



Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №4 по курсу
«Операционные системы»

Группа: М8О-215Б-23

Студент: Венгер Ирина Витальевна

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: _____

Москва, 2024.

Содержание

1. Постановка задачи.
2. Общие сведения о программе.
3. Общий метод и алгоритм решения.
4. Код программы.
5. Демонстрация работы программы.
6. Вывод.

Постановка задачи

Требуется создать динамические библиотеки, которые реализуют заданный вариант функционала. Далее использовать данные библиотеки 2-мя способами:

1. Во время компиляции (на этапе «линковки»/linking)
2. 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:

1. Динамические библиотеки, реализующие контракты, которые заданы вариантом;
2. Тестовая программа (*программа №1*), которая использует одну из библиотек, используя информацию полученную на этапе компиляции;
3. Тестовая программа (*программа №2*), которая загружает библиотеки, используя только их относительные пути и контракты.

Пользовательский ввод для обеих программ должен быть организован следующим образом:

1. Если пользователь вводит команду «0», то программа переключает одну реализацию контрактов на другую (необходимо только для *программы №2*).
2. «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;
3. «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.

Задание варианта:

- 1) Описание: Расчёт производной функции $\cos(x)$ в точке A с приращением δx ;
Сигнатура: `Float Derivative(float A, float deltaX)`;
Реализация 1: $f'(x) = (f(A + \delta x) - f(A))/\delta x$;
Реализация 2: $f'(x) = (f(A + \delta x) - f(A - \delta x))/(2 * \delta x)$.
- 2) Описание: Расчёт значения числа e (основание натурального логарифма);
Сигнатура: `Float E(int x)`;
Реализация 1: $(1 + 1/x)^x$;
Реализация 2: Сумма ряда по n от 0 до x , где элементы ряда равны: $(1/(n!))$.

Общие сведения о программе

Программа состоит из файлов: func.h, func1.c, func2.c, main1.c, main2.c и Makefile. Файл func.h содержит объявления функций для вычисления производной функции $\cos(x)$ и числа e . Файл func1.c реализует первую версию этих функций, используя методы для вычисления производной и числа e . В файле func2.c представлены альтернативные реализации этих функций с использованием других методов.

func1.c:

1. Реализация производной функции $\cos(x)$ с использованием формулы: $f'(x) = (f(A + \text{deltaX}) - f(A)) / \text{deltaX}$.
2. Реализация вычисления числа e с использованием формулы: $(1 + 1/x)^x$.

func2.c:

1. Реализация производной функции $\cos(x)$ с использованием формулы: $f'(x) = (f(A + \text{deltaX}) - f(A - \text{deltaX})) / (2 * \text{deltaX})$.
2. Реализация вычисления числа e с использованием суммы ряда: $e = \sum(1/n!, \text{от } n=0 \text{ до } x)$.

main1.c: Программа для использования библиотеки во время компиляции. В этой программе функции из библиотеки libfunc1.so загружаются на этапе компиляции и используются для вычислений производной и числа e .

main2.c: Программа для динамической загрузки библиотеки во время исполнения. В этой программе используется интерфейс ОС для загрузки библиотек libfunc1.dylib и libfunc2.dylib, переключение между ними осуществляется с помощью команды 0, а затем вызываются функции для вычислений.

Makefile компилирует две библиотеки libfunc1.dylib и libfunc2.dylib, а также две тестовые программы: main1 и main2. Для работы с математическими функциями в программе используется библиотека math.h, а для динамической загрузки библиотек — dlfcn.h.

1. **#include <stdio.h>** — для ввода-вывода. Из этого заголовка используются функции printf, fgets, sscanf, fprintf.
2. **#include <dlfcn.h>** — для динамической загрузки библиотек. Из этого заголовка используются функции dlopen, dlsym, dlclose, dlerror.
3. **#include <stdlib.h>** — для работы с памятью и управления процессами. Из этого заголовка используется функция exit.

