점유 - 대기 문제

→ 여러 방법 (동일 시간 분배, remain time 비교 등)

멈춰야하는 경우 존재한다.
(이때 해당 프로세스는 다른 task를 해야한다.)

CHAPTER 5
CPU SCHEDULING

Operating System Concepts

9. Basic Concepts & Scheduling Criteria

❖ Process Execution

- □ CPU execution
- □ I/O wait

메모리 프로그램 fetch 순간에도
context-switching

context switching
PCB 저장 → 복원 과정

•
•
•

load store
add store
read from file

•
•
•

I/o wait

CPU – burst
⇒ 부분부분에만 resource
많이 사용

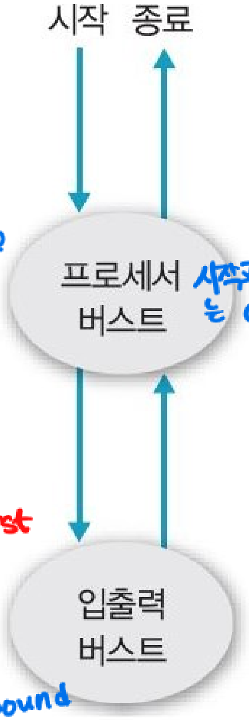
store increment
index
write to file

•
•
•

IO burst

load store
add store
read from file

•
•
•



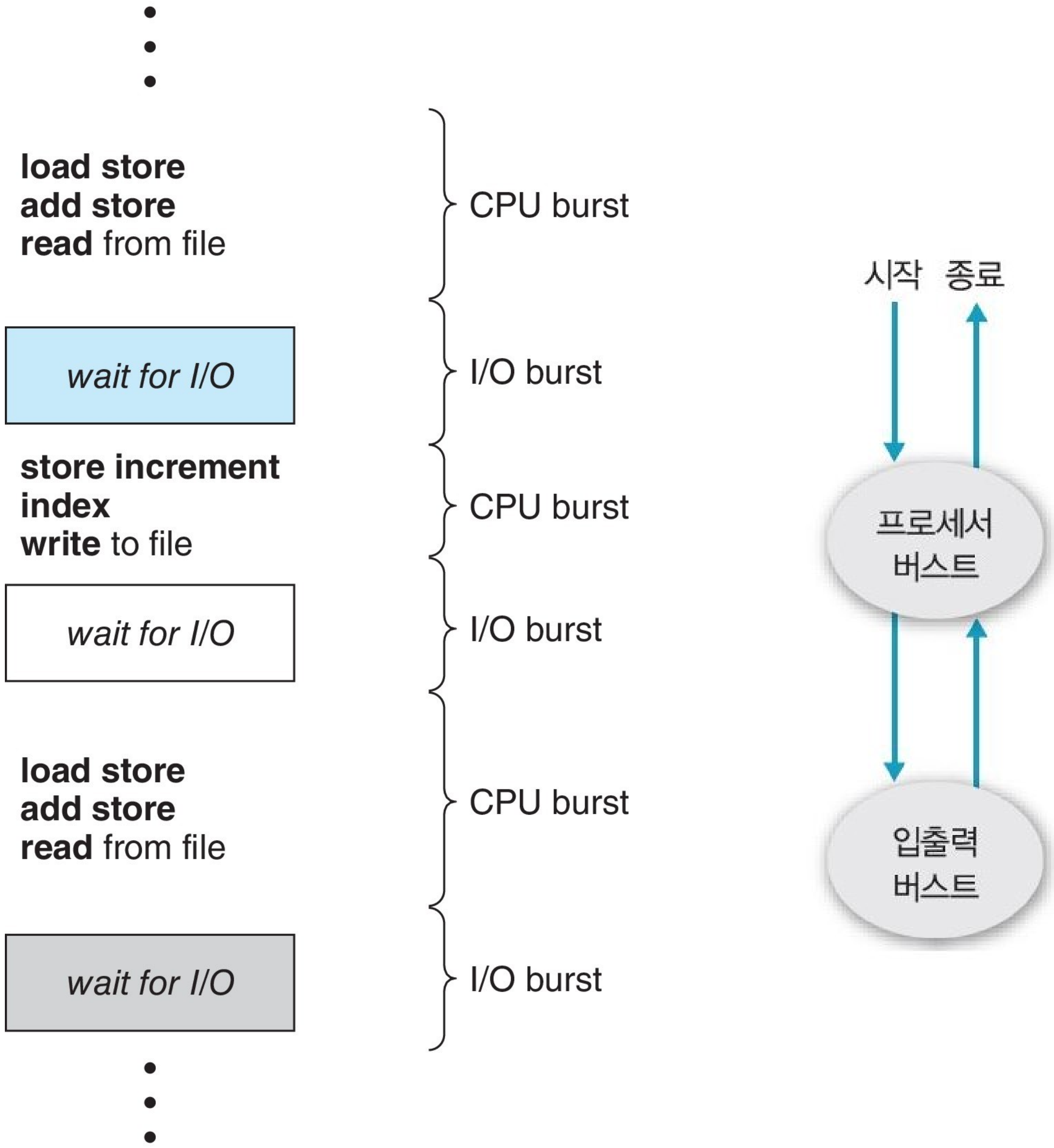
시작과 끝 end-point
는 CPU 버스트이다.

