Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

**Лабораторная работа №4 по курсу**

**«Операционные системы»**

Группа: М80-206Б-22

Студентка: Коломытцева Е. А.

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: 28.12.23

Москва, 2023

**Постановка задачи**

**Вариант 8.**

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 (static\_main.c), которая используют одну из библиотек, используя знания полученные на этапе компиляции;
* Тестовая программа (dynamic\_main.c), которая загружает библиотеки, используя только их местоположение и контракты.

Функции для варианта:

1. Расчет производной функции:
2. f'(x) = (f(A + deltaX) – f(A))/deltaX
3. f'(x) = (f(A + deltaX) –f(A-deltaX))/(2\*deltaX)

2. Расчет значения числа е(основание натурального логарифма):

1. (1 + 1/x) ^ x
2. Сумма ряда по n от 0 до x, где элементы ряда равны: (1/(n!))

**Общий метод и алгоритм решения**

Использованные системные вызовы:

* void \*dlopen(const char \* \_\_path, int \_\_mode) - подгружает динамическую библиотеку;
* void \*dlsym(void \*\_\_handle, const char \*\_\_symbol) - находит адресс в подгруженной библиотеке (по ее \_\_handle), с которого начинается \_\_symbol;
* int dlclose(void \*\_\_handle) - уменьшает на единицу счетчик ссылок на указатель \_\_handle, и если нет других загруженных библиотек, использующих ее символы и счетчик ссылок принимает нулевое значение, то динамическая библиотека выгружается.

Программа состоит из двух интерфейсов (main1.c и main2.c), каждый из них реализован по-разному, в соответствии с заданием. Также каждая реализация контрактов представляет из себя отдельный файл: lib1.c и lib2.c. Для объявления необходимых функций также используется заголовочный файл lib.h. Так как все собирается с помощью CMake, то в проекте присутствует CMakeLists.txt.

Описание CMakeLists.txt:

cmake\_minimum\_required(VERSION 3.8 FATAL\_ERROR)

project(main LANGUAGES C)

set(BUILD\_WITH\_ASAN 1)

add\_library(

lib1 SHARED

./include/lib.h

./src/lib1.c

)

add\_library(

lib2 SHARED

./include/lib.h

./src/lib2.c

)

add\_executable(main1 ./src/main1.c)

target\_include\_directories(main1 PRIVATE ./include)

target\_link\_libraries(main1 PRIVATE lib1 m)

add\_executable(main2 ./src/main1.c)

target\_include\_directories(main2 PRIVATE ./include)

target\_link\_libraries(main2 PRIVATE lib2 m)

add\_executable(main ./src/main2.c)

target\_include\_directories(main PRIVATE ./include m)

if (${BUILD\_WITH\_ASAN})

message("-- Adding sanitizers")

target\_compile\_options(main PRIVATE -fsanitize=address)

target\_link\_options(main PRIVATE -fsanitize=address)

target\_compile\_options(main1 PRIVATE -fsanitize=address)

target\_link\_options(main1 PRIVATE -fsanitize=address)

target\_compile\_options(main2 PRIVATE -fsanitize=address)

target\_link\_options(main2 PRIVATE -fsanitize=address)

endif()

// Создание библиотек

// Линковка исполняемых файлов

Объявим необходимые функции внутри файла lib.h. Используем спецификатор хранения extern, который сообщает компилятору, что находящиеся за ним типы и имена переменных объявляются где-то в другом месте. Так как по заданию необходимо подключать библиотеки на этапе линковки, то подключать lib.h в реализации lib1.c и lib2.c не следует. В этих файлах просто напишем логику работы необходимых функций. Важно, чтобы они назывались также, как и те, что объявлены в lib.h. Используемые алгоритмы:

Косинус — сумма ряда Тейлора;

Факториал — факториал «Деревом»;

Возведение в степень — алгоритм «бинарного» возведения в степень.

Интерфейс 1:

Подключаем lib.h и пользуемся функциями так, как будто библиотека обычная. Различия наступают в сборке программы. Если бы мы собирали такой код в терминале, то прописали бы gcc -c -fpic lib1.c. Опция -fpic - требует от компилятора, при создании объектных файлов, порождать позиционно-независимый код. Формат позиционно-независимого кода позволяет подключать исполняемые модули к коду основной программы в момент её загрузки. Далее gcc -shared -o liblib1.so lib1.o -lm. Опция -shared - указывает gcc, что в результате должен быть собран не исполняемый файл, а разделяемый объект — динамическая библиотека.

