

# HW1

- 제출 정보
  - 학과: 컴퓨터학과
  - 학번: 2020320156
  - 이름: 허준우
  - 제출 날짜: 2023/04/11
  - Freeday사용: 1
- 개발환경
  - Window
  - 나머지 환경은 0차과제와 동일
- 시스템 콜이란?
  - Linux OS의 프로세스 공간 구조
    - User Space
      - 사용자 애플리케이션과 같은 사용자 레벨의 소프트웨어가 실행되는 영역
      - CPU나 메모리 등 시스템 자원에 직접적인 접근이 불가능
      - 운영체제가 제공하는 API(Application Programming Interface)를 통해 시스템 자원을 사용
    - Kernel Space
      - 운영체제 커널(Kernel)과 드라이버, 시스템 서비스 등의 코드가 실행되는 영역
      - 컴퓨터 시스템의 핵심적인 기능을 제공하며, CPU나 메모리, 하드웨어 등의 시스템 자원에 직접적인 접근이 가능
  - Mode
    - User Mode
      - User Space 의 프로세스만 실행 가능한 모드
    - Kernel Mode
      - 모든 권한을 가진 모드

- 시스템 콜

- User Space 의 프로세스가 Kernel Space 의 리눅스 커널에게 프로세스를 요청할 때 사용되는 인터페이스

- 시스템 콜 루틴

1. 사용자 프로세스에서 시스템콜 함수 호출  
: User Space 에서 시스템콜을 호출하면, 해당 함수는 커널 공간에 위치한 시스템콜 테이블(System Call Table)의 인덱스 번호를 파라미터로 전달
2. 커널 모드 전환  
: 커널 스페이스를 사용하기 위해 사용자 모드에서 커널 모드로 mode switch
3. 인터럽트 처리  
: 커널 모드로 전환되면, CPU는 인터럽트를 발생. 이때 발생한 인터럽트는 커널 내부의 인터럽트 처리 루틴(Interrupt Service Routine, ISR)으로 이동하여 처리됩니다.
4. 시스템콜 함수 실행  
: 인터럽트 처리 루틴에서는 시스템콜 테이블에서 인덱스 번호에 해당하는 시스템콜 함수를 찾아 실행
5. 사용자 모드 복귀  
: 시스템콜 함수 실행이 완료되면, 커널 모드에서 사용자 모드로 다시 mode switch. 이후, 시스템콜 함수를 호출한 프로세스는 결과값을 받아 처리

- 관련파일

- syscall\_64.tbl

- /usr/src/linux-4.20.11/arch/x86/entry/syscalls/syscall\_64.tbl
- 시스템 콜 함수들의 이름에 대한 심볼정보(고유번호)를 모아 놓은 파일

```
...
335    common os2023_push    __x64_sys_os2023_push
336    common os2023_pop     __x64_sys_os2023_pop
```

- syscalls.h

- /usr/src/linux-4.20.11/include/linux/syscalls.h
- 추가한 시스템 콜 함수들의 prototype 정의

```
...
asmlinkage void sys_os2023_push(int)
asmlinkage int sys_os2023_pop(void)
```

- 시스템 콜 호출은 int 80인터럽트 핸들러에서 호출
- 인터럽트 핸들러는 assembly 코드로 작성됨
- asmlinkage 를 통해 assembly code에서도 C함수 호출이 가능해짐

#### ◦ oslab\_call\_stack.c

- /usr/src/linux-4.20.11/kernel/oslab\_call\_stack.c
- 시스템 콜의 소스

```
<linux/syscalls.h>
<linux/kernel.h>
<linux/linkage.h>

push, pop 함수 구현
• SYSCALL_DEFINE1(os2023_push, int, a){
...
}
• SYSCALL_DEFINE0(os2023_pop){
...
}
```

- 헤더들을 추가해야 사용가능
- SYSCALL\_DEFINE2(함수 이름, 첫번째 자료형, 첫번째 자료이름, 두번째 자료형, 두번째 자료이름)
  - linux/include/linux/syscalls.h 에 정의되어 있음

