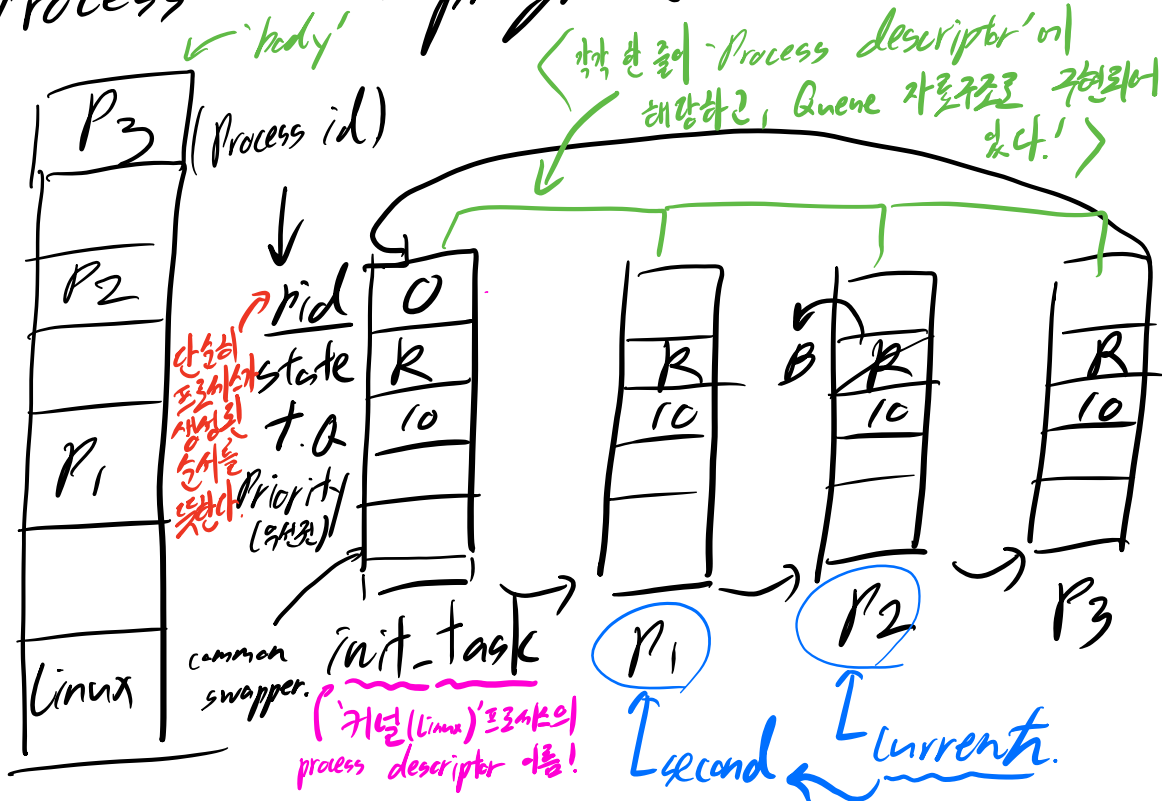Process = body + Process descriptor
-def   "소몸"   "영혼"
-process descriptor("task_struct{ }" 구조체로 선언되어 있음)
(process queue ← 특정 프로세스의 정보가 내포돼있는 구조체!!
    run queue   (ex) pid, ppid, state ···)

- current

- time slice (=time quarter)

- scheduler.

. Process = a program loaded in memory.
      'body'   <꽉 찬 줄이 'Process descriptor'에

      (Process id)   해당하고, Queue 자료구조로 구현되어
                              있다.'>

P3				
P2				
P1				
Linux				

pid → 단숨에 프로세스가 생성되는 순서를 뜻한다.
state
t.q
priority (우선권)

O R 10
B R 10
B R 10
R 10

common swapper. init_task
('커널(Linux)'프로세스의
process descriptor 이름!)

P1   P2   P3
second ← current.

＊run quene : runable state process만
　　　　실행하는 것.

＊process quene : 모든 state process를
　　　　실행하는 것.

P2
...

`scanf()` 함수를 호출하면,
해당 프로세스는 `block`
상태가 된다.

scanf(%d, &x)
...

`exit()`가 호출되면, 해당 프로세스는
zombie 프로세스가 된다.

exit(o)

⟵ 모든 프로세스는 `exit()`함수를 끝에
호출한다!!

・ process가 지닌 `priority`와 다른 변수를
고려하여, `schedule/reschedule` 된다.

→ `priority`가 높을수록, 앞선 순위로 schedule

`zombi process` : 해당 프로세스의 상태가
`Z`로 변함., `exit()`가
실행외면 해당 프로세스는
`zombi process`가 된다.

· Time-quater : 프로세스가 cpu를 점유할
수 있는 최대 시간.