

- process life cycle

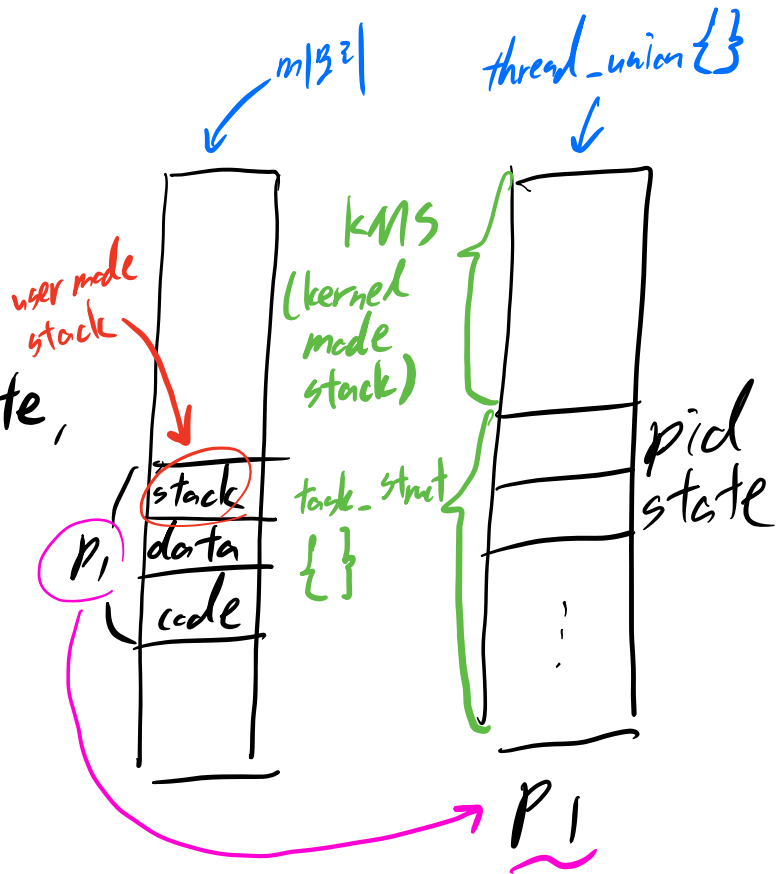
: fork (duplicate)
 exec (transform)
 exit (stop)

- thread - union

- fork, pthread_create,
 kernel - thread

- exec

- shell



• **fork** : duplicate (프로세스 복제. 생성) 하는 함수.
 ↳ 2개의 프로세스를 발생시키는 함수.

↳ parent process와
 child process가
 서로 독립적인
 관계가 된다.

```

{ex1.c}
ex) void main () {
    int x;
    x = fork();
  }
  