Интерфейс 2:

Воспользуемся системными вызовами из библиотеки . Функция dlopen открывает динамическую библиотеку (объект .so) по названию. Функция dlsym - обработчик динамически загруженного объекта вызовом dlopen. Функция dlclose, соответственно, закрывает динамическую библиотеку. Собираем с помощью gcc -L. -Wall -o main.out main2.c -llib2 -llib1. Флаг -L. Означает, что поиск файлов библиотек будет начинаться с текущей директории.

Система сборки: ASAN — это Address Sanitizer, инструмент, с помощью которого можно ловить RE связанные с неправильным обращением к памяти. Наиболее логичный способ их интеграции в СMake — интегрировать их как типы сборки CMake, чтобы программы были созданы оптимально для санитайзеров. Для получения оптимальных результатов эти типы сборки игнорируют все другие флаги компилятора.

**Код программы**

**lib.h**

#ifndef \_\_LIB\_H\_\_

#define \_\_LIB\_H\_\_

extern float Derivative(float A, float deltaX);

extern float E(int x);

#endif

**lib1.c**

#include <stdio.h>

const float PI = 3.1415926;

float Cos(float x)

{

int y = 100;

int div = (int) (x / PI);

x = x - (div \* PI);

char sign = 1;

if (div % 2 != 0) {

sign = -1;

}

float result = 1.0;

float inter = 1.0;

float num = x \* x;

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

float comp = 2.0 \* i;

float den = comp \* (comp - 1.0);

inter \*= num / den;

if (i % 2 == 0) {

result += inter;

} else {

result -= inter;

}

}

return sign \* result;

}

float Derivative(float A, float deltaX)

{

printf("\nCalculation of derivative function f(x) = Cos(x)\n");

printf("in point %f with approximation %f\n", A, deltaX);

printf("by formula f'(x) = (f(A + deltaX) – f(A))/deltaX\n");

printf("cos(A) = %f\n", Cos(A));

float dfdx = (Cos(A + deltaX) - Cos(A)) / deltaX;

return dfdx;

}

float binPow(float x, int y)

{

float z = 1.0;

while (y > 0) {

if (y % 2 != 0) {

z \*= x;

}

x \*= x;

y /= 2;

}

return z;

}

float E(int x)

{

printf("\nCalculation value of number e (base of natural logarithm)\n");

printf("with approximation %d\n", x);

printf("by formula e(x) = (1 + 1/x) ^ x\n");

float mant = (float) 1 + ((float) 1 / (float) x);

float e = binPow(mant, x);

return e;

}

**lib2.c**

#include <stdio.h>

const float PI = 3.1415926;

float Cos(float x)

{

int y = 100;

int div = (int) (x / PI);

x = x - (div \* PI);

char sign = 1;

if (div % 2 != 0) {

sign = -1;

}

float result = 1.0;

float inter = 1.0;

float num = x \* x;

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

float comp = 2.0 \* i;

float den = comp \* (comp - 1.0);

inter \*= num / den;

if (i % 2 == 0) {

result += inter;

} else {

result -= inter;

}

}

return sign \* result;

}

float Derivative(float A, float deltaX)

{

printf("\nCalculation of derivative function f(x) = cos(x)\n");

printf("in point %f with approximation %f\n", A, deltaX);

printf("by formula f'(x) = (f(A + deltaX) – f(A-deltaX))/(2\*deltaX)\n");

printf("cos(A) = %f\n", Cos(A));

float dfdx = (Cos(A + deltaX) - Cos(A - deltaX)) / (2 \* deltaX);

return dfdx;

}

int prodTree(int l, int r)

{

if (l > r) {

return 1;

}

if (l == r) {

return l;

}

if (l - r == 1) {

return l \* r;

}

int m = (l + r) / 2;

return prodTree(l, m) \* prodTree(m + 1, r);

}

int fact(int n)

{

if (n < 0) {

return 0;

}

if (n == 0) {

return 1;

