

학과: 서어서문학과

학번: 2015131406

이름: 박가은

제출 날짜: 2020년 5월 2일

Freeday 사용 일수: 2일

개발 환경: Oracle VM Virtual Machine, Linux-4.20.11, Ubuntu-18.04.2 (64bit)

리눅스의 시스템 콜: 운영체제는 커널 모드와 사용자 모드로 나뉜다. 사용자 모드에서 커널에 있는 기능을 사용할 수 있도록 도와주는 것이 시스템 콜이다. 즉, 시스템콜은 하드웨어의 기능을 사용자 모드에서도 편리하게 사용할 수 있는 인터페이스를 제공해준다. 시스템 콜 함수는 각각 고유의 번호를 가지고 있다. 사용자 모드에서 시스템 콜을 이용하고 싶다면 해당 번호로 호출하면 된다.

작업 과정:

```
332 common statx          __x64_sys_statx
333 common io_pgetevents  __x64_sys_io_pgetevents
334 common rseq           __x64_sys_rseq
335 common gaeunfalo_push __x64_sys_gaeunfalo_push
336 common gaeunfalo_pop  __x64_sys_gaeunfalo_pop
#
# x32-specific system call numbers start at 512 to avoid cache impact
# for native 64-bit operation. The __x32_compat_sys stubs are created
# on-the-fly for compat_sys_*( ) compatibility system calls if X86_X32
# is defined.
#
```

시스템콜의 주소를 저장하는 테이블(syscall_64.tbl)에 새로 추가할 시스템콜과 시스템콜의 번호를 선언했다.

```
if (personality != 0xffffffff)
    set_personality(personality);

return old;
}

asmlinkage int sys_gaeunfalo_push(int);
asmlinkage int sys_gaeunfalo_pop(void);

#endif
```

시스템콜 함수의 prototype을 선언했다. Asmlinkage를 쓴 이유는 시스템 콜은 어셈블리 코드로 작성된 인터럽트 핸들러에서 이루어진다. Asmlinkage를 쓰면 어셈블리 코드로 작성된 프로그램에서도 c언어로 작성된 함수를 불러올 수 있기 때문이다.

```
GAEUNFALO [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

Activities  Text Editor
Open  gaeunfalo_just/src/linu

#include<linux/syscalls.h>
#include<linux/kernel.h>
#include<linux/linkage.h>

#define MAXSIZE 1000

int stack[MAXSIZE];
int top = -1;

SYSCALL_DEFINE1(gaeunfalo_push, int, a)
{
    if(top == -1) // if stack is empty, Push 'a'
    {
        stack[++top] = a;
        printk(KERN_INFO "[System Call] gaeunfalo_push(): ---- %d", a); // Print stack in Kerr
        printk("Stack Top-----\n");
        int j; // to print stack in Kernel
        for(j = top; j >= 0; j--)
        {
            printk("%d\n", stack[j]);
        }
        printk("Stack Bottom-----\n");
        return a;
    }

    else if(top != MAXSIZE - 1) // If stack is not full
    {
        int i; // To check whether there exists overlapped value in stack or not
        for(i = 0; i <= top; i++)
        {
            if(stack[i] == a) // If there is already same value with a, then print Error
            {
                printk(KERN_INFO "[System Call] gaeunfalo_push(): ---- Error");
                printk("Stack Top-----\n");
                int j;
                for(j = top; j >= 0; j--)
                {
                    printk("%d", stack[j]);
                }
                printk("Stack Bottom-----\n");
                return -1;
            }
        }
        stack[++top] = a; // If there isn't, stack[top] = a
        printk(KERN_INFO "[System Call] gaeunfalo_push(): ---- %d", a);
        printk("Stack Top-----\n");
        int j;
        for(j = top; j >= 0; j--)
        {
            printk("%d\n", stack[j]);
        }
        printk("Stack Bottom-----\n");
        return a;
    }
}
```

```
GAEUNFALO [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

Activities  Text Editor
Open

        printk("%d", stack[j]);
    }
    printk("Stack Bottom-----\n");
    return -1;
}

stack[++top] = a; // If there isn't, stack[top] = a
printk(KERN_INFO "[System Call] gaeunfalo_push(): ---- %d", a);
printk("Stack Top-----\n");
int j;
for(j = top; j >= 0; j--)
{
    printk("%d", stack[j]);
}
printk("Stack Bottom-----\n");
return a;
}
printk(KERN_INFO "[System Call] gaeunfalo_push(): ---- Error");
printk("Stack Top-----\n");
int j;
for(j = top; j >= 0; j--)
{
    printk("%d", stack[j]);
}
printk("Stack Bottom-----\n");
return -1; // Stack is already full
}

SYSCALL_DEFINE0(gaeunfalo_pop)
{
    if(top == -1) // If stack is empty, print Error
    {
        printk(KERN_INFO "[System Call] gaeunfalo_pop(): ---- Error");
        printk("Stack Top-----\n");
        printk("Stack Bottom-----\n");
        return -1;
    }

    printk(KERN_INFO "[System Call] gaeunfalo_pop(): ---- %d", stack[top]);
    printk("Stack Top-----\n");
    int j;
    for(j = top; j >= 0; j--)
    {
        printk("%d", stack[j]);
    }
    printk("Stack Bottom-----\n");
    return stack[top--]; // If stack is not empty, pop stack[top]
}
```