상황에 따라 사용되는 burst
중복횟수가 달라진다.
I/O bounded

– Process는 어느 영역에 bound
되어 있는지 고려해야한다.

Basic Concepts

- Maximum CPU utilization obtained with multiprogramming
- CPU–I/O Burst Cycle
  - Process execution consists of a **cycle** of CPU execution and I/O wait
- **CPU burst** followed by **I/O burst**
- CPU burst distribution is of main concern



Histogram of CPU-burst Times

- Large number of short bursts
- Small number of longer bursts

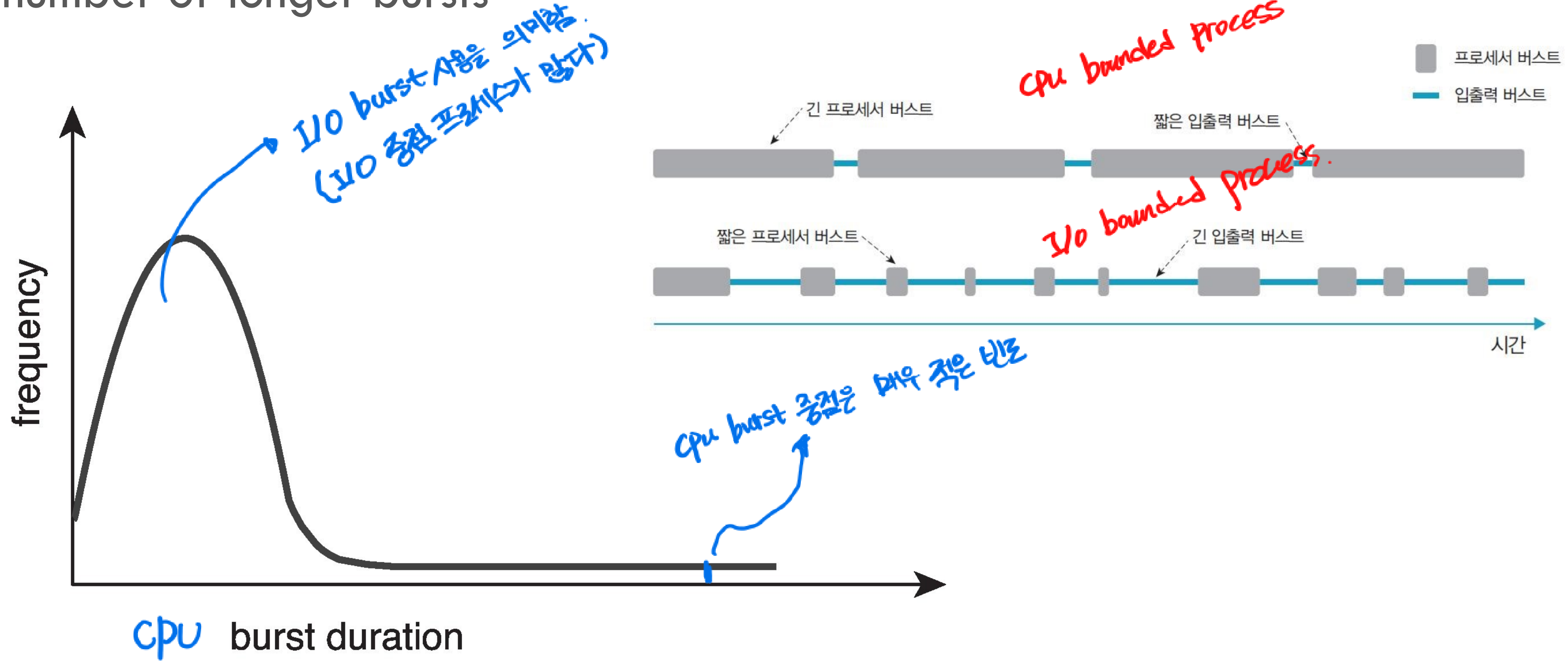


Figure 5.2 Histogram of CPU-burst durations.

CPU Scheduler

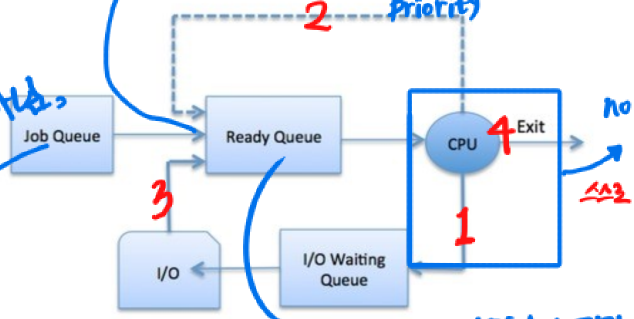
- The CPU scheduler selects from among the processes in ready queue, and allocates the a CPU core to one of them
  - □ Queue may be ordered in various ways
- CPU scheduling decisions may take place when a process:
  1. Switches from running to waiting state
  2. Switches from running to ready state
  3. Switches from waiting to ready
  4. Terminates
- Scheduling under 1 and 4 is nonpreemptive(cooperative)
- All other scheduling is preemptive – 현대의 대부분 스케줄러는 선점형 사용
  - □ Consider access to shared data(race condition) 동시접근 상황의 데이터 일관성 손상 상황. → 같은 input에 대해 다른 output 발생
  - □ Consider preemption while in kernel mode
  - □ Consider interrupts occurring during crucial OS activities

비슷한 경우

CPU ← Ready Queue

FIFO인 이유(queue 사용해서

SJF의 자료구조?

