운영체제 1차 과제 보고서

컴퓨터학과 202032012 조민규

2023-04-02 Freeday 미사용

개발환경

window10에 Oracle VM Virtual Machin을 설치해 Linux Ubnutu 설치

리눅스의 시스템콜

사용자의 응용프로그램이 보안등의 문제로 커널의 기능을 직접 사용하지 않고 시스템 콜을 호출하는 방법을 통해 시스템 콜은 응용프로그램과 커널 사이에 인터페이스 역할을 해준다.

syscall_64.tbl 파일에 시스템콜과 시스템콜 번호가 매핑되어 있어 응용프로그램에선 시스템콜 번호를 사용하여 호출할 수 있다.

syscalls.h에는 시스템콜의 프로토타입이 적혀있어 반환형과 인자들을 정의할 수 있다.

이후 DEFINEx매크로함수를 이용하여 실제 동작을 구현할 수 있다.

수정 및 작성한 부분과 설명

1. syscall_64.tbl 파일

```
342 331 common  pkey_free       __x64_sys_pkey_free
343 332 common  statx           __x64_sys_statx
344 333 common  io_pgetevents       __x64_sys_io_pgetevents
345 334 common  rseq            __x64_sys_rseq
346
347 #oslab
348 335 common  os2023_push     __x64_sys_os2023_push
349 336 common  os2023_pop      __x64_sys_os2023_pop
350
351 #
352 # x32-specific system call numbers start at 512 to avoid cache impact
353 # for native 64-bit operation. The __x32_compat_sys stubs are created
354 # on-the-fly for compat_sys_*() compatibility system calls if X86_X32
355 # is defined.
356 #
357 512 x32 rt_sigaction        __x32_compat_sys_rt_sigaction
358 513 x32 rt_sigreturn        sys32_x32_rt_sigreturn
359 514 x32 ioctl           __x32_compat_sys_ioctl
```

위의 348, 349행을 삽입하여 시스템콜 번호와 새로 만든 시스템콜을 매핑시켜준다

2. syscall.h 파일

```
1289 static inline unsigned int ksys_personality(unsigned int personality)
1290 {
1291     unsigned int old = current->personality;
1292
1293     if (personality != 0xffffffff)
1294         set_personality(personality);
1295
1296     return old;
1297 }
1298
1299 /*oslab*/
1300 asmlinkage void sys_os2023_push(int);
1301 asmlinkage int sys_os2023_pop(void);
1302 #endif
                                                    1302,1        Bot
```

새로 만든 시스템콜의 프로토타입을 설정한다.

3. oslab_my_stack.c

```
SYSCALL_DEFINE1(os2023_push, int, a){
    if(top>30){
        printk("no memory to push");
        return;
    }
    int i;
    for(i=0;i<top;i++){
        if(a==stack[i])
            return;
    }
    stack[top++]=a;
    showstack();
}
//stack pop
SYSCALL_DEFINE0(os2023_pop){
    if(top==0){
        printk("no element to pop");
        return -1;
    }
    int value = stack[--top];
    showstack();

    return value;
}
                                                    41,1          Bot
```

실제 시스템콜을 구현하는 부분으로 크기가 30짜리 int형 배열을 전역변수로 할당 후 push할 위치를 가리키는 int형 전역변수 top도 0으로 선언 및 초기화한다.

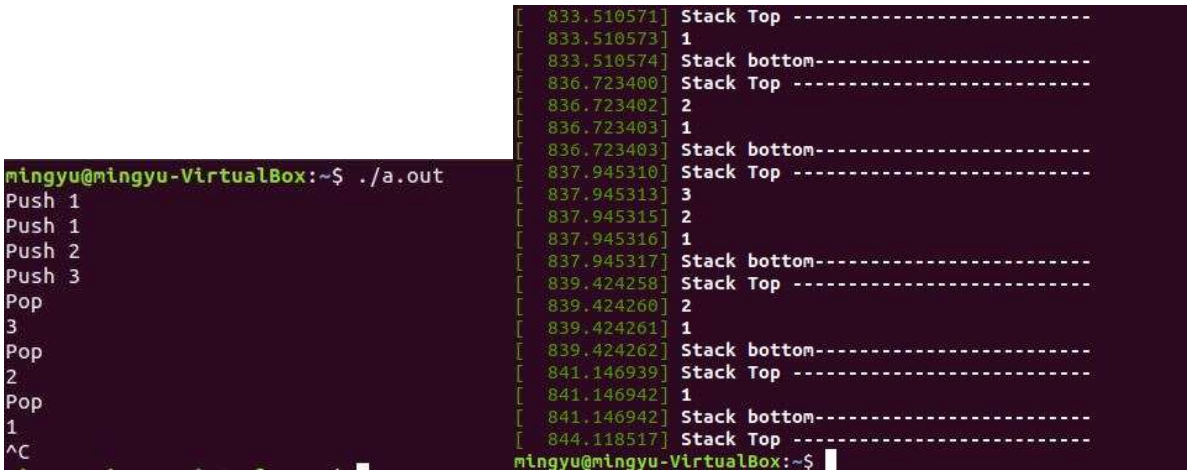
이후 중복을 피하는 코드, 스택 overhead가 일어났을 때, 스택이 empty시 pop하려는 경우 들을 간단하게 처리한다.

4. oslab_call_stack.c

```
int main(void){
    char inst[5];//string of instruction(push or pop)
    char integer[4];// integer to push(type string)
    int x;//integer to push(type integer)
    while(1){
        scanf("%s", inst);
        if(!strcmp(inst,"Push")){
            scanf("%s",integer);
            x=char2int(integer);
            syscall(my_stack_push, x);
        }
        else if(!strcmp(inst,"Pop")){
            int r = syscall(my_stack_pop);
            if(r==-1){
                printf("Pop Error");
                break;
            }
            printf("%d\n",r);
        }
        else{
            printf("unvalid instruction!\n");
        }
    }
}
```

실제 시스템콜을 호출하는 응용프로그램 부분으로 표준입력으로 명령어를 입력받아 수행한다. 따라서 명령어를 처리하는 부분 외에는 시스템콜이 처리한다.

실행결과



문제점과 해결방법

재부팅 시 계속 5.4.20.-generic 의 커널로 부팅되어 새로 작성한 시스템콜 호출에 실패했었는데 파일에 적힌대로 grub에서 linux-4.20.11.oslab커널로 부팅하여 해결했다.