그리고 시스템콜의 함수를 작성했다.

새로 만든 시스템콜을 호출하는 application도 만들었다. 중복을 허용하지 않는지 확인하기 위한 작업도 넣었다.

GAEUNFALO [실행 중] - Oracle VM VirtualBox

파일 머신 보기 입력 장치 도움말

현재 가상 머신의 게스트 운영 체제에서 **마우스 통합**을 지원합니다. 게스트 운영 체제에서 마우스를 사용하기 위해서 포인터를

```
Open
#include<unistd.h>
#include<stdio.h>

#define gaeunfalo_stack_push 335
#define gaeunfalo_stack_pop 336

int main()
{
    int i;
    int top = -1;

    for(i = 0; i < 5; i++)
    {
        int a;
        a = syscall(gaeunfalo_stack_push, (i + 1) / 2);
        if(a == -1)
        {
            printf("Push: Error\n");
        }
        else
        {
            printf("Push: %d\n", a);
            top++;
        }
    }

    for(i = top; i >= -1; i--)
    {
        int a;
        a = syscall(gaeunfalo_stack_pop);
        if(a == -1)
        {
            printf("Pop: Error\n");
        }
        else
        {
            printf("Pop: %d\n", a);
            top--;
        }
    }

    return 0;
}
```

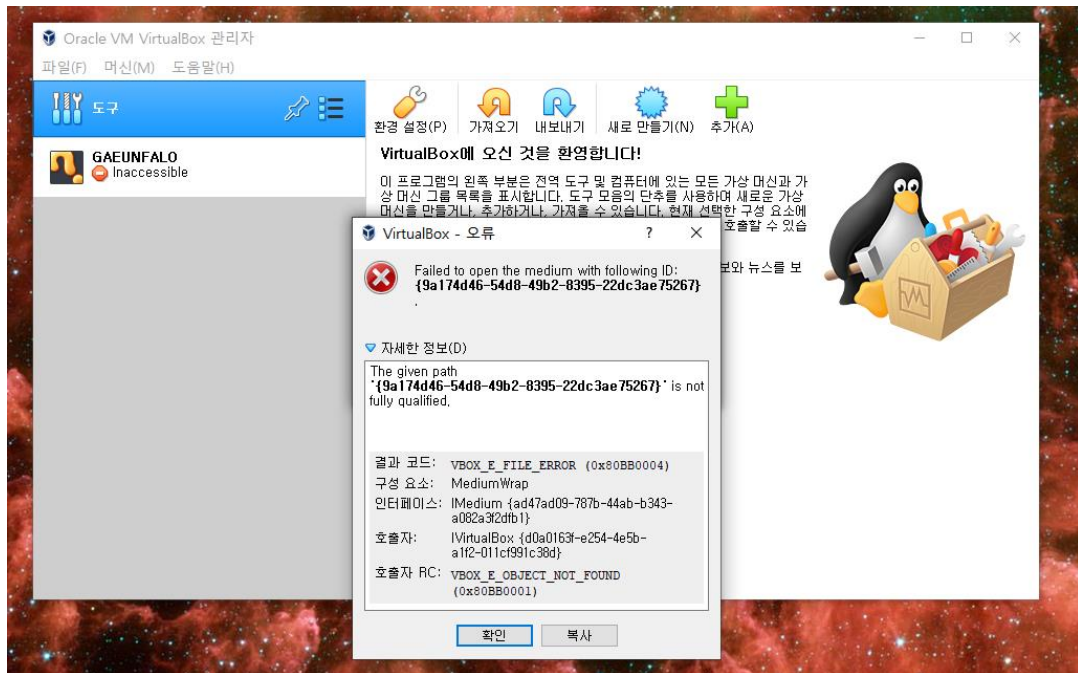
실행결과:

```
gaeunfalo@gaeunfalo-VirtualBox:~$ ./app_gaeunfalo
Push: 0
Push: 1
Push: Error
Push: 2
Push: Error
Pop: 2
Pop: 1
Pop: 0
Pop: Error
```

```
32.224687] [System Call] gaeunfalo_push(): ---- 0
32.224688] Stack Top-----
32.224689] 0
32.224689] Stack Bottom-----
32.224744] [System Call] gaeunfalo_push(): ---- 1
32.224745] Stack Top-----