```

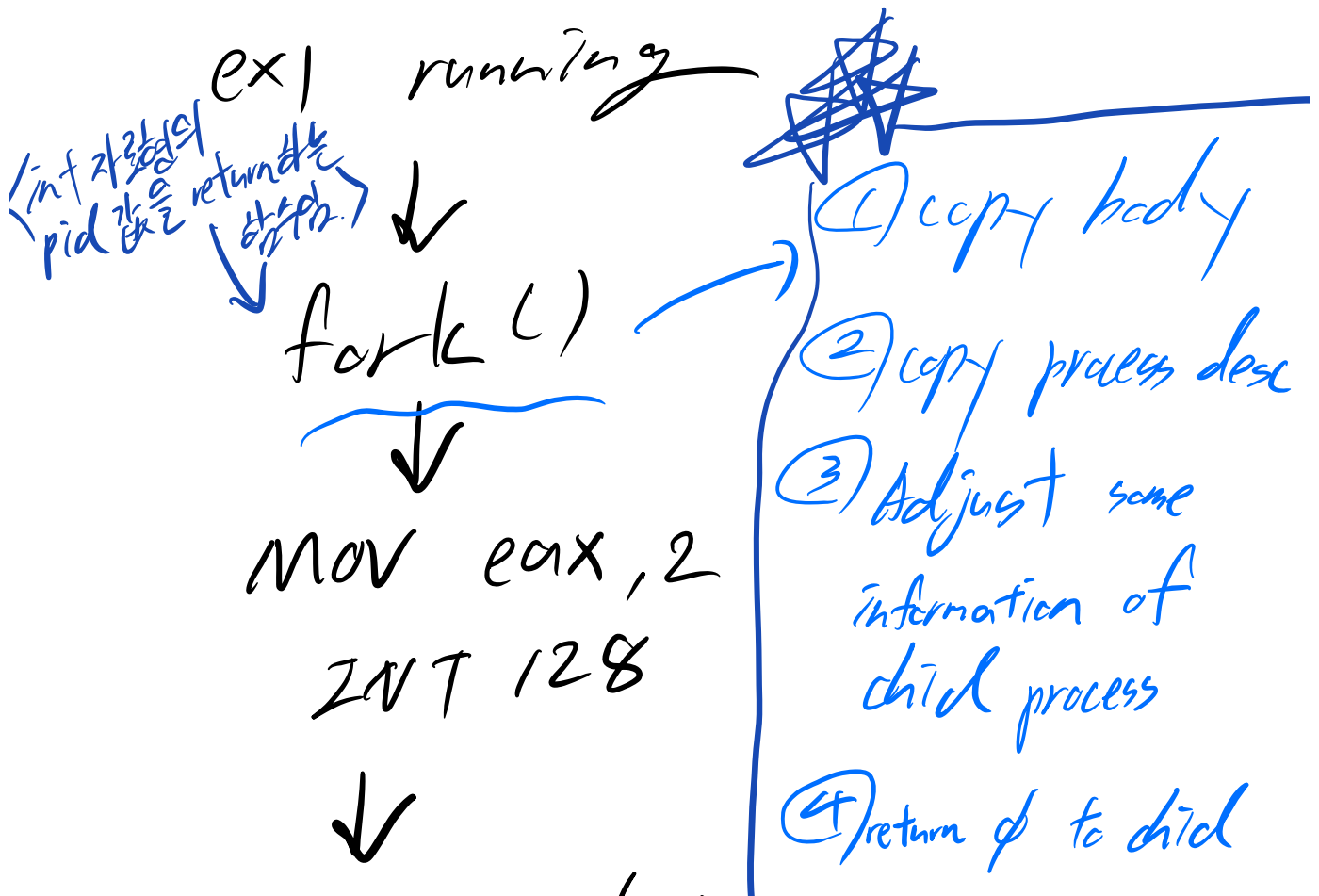
↳ 현재 프로세스를 복제하고
 이것이 child process가
 된다.

return 100;

printf("hello\n"); ← (A)

