

Федеральное агентство связи  
Сибирский государственный университет телекоммуникаций и информатики

Лабораторная работа №2

Выполнил: студент группы ИП-211  
Оганесян Альберт  
Лацук Андрей  
Проверил:  
Профессор кафедры ПМиК  
Малков Е. А.

Новосибирск 2024

**Цель работы:** Знакомство с инструментарием для разработки программ на платформе **Linux**

**Задание:**

1. Написать программу для манипуляции данными на основе рекурсивных структур, реализовать функции вставки, удаления и навигации для списка, реализовать сериализацию списка;
2. Провести трассировку программы с помощью GDB.

**Инструментарий:**

Редактор кода VIM, компилятор GCC, отладчик GDB.

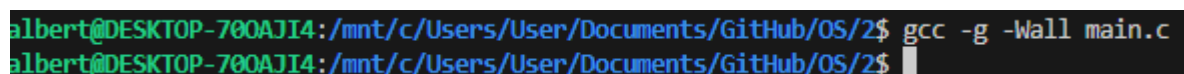
**Ход работы:**

1. В качестве программы для отладки, возьмем программ, написанную в первой лабораторной работе

```
int main() {  
    //  
    Student* studentList = NULL;  
  
    //  
    int grades1[] = {5, 4, 5, 5, 3};  
    int grades2[] = {3, 4, 2, 4, 5};  
  
    //  
    Student* student1 = createStudent("                ", grades1, " .          -          63.  
1");  
    addStudent(&studentList, student1);  
  
    Student* student2 = createStudent("                ", grades2, " .          9");  
    addStudent(&studentList, student2);  
  
    //  
    printStudents(studentList);  
  
    //  
    freeStudents(studentList);  
  
    return 0;  
}
```

1.1 Фрагмент программы, отрытый в редакторе VIM

2. Компилируем программу с флагом -g (Рис. 2.1), для последующего включения дебагинга GDB и запускаем ее. (Рис. 2.2)



```
albert@DESKTOP-700AJI4:/mnt/c/Users/User/Documents/GitHub/OS/2$ gcc -g -Wall main.c  
albert@DESKTOP-700AJI4:/mnt/c/Users/User/Documents/GitHub/OS/2$
```

Рис. 2.1 Команда для сборки программы

```

albert@DESKTOP-700AJI4:/mnt/c/Users/User/Documents/GitHub/OS/2$ ./a.out
ФИО: Лацук Андрей Юрьевич
Оценки: 3 4 2 4 5
Адрес: ул. Восход 9
-----
ФИО: Оганесян Альберт Самвелович
Оценки: 5 4 5 5 3
Адрес: ул. Бориса-богаткова 63/1

```

Рис. 2.2 Вывод программы

### 3. Используя GDB, проведем трассировку программы

```

(gdb) b main
Breakpoint 1 at 0x1455: file main.c, line 59.
(gdb) run
Starting program: /mnt/c/Users/User/Documents/GitHub/OS/2/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at main.c:59
59      int main() {
(gdb) █

```

Рис. 3.1 установка брейкпоинта и трассировка программы

```

(gdb) i locals
studentList = 0x0
grades1 = {5, 4, 5, 5, 3}
grades2 = {0, 0, 0, 0, 0}
student1 = 0x0
student2 = 0x0
(gdb) █

```

Рис. 3.2 просмотр локальных переменных программы

```

59      int main() {
(gdb) i frame
Stack level 0, frame at 0x7fffffffdf40:
 rip = 0x555555555455 in main (main.c:59); saved rip = 0x7ffff7db6d90
 source language c.
 Arglist at 0x7fffffffdf30, args:
 Locals at 0x7fffffffdf30, Previous frame's sp is 0x7fffffffdf40
 Saved registers:
  rbp at 0x7fffffffdf30, rip at 0x7fffffffdf38

```

Рис. 3.3 просмотр информации фрейма

```

Breakpoint 1, main () at main.c:59
59     int main() {
(gdb) next
61     Student* studentList = NULL;
(gdb) next
64     int grades1[] = {5, 4, 5, 5, 3};
(gdb) next
65     int grades2[] = {3, 4, 2, 4, 5};
(gdb) next
68     Student* student1 = createStudent("Оганесян Альберт Самвелович", gr
ades1, "ул. Бориса-богаткова 63/1");
(gdb) █

```

Рис. 3.4 выполнение программы по шагам

```

(gdb) ptype student2
type = struct Student {
    char name[100];
    int grades[5];
    char address[200];
    struct Student *next;
} *
(gdb) print student2
$1 = (Student *) 0x5555555593f0
(gdb) █

```

Рис. 3.5 просмотр типа и вывод переменной

4. Искусственно создадим в программе ошибку и попробуем отловить её.(поделим на ноль)

```
int a = 1/ 0;
```

Рис. 4.1 строка искусственной ошибкой

Теперь запустим отладку и посмотрим, что он нам выдаст

```

Program received signal SIGFPE, Arithmetic exception.
0x00005555555554f1 in main () at main.c:70
70     int a = 1/ 0;
(gdb) █

```

Рис. 4.2 вывод ошибки в консоль

Вывод: Мы настроили и познакомились с базовым функционалом gdb отладчика. Gdb имеет широкий инструментарий и множество настраиваемых параметров для гибкой работы с программами.