

2018. 4. 9 월 - 33 회차

과정 : TI, DSP, Xilinx Zynq FPGA, MCU 기반의 프로그래밍 전문가 과정

Prof. 이상훈
gcccompil3r@gmail.com
Stu. 정상용
fstopdg@gmail.com

Chapter 3 : 태스크관리

-> We are going to study about a task.(Definition, structure, producing and executing, context switching, thread, scheduling)

1. 프로세스와 스레드 그리고 태스크

-> 프로세스 : 동작중인 프로그램(running or runnable programme)

-> 스레드 : 수행의 단위

-> 태스크 : 자원소유권의 단위

-> 명령어 file : 어떤 파일인지 구체적으로 확인이 가능

리눅스실행파일포맷(Unix 도 마찬가지)

ELF

디버깅포맷

DWARF

cf. ELF 는 DWARF 를 싫어함,

윈도우실행파일포맷

PE

-> ./a.out & 에서 &의미 : 백그라운드에서 실행-> 예제를 통한 프로그램과 프로세스의 차이

ps 에서 나온 test : process(동적)

ls 에서 나온 test : program(수동적)

-> process & thread

process : pid = tgid

thread : pid = tgid -> thread leader(process)

pid != tgid -> thread component

e.g

FunnyJungle employee : tgid != pid(tgid = Funny Jungle, pid = ID)

Funny Jungle CEO : tgid = pid -> process

cf.

fork = process(pid = tgid)

-> Finding where pid, tgid are

vi -t task_struct -> 144 -> 1 -> :1481

2. 사용자입장에서 프로세스구조

-> 'vm_area_struct' or 'segment'
 -> where is 'vm_area_struct'?
 task_struct -> mm_struct -> :433, 434
 start_code, end_code : TEXT
 start_data, end_data : DATA
 start_brk, end_brk : HEAP(brk = HEAP)
 start_stack : STACK
 Therefore, a size of vm_area_struct : 28byte(= 4byte x 7)
 -> pgd(Page Global Directory)
 starting point at second vm layout
 pgd is placed under mm_struct

3. 프로세스와 스레드의 생성과 수행

-> How to make process : There are 2 ways, fork() and vfork().

-> The difference between fork() and vfork() : vfork() is used when does not need a parent's memory
 (i.e when use fork() with exe() -> fork() + exe() = vfork()), fork() + exe() : useless copy

-> How to make thread : Also 2ways, clone() and pthread()
 -> clone() : it will be a process or a thread depending on an option.

p.54
 code
 clone() shares memory, therefore global value is 3

-> A thread's character : parent's thread and child's thread share the memory
 It can occur a problem on critical section. So we use mutex, semaphore or spin-lock.

homework

1. sys_fork 분석
2. NPTL 조사하기

kernel/linux-4.4/include/linux/sched.h
 -> 1378 ~ 1824 : task_struct

kernel/linux-4.4/kernel/fork.c
 -> 1693 : do_fork
 -> 1787 : SYSCALL_DEFINE0(fork) = sys_fork
 -> a code about sys_fork

```
#ifdef __ARCH_WANT_SYS_FORK
SYSCALL_DEFINE0(fork)
{
#ifdef CONFIG_MMU
return _do_fork(SIGCHLD, 0, 0, NULL, NULL, 0);
```

```
#else
    /* can not support in nommu mode */
    return -EINVAL;
#endif
}
```

: ##을 제거하고 앞과 뒤의 내용을 합쳐 함수를 생성.

cf. 원래는 do_fork() 먼저 못 찾음. 하위구조체에서 상위구조체 혹은 함수를 찾지 못함.