

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Вычислительной Техники

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Организация процессов и программирование в среде
Linux»
ТЕМА: «ОБРАБОТКА СИГНАЛОВ»

Студент гр. 8308

Тайсумов И.И.

Преподаватель

Разумовский Г.В.

Санкт-Петербург

2021

Цель работы.

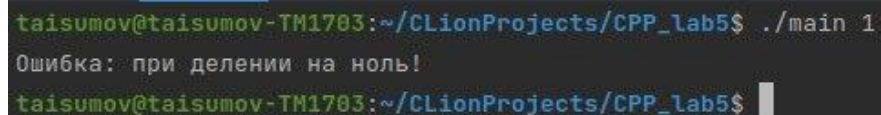
Знакомство с механизмом сигналов и способами их обработки.

Задание.

1. Написать программу, которая реагирует на ошибки при выполнении операции деления и неверном использовании указателя (деление на ноль, нарушение защиты памяти). При обнаружении ошибки программа должна передать управление функции, которая выведет сообщение и завершит работу программы с кодом ошибки (1 или 2). Тип ошибки, который должна зафиксировать программа, задается как параметр при ее запуске.
2. Откомпилировать программу и дважды запустить ее с разными значениями типа ошибки.

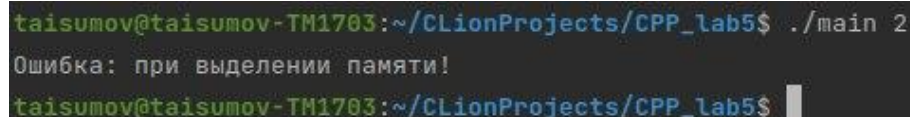
Обработка результатов эксперимента.

Ниже приведены выводы программы для случая, когда программе поступает сигнал о делении на ноль, а также второй случай, когда в программе появляется ошибка при выделении памяти:



```
taisumov@taisumov-TM1703:~/CLionProjects/CPP_lab5$ ./main 1
Ошибка: при делении на ноль!
taisumov@taisumov-TM1703:~/CLionProjects/CPP_lab5$
```

Рисунок 1. Обработка ошибки деления на ноль



```
taisumov@taisumov-TM1703:~/CLionProjects/CPP_lab5$ ./main 2
Ошибка: при выделении памяти!
taisumov@taisumov-TM1703:~/CLionProjects/CPP_lab5$
```

Рисунок 2. Обработка ошибки при выделении памяти

Текст программы и распечатки вывода приведены в приложении.

Вывод.

В ходе выполнения лабораторной работы были изучены и использованы системные функции, обеспечивающие обработку сигналов; разработана программа, порождающая две типа ошибок на выбор пользователя и завершающая работу программы с определенным кодом в зависимости от типа ошибки.

ПРИЛОЖЕНИЕ

main.cpp:

```
/**
 * Ислам Тайсумов, группа 8308
 *
 * Компиляция программы:
 * 1. g++ -o main main.cpp
 * 2.1 ./main 1 (для деления на ноль)
 * 2.2 ./main 2 (для ошибки памяти)
 */

#include <iostream>
#include <signal.h>

using namespace std;

void catchError(int err) {
    cout << "Ошибка: ";
    switch(err){
        case SIGFPE:
            cout << "при делении на ноль!" << endl;
            exit(1);
            break;
        case SIGSEGV:
            cout << "при выделении памяти!" << endl;
            exit(2);
            break;
        default:
            break;
    }
}

int main(int arc, char* argv[]) {
    //SIGFPE(8) - неверная операция (переполнение, деление на 0)
    signal(SIGFPE, catchError);
    //SIGSEGV(11) - нарушение защиты памяти
    signal(SIGSEGV, catchError);

    int numOfError = atoi(argv[1]);
    int err;
    char* c = "Hello";

    switch (numOfError) {
        case 1:
            err = 228 / 0;
            cout << "Деление успешно завершено!";
            break;
        case 2:
            c[10] = 'z';
    }
}
```

```
        cout << "Преобразование строки произошло успешно!";  
        break;  
    default:  
        break;  
}  
  
cout << "all OK";  
  
return 0;  
}
```