

2016025469 컴퓨터공학과 서건식

운영 체제 HW#7

제출 일자 : 2020/05/14

A. 과제 A

1. 시스템 콜 추가 과정 설명

Step 1 시스템 호출 등록

```
319      common memfd_create      sys_memfd_create
320      common kexec_file_load    sys_kexec_file_load
321      common bpf                 sys_bpf
322      common helloworld          sys_helloworld
323      common get_task_info       sys_get_task_info
```

323 번째 system call로 추가했습니다.

Step 2 시스템 호출 함수 선언

```
asmlinkage long sys_bpf(int cmd, union bpf_attr *attr,
                        unsigned int size);
asmlinkage long sys_helloworld(void);
asmlinkage long sys_get_task_info(void);
```

시스템 호출함수를 선언했습니다.

Step 3 kernel 내부에 시스템 호출 함수 선언

```
#include <linux/linkage.h>
#include <linux/sched.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/unistd.h>

asmlinkage long sys_get_task_info(void)
{
    struct task_struct *info;
    info = get_current();
    printk("-----\n");
    printk("PID : %d\n", info->pid);
    printk("TGID : %d\n", info->tgid);
    printk("PPID : %d\n", info->parent->pid);
    printk("STATE : %d\n", info->state);
    printk("PRIORITY : %d\n", info->prio);
    printk("STATIC_PRIO : %d\n", info->static_prio);
    printk("NORMAL_PRIO : %d\n", info->normal_prio);
    printk("RT_PRIORITY : %d\n", info->rt_priority);
    printk("POLICY : %d\n", info->policy);
    printk("Session ID : %d\n", info->sessionid);

    return 0;
}
```

자세한 내용은 2번에서 다루겠습니다.

```
#
# Makefile for the linux kernel.
#

obj-y      = fork.o exec_domain.o panic.o \
             cpu.o exit.o softirq.o resource.o \
             sysctl.o sysctl_binary.o capability.o ptrace.o user.o \
             signal.o sys.o kmod.o workqueue.o pid.o task_work.o \
             extable.o params.o \
             kthread.o sys_ni.o nsproxy.o \
             notifier.o ksysfs.o cred.o reboot.o \
             async.o range.o groups.o smpboot.o helloworld.o get_task_info.o\

ifdef CONFIG_FUNCTION_TRACER
# Do not trace debug files and internal ftrace files
CFLAGS_REMOVE_cgroup-debug.o = -pg
```

Makefile에 추가

Step 4 compile

```
obj-y += sched/
obj-y += locking/
root@2020osclass:/usr/src/linux-3.18.27/kernel# cd ..
root@2020osclass:/usr/src/linux-3.18.27# ls
arch          drivers      Kconfig      modules.order  secur
block         firmware    kernel        Module.symvers  sound
compile.sh    fs          lib           net             Syste
COPYING       include     MAINTAINERS   README          tags
CREDITS       init        Makefile      REPORTING-BUGS  tools
crypto        ipc         mm            samples         usr
Documentation Kbuild     modules.builtin scripts         virt
root@2020osclass:/usr/src/linux-3.18.27# sh compile.sh
```

Sh compile.sh 명령어 사용

```
#include <stdio.h>
#include <linux/unistd.h>

int main(int argc, char *argv[])
{
    syscall(323);
    return 0;
}
```

테스트 프로그램은 다음과 같습니다.

실행결과:

```
root@2020osclass:~# dmesg --clear
root@2020osclass:~# ./get_task_info
root@2020osclass:~# dmesg
[ 22.955559] -----
[ 22.955568] PID : 1851
[ 22.955574] TGID : 1851
[ 22.955579] PPID : 1824
[ 22.955584] STATE : 0
[ 22.955589] PRIORITY : 120
[ 22.955594] STATIC_PRIO : 120
[ 22.955599] NORMAL_PRIO : 120
[ 22.955604] RT_PRIORITY : 0
[ 22.955609] POLICY : 0
[ 22.955614] Session ID : 1
```

2. 추가한 시스템 콜 설명

```
#include <linux/linkage.h>
#include <linux/sched.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/unistd.h>

asmlinkage long sys_get_task_info(void)
{
    struct task_struct *info;
    info = get_current();
    printk("-----\n");
    printk("PID : %d\n", info->pid);
    printk("TGID : %d\n", info->tgid);
    printk("PPID : %d\n", info->parent->pid);
    printk("STATE : %ld\n", info->state);
    printk("PRIORITY : %d\n", info->prio);
    printk("STATIC_PRIO : %d\n", info->static_prio);
    printk("NORMAL_PRIO : %d\n", info->normal_prio);
    printk("RT_PRIORITY : %d\n", info->rt_priority);
    printk("POLICY : %d\n", info->policy);
    printk("Session ID : %d\n", info->sessionid);

    return 0;
}
```

Sched.h을 include하여 get_current()함수를 호출할 수 있도록 하였습니다. 실제 task_struct의 변수명들을 확인해보면서 각각 맞는 데이터를 출력할수 있도록 printk 함수를 사용했습니다

Info 포인터 변수에 해당 task_struct의 주소값을 받습니다.

구글 드라이브 주소:

<https://drive.google.com/file/d/1x8xwPOEV2Uer4g24-iAyWkuSP7dzcmcQ/view?usp=sharing>