[2023] 운영체제 1차 과제 보고서

학과 : 컴퓨터학과

학번 : 2021320126

이름 : 장지윤

제출 날짜 : 2023-04-09

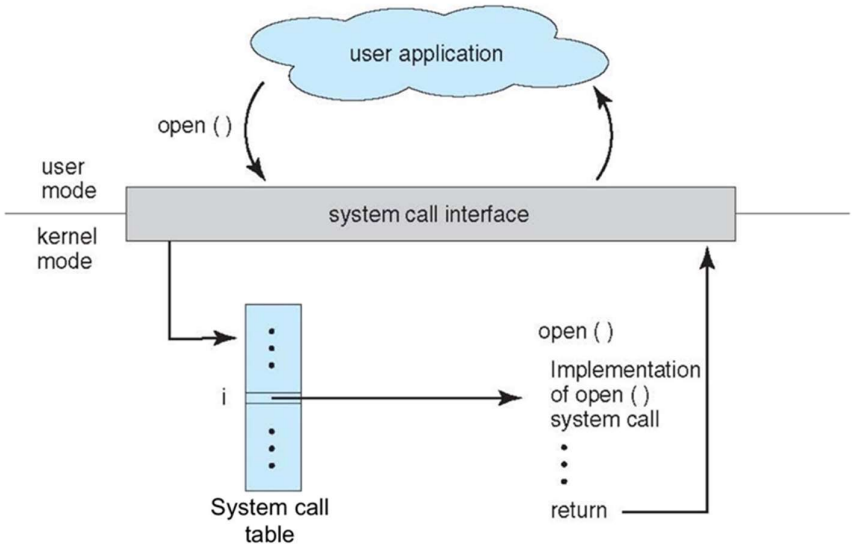
Freeday 사용 일수 : 0일

1. 개발환경

- Host OS : Window 11
- Virtual Box 7.0.6
- Ubuntu 18.04.02 (64bit)
- Linux Kernel 4.20.11

2. Linux System Call

System Call이란 OS Kernel이 제공하는 서비스에 대해, 응용 프로그램의 요청에 따라 커널에 접근하기 위한 인터페이스이다. CPU에는 크게 Kernel mode와 User mode가 존재한다. 사용자가 악의적으로 하드웨어에 접근해 문제가 생기는 경우를 방지하고 시스템을 보호하기 위하여 두 가지 모드는 서로 다른 권한을 갖는다. 일반적인 User application은 User mode에서 동작하는데, 이 Application process에서 파일을 읽고 쓰기 위해서, 커널에서 제공하는 protected service를 이용하기 위해서는 하드웨어에 접근을 해야 한다. 이 접근을 가능하게 하는 통로를 바로 System Call이라고 한다.



위 그림은 User Application에서 파일을 열기 위하여 open()이라는 System Call 함수를 실행한 예시이다. User mode에서 System Call을 호출하면 System Call Interface를 지나 mode의 변경이 일어나며, Kernel area에 존재하는 System call table에서 호출된 함수를 실행하고 난 후 다시 mode를 User mode로 전환하게 된다.

3. 수정 및 작성한 부분과 설명

- (linux)/arch/x86/entry/syscalls/syscall_64.tbl (수정)

Linux kernel에 있는 System Call 함수들의 이름과 심볼의 정보를 담고 있는 파일이다. 스택 구현을 위해 추가할 두 System Call(push, pop)을 각각 Symbol 335, 336번으로 추가했다.

```
347 #oslab
348 335 common  os2023_push      __x64_sys_os2023_push
349 336 common  os2023_pop       __x64_sys_os2023_pop
350
```

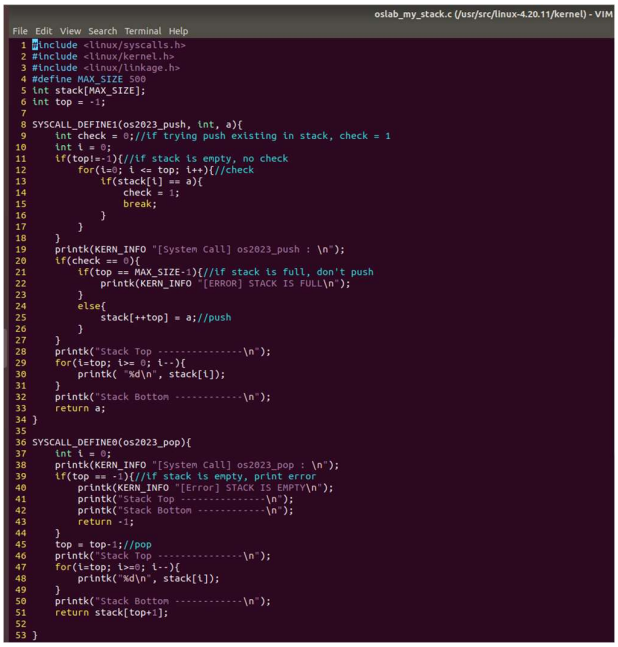
- (linux)/kernel/include/linux/syscalls.h (수정)

System Call 함수들의 Prototype이 정의되는 곳이다. 위에서 추가한 두 개의 함수의 Prototype을 추가해주었다. Assembly에서 C언어 함수 호출이 가능할 수 있도록 asmlinkage를 이용하여 코드를 추가하였다.

```
1299 /*oslab*/
1300 asmlinkage void sys_os2023_push(int);
1301 asmlinkage int sys_os2023_pop(void);
1302
1303
1304 #endif
```

