

Федеральное агентство связи
Сибирский государственный университет телекоммуникаций и информатики

Лабораторная работа №5

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

Новосибирск 2024

Цель: получение навыков разработки библиотек динамической компоновки на платформе Linux

Задание: разделите программу лабораторной 1 на модули.

Реализуйте функции работы со списками в библиотеке совместного доступа **.so**

Ход работы:

1. Разделим программу из 1 лабораторной на модуль библиотеки lib.c:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include "list.h"
```

```
// Функция для создания нового студента
```

```
Student* createStudent(const char* name, int grades[], const char* address) {
```

```
    Student* newStudent = (Student*)malloc(sizeof(Student));
```

```
    if (newStudent == NULL) {
```

```
        printf("Ошибка выделения памяти\n");
```

```
        exit(1);
```

```
    }
```

```
    strncpy(newStudent->name, name, sizeof(newStudent->name));
```

```
    memcpy(newStudent->grades, grades, sizeof(newStudent->grades));
```

```
    strncpy(newStudent->address, address, sizeof(newStudent->address));
```

```
    newStudent->next = NULL;
```

```

    return newStudent;
}

// Функция для добавления студента в начало списка
void addStudent(Student** head, Student* newStudent) {
    newStudent->next = *head;
    *head = newStudent;
}

// Функция для вывода информации о студентах
void printStudents(const Student* head) {
    const Student* current = head;
    while (current != NULL) {
        printf("ФИО: %s\n", current->name);
        printf("Оценки: ");
        for (int i = 0; i < 5; i++) {
            printf("%d ", current->grades[i]);
        }
        printf("\n");
        printf("Адрес: %s\n", current->address);
        printf("-----\n");
        current = current->next;
    }
}

```

// Функция для освобождения памяти списка

```
void freeStudents(Student* head) {
```

```
    Student* current = head;
```

```
    while (current != NULL) {
```

```
        Student* temp = current;
```

```
        current = current->next;
```

```
        free(temp);
```

```
    }
```

```
}
```

И исполняемый файл main.c:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include "list.h"
```

```
int main() {
```

```
    // Создание пустого списка
```

```
    Student* studentList = NULL;
```

```
    // Пример данных для студентов
```

```
    int grades1[] = {5, 4, 5, 5, 3};
```

```
    int grades2[] = {3, 4, 2, 4, 5};
```

```

// Создание и добавление студентов в список

Student* student1 = createStudent("Оганесян Альберт Самвелович", grades1,
"ул. Бориса-богаткова 63/1");

addStudent(&studentList, student1);


Student* student2 = createStudent("Лацук Андрей Юрьевич", grades2, "ул.
Восход 9");

addStudent(&studentList, student2);


// Вывод информации о студентах

printStudents(studentList);


// Освобождение памяти

freeStudents(studentList);


return 0;

}

```

2. Собираем объектный файл с позиционно-независимым кодом (-fPIC):

```
gcc -c -fPIC list.c -o list.o
```

3. Собираем динамическую библиотеку из объектника при помощи флага (-shared):

```
gcc -shared -o liblist.so list.o
```

4. Собираем исполняемый файл при помощи динамической библиотеки и запускаем его (рис 4.1)

```
• [stud@localhost 5]$ gcc -o main main.c ./liblist.so
• [stud@localhost 5]$ ./main
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 4.1 результат сборки и запуска программы

Вывод: мы научились основам разработки библиотек динамической компоновки