32.224745] 1
32.224745] 0
32.224746] Stack Bottom-----
32.224747] [System Call] gaeunfalo_push(): ---- Error
32.224748] Stack Top-----
32.224748] 1
32.224748] 0
32.224749] Stack Bottom-----
32.224750] [System Call] gaeunfalo_push(): ---- 2
32.224750] Stack Top-----
32.224751] 2
32.224751] 1
32.224751] 0
32.224752] Stack Bottom-----
32.224753] [System Call] gaeunfalo_push(): ---- Error
32.224753] Stack Top-----
32.224753] 2
32.224754] 1
32.224754] 0
32.224754] Stack Bottom-----
32.224756] [System Call] gaeunfalo_pop(): ---- 2
32.224756] Stack Top-----
32.224756] 1
32.224756] 0
32.224757] Stack Bottom-----
32.224758] [System Call] gaeunfalo_pop(): ---- 1
32.224758] Stack Top-----
32.224758] 0
32.224759] Stack Bottom-----
32.224760] [System Call] gaeunfalo_pop(): ---- 0
32.224760] Stack Top-----
32.224760] Stack Bottom-----
32.224761] [System Call] gaeunfalo_pop(): ---- Error
32.224762] Stack Top-----
32.224762] Stack Bottom-----
gaeunfalo@gaeunfalo-VirtualBox:~$
```

실행과정 중 어려움:

- 1) 갑작스러운 가상머신의 inaccessible 상태. 아무리 검색해보도 뚜렷한 해결책이 나오지 않아 리눅스 재설치만 3번을 했다. 아직도 원인을 모르고 있다. 실수로 이미지 파일을 두 번 설치했는데 그것 때문에 그럴 수도 있겠다는 생각을 했다.



- 2) 시스템콜 함수를 작성하고 Makefile에 오브젝트 파일도 추가한 뒤에 `sudo make`, `sudo make install`을 해도 push를 하면 결과로 -1만 출력됐다. 결국 `sudo make -j 4`, `sudo make install`을 한 뒤에 `sudo reboot`로 재부팅을 하면 된다는 사실을 깨달았다.