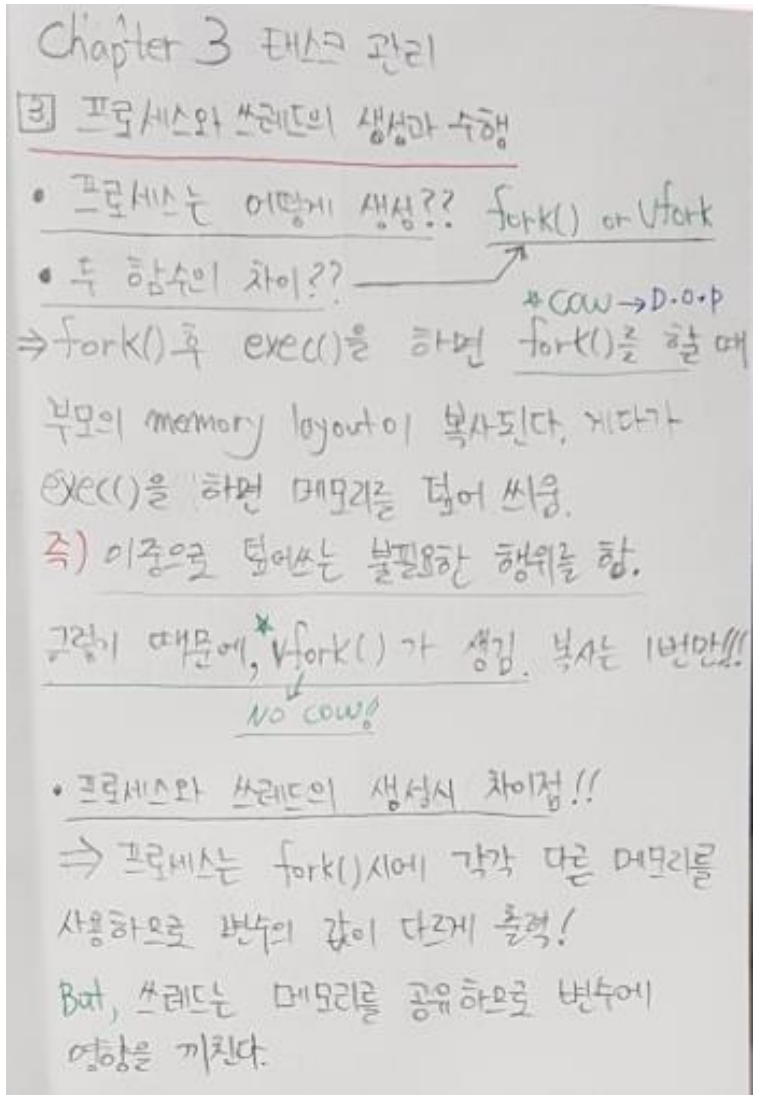
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – GJ (박현우)
uc820@naver.com

1. 리눅스 커널 내부 구조 – Chapter 3 (태스크 관리)



