os1_2018320161_송대선_보고서

학과 : 컴퓨터학과

학번 : 2018320161

이름 : 송대선

제출 날짜 : 2020-04-30

Freeday 사용 일수 : 0일

개발환경

개발환경

노트북 기종 : XPS 15 9550

CPU : intel i7 inside

RAM : 32GB

OS : Window 10 home 위의 Oracle VM Virtual Box 6.1.6의 Ubuntu 18.04.2

리눅스의 시스템 콜 (호출 루틴 포함)에 대한 설명

시스템 콜에 대한 기본적인 이해는 다음과 같다. 운영체제는 kernel mode와 user mode로 나뉘어 구동되는데, user mode에서 구동되는 user application은 hardware에 직접적으로 접근할 수 없다. Hardware에 직접 접근 할 수 있는 것은 kernel mode의 kernel이다. user application이 하드웨어에 간접적으로 접근하는 그 통로가 system call인 것이다. system call interface를 통해 커널 영역의 기능을 user mode가 사용하는 것이 가능해진다. 즉, system call은 응용프로그램에서 운영체제에게 기능(시스템 자원)을 수행해달라고 하는 하나의 수단인 것이다.

사용자 프로세서가 system call을 요청하면 제어가 커널로 넘어간다. 즉, user mode에서 kernel mode로 전환된다는 것이다. 커널은 내부적으로 각각의 시스템콜을 구분하기 위해 기능별로 고유번호를 할당해 놓는다. 해당번호는 커널내부에 제어루틴을 정의한다. 커널은 요청받은 system call에 대응하는 기능번호를 확인한다. 커널은 그 번호에 맞는 서비스 루틴을 호출하게 된다. 서비스 루틴을 모두 처리하고나면 kernel mode에서 user mode로 다시 넘어온다.

수정 및 작성한 부분과 설명 (이유)

syscall_64.tbl

syscall_64.tbl 파일에 다음의 코드를 추가하였습니다.

```
335 common  oslab_push  __x64_sys_oslab_push
336 common  oslab_pop   __x64_sys_oslab_pop
```

그 이유는 저의 system call 함수인 oslab_push과 oslab_pop에게 고유번호를 부여하고 정의해주기 위함입니다.

syscalls.h

syscalls.h 파일에 다음의 코드를 추가하였습니다.

```
asmlinkage int sys_oslab_push(int);
asmlinkage int sys_oslab_pop(void);
```

asmlinkage를 함수 앞에 선언함으로써 assembly code에서도 C함수 호출을 가능하게 했습니다.

sys_oslab_push, sys_oslab_pop의 prototype을 정의했습니다

my_stack_syscall.c

my_stack_syscall.c 파일은 직접 작성했고, 다음과 같은 흐름으로 진행됩니다.

oslab_push의 세부적인 구현은 다음과 같습니다.

1. 일단 입력받은 int값이 이미 stack에 존재하는지의 여부를 판단하고, 이미 존재한다면 stack에 추가하지 않습니다. 만일 stack에 존재하지 않는다면 step 2로 넘어갑니다.
2. 지금의 stack이 full인지의 여부를 판단합니다. 이는 int variable top과 MAXSIZE의 크기비교로 정합니다. 만일 stack이 full이 아니라면, step 3로 넘어갑니다.
3. 입력받은 int 값을 stack에 추가하고 top의 값을 1 증가시킵니다.

oslab_pop의 세부적인 구현은 다음과 같습니다.

1. stack이 empty인지의 여부를 판단합니다. top과 int -1의 비교를 통해 이를 정합니다. 만일 stack이 empty라면 -1을 return합니다. 만일 stack이 empty가 아니라면 step 2로 넘어갑니다.
2. stack에서 top에 해당하는 값을 저장해두었다가 top을 1만큼 감소시키고 저장해 두었던 값을 return합니다.

Makefile

Makefile의 obj-y부분에 my_stack_syscall.o를 추가하였습니다.

```
...
async.o range.o smpboot.o ucount.o my_stack_syscall.o
```

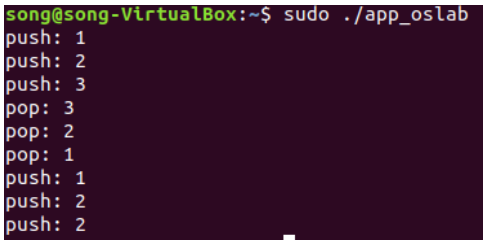
kernel make에서 제가 작성한 my_stack_syscall.c이 object로 같이 커널로 컴파일되게 하려 함입니다.

app_oslab.c

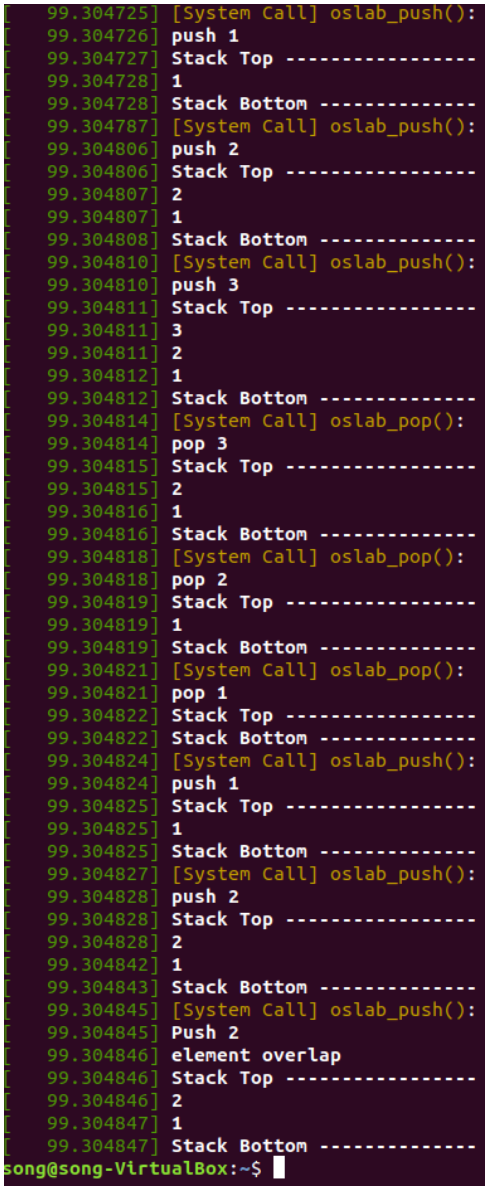
User Application인 app_oslab.c을 작성했고, 다음과 같은 흐름으로 진행됩니다.

1. 1, 2, 3을 순서대로 oslab_push합니다.
2. 3번을 연속으로 oslab_pop합니다.
3. 1, 2, 2을 순서대로 oslab_push합니다.

실행 결과 스냅샷



user application 실행결과



sudo dmesg 결과

숙제 수행 과정 중 발생한 문제점과 해결방법

커널 컴파일이 전혀 되지 않았던 문제가 있었는데, 이는 그 컴퓨터 시스템의 CPU가 AMD 기반이었기에 생긴 문제로 추정됩니다. INTEL기반의 CPU를 사용하는 노트북에서 같은 작업을 실행하였더니 정상적으로 작동하였습니다.