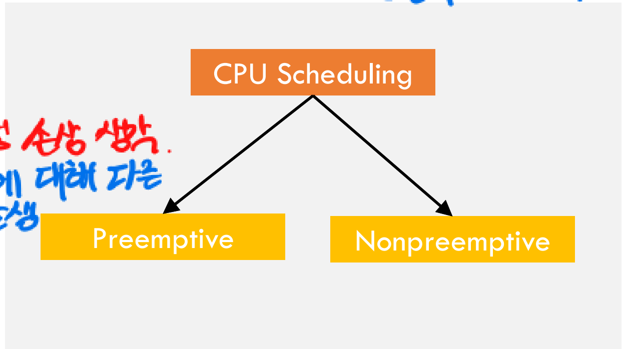


Memory에 적재된 것의 상태

CPU scheduler가 아님, (장기 스케줄러)

nonpreemptive 스스로 비켜나옴

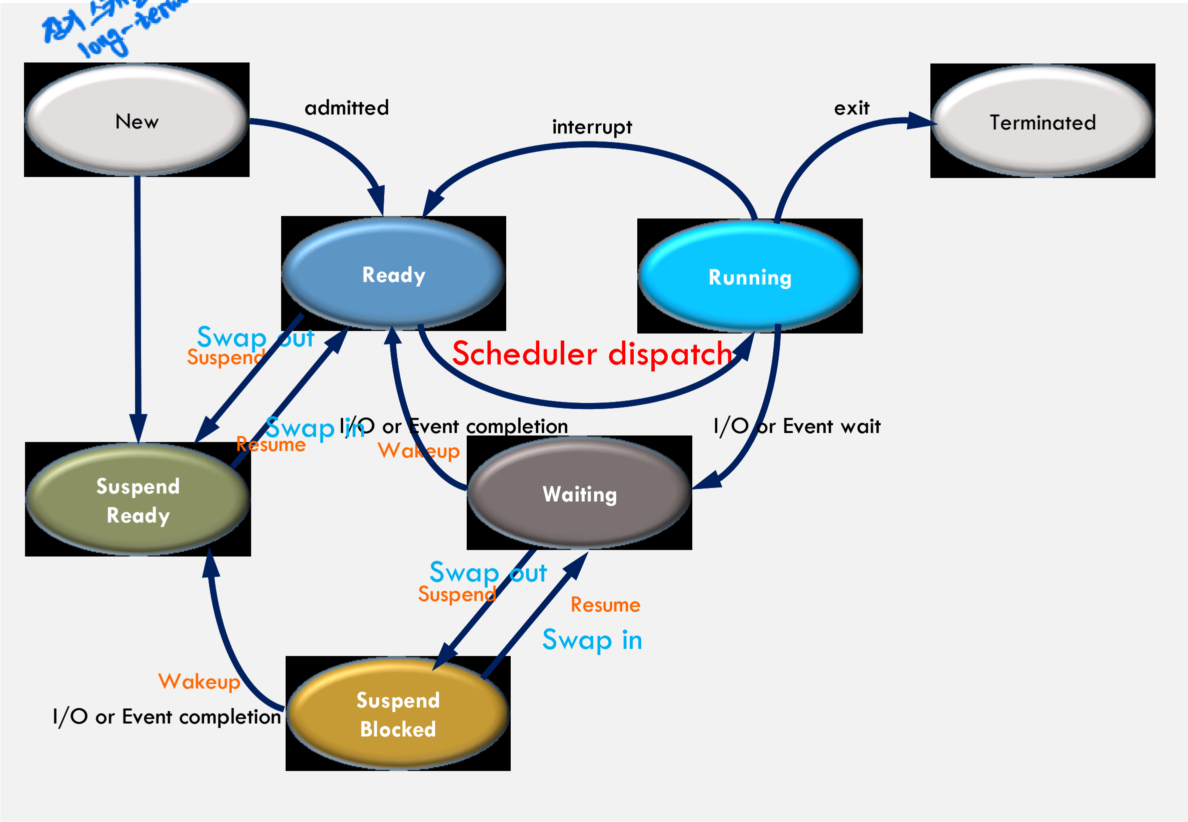
Swap 사용하여 가상 (중기 스케줄러)