Chapter 3 태스크 관리

[3] 프로세스와 쓰레드의 생성과 수행

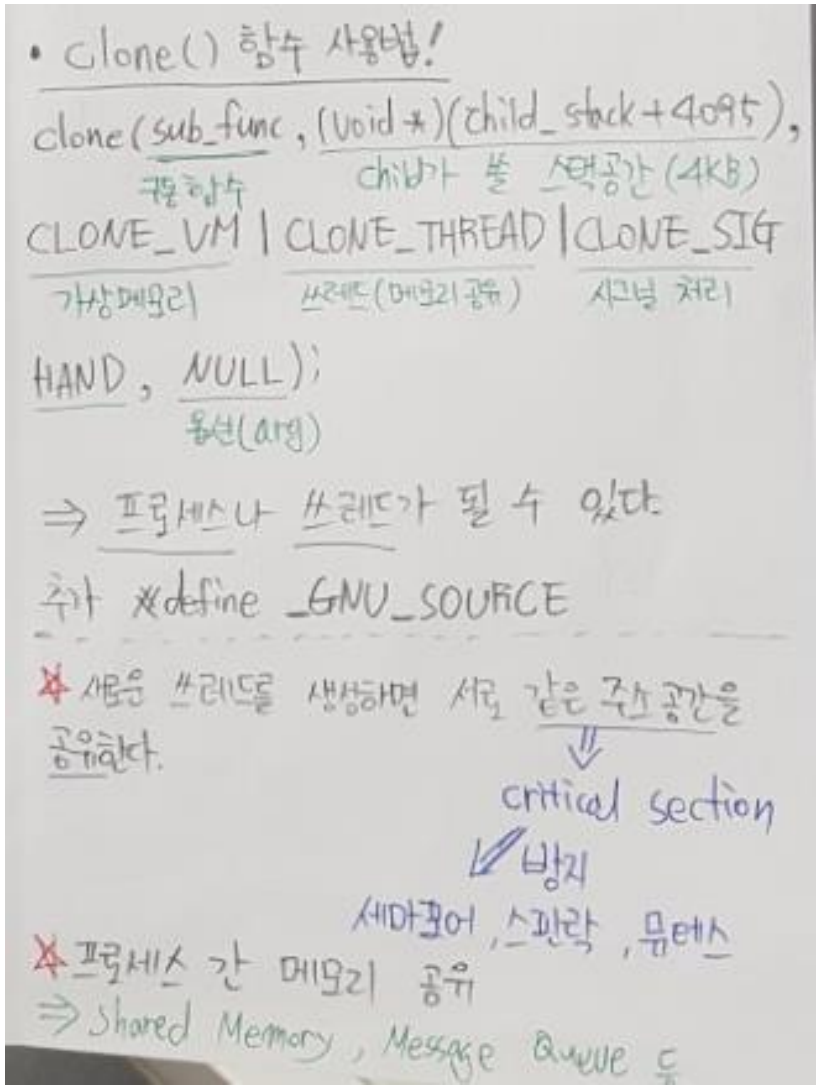
- 프로세스는 어떻게 생성?? fork() or Vfork
- 두 함수의 차이?? *COW → D.O.P

⇒ fork() 혹 exec()을 하면 fork()를 할 때 부모의 memory layout이 복사된다. 하지만 exec()을 하면 메모리를 덮어 씌움.
즉) 이중으로 덮어쓰는 불필요한 행위를 함.

그렇기 때문에, *Vfork() 가 생김. 복사는 1번만!!!
No COW!

- 프로세스와 쓰레드의 생성시 차이점!!

⇒ 프로세스는 fork()시에 각각 다른 메모리를 사용하므로 변수의 값이 다르게 출력!
But, 쓰레드는 메모리를 공유하므로 변수에 영향을 끼친다.



- clone() 함수 사용법!

clone(sub_func, (void *)(child_stack+4095),
함수 / child가 쓸 스택공간(4KB)
CLONE_VM | CLONE_THREAD | CLONE_SIG
가상메모리 / 쓰레드(메모리공유) / 시그널 처리
HAND, NULL);
옵션(arg)