4. **#include <math.h>** — для математических вычислений. Из этого заголовка используются функции `cos` (для вычисления косинуса), `pow` (для вычисления степени).

Общий метод и алгоритм решения

Метод решения задачи:

1. Создание динамических библиотек:

Для решения задачи создаются две динамические библиотеки (`libfunc1.dylib` и `libfunc2.dylib`), каждая из которых реализует два математических вычисления: производную функции $\cos(x)$ и вычисление числа e . Каждая библиотека предоставляет альтернативные реализации этих вычислений.

2. Реализация математических функций:

В библиотеке `libfunc1.dylib` реализуется первая версия функций:

- **Вычисление производной функции $\cos(x)$** с использованием формулы: $f'(x) = (\cos(A + \Delta x) - \cos(A)) / \Delta x$
- **Вычисление числа e** с использованием формулы: $e = (1 + 1/x)^x$

В библиотеке `libfunc2.dylib` реализуется вторая версия функций:

- **Вычисление производной функции $\cos(x)$** с использованием формулы: $f'(x) = (\cos(A + \Delta x) - \cos(A - \Delta x)) / (2 * \Delta x)$
- **Вычисление числа e** с использованием суммы ряда: $e = \sum (1 / n!)$ для n от 0 до x

3. Тестовые программы:

В первой тестовой программе (`main1.c`) используется статическая привязка к библиотеке `libfunc1.dylib`, и функции для вычисления производной и числа e вызываются напрямую.

Во второй тестовой программе (`main2.c`) динамическая загрузка библиотек с помощью интерфейса ОС (функции `dlopen`, `dlsym`, `dlclose`) позволяет загружать и использовать библиотеки во время выполнения программы. В этой программе также реализована возможность переключения между библиотеками с помощью команды "0", чтобы динамически выбирать одну из двух реализаций.

4. Взаимодействие с пользователем:

Пользователь взаимодействует с программой через командный ввод. Для выполнения функций необходимо ввести команду:

- "1 arg1 arg2" — для вычисления производной с переданными аргументами (A , Δx).
- "2 arg1" — для вычисления числа e с переданным значением x .
- "0" — для переключения между библиотеками `libfunc1.dylib` и `libfunc2.dylib` во второй программе.

Алгоритм решения:

1. Компиляция библиотеки:

Для каждой библиотеки (файлы func1.c и func2.c) компилируются динамические библиотеки с использованием флага -fPIC для создания объектных файлов с независимым кодом и флага -shared для создания динамических библиотек. С помощью Makefile библиотеки компилируются и связываются с соответствующими программами.

2. Основная программа (main1.c):

В этой программе используется статическое связывание с библиотекой libfunc1.dylib. При запуске программы пользователю предлагается вводить команды:

- При вводе команды "1" программа запрашивает значения для вычисления производной функции $\cos(x)$.
- При вводе команды "2" программа запрашивает значение для вычисления числа e .
- Все вычисления выполняются с использованием функций, доступных из библиотеки libfunc1.dylib.

3. Основная программа с динамической загрузкой (main2.c):

Программа динамически загружает библиотеку (libfunc1.dylib) с помощью функции dlopen. После загрузки библиотеки функции для вычисления производной и числа e извлекаются через dlsym. Программа выполняет вычисления по выбранной пользователем команде.

Вторая программа позволяет переключаться между библиотеками. При вводе команды "0" текущая библиотека закрывается через dlclose, и затем загружается другая библиотека (например, libfunc2.dylib).

4. Обработка команд:

В программе с динамической загрузкой пользователь может ввести команду "0" для переключения между библиотеками и команды "1" и "2" для выполнения соответствующих вычислений (производной и числа e).

Результаты вычислений выводятся на экран после выполнения команд.

5. Завершение работы программы:

Для завершения работы программы пользователю предлагается ввести команду "exit". В это время открытые библиотеки закрываются через dlclose, и программа завершает выполнение.

