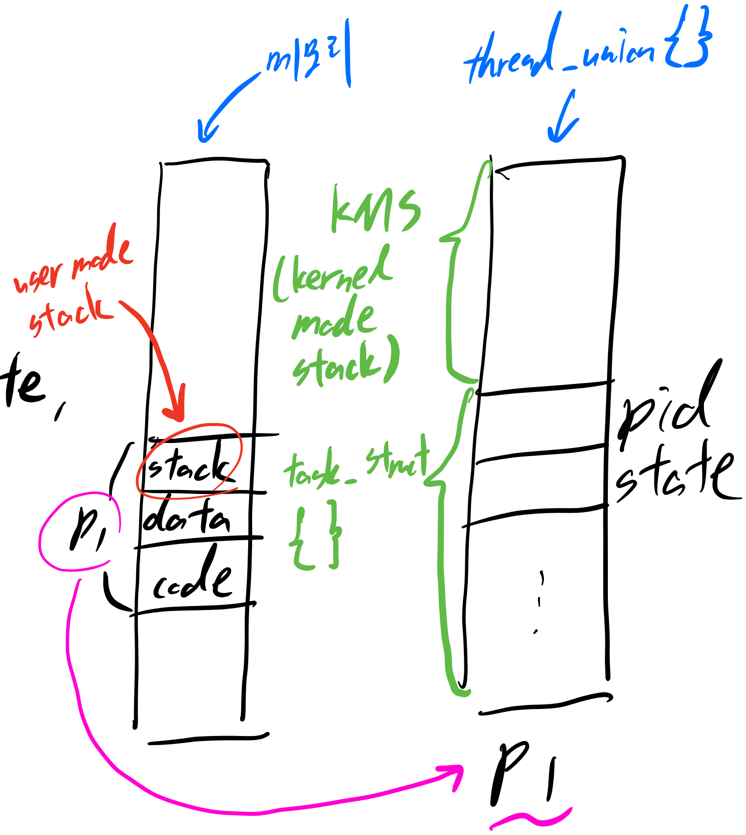
- process life cycle
  : fork (duplicate)
  exec (transform)
  exit (stop)

- thread_union
- fork, pthread_create, kernel_thread
- exec
- shell



• fork : duplicate (프로세스 복제·생성) 하는 함수.
  └ 2^N개를 발생시키는 함수.
  └> parent process와 child process가 서로 독립적인 관계가 된다.

(현재 프로세스를 복제하고 이것이 child process가 된다.)