}

if (n == 1 || n == 2) {

return n;

}

return prodTree(2, n);

}

float machineEpsilon(void)

{

float e = 1.0f;

while (1.0f + e / 2.0f > 1.0f)

e /= 2.0f;

return e;

}

float E(int x)

{

printf("\nCalculation value of number e (base of natural logarithm)\n");

printf("with approximation %d\n", x);

printf("by sum of row by n from 0 to x f(n) = (1/(n!))\n");

float e = 0;

for (int n = 0; n <= x; n++) {

float tmp = ((float) 1 / fact(n));

float ftmp = tmp > 0 ? tmp : (float) (-1) \* tmp;

if (ftmp <= machineEpsilon()) {

printf("Approximation can not work because of mashine Epsilon of float is %.8f\n", machineEpsilon());

break;

}

e += tmp;

}

return e;

}

**main1.c**

#include <stdio.h>

#include "lib.h"

int main(int argc, char const \*argv[])

{

printf("\nWrite:\n [command] [arg1] ... [argN]\n");

printf("\nIf you want to take derivation of f(x) = cos(x), write 1 [point] [delta]\n");

printf("\nIf you want to calculate number e (base of natural logarithm), write 2 [approximation]\n\n");

int command = 0;

while (scanf("%d", &command) != EOF) {

switch (command) {

case 1:

float A, deltaX;

scanf("%f%f", &A, &deltaX);

printf("Answer: %f\n", Derivative(A, deltaX));

break;

case 2:

int x;

scanf("%d", &x);

printf("Answer: %f\n", E(x));

break;

default:

printf("wrong command\n");

break;

}

printf("\nWrite:\n [command] [arg1] ... [argN]\n");

printf("\nIf you want to take derivation of f(x) = cos(x), write 1 [point] [delta]\n");

printf("\nIf you want to calculate number e (base of natural logarithm), write 2 [approximation]\n\n");

}

return 0;

}

**main2.c**

#include <stdio.h>

#include <dlfcn.h>

#include "lib.h"

const char\* lib1 = "./liblib1.so";

const char\* lib2 = "./liblib2.so";

int main(int argc, char const \*argv[])

{

printf("\nWrite:\n [command] [arg1] ... [argN]\n");

printf("\nIf you want to change methods of calculation, write 0\n");

printf("\nIf you want to take derivation of f(x) = cos(x), write 1 [point] [delta]\n");

printf("\nIf you want to calculate number e (base of natural logarithm), write 2 [approximation]\n");

int command = 0;

int link = 0;

void \*currentLib = dlopen(lib1, RTLD\_LAZY);

printf("\nCurrent lib is %d\n\n", link);

float (\*Derivative)(float A, float deltaX);

float (\*E)(int x);

Derivative = dlsym(currentLib, "Derivative");

E = dlsym(currentLib, "E");

while (scanf("%d", &command) != EOF) {

switch (command) {

case 0:

dlclose(currentLib);

if (link == 0) {

currentLib = dlopen(lib2, RTLD\_LAZY);

} else {

currentLib = dlopen(lib1, RTLD\_LAZY);

}

link = !link;

Derivative = dlsym(currentLib, "Derivative");

E = dlsym(currentLib, "E");

break;

case 1:

float A, deltaX;

scanf("%f%f", &A, &deltaX);

printf("Answer: %f\n", Derivative(A, deltaX));

break;

case 2:

int x;

scanf("%d", &x);

printf("Answer: %f\n", E(x));

break;

default:

printf("wrong command\n");

break;

}

printf("\nWrite:\n [command] [arg1] ... [argN]\n");

printf("\nIf you want to change methods of calculation, write 0\n");

printf("\nIf you want to take derivation of f(x) = cos(x), write 1 [point] [delta]\n");

printf("\nIf you want to calculate number e (base of natural logarithm), write 2 [approximation]\n");

printf("\nCurrent lib is %d\n\n", link);

}

return 0;

}

**Протокол работы программы**

**Тестирование:**

./main

Write:

[command] [arg1] ... [argN]

If you want to change methods of calculation, write 0

If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

If you want to calculate number e (base of natural logarithm), write 2 [approximation]

Current lib is 0

0

Write:

[command] [arg1] ... [argN]