#### ◦ Makefile

- /usr/src/linux-4.20.11/kernel/Makefile
- oslab\_call\_stack.o 오브젝트 추가 (.c 파일과 이름이 동일해야함)

```
oslab@oslab-VirtualBox: /usr/src/linux-4.20.11/kernel
File Edit View Search Terminal Help
# SPDX-License-Identifier: GPL-2.0
#
# 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 umh.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 smpboot.o ucount.o oslab_my_stack.o
obj-$(CONFIG_MODULES) += kmod.o
obj-$(CONFIG_MULTIUSER) += groups.o
```

- make

- /usr/src/linux-4.20.11 에서 시행

```
sudo make
sudo make install
```

- 이후 리부트

- test.c

- /아무곳이나 가능
- 정의한 system call 을 유저스페이스 사용하는 테스트 코드

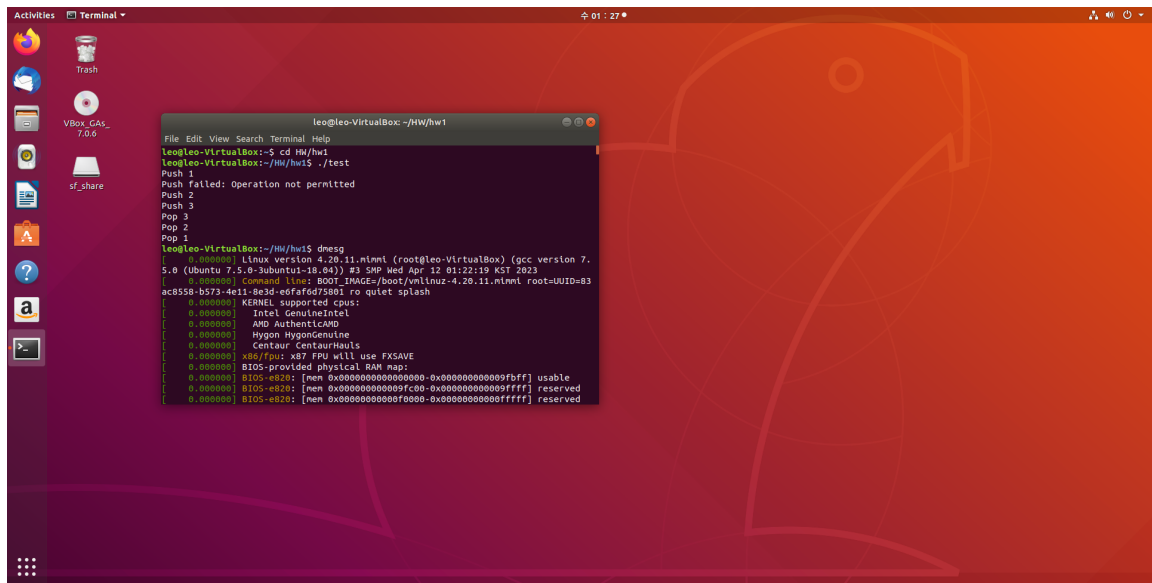
```
#include <stdio.h>
#include <unistd.h>
#include <sys/syscall.h>

#define OS2023_PUSH_SYSCALL_NUM 335
#define OS2023_POP_SYSCALL_NUM 336

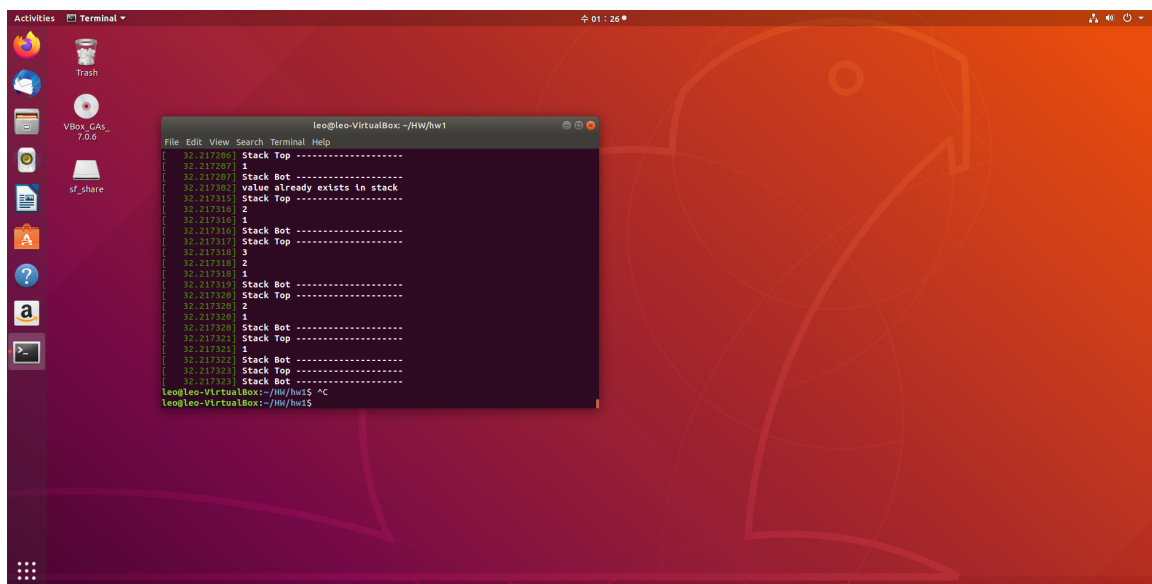
void push(int v) {
    int ret = syscall(OS2023_PUSH_SYSCALL_NUM, v);
    ...
}

void pop() {
    int value = syscall(OS2023_POP_SYSCALL_NUM);
    ...
}
```