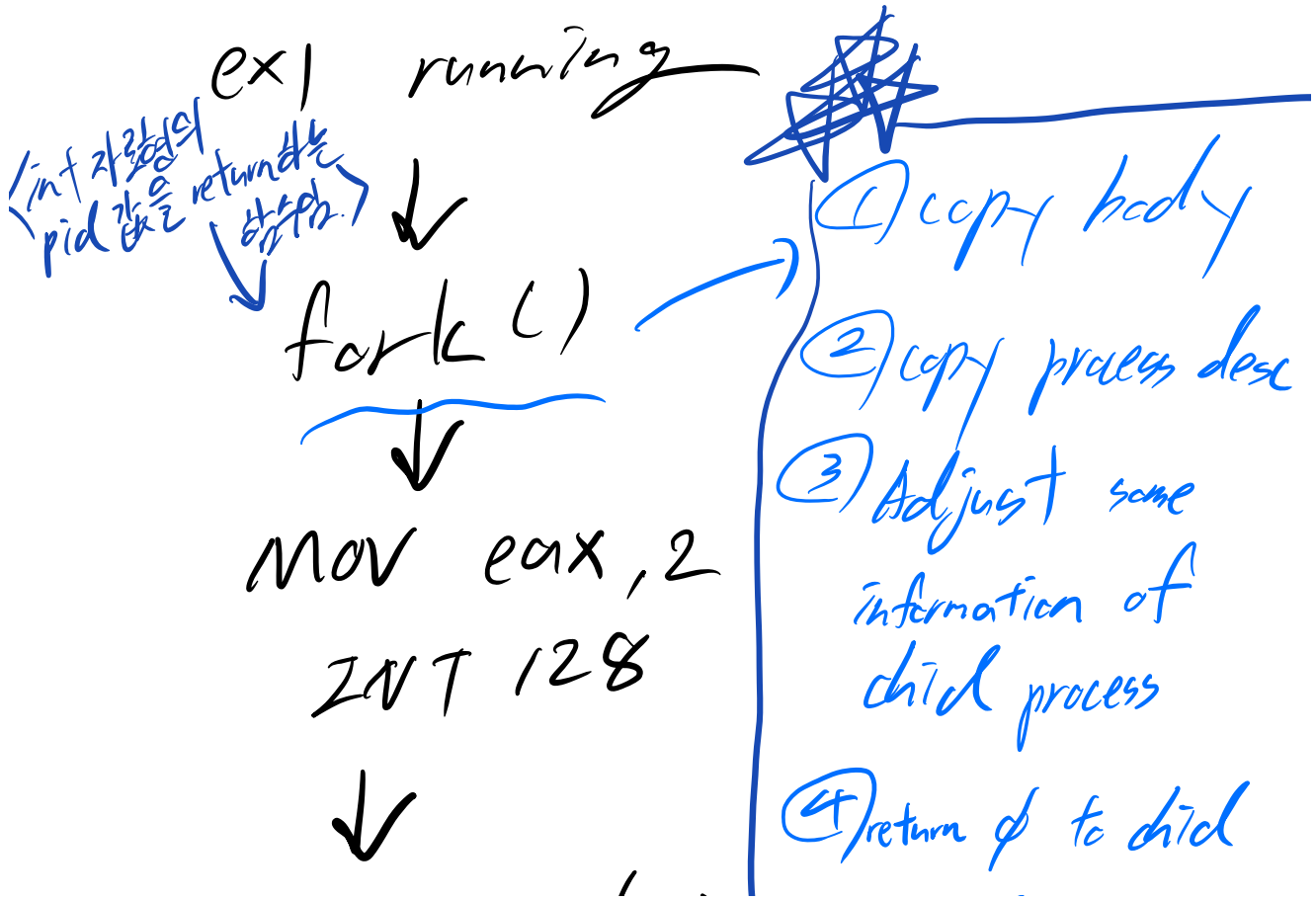
ex) ex1.c

```
void main() {
    int x;
    x = fork();
```

printf("hello\n"); ← (A)

2

결과 : hello
hello



system_call :

...