If you want to change methods of calculation, write 0

If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

If you want to calculate number e (base of natural logarithm), write 2 [approximation]

Current lib is 1

1 2 0.0001

Calculation of derivative function f(x) = cos(x)

in point 2.000000 with approximation 0.000100

by formula f'(x) = (f(A + deltaX) – f(A-deltaX))/(2\*deltaX)

cos(A) = -0.416147

Answer: -0.908971

Write:

[command] [arg1] ... [argN]

If you want to change methods of calculation, write 0

If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

If you want to calculate number e (base of natural logarithm), write 2 [approximation]

Current lib is 1

2 1000

Calculation value of number e (base of natural logarithm)

with approximation 1000

by sum of row by n from 0 to x f(n) = (1/(n!))

Approximation can not work because of mashine Epsilon of float is 0.00000012

Answer: 2.718282

Write:

[command] [arg1] ... [argN]

If you want to change methods of calculation, write 0

If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

If you want to calculate number e (base of natural logarithm), write 2 [approximation]

Current lib is 1

0

Write:

[command] [arg1] ... [argN]

If you want to change methods of calculation, write 0

If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

If you want to calculate number e (base of natural logarithm), write 2 [approximation]

Current lib is 0

2 1000

Calculation value of number e (base of natural logarithm)

with approximation 1000

by formula e(x) = (1 + 1/x) ^ x

Answer: 2.717042

Write:

[command] [arg1] ... [argN]

If you want to change methods of calculation, write 0

If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

If you want to calculate number e (base of natural logarithm), write 2 [approximation]

Current lib is 0

1 2 0.0001

Calculation of derivative function f(x) = Cos(x)

in point 2.000000 with approximation 0.000100

by formula f'(x) = (f(A + deltaX) – f(A))/deltaX

cos(A) = -0.416147

Answer: -0.908673

Write:

[command] [arg1] ... [argN]

If you want to change methods of calculation, write 0

If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

If you want to calculate number e (base of natural logarithm), write 2 [approximation]

Current lib is 0

**Strace:**

execve("./main", ["./main"], 0x7ffff87357b8 /\* 47 vars \*/) = 0

brk(NULL) = 0x55899a1f6000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffc96f72360) = -1 EINVAL (Недопустимый аргумент)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8ea8000

access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=119923, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 119923, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fdba8e8a000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libasan.so.8", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=10108112, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 6961512, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdba8600000

mmap(0x7fdba8625000, 1110016, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25000) = 0x7fdba8625000

mmap(0x7fdba8734000, 217088, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x134000) = 0x7fdba8734000

mmap(0x7fdba8769000, 28672, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x168000) = 0x7fdba8769000

mmap(0x7fdba8770000, 5454184, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8770000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P<\2\0\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2072888, ...}, AT\_EMPTY\_PATH) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

mmap(NULL, 2117488, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdba8200000

mmap(0x7fdba8222000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7fdba8222000

mmap(0x7fdba839a000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19a000) = 0x7fdba839a000

mmap(0x7fdba83f2000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1f1000) = 0x7fdba83f2000

mmap(0x7fdba83f8000, 53104, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fdba83f8000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=948816, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 950520, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdba8da1000

mmap(0x7fdba8daf000, 516096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7fdba8daf000

mmap(0x7fdba8e2d000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8c000) = 0x7fdba8e2d000

mmap(0x7fdba8e88000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe6000) = 0x7fdba8e88000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=141872, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 144232, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdba8d7d000

mmap(0x7fdba8d80000, 110592, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7fdba8d80000

mmap(0x7fdba8d9b000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e000) = 0x7fdba8d9b000

mmap(0x7fdba8d9f000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21000) = 0x7fdba8d9f000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8d7b000

arch\_prctl(ARCH\_SET\_FS, 0x7fdba8d7be80) = 0

set\_tid\_address(0x7fdba8d7c150) = 38562

set\_robust\_list(0x7fdba8d7c160, 24) = 0

rseq(0x7fdba8d7c7a0, 0x20, 0, 0x53053053) = 0

mprotect(0x7fdba83f2000, 16384, PROT\_READ) = 0

mprotect(0x7fdba8d9f000, 4096, PROT\_READ) = 0

mprotect(0x7fdba8e88000, 4096, PROT\_READ) = 0

mprotect(0x7fdba8769000, 16384, PROT\_READ) = 0

mprotect(0x558998b4d000, 4096, PROT\_READ) = 0

mprotect(0x7fdba8edd000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0x7fdba8e8a000, 119923) = 0

readlinkat(AT\_FDCWD, "/proc/self/exe", "/home/katya/MAI\_2/OS/github/OS\_M"..., 4096) = 50