Scheduler

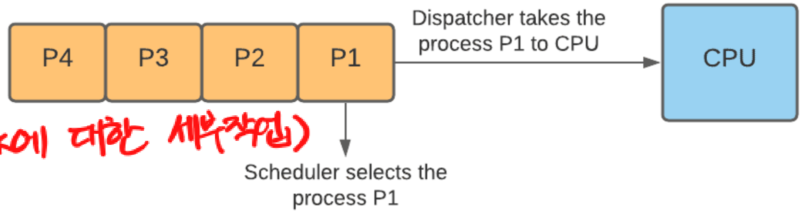
- Long-term Scheduler(Job scheduler)
  - 스케줄링에 따라 디스크에서 메모리로 작업 가져와 처리할 순서 결정(ready queue)
  - Process 수 제어
- Short-term Scheduler (CPU scheduler)
  - Process 의 CPU 할당
  - 메모리에 적재된 프로세스 중 프로세서를 할당하여 실행 상태가 되도록 결정하는 프로세스
- Medium-term Scheduler (Swapper)
  - Swap-in 과 swap-out 결정
  - Process 수 제어

장기 스케줄러
long-term scheduler



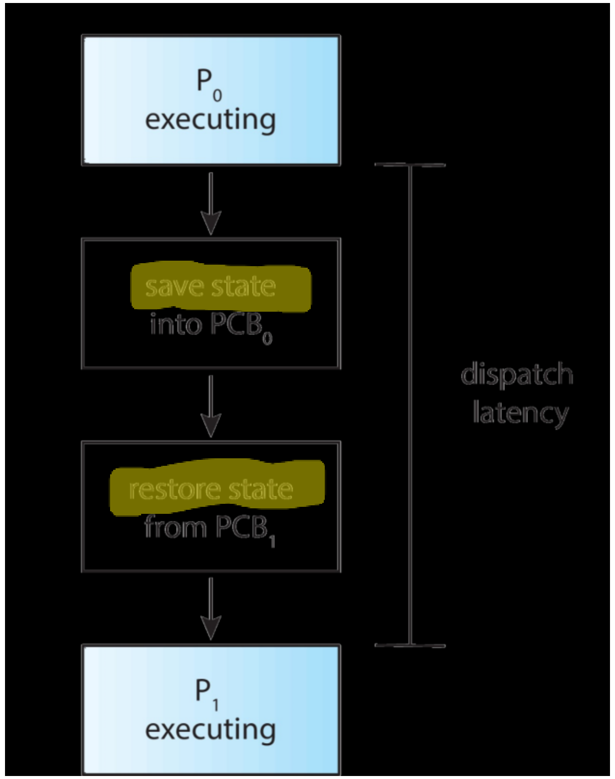
Dispatcher

- Dispatcher module gives control of the CPU to the process selected by the short-term scheduler; this involves: 구체적으로 행하는 모듈 (Scheduling 된 task에 대한 세부작업)
  - □ Switching context Privileged mode 으로 사용해야 하므로.
  - □ Switching to user mode
  - □ Jumping to the proper location in the user program to restart that program

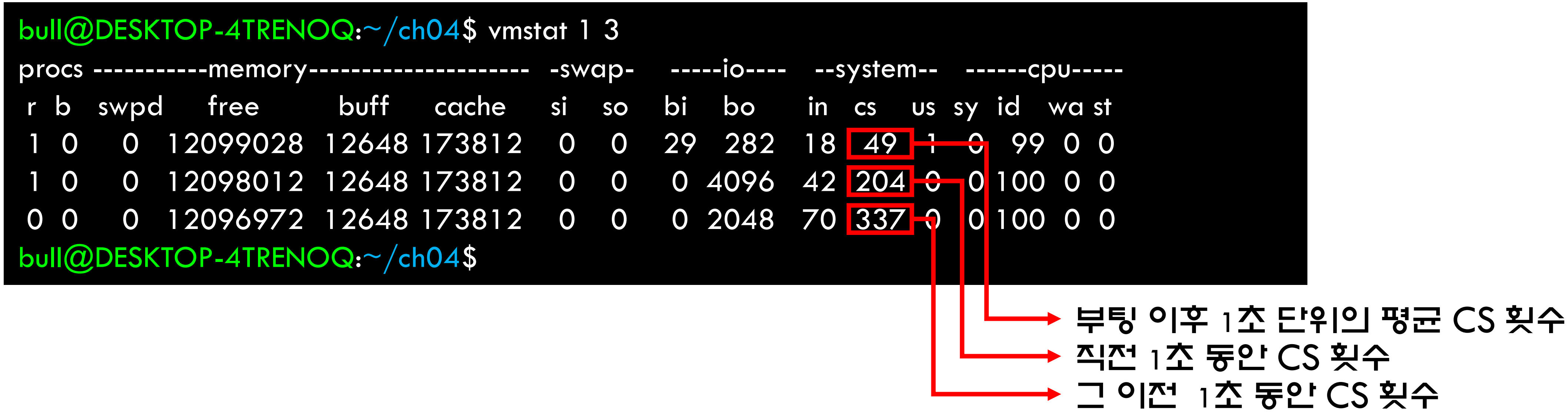

P4
P3
P2
P1
Dispatcher takes the process P1 to CPU
CPU
Scheduler selects the process P1

