Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №5 по курсу**

**«Операционные системы»**

Студент: Боев Савелий Сергеевич

Группа: М8О-207Б-21

Вариант: 34

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**Содержание**

[Репозиторий 3](file:///C:\Users\noobl\Downloads\Osi4%20(1).docx#_Toc123904211)

[Постановка задачи 3](file:///C:\Users\noobl\Downloads\Osi4%20(1).docx#_Toc123904212)

[Цель работы 3](file:///C:\Users\noobl\Downloads\Osi4%20(1).docx#_Toc123904213)

[Задание 3](file:///C:\Users\noobl\Downloads\Osi4%20(1).docx#_Toc123904214)

[Общие сведения о программе 4](file:///C:\Users\noobl\Downloads\Osi4%20(1).docx#_Toc123904215)

[Общий метод и алгоритм решения 5](file:///C:\Users\noobl\Downloads\Osi4%20(1).docx#_Toc123904216)

[Исходный код 6](file:///C:\Users\noobl\Downloads\Osi4%20(1).docx#_Toc123904217)

[Демонстрация работы программы 6](file:///C:\Users\noobl\Downloads\Osi4%20(1).docx#_Toc123904218)

[Выводы 9](file:///C:\Users\noobl\Downloads\Osi4%20(1).docx#_Toc123904219)

# **Репозиторий**

<https://github.com/IamNoobLEL/Labs-OSi>

# **Постановка задачи**

## **Цель работы**

Изучение операционных систем

## **Задание**

Задача: реализовать 2 динамические библиотеки и 2 программы для работы с ними. Первая программа будет загружать библиотеку (одну) на этапе компиляции при помощи ключа -lmylib, а вторая программа будет подключать две динамические библиотеки при помощи dl-функций в самом коде.

# **Общие сведения о программе**

Для выполнения данной лабораторной работы я предварительно создал 4 файла: первые два – lib1.cpp и lib2.cpp являются исходным кодом для наших динамических библиотек. Файлы first.cpp и second.cpp являются двумя программами, которые нужно было реализовать по заданию. first.cpp является программой, к которой библиотека подгружается на этапе компиляции, а second.cpp является программой, к которой библиотека подключается непосредственно в самом коде.

1) **add\_library(d1 SHARED lib1.cpp)** - создает целевую библиотеку **d1**, которая собирается из исходного файла **lib1.cpp**. Ключевое слово **SHARED** указывает, что это динамическая библиотека (shared library).

2) **target\_link\_libraries(main1 d1 -Wl,-rpath,.)** - добавляет библиотеку **d1** в целевой файл **main1**. **-Wl,-rpath,.** задает путь к директории, где будут искаться зависимости во время исполнения.

3) **target\_link\_libraries(main2 d1)** - добавляет библиотеку **d1** в целевой файл **main2**. **D1** - это библиотека для загрузки динамических библиотек во время исполнения программы.

4) **set\_target\_properties(d1 PROPERTIES OUTPUT\_NAME "d1")** - задает имя выходного файла для библиотеки **d1**.

5) **set\_target\_properties(d1 PROPERTIES PREFIX "")** - удаляет префикс **lib** из имени выходного файла библиотеки **d1**.

6) **set\_target\_properties(d1 PROPERTIES SUFFIX ".so")** - задает суффикс **.so** для выходного файла библиотеки **d1**. Для Unix-подобных систем это означает, что это динамическая библиотека.

void\* dlopen(...) - вгружает нашу библиотеку;

void\* dlsym(...) - присваивает указателю на функцию ее адрес в библиотеке

int dlclose(...) - освобождает указатель на библиотеку

# **Общий метод и алгоритм решения**

В самом начале выполнения лабораторной работы я реализовал две библиотеки: lib1.cpp и lib2.cpp. Там реализовал простейший подсчет площади квадрата и прямоугольного треугольника, а также алгоритмы перевода числа из десятичной системы в двоичную и троичную. В first.cpp через конструкцию if обрабатывал команды, которые вводит пользователь и выдавал ожидаемый результат. В second.cpp подгружал библиотеки через

void\* dlopen(...), через void\* dlsym(...) делал функции из динамических библиотек видными для second.cpp.