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8ea7000

openat(AT\_FDCWD, "/proc/self/cmdline", O\_RDONLY) = 3

read(3, "./main\0", 4096) = 7

read(3, "", 4089) = 0

close(3) = 0

munmap(0x7fdba8ea7000, 4096) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8ea7000

openat(AT\_FDCWD, "/proc/self/environ", O\_RDONLY) = 3

read(3, "SHELL=/bin/bash\0SESSION\_MANAGER="..., 4096) = 3323

read(3, "", 773) = 0

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8ea5000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8ea3000

mmap(NULL, 3727360, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba7e72000

mmap(NULL, 2097152, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba7c72000

munmap(0x7fdba7c72000, 581632) = 0

munmap(0x7fdba7e00000, 466944) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8ea2000

mmap(NULL, 2097152, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba7b00000

munmap(0x7fdba7c00000, 1048576) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8ea1000

mmap(NULL, 2097152, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba7900000

munmap(0x7fdba7a00000, 1048576) = 0

mmap(NULL, 2097152, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba7700000

munmap(0x7fdba7800000, 1048576) = 0

mmap(NULL, 2097152, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba7500000

munmap(0x7fdba7600000, 1048576) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8ea0000

prlimit64(0, RLIMIT\_CORE, NULL, {rlim\_cur=0, rlim\_max=RLIM64\_INFINITY}) = 0

prlimit64(0, RLIMIT\_CORE, {rlim\_cur=0, rlim\_max=RLIM64\_INFINITY}, NULL) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9f000

openat(AT\_FDCWD, "/proc/self/maps", O\_RDONLY) = 3

read(3, "558998b4a000-558998b4b000 r--p 0"..., 4096) = 4026

read(3, "7ffc96f9d000-7ffc96fa1000 r--p 0"..., 70) = 70

close(3) = 0

munmap(0x7fdba8e9f000, 4096) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9e000

openat(AT\_FDCWD, "/proc/self/maps", O\_RDONLY) = 3

read(3, "558998b4a000-558998b4b000 r--p 0"..., 8192) = 4026

read(3, "7ffc96f9d000-7ffc96fa1000 r--p 0"..., 4166) = 244

read(3, "", 3922) = 0

close(3) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9d000

openat(AT\_FDCWD, "/proc/self/maps", O\_RDONLY) = 3

read(3, "558998b4a000-558998b4b000 r--p 0"..., 4096) = 4026

read(3, "7ffc96f9d000-7ffc96fa1000 r--p 0"..., 70) = 70

close(3) = 0

munmap(0x7fdba8e9d000, 4096) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9c000

openat(AT\_FDCWD, "/proc/self/maps", O\_RDONLY) = 3

read(3, "558998b4a000-558998b4b000 r--p 0"..., 8192) = 4026

read(3, "7ffc96f9d000-7ffc96fa1000 r--p 0"..., 4166) = 244

read(3, "", 3922) = 0

close(3) = 0

munmap(0x7fdba8e9c000, 8192) = 0

mmap(0x7fff7000, 268435456, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS|MAP\_NORESERVE, -1, 0) = 0x7fff7000

madvise(0x7fff7000, 268435456, MADV\_NOHUGEPAGE) = 0

madvise(0x7fff7000, 268435456, MADV\_DONTDUMP) = 0

mmap(0x2008fff7000, 15392894357504, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS|MAP\_NORESERVE, -1, 0) = 0x2008fff7000

madvise(0x2008fff7000, 15392894357504, MADV\_NOHUGEPAGE) = 0

madvise(0x2008fff7000, 15392894357504, MADV\_DONTDUMP) = 0

mmap(0x8fff7000, 2199023255552, PROT\_NONE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS|MAP\_NORESERVE, -1, 0) = 0x8fff7000