- Dispatch latency – time it takes for the dispatcher to stop one process and start another running

= context-switching 하는 지연 시간
PCB 값 저장 및 복원 과정.
→ 너무 잦은 경우



Dispatcher

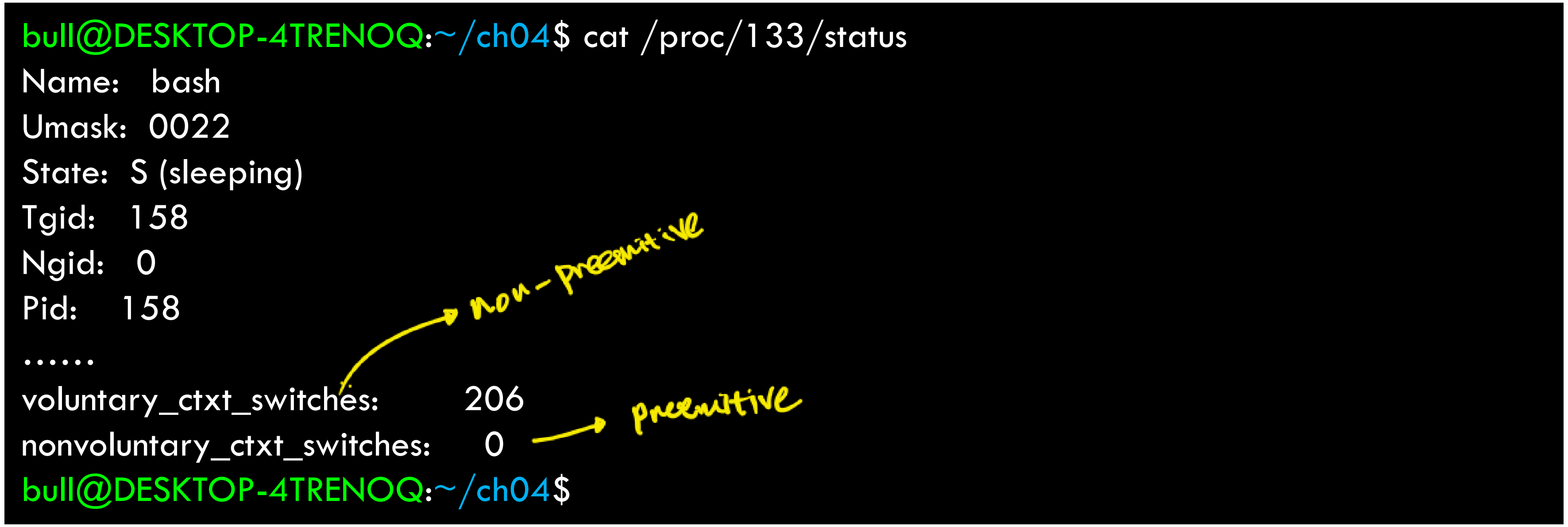


```
bull@DESKTOP-4TRENOQ:~/ch04$ vmstat 1 3
procs -----------memory---------------------- -swap-   -----io----  --system--   ------cpu-----
 r  b   swpd     free        buff    cache    si    so    bi    bo     in   cs    us  sy  id   wa st
 1  0      0  12099028  12648 173812   0     0    29   282    18   49    1   0   99   0  0
 1  0      0  12098012  12648 173812   0     0     0  4096    42  204    0   0  100   0  0
 0  0      0  12096972  12648 173812   0     0     0  2048    70  337    0   0  100   0  0
bull@DESKTOP-4TRENOQ:~/ch04$
```

부팅 이후 1초 단위의 평균 CS 횟수

직전 1초 동안 CS 횟수

그 이전 1초 동안 CS 횟수



```
bull@DESKTOP-4TRENOQ:~/ch04$ cat /proc/133/status
Name:   bash
Umask:  0022
State:  S (sleeping)
Tgid:   158
Ngid:   0
Pid:    158
……
voluntary_ctxt_switches:        206
nonvoluntary_ctxt_switches:     0
bull@DESKTOP-4TRENOQ:~/ch04$
```

non-preemitive

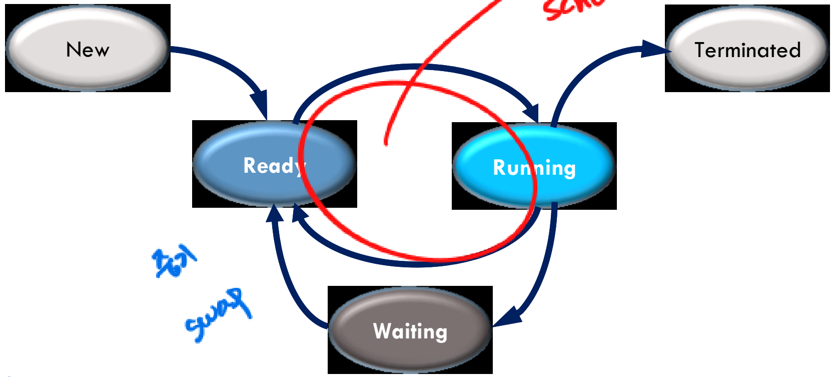
preemitive

성능 척도를 잘 정하자 (기준을 정하자)