↓

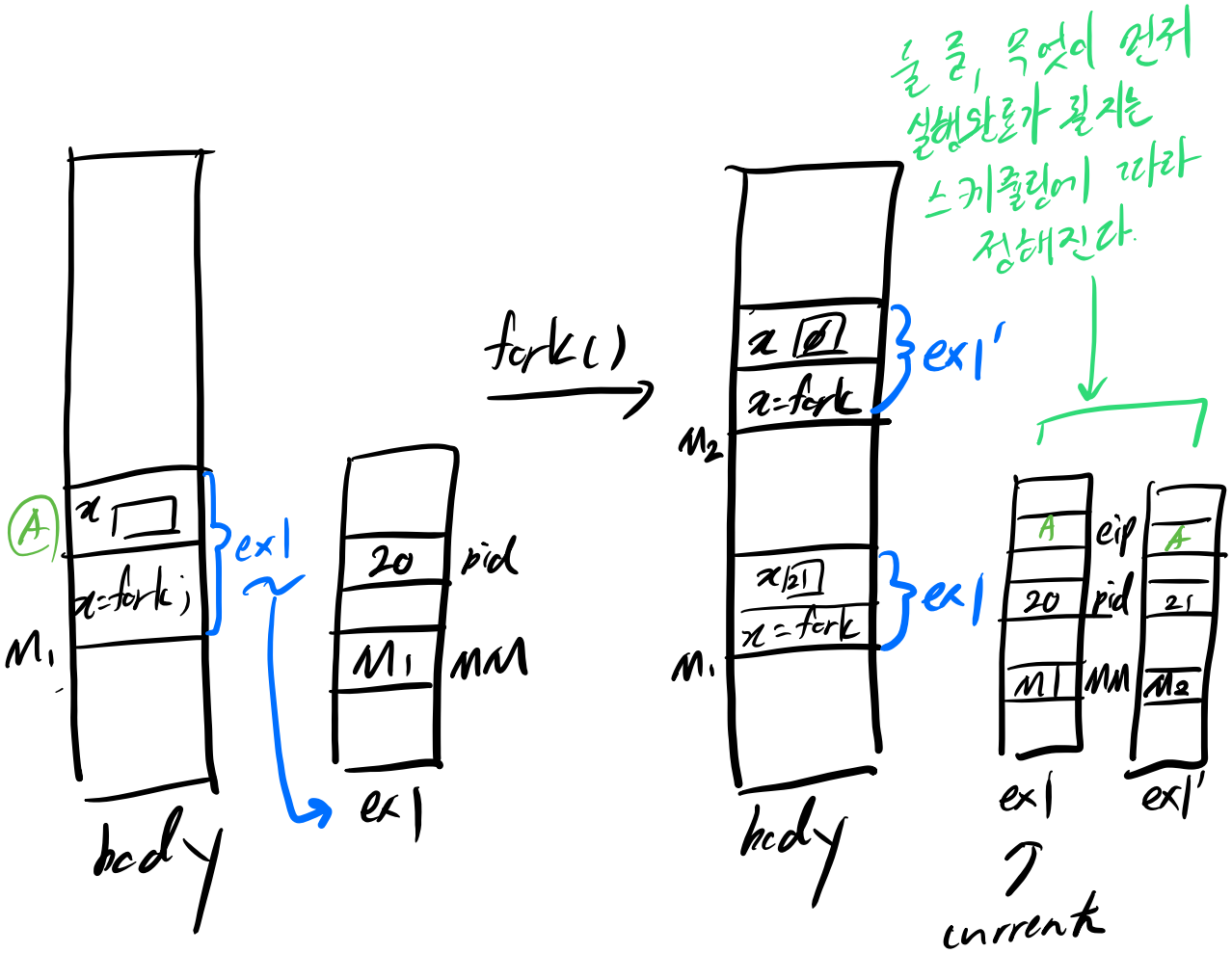
sys_fork()

↓

do_fork()

process

- parrent process 내 x의 변수에는 child pid의 값이 들어간다.



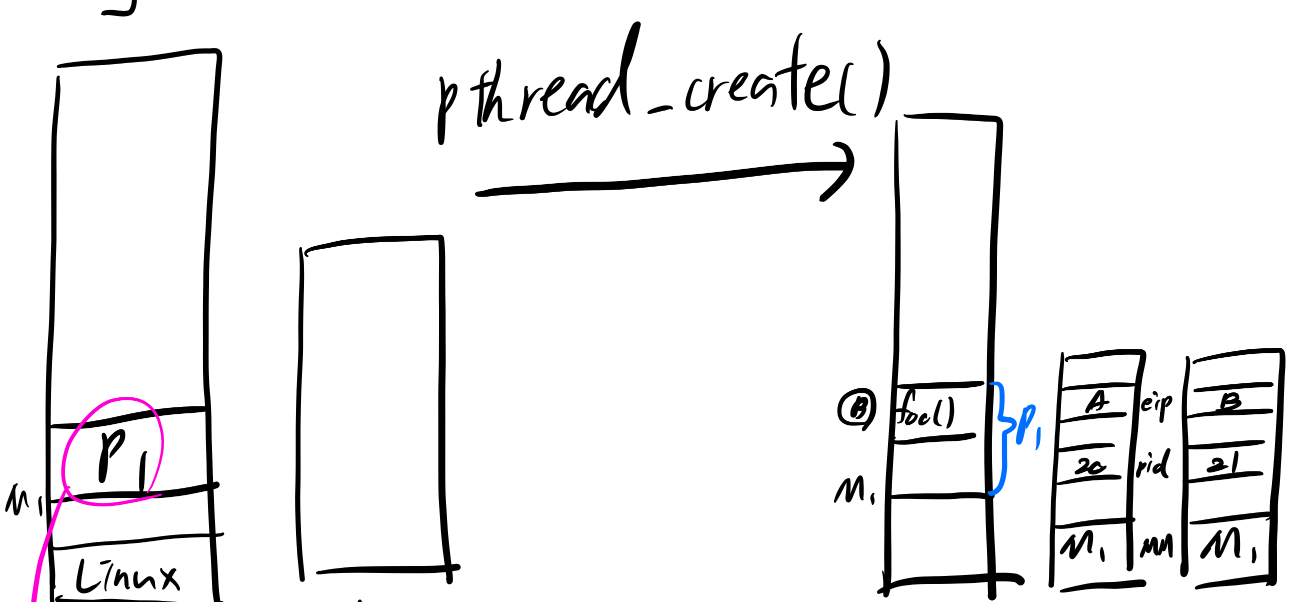
• pthread_create : body는 공유하고 process descriptor만 복제하는 함수 ⇒ 쓰레드를 생성하는 함수.