sigaltstack(NULL, {ss\_sp=NULL, ss\_flags=SS\_DISABLE, ss\_size=0}) = 0

mmap(NULL, 32768, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e96000

sigaltstack({ss\_sp=0x7fdba8e96000, ss\_flags=0, ss\_size=32768}, NULL) = 0

rt\_sigaction(SIGSEGV, {sa\_handler=0x7fdba86e4580, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_ONSTACK|SA\_NODEFER|SA\_SIGINFO, sa\_restorer=0x7fdba823c460}, NULL, 8) = 0

rt\_sigaction(SIGBUS, {sa\_handler=0x7fdba86e4580, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_ONSTACK|SA\_NODEFER|SA\_SIGINFO, sa\_restorer=0x7fdba823c460}, NULL, 8) = 0

rt\_sigaction(SIGFPE, {sa\_handler=0x7fdba86e4580, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_ONSTACK|SA\_NODEFER|SA\_SIGINFO, sa\_restorer=0x7fdba823c460}, NULL, 8) = 0

mmap(0x600000000000, 4398046519296, PROT\_NONE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS|MAP\_NORESERVE, -1, 0) = 0x600000000000

mmap(0x640000000000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x640000000000

mmap(NULL, 8388608, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_NORESERVE, -1, 0) = 0x7fdba6d00000

mmap(NULL, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8d6d000

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e95000

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e94000

getpid() = 38562

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e93000

openat(AT\_FDCWD, "/proc/self/maps", O\_RDONLY) = 3

read(3, "7fff7000-8fff7000 rw-p 00000000 "..., 4096) = 3985

read(3, "7fdba8edd000-7fdba8edf000 r--p 0"..., 111) = 111

close(3) = 0

munmap(0x7fdba8e93000, 4096) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e92000

openat(AT\_FDCWD, "/proc/self/maps", O\_RDONLY) = 3

read(3, "7fff7000-8fff7000 rw-p 00000000 "..., 8192) = 3985

read(3, "7fdba8edd000-7fdba8edf000 r--p 0"..., 4207) = 565

read(3, "", 3642) = 0

close(3) = 0

munmap(0x7fdba8e9e000, 8192) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9f000

openat(AT\_FDCWD, "/proc/self/maps", O\_RDONLY) = 3

read(3, "7fff7000-8fff7000 rw-p 00000000 "..., 4096) = 4034

read(3, "7fdba8edd000-7fdba8edf000 r--p 0"..., 62) = 62

close(3) = 0

munmap(0x7fdba8e9f000, 4096) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9e000

openat(AT\_FDCWD, "/proc/self/maps", O\_RDONLY) = 3

read(3, "7fff7000-8fff7000 rw-p 00000000 "..., 8192) = 3985

read(3, "7fdba8edd000-7fdba8edf000 r--p 0"..., 4207) = 565

read(3, "", 3642) = 0

close(3) = 0

munmap(0x7fdba8e9e000, 8192) = 0

mmap(0x100012ce7000, 1044480, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS|MAP\_NORESERVE, -1, 0) = 0x100012ce7000

madvise(0x100012ce7000, 1044480, MADV\_NOHUGEPAGE) = 0

madvise(0x100012ce7000, 1044480, MADV\_DONTDUMP) = 0

mmap(NULL, 11571200, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba61f7000

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9f000

sigaltstack(NULL, {ss\_sp=0x7fdba8e96000, ss\_flags=0, ss\_size=32768}) = 0

mmap(NULL, 1703936, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8460000

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9e000

mmap(NULL, 2097152, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba5ff7000

munmap(0x7fdba5ff7000, 36864) = 0

munmap(0x7fdba6100000, 1011712) = 0

munmap(0x7fdba8e9e000, 4096) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9e000

mmap(NULL, 2097152, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba5e00000

munmap(0x7fdba5f00000, 1048576) = 0

mmap(NULL, 2097152, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba5c00000

munmap(0x7fdba5d00000, 1048576) = 0

munmap(0x7fdba8e9e000, 4096) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9e000

munmap(0x7fdba8e9e000, 4096) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9e000

munmap(0x7fdba8e9e000, 4096) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9e000

munmap(0x7fdba8e9e000, 4096) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9e000

munmap(0x7fdba8e9e000, 4096) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9e000

munmap(0x7fdba8e9e000, 4096) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e9e000

