

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
среде Linux»
Тема: Обработка сигналов

Студент гр. 9308

Яловега Н.В.

Преподаватель

Разумовский Г.В.

Санкт-Петербург

2022

Цель работы.

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.

Выполнение работы

Компиляция программы:

```
# kivyfreakt @ hpomen in ~/documents/eltech/linux/lab5 on git:master x [12:33:52]
$ g++ main.cpp -o main
main.cpp: In function 'int main(int, char**)':
main.cpp:23:17: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
   23 |     char *str = "test";
      |               ^~~~~~
main.cpp:34:17: warning: division by zero [-Wdiv-by-zero]
   34 |     num /= 0; // деление на ноль )
      |           ^~
```

Рисунок 1. Компиляция программ

Как видим, компилятор сообщает о возможных ошибках.

Запуск программы с разными входными параметрами.

```
# kivyfreakt @ hpomen in ~/documents/eltech/linux/lab5 on git:master x [12:33:59]
$ ./main
Incorrect number of arguments!
Example: ./main error_type

# kivyfreakt @ hpomen in ~/documents/eltech/linux/lab5 on git:master x [12:35:26]
$ ./main 1
SIGFPE signal; return 1

# kivyfreakt @ hpomen in ~/documents/eltech/linux/lab5 on git:master x [12:35:29] C:1
$ ./main 2
SIGSEGV signal; return 2

# kivyfreakt @ hpomen in ~/documents/eltech/linux/lab5 on git:master x [12:35:31] C:2
$ |
```

Рисунок 2. Запуск программы

Как видим, программа при получении сигнала SIGFPE возвращает код 1, а при получении сигнала SIGSEGV возвращает код 2.

Вывод

В ходе работы были изучены механизмы отправки и обработки сигналов в программах.

Приложение

main.cpp

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

void signal_handler(int sig)
{
    switch (sig)
    {
        case SIGFPE:
            std::cout << "SIGFPE signal; return 1" << std::endl;
            exit(1);
            break;
        case SIGSEGV:
            std::cout << "SIGSEGV signal; return 2" << std::endl;
            exit(2);
            break;
    }
}

int main(int argc, char* argv[])
{
    int error_type,
        num = 1917;
    char *str = "test";

    // установка обработчиков сигналов
    signal(SIGFPE, signal_handler);
    signal(SIGSEGV, signal_handler);

    if(argc == 2)
    {
        error_type = atoi(argv[1]);

        if(error_type == 1)
            num /= 0; // деление на ноль )
        else if(error_type == 2)
            str[num] = 'y'; // запись в раздел памяти только для чтения
        else
            std::cout << "Illigal agrument! error_type should be 1 or 2" <<
std::endl;
    }
    else
```

```
    {
        std::cout << "Incorrect number of arguments!" << std::endl;
        std::cout << "Example: " << argv[0] << " error_type" << std::endl;
    }

    return 0;
}
```