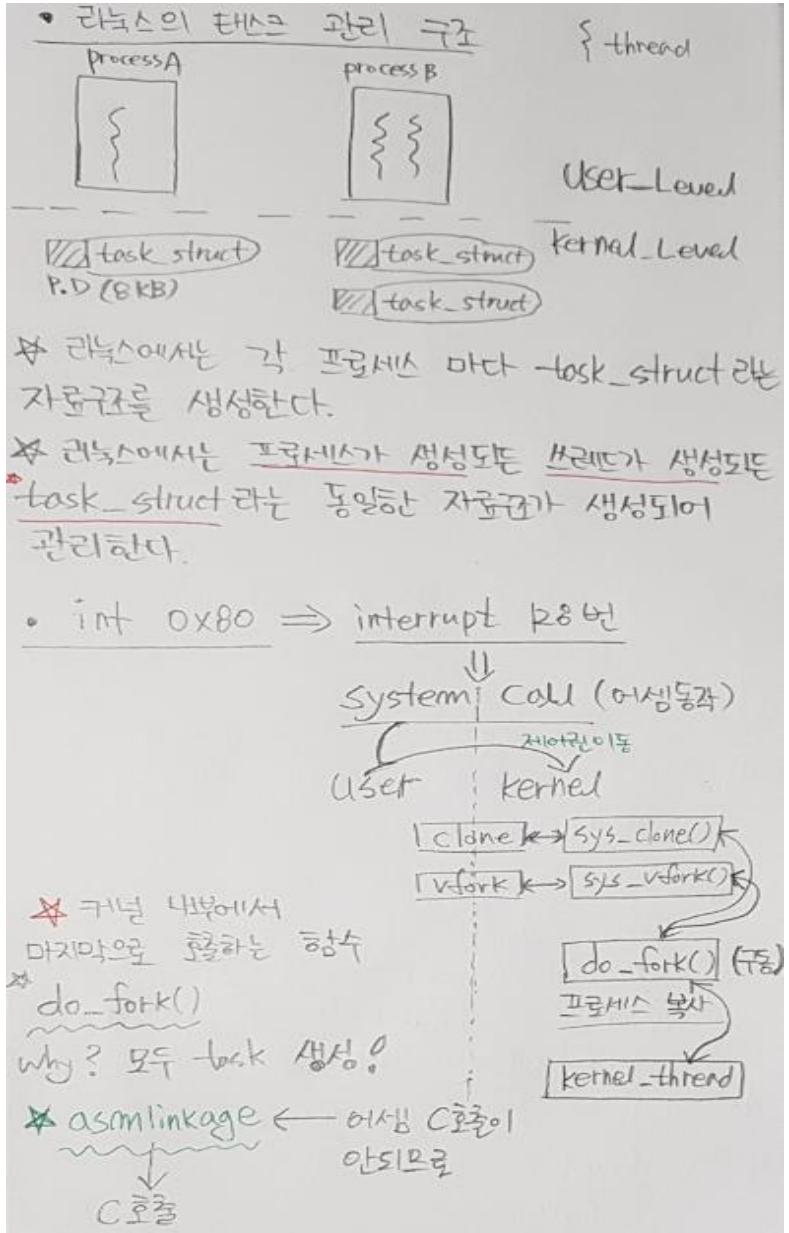
⇒ 프로세스나 쓰레드가 될 수 있다.