Код программы

Код программы приведён в приложении 1.

Использование утилиты strace

Листинг утилиты strace приведен в приложении 2. В нем можно увидеть системные вызовы:

main1.c

1. `execve("./main1", ["./main1"], ...)`

Этот системный вызов запускает программу main1 с передачей в нее аргументов. Вернувшийся результат = 0 означает, что процесс был успешно запущен.

2. `mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)`

Выделение памяти для процесса (8192 байта). Это используется для размещения данных в процессе.

3. `openat(AT_FDCWD, "./lib/libfunc1.so", O_RDONLY|O_CLOEXEC)`

Попытка открыть динамическую библиотеку libfunc1.so для загрузки, но она не найдена, о чем сообщает ошибка ENOENT. Строка повторяется для других путей, пока не будет найдено успешное открытие библиотеки.

4. `mmap(0x7ff94b29e000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000)`

Загрузка библиотеки в память, с установкой прав доступа (чтение и выполнение) для адреса в памяти.

5. `close(3)`

Закрытие файлового дескриптора после завершения работы с библиотекой.

6. `openat(AT_FDCWD, "/lib/libm.so.6", O_RDONLY|O_CLOEXEC)`

Открытие системной библиотеки libm.so.6, что является обязательным для большинства программ, использующих стандартные функции.

7. `mmap(0x7ff94b06e000, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0)`

Загрузка библиотеки libm.so.6 в память. Процесс также меняет права доступа и фиксирует области памяти.

8. +++ exited with 0 +++

Процесс завершился с кодом возврата 0, что означает успешное выполнение программы.

main2.c

1. `execve("./main2", ["./main2"], ...)`

Запуск программы main2, аналогично первому случаю. Программа запускается без ошибок.

2. `mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)`

Как и в main1, происходит выделение памяти для программы.

3. `openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC)`

Открытие кэша динамических библиотек для загрузки необходимых зависимостей.

4. `openat(AT_FDCWD, "/lib/aarch64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC)`

Попытка загрузки системной библиотеки libc.so.6. Эта библиотека критична для работы программы, она загружается в память.

5. `mmap(0x7fe7cc965000, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0)`

Загружается libc.so.6 с изменением прав доступа и установкой соответствующих областей памяти.

6. `mprotect(0xffff92bbd000, 77824, PROT_NONE)`

Операция, блокирующая доступ к части памяти для защиты от изменений (для безопасности).

7. `openat(AT_FDCWD, "./lib/libfunc1.so", O_RDONLY|O_CLOEXEC)`

Открытие библиотеки libfunc1.so для загрузки. Система пытается найти ее в различных путях, включая локальные каталоги.

8. `mmap(0x7fe7ccb90000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000)`

Загрузка libfunc1.so в память с правами доступа на чтение и выполнение.

9. `openat(AT_FDCWD, "/lib/aarch64-linux-gnu/libc.so.6",
O_RDONLY|O_CLOEXEC)`

Загрузка математической библиотеки `libm.so.6`, которая обычно требуется для выполнения математических операций.

10. `openat(AT_FDCWD, "./lib/libfunc2.so", O_RDONLY|O_CLOEXEC)`

Попытка загрузки другой динамической библиотеки `libfunc2.so`, возможно, для использования других функций, специфичных для программы.

11. `+++ exited with 0 +++`

Программа завершена успешно с кодом 0.

Демонстрация работы программы

(base) boopie@MacBook-Air-Irina lab4 % ./main1

Введите 'выход' для выхода.

Введите команду: 1

Введите параметры (A deltaX): 0.5 0.9

Результат производной: -0.786239

Введите команду: 2

Введите параметр (x): -1

Результат E: inf

Введите команду: выход

(base) boopie@MacBook-Air-Irina lab4 % ./main2

Введите 'выход' для выхода.

Введите '0', чтобы переключиться между библиотеками.

