운영체제

과목명	운영체제
교수명	김철홍
학 과	컴퓨터학부
학 번	20192393
이 름	김현우
제출일	2023.10.05

1 . 제출하는 소스코드 파일 리스트

- syscall_64.tbl : /usr/src/linux/linux-5.15.120/arch/x86/entry/syscalls
- syscalls.h : /usr/src/linux/linux-5.15.120/include/linux
- sys_print_reverse.c : /usr/src/linux/linux-5.15.120/kernel/
- sys_plus.c : /usr/src/linux/linux-5.15.120/kernel/
- sys_minus.c : /usr/src/linux/linux-5.15.120/kernel/
- Makefile : /usr/src/linux/linux-5.15.120/kernel/
- unistd.h : /usr/src/linux/linux-5.15.120/include/uapi/asm-generic/
- syscall_test.c : /home/os20192393

2. 작업 설명

커널을 변경하여 커널 내부에 새로운 시스템 콜 함수를 추가한다.

추가한 시스템 콜 함수은 string 입력 시 역순 출력하는 함수, 덧셈 기호와 숫자 두개 입력 시 뺄셈하는 함수, 뺄셈 기호와 숫자 두개 입력 시 덧셈하는 함수 3 개를 추가한다.

추가한 시스템 콜을 호출해서 동작하는 프로그램 만든다.

syscall_test.c

```c
#include <stdio.h>
#include <linux/kernel.h>
#include <sys/syscall.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define BUFFER_SIZE 1024

#define __NR_print_hello 449
#define __NR_print_reverse 450 // 역순 출력하는 시스템 콜 테이블에 추가한 번호
#define __NR_plus 451 //  뺄셈하는 시스템 콜 테이블에 추가한 번호
#define __NR_minus 452 // 덧셈하는 시스템 콜 테이블에 추가한 번호

void delete_space(char s[]); // 공백을 제거하는 함수

int main() {

    char input[BUFFER_SIZE]; // 입력 문자열을 저장할 배열
    char num1[BUFFER_SIZE]; // 첫 번째 숫자를 저장할 배열
    char num2[BUFFER_SIZE]; // 두 번째 숫자를 저장할 배열
    char operator; // 연산자를 저장할 변수
    int i,j = 0;

    while(1) {
        printf("Input: ");
        fgets(input, sizeof(input), stdin); // 사용자로부터 입력 받음

        // 개행 문자 제거
        input[strcspn(input, "\n")] = '\0';

        // 공백 제거
        for (i = 0; input[i] != '\0'; i++) {
            if (input[i] != ' ') {
                break;
            }
        }
```

```
// 공백을 제거
delete_space(input);

// 개행 입력시 종료
if(strcmp(&input[0], "") == 0) {
    exit(0);
}


int operatorNum = 0; // 연산자 개수 저장하는 변수
// 연산자 개수 찾기
for (i = 0; input[i] != '\0'; i++) {

    if(isdigit(input[i]) == 0) {
        operatorNum++;
    }
}
// 연산자 위치 찾기
for (i = 0; input[i] != '\0'; i++) {

    if(isdigit(input[i]) == 0) {
        operator = input[i];
        break;
    }
}

// 연산자를 찾았을 경우
if (input[i] != '\0') {
    // 숫자 부분을 추출
    for (j = 0; j < i; j++) {
        num1[j] = input[j];
    }
    num1[j] = '\0'; // 문자열 끝을 표시

    // 두 번째 숫자 부분 추출
    for (i = i + 1, j = 0; input[i] != '\0'; i++) {
        num2[j] = input[i];
        j++;
    }
    num2[j] = '\0'; // 문자열 끝을 표시
}
// 연산자를 찾지 못한 경우
else {
    strcpy(num1, input);
    operator = '\0';
    strcpy(num2, "");
}
```

```
        // 문자열을 정수로 변환하여 계산
        int operand1 = atoi(num1);
        int operand2 = atoi(num2);
        int result;
        char resultString[BUFFER_SIZE];


        if( operator == '\0') { // 연산자가 없는 경우
            syscall(__NR_print_reverse,num1,resultString); // 역순 출력 시스템 콜 호출

            // 결과 출력
            printf("Output: %s\n", resultString);
        }
        else { // 연산자가 있는 경우
            if(strcmp(num2, "") == 0) {
                printf("Wrong Input!\n");
            }
               else if(strcmp(num1, "") == 0) {
                      printf("Wrong Input!\n");
            }
            else {
                if(operatorNum != 1) {
                    printf("Wrong Input!\n");
                }
                else {
                    if (operator == '+') {
                        syscall(__NR_plus,operand1,operand2,&result); // 덧셈 시스템
콜 호출

                        // 결과 출력
                        printf("Output: %d\n", result);
                    }
                    else if (operator == '-') {
                        syscall(__NR_minus,operand1,operand2,&result); // 뺄셈 시스템
콜 호출

                        // 결과 출력
                        printf("Output: %d\n", result);
                    }
                    else {
                        printf("Wrong Input!\n");
                    }
                }
            }
        }
    }

    return 0;
}
```

```c
void delete_space(char s[]) { //공백을 제거하는 함수
    char tmp[BUFFER_SIZE];
    int i, k = 0;

    for (i = 0; i < (int)strlen(s); i++) {
        if (!isspace((unsigned char)s[i])) { //만약 공백을 만난다면
            tmp[k++] = s[i]; //공백대신 문자를 저장
        }
    }

    tmp[k] = '\0';
    strcpy(s, tmp);
}
```

