

- shell

- linux initialization

* 'cpu가 어떤 코드를 실행하고
있느냐!'가 제일 중요하다.

~~shell~~ shell (bash.c)

← { bash 명령어를 실행시키면, 해당 c파일이
실행된다. }

for (i;) {

소문만루프!

printf("# ");

scanf("%s", buf)

← bash 명령어를 입력 받은
ex) ". / ex1"

x = fork();

← 'bash.c' 프로세스가 복사됨.

해당 함수가
호출되면,
사용자가
bash 명령어를
입력하기 전까지
해당 'bash' 프로세스는
'sleep' 상태로 됨.

if (x == 0) {

execve(buf, ...);

← 복사된 자식 프로세스 (bash.c)가
이제 막 실행될 프로세스로 전환됨.

} else {

wait();

}

← 해당 함수를 통해, child 프로세스가 끝나기 전까지, parent 프로세스는
'sleep' 상태로 기다리고 있음.

}

↑ 모든 프로그램은 이렇게 실행된다!

start_kernel()



rest_init



kernel_thread(kernel_init, ...);

kernel_thread(kthread, ...);



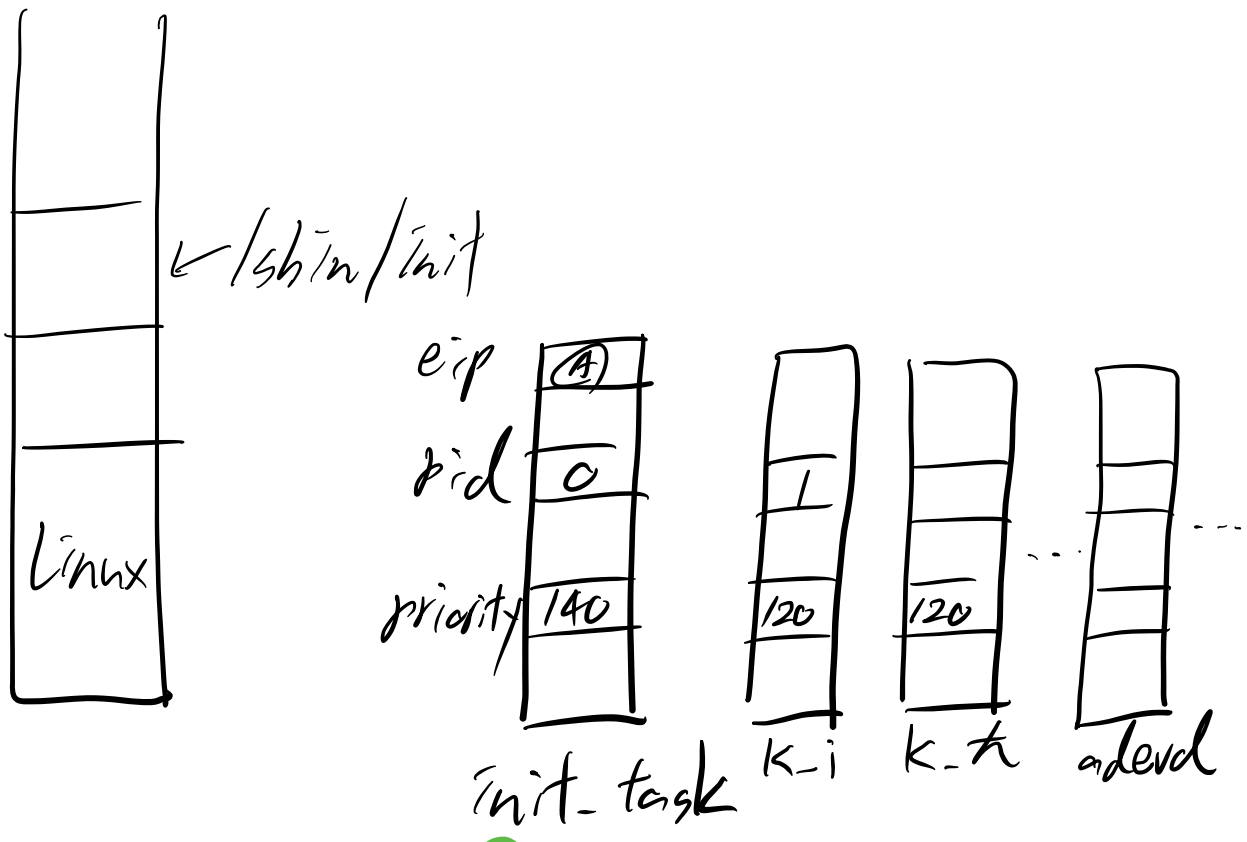
cpu_idle



(A) schedule

↓
kernel_init() 호출이 되면 scheduled이기 때문에, return이 없다.

↓
...
execve("/sbin/init", ...);



자란형이 구조체(struct-task)
인 변수