Введите команду: 1

Введите параметры (A deltaX): 0.3 1

Результат производной: -0.687838

Введите команду: 2

Введите параметр (x): 10

Результат E: 2.593742

Введите команду: 0

переключиться в библиотеку: ./lib/libfunc2.so

Введите команду: 1

Введите параметры (A deltaX): 0.3 1

Результат производной: -0.248672

Введите команду: 2

Введите параметр (x): 9

Результат E: 2.718282

Введите команду: выход

Вывод

В ходе выполнения лабораторной работы была реализована программа, демонстрирующая два подхода к вычислению производной функции $\cos(x)$ и числа “e” с использованием динамических библиотек. Первый подход, реализованный в файле `main1.c`, использует подключение библиотеки на этапе компиляции. Этот метод является простым и надежным, так как все функции подключаются заранее, что гарантирует корректность работы программы.

Второй подход, реализованный в файле `main2.c`, использует динамическую загрузку библиотеки с помощью функций из библиотеки `dlfcn.h`. Этот метод предоставляет гибкость, позволяя загружать библиотеку в процессе выполнения программы, что делает возможным обновление или замену библиотеки без необходимости перекомпиляции всей программы.

Оба подхода продемонстрировали свою эффективность и корректность в вычислениях, а также показали различия в подходах к взаимодействию с библиотеками. Работа над лабораторной была полезна для изучения принципов работы с динамическими библиотеками и их применения в реальных задачах.

Приложение 1

src/func.h

```
#ifndef FUNC_H
#define FUNC_H

float Derivative(float A, float deltaX);
float E(int x);

#endif // FUNC_H
```

src/func1.c

```
#include <math.h>
#include "func.h"

// Первая реализация производной
float Derivative(float A, float deltaX) {
    return (cos(A + deltaX) - cos(A)) / deltaX;
}

// Первая реализация e
float E(int x) {
    return pow(1 + 1.0 / x, x);
}
```

src/func2.c

```
#include <math.h>
#include "func.h"

// Вторая реализация производной
float Derivative(float A, float deltaX) {
    return (cos(A + deltaX) - cos(A - deltaX)) / (2 * deltaX);
}

// Вторая реализация e
float E(int x) {
    float result = 1.0, term = 1.0;
    for (int n = 1; n <= x; n++) {
        term /= n;
        result += term;
    }
    return result;
}
```

src/main1.c

```
#include <stdio.h>
#include <string.h>
#include "func.h"

int main() {
    char command[256];
```

```

printf("Введите 'выход' для выхода.\n");

while (1) {
    printf("Введите команду: ");
    scanf("%s", command);

    if (strcmp(command, "выход") == 0) {
        break;
    }

    if (strcmp(command, "1") == 0) {
        float A, deltaX;
        printf("Введите параметры (A deltaX): ");
        scanf("%f %f", &A, &deltaX);
        printf("Результат производной: %f\n", Derivative(A, deltaX));
    } else if (strcmp(command, "2") == 0) {
        int x;
        printf("Введите параметр (x): ");
        scanf("%d", &x);
        printf("Результат E: %f\n", E(x));
    } else {
        printf("Неизвестная команда.\n");
    }
}

return 0;
}

```

src/main2.c

```

#include <stdio.h>
#include <stdlib.h>
#include <dlfcn.h>
#include <string.h>
#include "func.h"

int main() {
    void *libHandle;
    float (*Derivative)(float, float);
    float (*E)(int);
    char command[256];
    //char currentLib[256] = "./lib/libfunc1.so";
    char currentLib[256] = "./lib/libfunc1.dylib";

    printf("Введите 'выход' для выхода.\n");
    printf("Введите '0', чтобы переключиться между библиотеками.\n");

    // Загрузка первой библиотеки
    libHandle = dlopen(currentLib, RTLD_LAZY);
    if (!libHandle) {
        fprintf(stderr, "Ошибка загрузки библиотеки: %s\n", dlerror());
        return 1;
    }
}