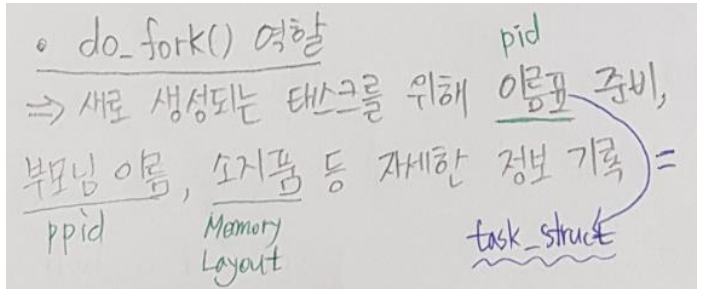
추가 #define _GNU_SOURCE

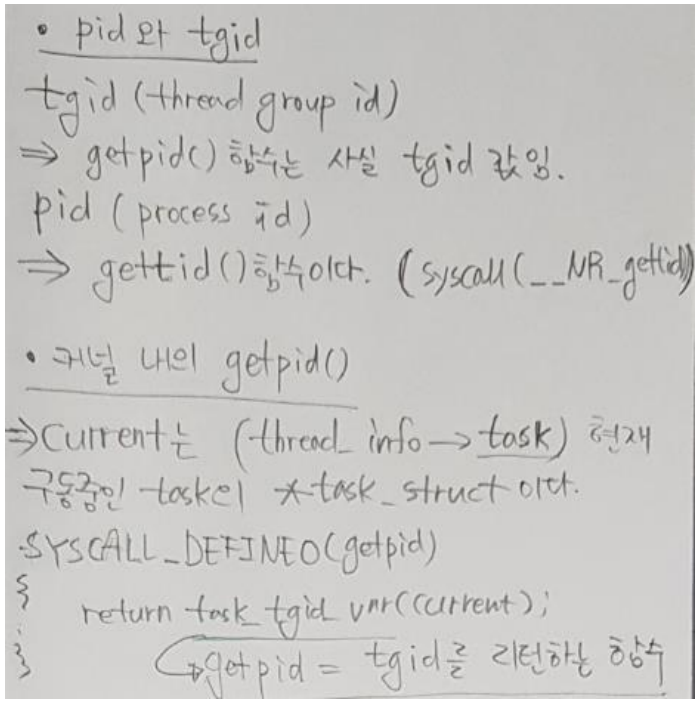
* 새롭은 쓰레드를 생성하면 서로 같은 주소공간을 공유한다.
⇓
critical section
⇓ 방지
세마포어, 스핀락, 뮤텍스

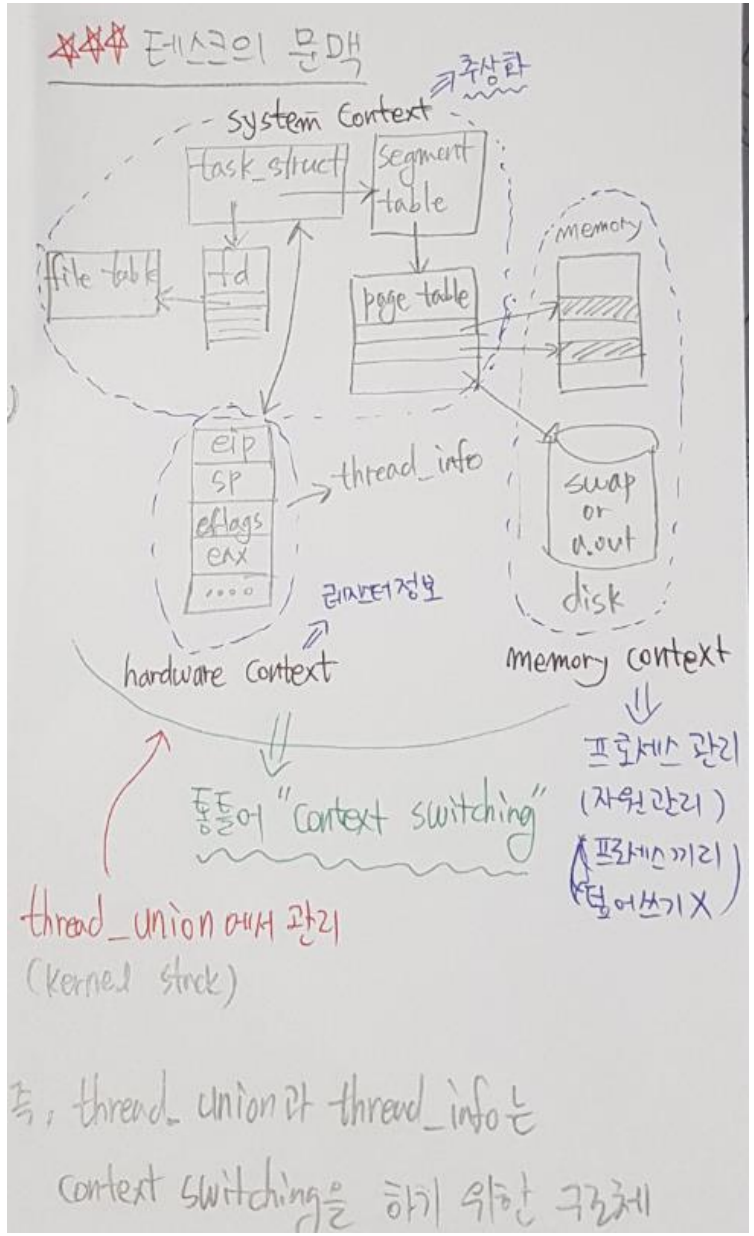
* 프로세스 간 메모리 공유
⇒ Shared Memory, Message Queue 등

