Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №4 по курсу**

**«Операционные системы»**

**Динамические библиотеки**

Студент: Климов Иван Павлович

Группа: М8О–212Б–22

Вариант: 23

Преподаватель: Соколов Андрей Алексеевич

Оценка: \_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_

Москва, 2023.

**Постановка задачи**

Требуется создать динамические библиотеки, которые реализуют определенный функционал.

Далее использовать данные библиотеки 2-мя способами:

Во время компиляции (на этапе «линковки»/linking)

Во время исполнения программы. Библиотеки загружаются в память с помощью

интерфейса ОС для работы с динамическими библиотеками

В конечном итоге, в лабораторной работе необходимо получить следующие части:

Динамические библиотеки, реализующие контракты, которые заданы вариантом;

Тестовая программа (программа №1), которая используют одну из библиотек, используя

знания полученные на этапе компиляции;

Тестовая программа (программа №2), которая загружает библиотеки, используя только их

местоположение и контракты.

Провести анализ двух типов использования библиотек.

Пользовательский ввод для обоих программ должен быть организован следующим образом:

Если пользователь вводит команду «0», то программа переключает одну реализацию

контрактов на другую (необходимо только для программы №2). Можно реализовать

лабораторную работу без данной функции, но максимальная оценка в этом случае будет

«хорошо»;

«1 arg1 arg2 … argN», где после «1» идут аргументы для первой функции, предусмотренной

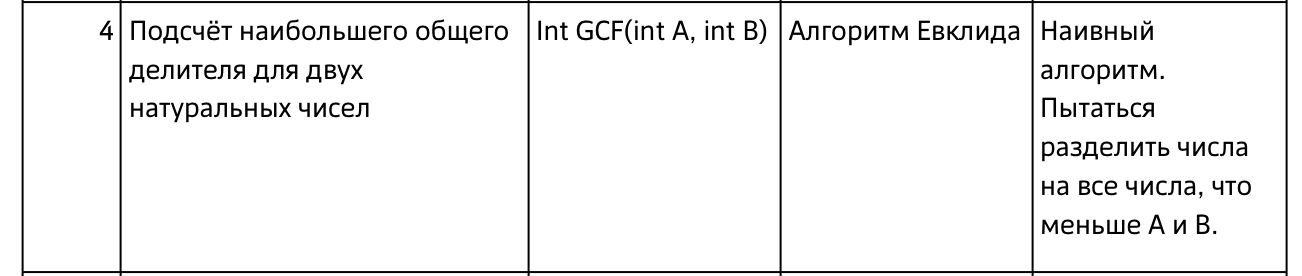
контрактами. После ввода команды происходит вызов первой функции, и на экране

появляется результат её выполнения;

«2 arg1 arg2 … argM», где после «2» идут аргументы для второй функции,

предусмотренной контрактами. После ввода команды происходит вызов второй функции,

и на экране появляется результат её выполнения.





**Общие сведения о программе**

Проект состоит из четырех частей: main\_1.c, main\_2.c, Evklid.c, calculateE.c . В первам мейне мы просто вводим необходимые параметры, чтобы проверить работы функций. Во втором мейне мы можем выбирать, каким контрактом пользоваться.

**Общий метод и алгоритм решения**.

Для первой части задания мы создаем динамические библиотеки на стадии компиляции: используем следующие команды:

ivanklimov@MacBook-Air-Ivan-2 os\_lab\_4 % gcc -o libEvklid.so -shared Evklid.c

ivanklimov@MacBook-Air-Ivan-2 os\_lab\_4 % gcc -o libCalculateE.so -shared calculateE.c

ivanklimov@MacBook-Air-Ivan-2 os\_lab\_4 % gcc main.c -L. -lEvklid -lCalculateE -o main

ivanklimov@MacBook-Air-Ivan-2 os\_lab\_4 % ./main

Затем вводим числа и проверяем результат.

Во второй части задания мы подключаем библиотеку уже в рантайме. Для этого нам необходимо подключить библиотеку dlfcn.h. Затем при помощи dlopen() открываем библиотеку по адресу, dlsym() нам помогает вытащить нужную функцию по названию этой функции, dlclose() закрывает библиотеку

**main\_1.c**

#include <stdio.h>

#include "Evklid.h"

#include "calculateE.h"

int main(){

int ans, a, b, n;

float ans2;

while(1){

printf("Введите 2 числа:\n");

scanf("%d%d", &a, &b);

if(a == -1){break;}

ans = gcf(a, b);

printf("Ответ:\n");

printf("%d\n", ans);

printf("~~~~~~~~~~~~~~~\n");

printf("Введите число:\n");

scanf("%d", &n);

ans2 = calculateE(n);

printf("Ответ: %lf\n", ans2);

printf("~~~~~~~~~~~~~~~\n");

}

return 0;

}

**main\_2.c**

#include <stdio.h>

#include <dlfcn.h>

int main(){

int a, b, n, i, k = 0;

void \*lib\_evklid;

void \*lib\_E;

int (\*func\_evklid)(int x, int y);

float (\*func\_E)(int z);

while(1){

printf("Выбор контракта(выход: -1): ");

scanf("%d", &i);//=1

if(i == -1){break;}

if(i == 0){ k = (k+1)%2;}

switch (k)

{

case(0): //evklid

printf("Введите 2 числа:\n");

scanf("%d%d", &a, &b);

lib\_evklid = dlopen("libEvklid.so", RTLD\_LAZY);//RTLD\_LAZY, подразумевающим разрешение неопределенных символов в виде кода, содержащегося в исполняемой динамической библиотеке

if (!lib\_evklid){

//если ошибка, то вывести ее на экран

fprintf(stderr, "dlopen() error: %s\n", dlerror());

return 1;

}

func\_evklid = dlsym(lib\_evklid, "gcf");

printf("ans\_1 = %d\n", (\*func\_evklid)(a, b));

dlclose(lib\_evklid);

break;

case(1): //E

printf("Введите число:\n");

scanf("%d", &n);

lib\_E = dlopen("libCalculateE.so", RTLD\_LAZY);

if (!lib\_E){

//если ошибка, то вывести ее на экран

fprintf(stderr, "dlopen() error: %s\n", dlerror());

return 1;

}

func\_E = dlsym(lib\_E, "calculateE");

printf("ans\_2 = %lf\n", (\*func\_E)(n));

dlclose(lib\_E);

break;

}

}

return 0;

}

**Evklid.c**

#include <stdio.h>

#include "Evklid.h"

// Функция для вычисления наибольшего общего делителя (НОД) двух чисел

int gcf(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

**calculateE.c**

#include <stdio.h>

#include "calculateE.h"

long long factorial(int n) {

if (n == 0 || n == 1) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

// Функция для вычисления значения числа e с использованием ряда Маклорена

float calculateE(int n) {

float e = 1.0;

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

e += 1.0 / factorial(i);

}

return e;

}

Вывод:

В ходе работы надо этой программой я познакомился с динамическими библиотеками, реализовал переключатель между заданиями. Создал свои библиотеки.