Scheduling Criteria

성능을 보여줄수 있는 척도

- CPU utilization – keep the CPU as busy as possible
  - Throughput – # of processes that complete their execution per time unit 처리량 (단위 시간 당 process 최대한 많이)
  - Turnaround time – amount of time to execute a particular process CPU 사용 시간 외 모든 것 (Waiting, responding 등) 포함 시간
  - Waiting time – amount of time a process has been waiting in the ready queue Preemption 으로 Ready-queue에 저장하여 Ready에서 다시 돌아올 때까지의 시간
  - Response time – amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment) 처음으로 CPU가 응답할 때까지의 시간

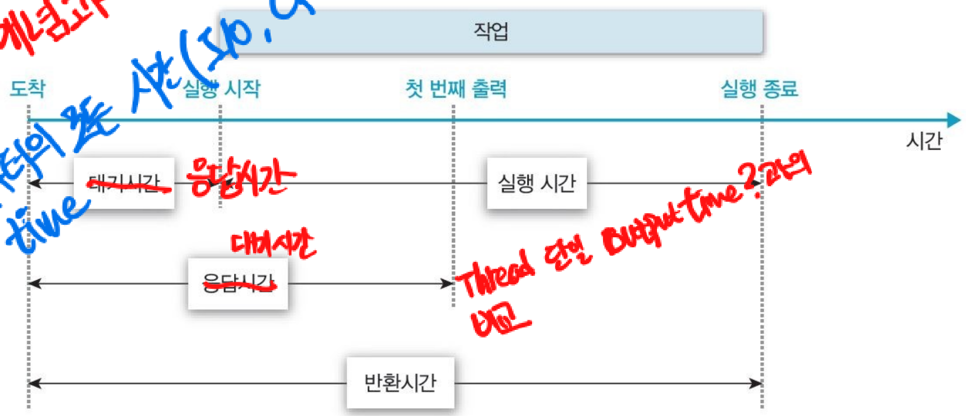


Short-term Scheduler 중점적

중기 swap

컴구조의 응답 개념과 다르다.

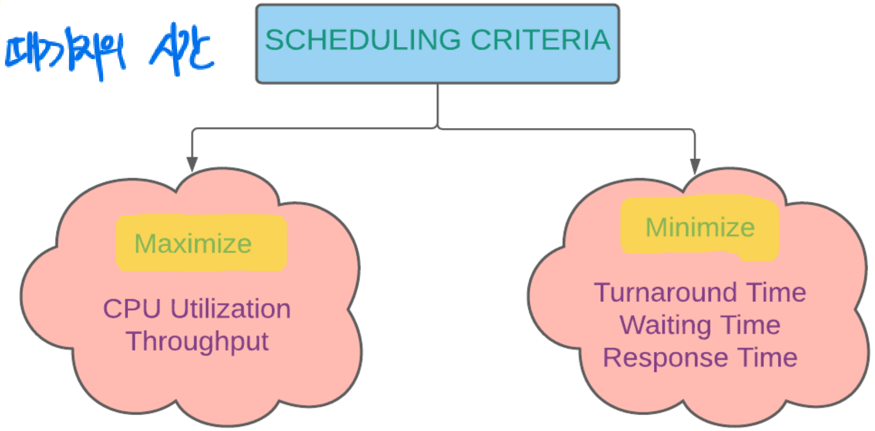
→ 프로그램 사용자의 총 시간 (I/O, CPU 등) elapse-time