}

결과: hello
hello



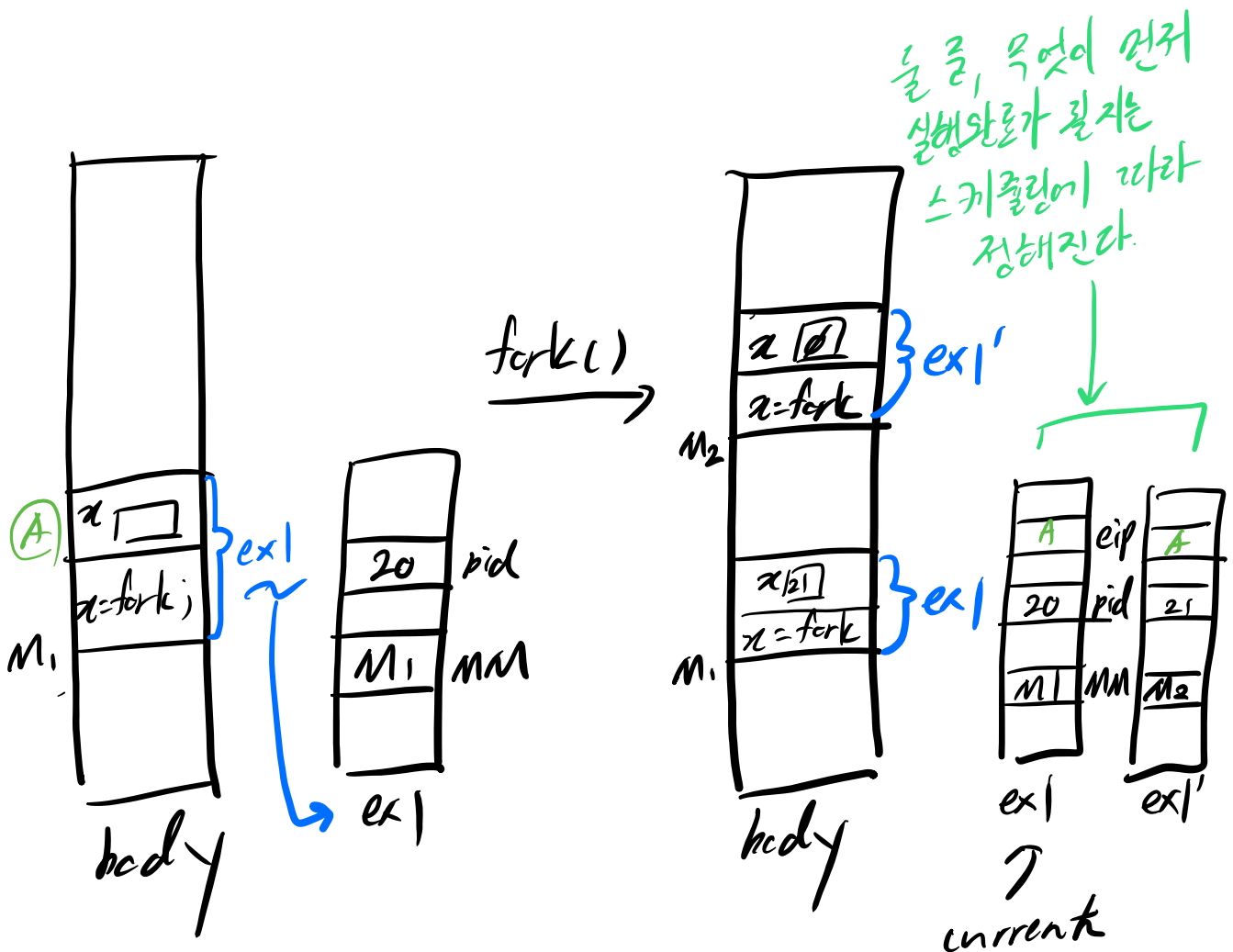
system - case:

↓
sys_fork()

↓
do_fork()

process

X parent process 나
자의 번식에는
child pid의 갯수 같이
들어간다.

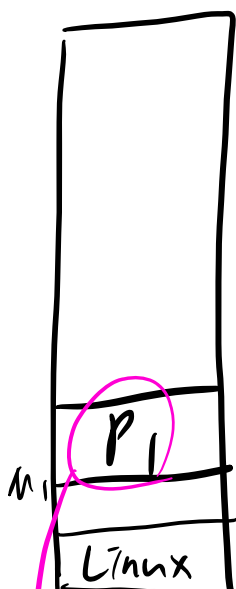


- pthread_create : body는 공유하고
 ↑ INT를 발생시키는 함수. process descriptor 안 복제
 하는 함수 \Rightarrow 쓰레드를
 생성하는 함수.