sys_print_reverse.c

```c
#include <linux/kernel.h>
#include <linux/linkage.h>
#include <linux/syscalls.h>
#include <linux/init.h>
#include <linux/fs.h>
#include <linux/miscdevice.h>
#include <linux/mutex.h>
#include <asm/uaccess.h>
#include <linux/slab.h>

#define BUFFER_SIZE 1024

SYSCALL_DEFINE2(print_reverse,char *,from_user,char *,to_user){ // 시스템콜 함수 호출할때 인자가 2 개 필요 인자로 데이터를 가져오는 공간과 데이터를 써 넣을 사용자 공간
    char reversed_str[BUFFER_SIZE]; // 입력 문자열을 저장하는 배열
    char result_str[BUFFER_SIZE] = ""; // 역순 문자열을 저장하는 배열

    copy_from_user(reversed_str, from_user, BUFFER_SIZE); // 사용자 메모리 블록 데이터를 커널 메모리 블록 데이터에 써넣기

    int length;
    length = strlen(reversed_str);

    int i = 0;
    for (i = length - 1; i >= 0; i--) { // 입력 문자열 역순으로 저장하기
        strncat(result_str, &reversed_str[i], 1);
    }

    copy_to_user(to_user, result_str, BUFFER_SIZE); // 커널 메모리 블록 데이터를 사용자 메모리 블록 데이터에 써넣기

    return 0;
}
```

sys_plus.c

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

SYSCALL_DEFINE3(plus,int,x,int,y,int *,to_user){ // 시스템콜 함수 호출할때 인자가 3 개 필요 인자로 계산할 int 값 2 개와 데이터를 써 넣을 사용자 공간
        int res = x-y; // 뺄셈하기
        put_user(res,to_user); // 사용자 공간의 데이터를 써 넣기
        return res;
}
```

sys_minus.c

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

SYSCALL_DEFINE3(minus,int,x,int,y,int *,to_user){ // 시스템콜 함수 호출할때 인자가 3 개 필요 인자로 계산할 int 값 2 개와 데이터를 써 넣을 사용자 공간
        int res = x+y; // 덧셈하기
        put_user(res,to_user); // 사용자 공간의 데이터를 써 넣기
        return res;
}
```

3. 실행 화면 캡쳐

자릿수 역순 변경 실행 화면

```
root@20192393:/home/os20192393# ./syscall_test
Input: 123
Output: 321
Input: 348957
Output: 759843
Input:
root@20192393:/home/os20192393#
```

부호 바꾸어 덧셈, 뺄셈 실행 화면

```
root@20192393:/home/os20192393# ./syscall_test
Input: 123+111
Output: 12
Input: 12350-13253
Output: 25603
Input:
root@20192393:/home/os20192393#
```

앞의 두가지 경우 외 스트림 입력시 실행 화면

```
root@20192393:/home/os20192393# ./syscall_test
Input: abcde
Wrong Input!
Input: 12++
Wrong Input!
Input: 1023+ -1023
Wrong Input!
Input: 123+123-123
Wrong Input!
Input:
```