

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

Петров Г.А.

Преподаватель

Разумовский Г.В.

Санкт-Петербург

2021

Цель работы

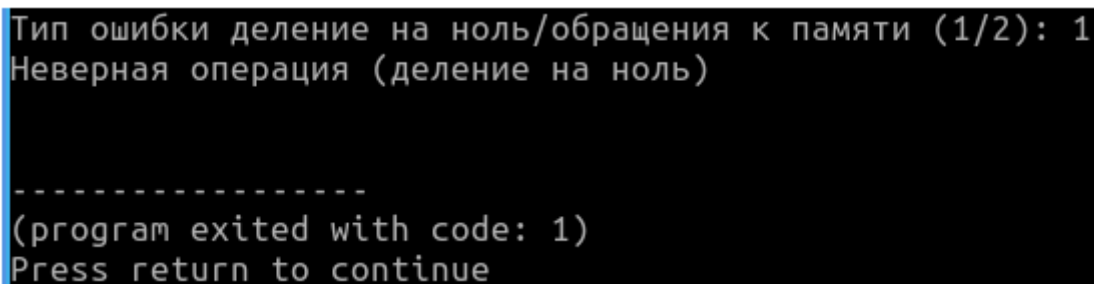
Целью лабораторной работы является знакомство с механизмом сигналов и способами их обработки.

Задание

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.

Примеры выполнения программы

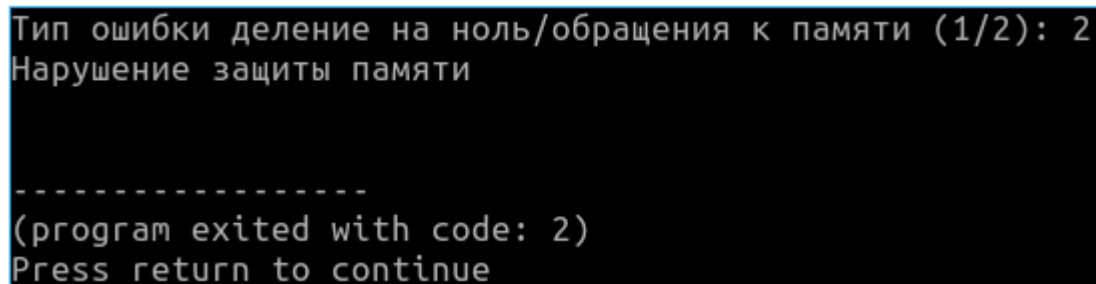
Программа была разработана и откомпилирована. После чего программа была запущена на двух вариантах входных данных (деление на ноль и неверное обращение к памяти). Результаты работы программы в обоих случаях приведены на рисунках 1 и 2.



```
Тип ошибки деление на ноль/обращения к памяти (1/2): 1
Неверная операция (деление на ноль)

-----
(program exited with code: 1)
Press return to continue
```

Рисунок 1



```
Тип ошибки деление на ноль/обращения к памяти (1/2): 2
Нарушение защиты памяти

-----
(program exited with code: 2)
Press return to continue
```

Рисунок 2

Исходный код программ

lab4.cpp

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

//ф-ия обработки сигнала
void mysighandler(int signal)
{
    switch(signal)
    {
        case SIGFPE:
            std::cout<<"Неверная операция (деление на ноль)"<<std::endl;
            exit(1);
            break;
        case SIGSEGV:
            std::cout<<"Нарушение защиты памяти"<<std::endl;
            exit(2);
            break;
    }
}

int main()
{
    //переназначение обработки сигнала на новую функцию
    signal(SIGFPE, mysighandler);
    signal(SIGSEGV, mysighandler);

    int Z;
    int i=0;
    int *p=NULL;
    std::cout<<"Тип ошибки деление на ноль/обращения к памяти (1/2): ";
    std::cin>>Z;
    switch(Z)
    {
        case 1:
            i=0/0;
            break;
        case 2:
            *p=555;
            break;
    }
}
```

Вывод

При выполнении лабораторной работы изучены и использованы системные функции, обеспечивающие обработку сигналов; разработана программа, порождающая два типа ошибки на выбор пользователя и завершающая работу программы с определенным кодом в зависимости от типа ошибки.