- 스냅샷



```
leo@leo-VirtualBox: ~/HW/hw1
File Edit View Search Terminal Help
leo@leo-VirtualBox:~$ cd HW/hw1
leo@leo-VirtualBox:~/HW/hw1$ ./test
Push 1
Push Failed: Operation not permitted
Push 2
Push 3
Pop 3
Pop 2
Pop 1
leo@leo-VirtualBox:~/HW/hw1$ dmesg
[ 0.000000] Linux version 4.20.11.mnml (root@leo-VirtualBox) (gcc version 7.
5.0 (Ubuntu 7.5.0-3ubuntu1-18.04)) #3 SMP Wed Apr 12 01:22:19 KST 2023
[ 0.217200] Command line: BOOT_IMAGE=/boot/vmlinuz-4.20.11.mnml root=UUID=83
ac8558-b573-4e11-8e3d-e0faf6d75801 ro quiet splash
[ 0.000000] KERNEL supported cpus:
[ 0.000000] Intel GenuineIntel
[ 0.000000] AMD AuthenticAMD
[ 0.000000] Hygon HygonGenuine
[ 0.000000] Centaur CentaurHauls
[ 0.000000] x86/fpu: x87 FPU will use FXSAVE
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009f000] usable
[ 0.000000] BIOS-e820: [mem 0x000000000009f000-0x000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000000000] reserved
```



```
leo@leo-VirtualBox: ~/HW/hw1
File Edit View Search Terminal Help
[ 0.217200] Stack Top -----
[ 0.217207] 1
[ 0.217207] Stack Bot -----
[ 0.217208] value already exists in stack
[ 0.217315] Stack Top -----
[ 0.217316] 2
[ 0.217316] 1
[ 0.217316] Stack Bot -----
[ 0.217317] Stack Top -----
[ 0.217318] 3
[ 0.217318] 2
[ 0.217318] 1
[ 0.217319] Stack Bot -----
[ 0.217320] Stack Top -----
[ 0.217320] 2
[ 0.217320] 1
[ 0.217320] Stack Bot -----
[ 0.217321] Stack Top -----
[ 0.217321] 1
[ 0.217322] Stack Top -----
[ 0.217323] Stack Bot -----
[ 0.217323] Stack Top -----
[ 0.217323] Stack Bot -----
leo@leo-VirtualBox:~/HW/hw1$ ^C
leo@leo-VirtualBox:~/HW/hw1$
```

- 문제점 및 해결방법
  - 별다른 문제점이 없었다.