↑ INT를 발생시키는 함수.

ex) {p1.c}

```
void foo(){ Ⓑ
void main(){
    ...
    ...
    pthread_create(..., foo, ...);
  }
}
```

이 다음에 수행할 명령 위치를 설정!



p1 p2

• kernel_thread : `pthread_create'와 동일하게 작동하는 함수이지만, INT를 발생시키지 않는다.

<kernel_thread에 의해 실행됨.>

ex) void kernel_init() {

printf("...");

...

rest_init() {

....

kernel_thread(kernel_init, ...);

....

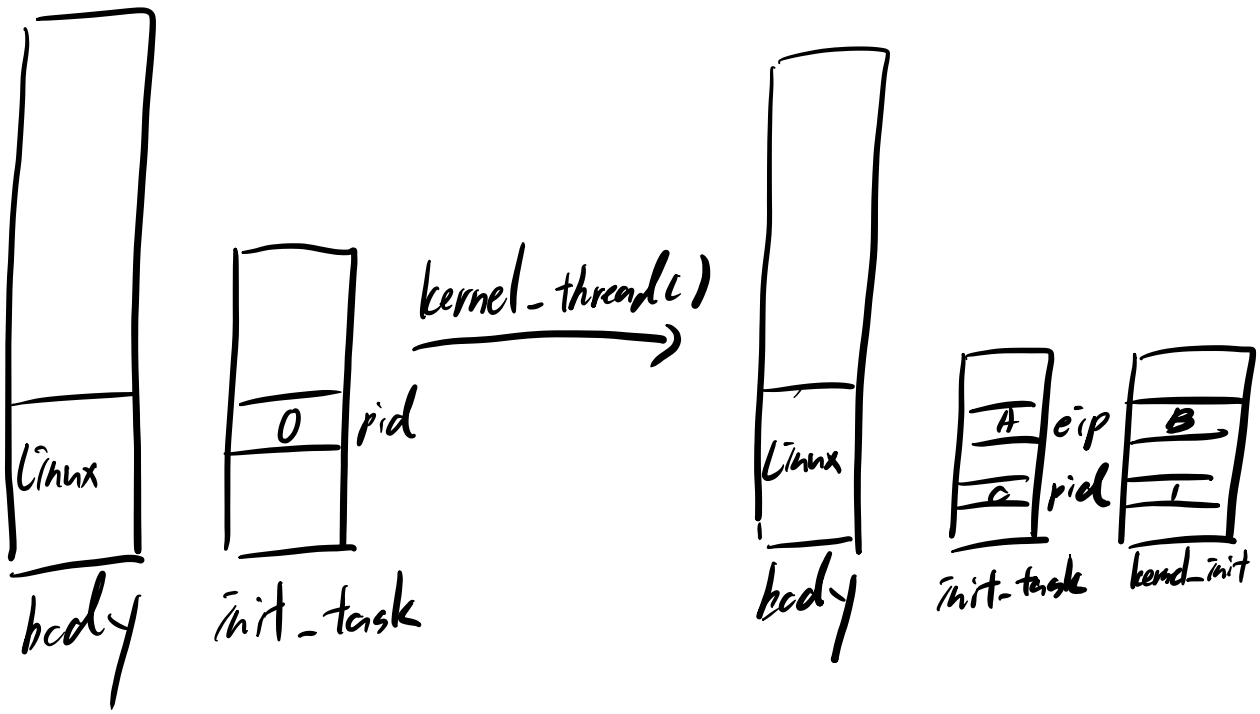
cpu_idle()