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e91000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e8f000

clock\_gettime(CLOCK\_MONOTONIC, {tv\_sec=13712, tv\_nsec=486994678}) = 0

mmap(0x607000000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x607000000000

mmap(0x607e00000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x607e00000000

mmap(NULL, 1048576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba7c00000

mmap(NULL, 8388608, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_NORESERVE, -1, 0) = 0x7fdba5400000

clock\_gettime(CLOCK\_MONOTONIC, {tv\_sec=13712, tv\_nsec=487407556}) = 0

mmap(0x603000000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x603000000000

mmap(0x603e00000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x603e00000000

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e8e000

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdba8e8d000

newfstatat(1, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x1), ...}, AT\_EMPTY\_PATH) = 0

mmap(0x619000000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x619000000000

mmap(0x619e00000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x619e00000000

write(1, "\n", 1

) = 1

write(1, "Write:\n", 7Write:

) = 7

write(1, " [command] [arg1] ... [argN]\n", 29 [command] [arg1] ... [argN]

) = 29

write(1, "\n", 1

) = 1

write(1, "If you want to change methods of"..., 54If you want to change methods of calculation, write 0

) = 54

write(1, "\n", 1

) = 1

write(1, "If you want to take derivation o"..., 73If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

) = 73

write(1, "\n", 1

) = 1

write(1, "If you want to calculate number "..., 87If you want to calculate number e (base of natural logarithm), write 2 [approximation]

) = 87

mmap(0x624000000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x624000000000

mmap(0x624e00000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x624e00000000

mmap(0x602000000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x602000000000

mmap(0x602e00000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x602e00000000

openat(AT\_FDCWD, "./liblib1.so", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0775, st\_size=15728, ...}, AT\_EMPTY\_PATH) = 0

mmap(0x61a000000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x61a000000000

mmap(0x61ae00000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x61ae00000000

mmap(0x60d000000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x60d000000000

mmap(0x60de00000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x60de00000000

getcwd("/home/katya/MAI\_2/OS/github/OS\_MAI/lab4/build", 128) = 46

mmap(NULL, 16432, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdba8d68000

mmap(0x7fdba8d69000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7fdba8d69000

mmap(0x7fdba8d6a000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7fdba8d6a000

mmap(0x7fdba8d6b000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7fdba8d6b000

close(3) = 0

mmap(0x606000000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x606000000000

mmap(0x606e00000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x606e00000000

mprotect(0x7fdba8d6b000, 4096, PROT\_READ) = 0

mmap(0x61d000000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x61d000000000

mmap(0x61de00000000, 65536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x61de00000000

write(1, "\nCurrent lib is 0\n\n", 19

Current lib is 0

) = 19

newfstatat(0, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x1), ...}, AT\_EMPTY\_PATH) = 0

read(0, 0

"0\n", 1024) = 2

munmap(0x7fdba8d68000, 16432) = 0

openat(AT\_FDCWD, "./liblib2.so", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0775, st\_size=15856, ...}, AT\_EMPTY\_PATH) = 0

getcwd("/home/katya/MAI\_2/OS/github/OS\_MAI/lab4/build", 128) = 46

mmap(NULL, 16448, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdba8d68000

mmap(0x7fdba8d69000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7fdba8d69000

mmap(0x7fdba8d6a000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7fdba8d6a000

mmap(0x7fdba8d6b000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7fdba8d6b000

close(3) = 0

mprotect(0x7fdba8d6b000, 4096, PROT\_READ) = 0

write(1, "\nWrite:\n", 8

Write:

) = 8

write(1, " [command] [arg1] ... [argN]\n", 29 [command] [arg1] ... [argN]

) = 29

write(1, "\n", 1

) = 1

write(1, "If you want to change methods of"..., 54If you want to change methods of calculation, write 0

) = 54

write(1, "\n", 1

) = 1

write(1, "If you want to take derivation o"..., 73If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

) = 73

write(1, "\n", 1

) = 1

write(1, "If you want to calculate number "..., 87If you want to calculate number e (base of natural logarithm), write 2 [approximation]