- (linux)/kernel/oslab_my_stack.c (작성)

추가한 System Call(push, pop) 함수들을 구현한 파일이다. 상단에 <Syscalls.h>, <Kernel.h>, <Linkage.h>를 include하였으며, int형 변수 top을 이용하여 stack을 구현하였다. Push 함수에 경우에는 현재 stack에 추가하려는 값이 이미 stack에 존재하는 값인지 판단하여 존재하지 않는 값인 경우 추가하였다. 또한, pop에 경우에는 현재 top에 있는 값을 stack에서 제외하였으며, 만약 stack에 아무 값이 없는 경우에 pop 하는 것을 막기 위해서 에러 메시지를 작성하였다.



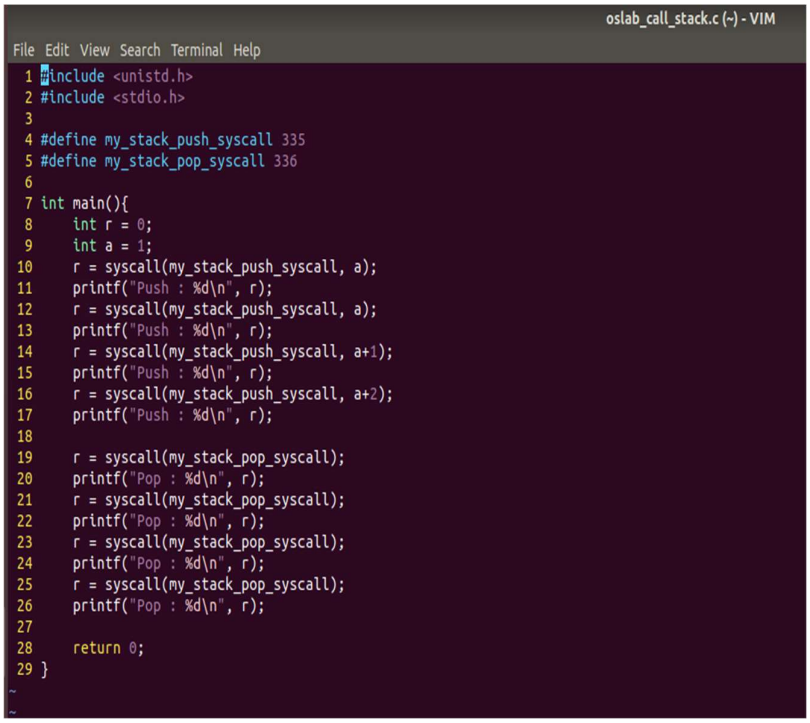
- (linux)/kernel/Makefile (수정)

  Kernel build 과정에서 oslab_my_stack.c가 포함될 수 있도록 obj-y에 해당 오브젝트 파일을 추가하였다.

```
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
```

- (home)/oslab_call_stack.c (작성)

  추가한 System Call을 사용하기 위해 User Application을 작성하였다. Application을 작성하며 중복 값을 push하는 경우를 고려하여 예외 처리가 이루어지는지 확인하였다. 또한 비어 있는 stack에 pop을 하는 경우를 고려하여 예외 처리가 이루어지는지 확인하였다.



4. 실행 결과

-User Application 실행 결과

```
jiyun@jiyun-VirtualBox:~$ ./oslab_call_stack
Push : 1
Push : 1
Push : 2
Push : 3
Pop : 3
Pop : 2
Pop : 1
Pop : -1
```

- Kernel log 출력 (dmesg 실행 결과)

```
[  124.291590] [System Call] os2023_push :
[  124.291591] Stack Top ---------------
[  124.291592] 1
[  124.291592] Stack Bottom ------------
[  124.291696] [System Call] os2023_push :
[  124.291696] Stack Top ---------------
[  124.291697] 1
[  124.291697] Stack Bottom ------------
[  124.291700] [System Call] os2023_push :
[  124.291711] Stack Top ---------------
[  124.291712] 2
[  124.291712] 1
[  124.291713] Stack Bottom ------------
[  124.291715] [System Call] os2023_push :
[  124.291716] Stack Top ---------------
[  124.291716] 3
[  124.291717] 2
[  124.291717] 1
[  124.291717] Stack Bottom ------------
[  124.291719] [System Call] os2023_pop :
[  124.291720] Stack Top ---------------
[  124.291720] 2
[  124.291721] 1
[  124.291721] Stack Bottom ------------
[  124.291723] [System Call] os2023_pop :
[  124.291723] Stack Top ---------------
[  124.291724] 1
[  124.291724] Stack Bottom ------------
[  124.291726] [System Call] os2023_pop :
[  124.291727] Stack Top ---------------
[  124.291727] Stack Bottom ------------
[  124.291729] [System Call] os2023_pop :
[  124.291729] [Error] STACK IS EMPTY
[  124.291730] Stack Top ---------------
[  124.291730] Stack Bottom ------------
jiyun@jiyun-VirtualBox:~$
```

5. 숙제 수행 과정 중 발생한 문제점과 해결방법

반복문에서 새 변수를 선언하여 컴파일 에러가 발생하였다. 반복문 외에서 변수를 선언하여 컴파일 에러를 해결하였다.