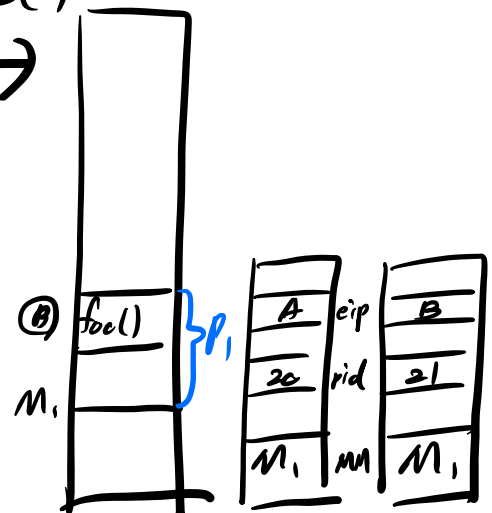
ex) ^{P1.C} void foo() { ③
 void main() {

pthread_create(..., foo, ...);
 }

이 다음에 수행할
 명령 위치를 설정!



pthread_create()



hcd → P₁

P₁ P₂

- kernel_thread : 'pthread_create'와 동일하게 작동하는 함수이지만, INT를 발생시키지 않는다.
- < kernel_thread에 의해 실행됨. >

ex) void kernel_init() { (B)

printf("...");

...

rest_init();

...

kernel_thread(kernel_init, ...);

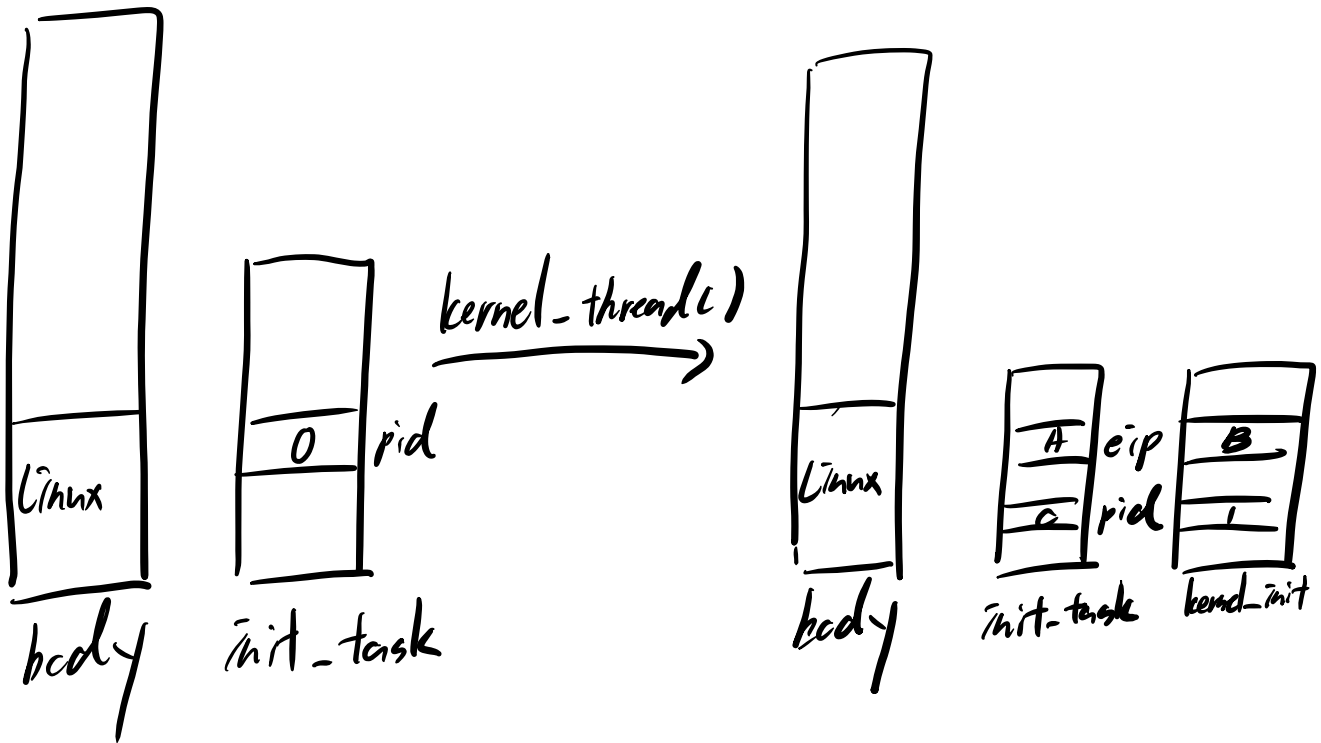
...