) = 87

write(1, "\nCurrent lib is 1\n\n", 19

Current lib is 1

) = 19

read(0, 1 2000

"1 2000\n", 1024) = 7

read(0, 1 3 4

"1 3 4\n", 1024) = 6

write(1, "\n", 1

) = 1

write(1, "Calculation of derivative functi"..., 49Calculation of derivative function f(x) = cos(x)

) = 49

write(1, "in point 2000.000000 with approx"..., 49in point 2000.000000 with approximation 1.000000

) = 49

write(1, "by formula f'(x) = (f(A + deltaX"..., 62by formula f'(x) = (f(A + deltaX) – f(A-deltaX))/(2\*deltaX)

) = 62

write(1, "cos(A) = -0.367526\n", 19cos(A) = -0.367526

) = 19

write(1, "Answer: -0.782579\n", 18Answer: -0.782579

) = 18

write(1, "\nWrite:\n", 8

Write:

) = 8

write(1, " [command] [arg1] ... [argN]\n", 29 [command] [arg1] ... [argN]

) = 29

write(1, "\n", 1

) = 1

write(1, "If you want to change methods of"..., 54If you want to change methods of calculation, write 0

) = 54

write(1, "\n", 1

) = 1

write(1, "If you want to take derivation o"..., 73If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

) = 73

write(1, "\n", 1

) = 1

write(1, "If you want to calculate number "..., 87If you want to calculate number e (base of natural logarithm), write 2 [approximation]

) = 87

write(1, "\nCurrent lib is 1\n\n", 19

Current lib is 1

) = 19

write(1, "wrong command\n", 14wrong command

) = 14

write(1, "\nWrite:\n", 8

Write:

) = 8

write(1, " [command] [arg1] ... [argN]\n", 29 [command] [arg1] ... [argN]

) = 29

write(1, "\n", 1

) = 1

write(1, "If you want to change methods of"..., 54If you want to change methods of calculation, write 0

) = 54

write(1, "\n", 1

) = 1

write(1, "If you want to take derivation o"..., 73If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

) = 73

write(1, "\n", 1

) = 1

write(1, "If you want to calculate number "..., 87If you want to calculate number e (base of natural logarithm), write 2 [approximation]

) = 87

write(1, "\nCurrent lib is 1\n\n", 19

Current lib is 1

) = 19

write(1, "wrong command\n", 14wrong command

) = 14

write(1, "\nWrite:\n", 8

Write:

) = 8

write(1, " [command] [arg1] ... [argN]\n", 29 [command] [arg1] ... [argN]

) = 29

write(1, "\n", 1

) = 1

write(1, "If you want to change methods of"..., 54If you want to change methods of calculation, write 0

) = 54

write(1, "\n", 1

) = 1

write(1, "If you want to take derivation o"..., 73If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

) = 73

write(1, "\n", 1

) = 1

write(1, "If you want to calculate number "..., 87If you want to calculate number e (base of natural logarithm), write 2 [approximation]

) = 87

write(1, "\nCurrent lib is 1\n\n", 19

Current lib is 1

) = 19

read(0, 3

"3\n", 1024) = 2

write(1, "wrong command\n", 14wrong command

) = 14

write(1, "\nWrite:\n", 8

Write:

) = 8

write(1, " [command] [arg1] ... [argN]\n", 29 [command] [arg1] ... [argN]

) = 29

write(1, "\n", 1

) = 1

write(1, "If you want to change methods of"..., 54If you want to change methods of calculation, write 0

) = 54

write(1, "\n", 1

) = 1

write(1, "If you want to take derivation o"..., 73If you want to take derivation of f(x) = cos(x), write 1 [point] [delta]

) = 73

write(1, "\n", 1

) = 1

write(1, "If you want to calculate number "..., 87If you want to calculate number e (base of natural logarithm), write 2 [approximation]

) = 87

write(1, "\nCurrent lib is 1\n\n", 19

Current lib is 1

) = 19

**Вывод**

В ходе лабораторной работы я познакомилась с созданием динамических библиотек в ОС Linux, а также с возможностью загружать эти библиотеки в ходе выполнения программы. Их загрузка во время выполнения программы упрощает компиляцию программы, а также уменьшает размер исполняемых файлов.