# **Исходный код**

**first.cpp**

#include <iostream>

extern "C" float Square(float A, float B);

extern "C" char\* translation(long x);

int main(){

int command;

while((std::cout << "Enter command: ") && (std::cin >> command)){

if(command == 1){

std::cout << "Enter lengths A and B: ";

float A, B;

std::cin >> A >> B;

std::cout << "Square is " << Square(A, B) << std::endl;

}

else if(command == 2){

long x;

std::cout << "Enter decimal number: ";

std::cin >> x;

char\* memory = translation(x);

std::cout << "Binary number is " << memory << std::endl;

free(memory);

}

else

std::cout << "Commands can be 1 and 2 ";

}

}

**second.cpp**

extern "C" float E (int x);

extern "C" float Square (float A, float B);

float E (int x) {

if (x < 0) {

return -1;

}

float e = 1.0;

float term = 1.0;

for (int i = 1; i <= x; ++i) {

term /= i;

e += term;

}

return e;

}

float Square (float A, float B) {

if (A <= 0 || B <= 0) {

return -1;

}

float square = 0.5 \* A \* B;

return square;

}

**lib1.cpp**#include <iostream>

#include <algorithm>

#include <cstring>

extern "C" float Square(float A, float B);

extern "C" char\* translation(long x);

float Square(float A, float B){

return A \* B;

}

char\* translation(long x){

std::string bin;

if(x == 0) bin += "0";

while(x != 0){

if(x % 2 ==0)

{

x/=2;

bin += "0";

}

else

if(x % 2 != 0)

{

x /= 2;

bin += "1";

}

}

std::string number = bin;

std::reverse(number.begin(), number.end());

char\* answer = (char\*) malloc((number.size() + 1) \* sizeof(char));

strcpy(answer, number.c\_str());

return answer;

}

**lib2.cpp**#include <iostream>

#include <algorithm>

#include <cstring>

extern "C" float Square(float A, float B);

extern "C" char\* translation(long x);

float Square(float A, float B){

return 0.5 \* (A \* B);

}

char\* translation(long x) {

std::string bin;

if(x == 0) bin += "0";

while (x > 0) {

bin += std::to\_string(x % 3);

x /= 3;

}

std::string number = bin;

std::reverse(number.begin(), number.end());

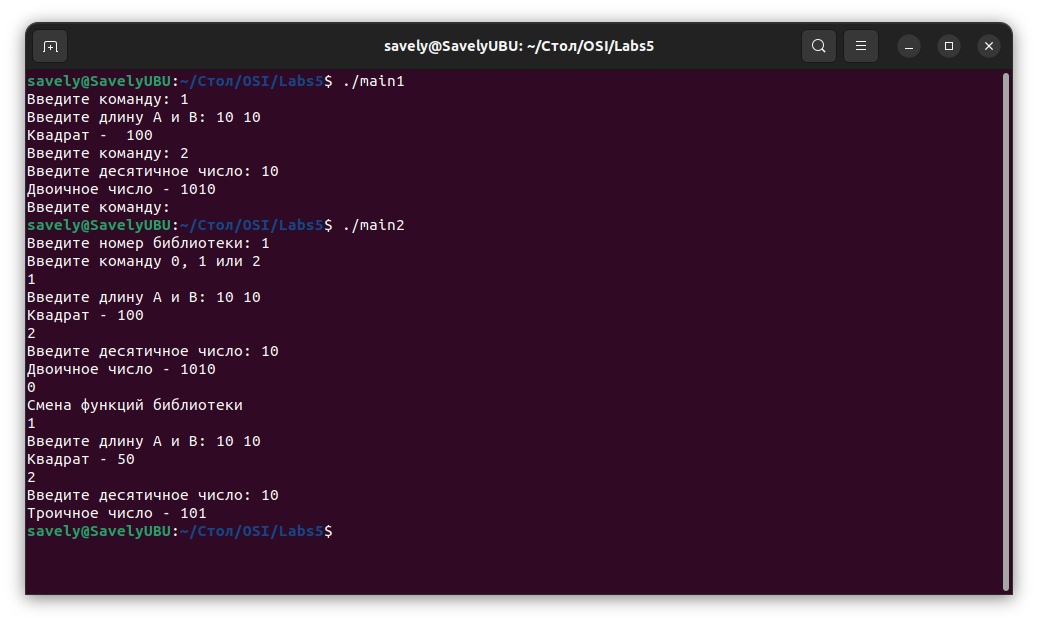
char \*answer = (char \*) malloc((number.size() + 1) \* sizeof(char));

strcpy(answer, number.c\_str());

return answer;

}

## **Демонстрация работы программы**



# **Выводы**

В данной работе я научился работать с динамическими библиотеками, изучил теорию их использования, осознал их отличия от статических библиотек и успешно реализовал динамические библиотеки, требуемые по заданию.