1. 리눅스 커널 내부 구조 – Chapter 3 (태스크 관리)

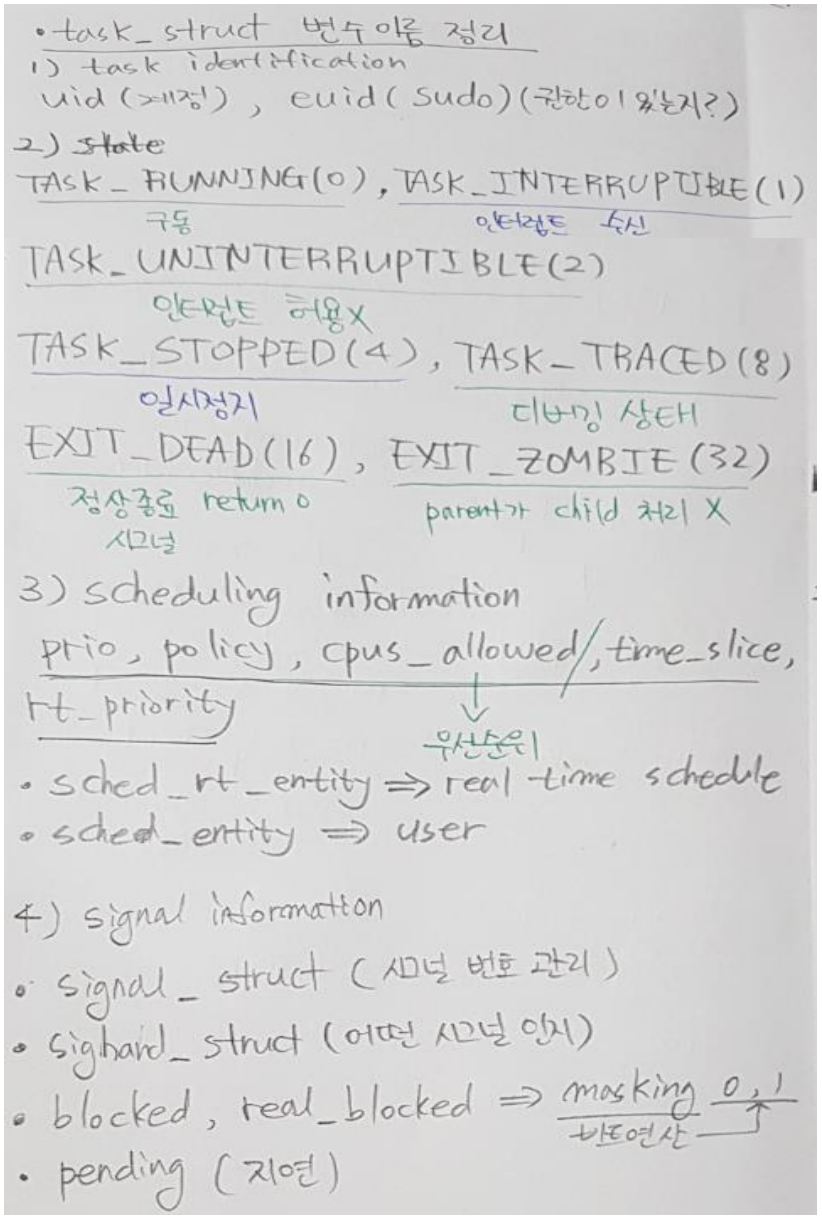


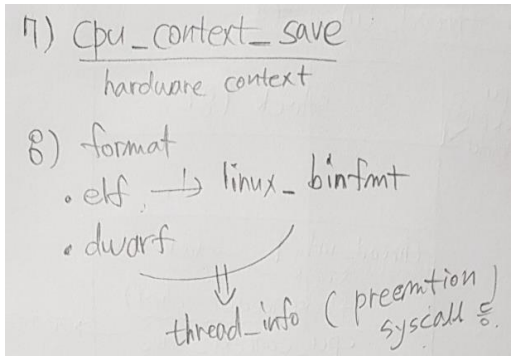