while(1) {

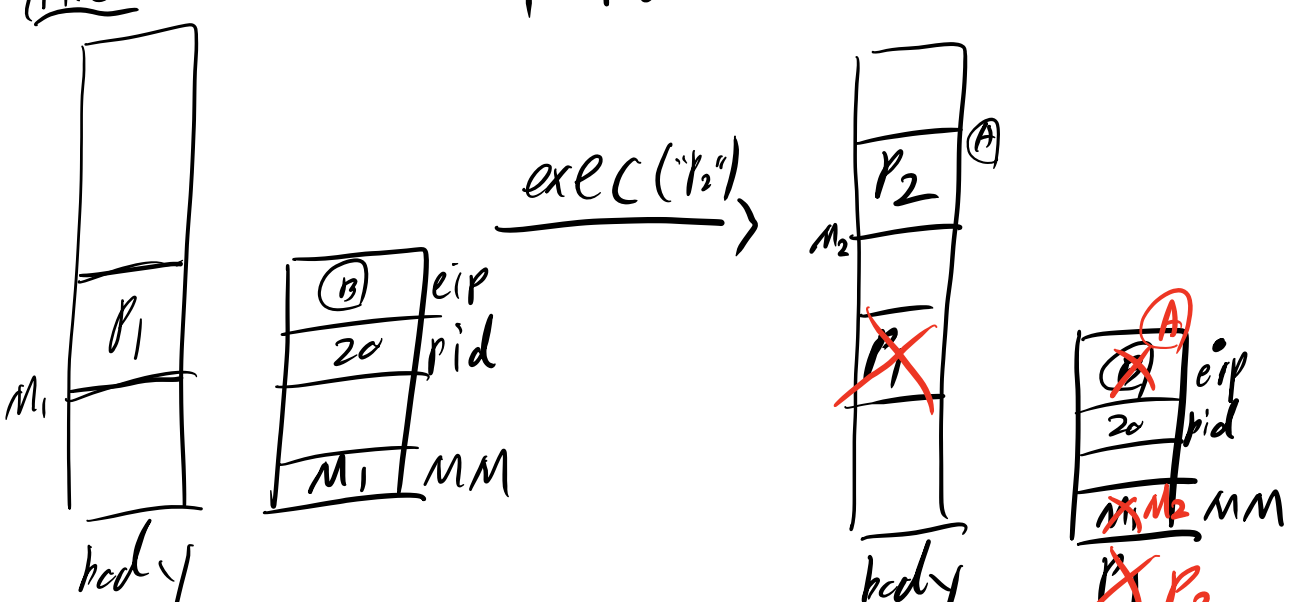
...

}

sched ( ) 3



• exec : 하나의 프로세스를 다른 프로세스로 바꿀때 사용되는 함수.



↑ body 상에선 바뀌지만, pid는 같으므로 프로세스 변경 X.

X interrupt처리 시에는 CPU가 프로세스를 reschedule 한다.

X pid가 같으면, system 상에선 같은 프로세스이다.

X 모든 운영체제에서 새로운 프로세스를 만드는 과정은 기존에 있던 프로세스를 복제한 후 (parent process)
해당 복제된 프로세스의 body를 새로운 프로세스의 body로 바꾼다.