응답시간

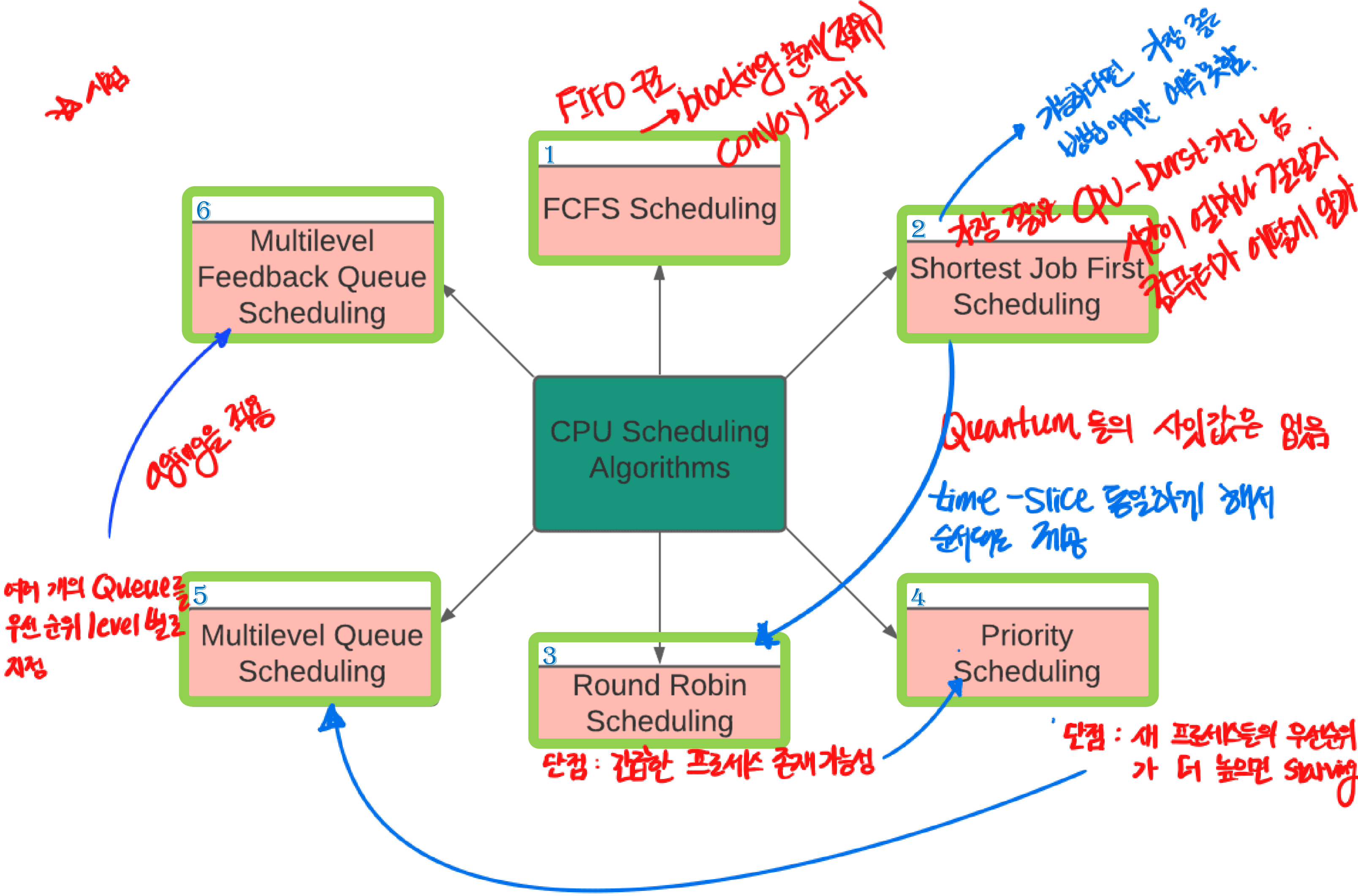
대기시간

Thread 단일 output time? 과의 비교

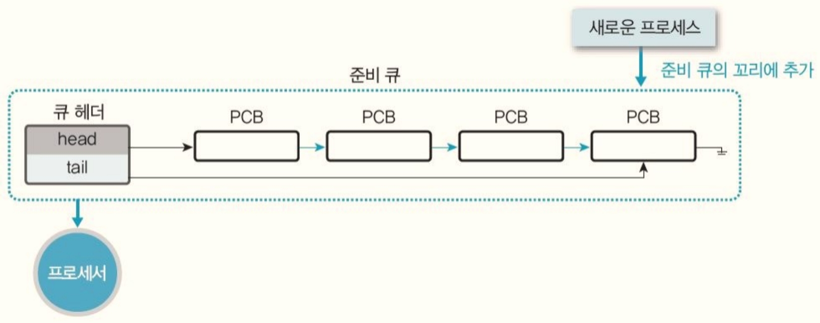


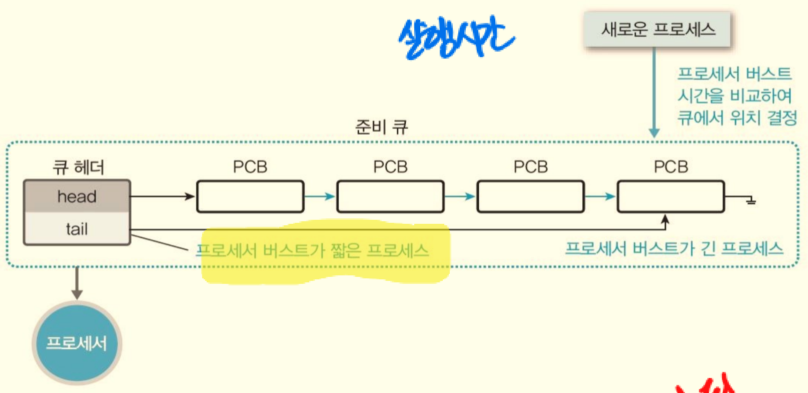
Scheduling Algorithm Optimization Criteria