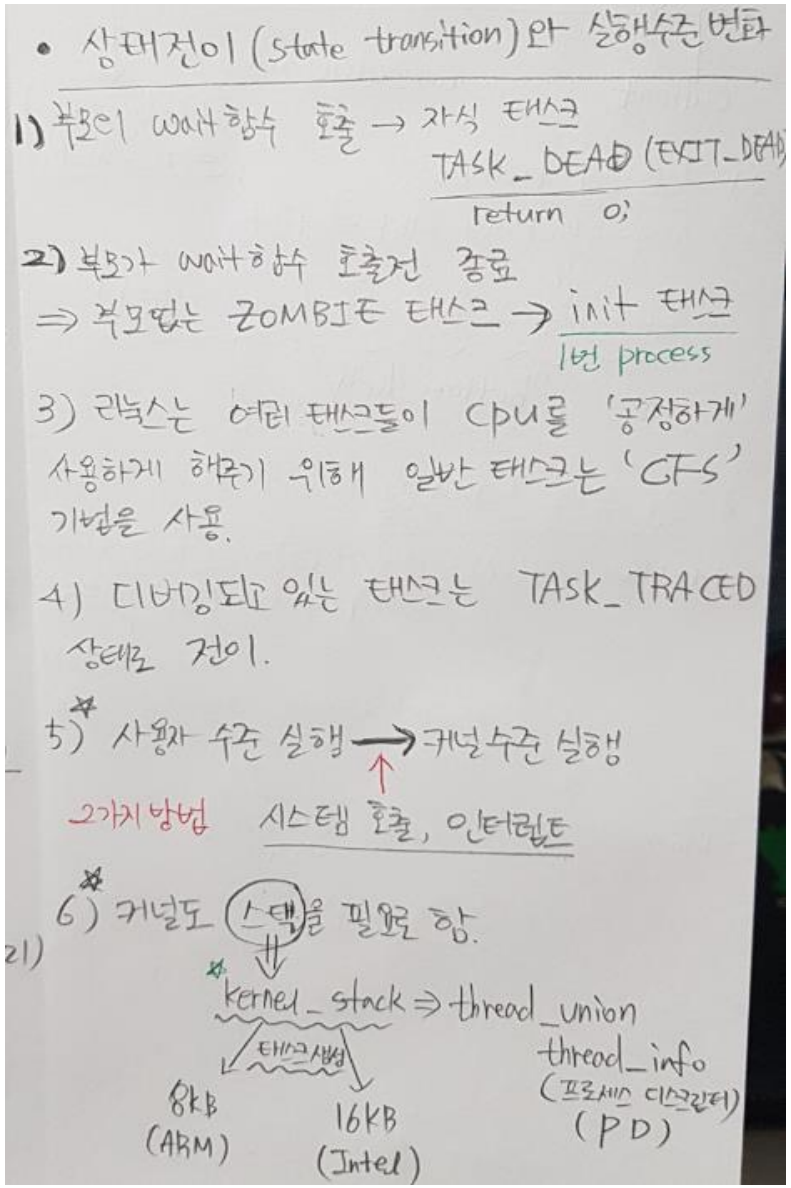




1. 리눅스 커널 내부 구조 – Chapter 3 (태스크 관리)







1. 리눅스 커널 내부 구조 – Chapter 3 (태스크 관리)