cpu_idle()

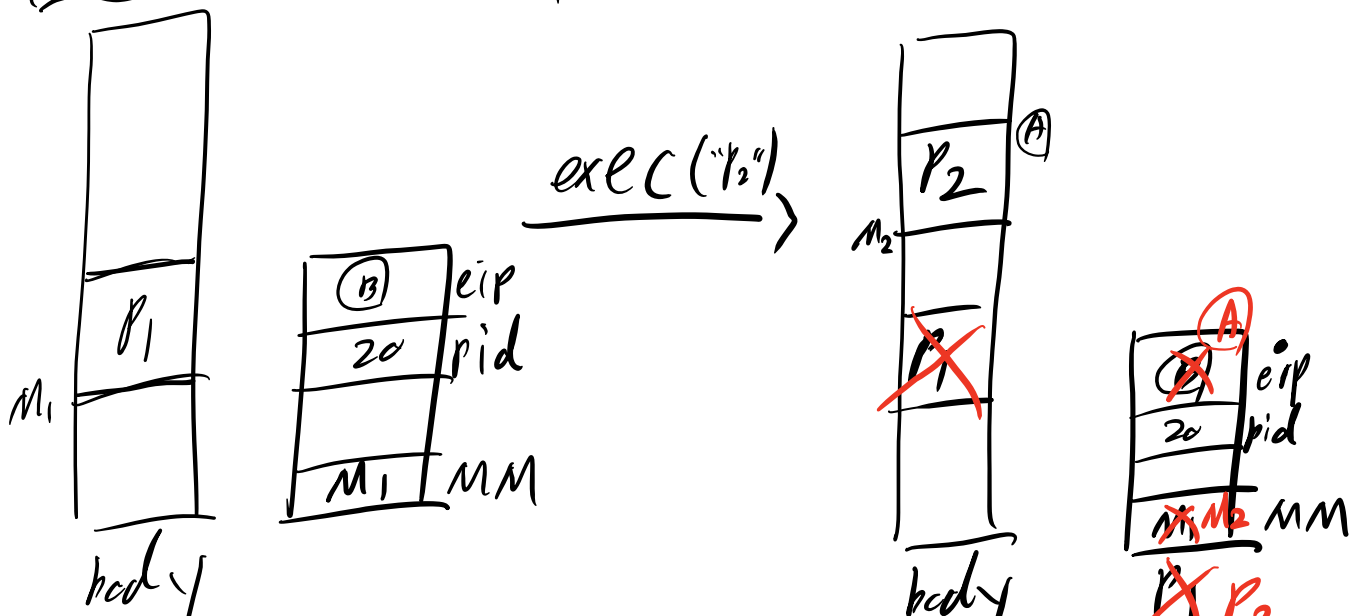
→ while(1) {

... }

sched(1)3



• `exec` : 하나의 프로세스를 다른 프로세스로 바꿀때 사용되는 함수.
 {P.L.C}



body 상이한 바깥쪽만,
pid는 같으므로 프로세스
변경 X.

X interrupt 처리 후에는 cpu가
프로세스를 reschedule 한다.

X pid가 같으면, system 입장에선
같은 프로세스이다.

X 모든 운영체제에서 새로운 프로세스를 만드는
과정은 기존에 있던 프로세스를 복제한 후
(parent process)

해당 복제한 프로세스의 body를 새로운 프로세스의
body로 바꾼다.