, 자주

I/O에 시간이 더 많이 필요한 job

길게, 드물게

계산위주 job

CPU scheduler

- Ready 상태의 프로세스 중에서 어떤에 CPU에 줄 프로세스를 고른다.

Dispatcher

- CPU의 제어권을 CPU scheduler 에 의해 선택된 프로세스에게 넘긴다.
- 이를 "context switch" 라고 한다.

CPU Scheduling은 언제 필요한가?

1. Running → Blocked (I/O request)
2. Running → Ready (timer interrupt)
3. Blocked → Ready (I/O request complete)
4. Terminate

자진반납: nonpreemptive

강제로빼앗음: preemptive

Scheduling criteria – 성능 척도

CPU utilization (이용률)

→ CPU를 최대한 오래 사용하자 ↑

Throughput (처리량)

→ 일정 시간동안 CPU가 처리한 프로세스 숫자. ↑

Turnaround time (소요시간, 반환시간)

CPU를 사용한 시간 + CPU를 기다린 시간

→ CPU burst를 요청한 때부터 I/O burst를 하러 가기까지 걸린 시간

Waiting time (대기시간)

→ 프로세스가 Ready queue 에서 CPU를 사용하기 위해 기다릴 시간

Response time (응답시간)

→ CPU를 쓰려고 들어와서, CPU를 실제로 쓰기까지 걸린 시간