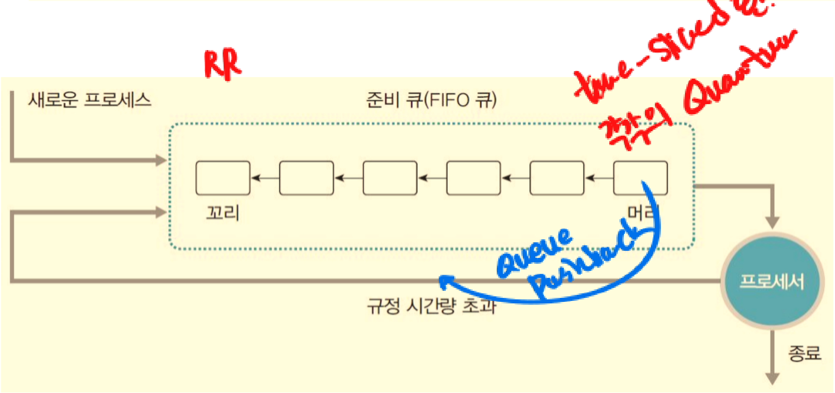
- Max CPU utilization
- Max throughput
- Min turnaround time
- Min waiting time
- Min response time

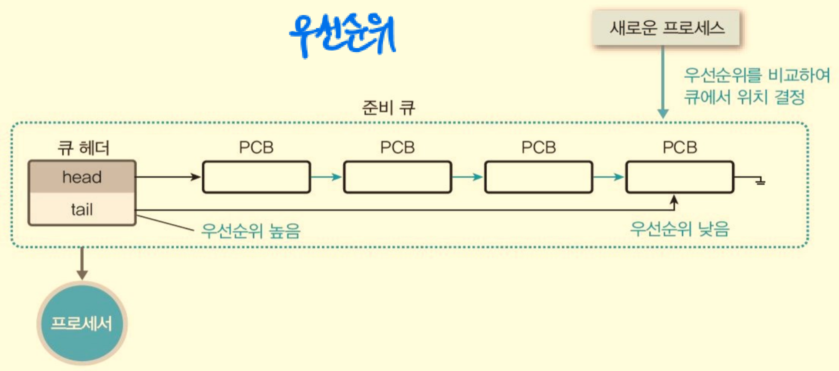


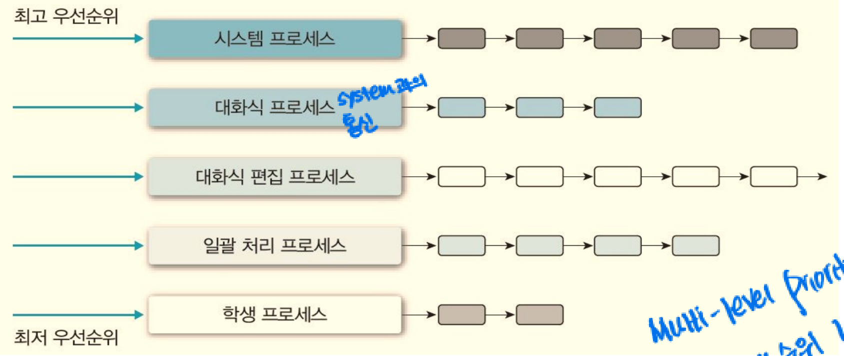
Scheduling Algorithm



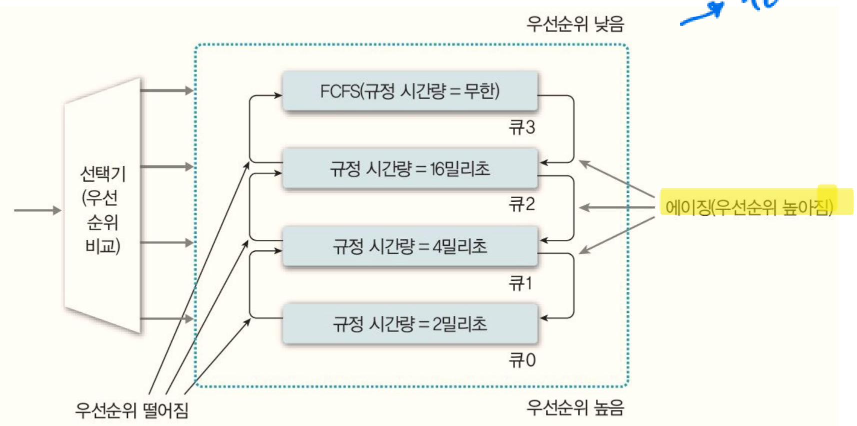

실행시간
새로운 프로세스
프로세서 버스트 시간을 비교하여 큐에서 위치 결정
준비 큐
큐 헤더
head
tail
PCB
PCB
PCB
PCB
프로세서 버스트가 짧은 프로세스
프로세서 버스트가 긴 프로세스
프로세서







Multi-level Priority queue
→ 우선순위 낮은 queue 기아상태 가능성


우선순위 낮음
FCFS(규정 시간량 = 무한)
큐3
규정 시간량 = 16밀리초
큐2
규정 시간량 = 4밀리초
큐1
규정 시간량 = 2밀리초
큐0
선택기 (우선 순위 비교)
에이징(우선순위 높아짐)
우선순위 떨어짐
우선순위 높음