```

```

while (1) {
    printf("Введите команду: ");
    scanf("%s", command);

    if (strcmp(command, "ВЫХОД") == 0) {
        break;
    }

    if (strcmp(command, "0") == 0) {
        dlclose(libHandle);
        //if (strcmp(currentLib, "./lib/libfunc1.so") == 0) {
        //    strcpy(currentLib, "./lib/libfunc2.so");
        //} else {
        //    strcpy(currentLib, "./lib/libfunc1.so");
        //}

        if (strcmp(currentLib, "./lib/libfunc1.dylib") == 0) {
            strcpy(currentLib, "./lib/libfunc2.dylib");
        } else {
            strcpy(currentLib, "./lib/libfunc1.dylib");
        }

        libHandle = dlopen(currentLib, RTLD_LAZY);
        if (!libHandle) {
            fprintf(stderr, "Ошибка загрузки библиотеки: %s\n", dlerror());
            return 1;
        }

        printf("переключиться в библиотеку: %s\n", currentLib);
        continue;
    }

    // Загрузка функций из библиотеки
    Derivative = dlsym(libHandle, "Derivative");
    E = dlsym(libHandle, "E");

    if (!Derivative || !E) {
        fprintf(stderr, "Функции не найдены: %s\n", dlerror());
        dlclose(libHandle);
        return 1;
    }

    if (strcmp(command, "1") == 0) {
        float A, deltaX;
        printf("Введите параметры (A deltaX): ");
        scanf("%f %f", &A, &deltaX);
        printf("Результат производной: %f\n", Derivative(A, deltaX));
    } else if (strcmp(command, "2") == 0) {
        int x;
        printf("Введите параметр (x): ");
        scanf("%d", &x);
        printf("Результат E: %f\n", E(x));
    } else {
        printf("Неизвестная команда.\n");
    }
}

// Заккрытие библиотеки перед выходом

```

```
    dlclose(libHandle);  
    return 0;  
}
```

Makefile

```
# Компилятор  
CC = gcc  
  
# Флаги для создания динамических библиотек  
CFLAGS = -fPIC -shared  
LDFLAGS = -lm # Подключение math.h  
  
# Правило по умолчанию  
all: lib/libfunc1.dylib lib/libfunc2.dylib main1 main2  
  
#all: lib/libfunc1.so lib/libfunc2.so main1 main2  
  
# Создание динамической библиотеки libfunc1.dylib  
lib/libfunc1.dylib: src/func1.c src/func.h  
    $(CC) $(CFLAGS) -o $@ $< $(LDFLAGS)  
  
# Создание динамической библиотеки libfunc2.dylib  
lib/libfunc2.dylib: src/func2.c src/func.h  
    $(CC) $(CFLAGS) -o $@ $< $(LDFLAGS)  
  
# Создание динамической библиотеки libfunc1.so  
#lib/libfunc1.so: src/func1.c src/func.h  
#    $(CC) $(CFLAGS) -o $@ $< $(LDFLAGS) # <- Добавлен -lm  
  
# Создание динамической библиотеки libfunc2.so  
#lib/libfunc2.so: src/func2.c src/func.h  
#    $(CC) $(CFLAGS) -o $@ $< $(LDFLAGS) # <- Добавлен -lm  
  
# Компиляция main1 с использованием libfunc1.dylib  
main1: src/main1.c  
    $(CC) -o $@ $< -L./lib -lfunc1 $(LDFLAGS) -Wl,-rpath,./lib  
  
# Компиляция main2 (если требуется динамическая загрузка)  
main2: src/main2.c  
    $(CC) -o $@ $< $(LDFLAGS)  
  
# Компиляция main1 с использованием libfunc1.so  
#main1: src/main1.c  
#    $(CC) -o $@ $< -L./lib -lfunc1 $(LDFLAGS) -Wl,-rpath=./lib # <- Добавлен -lm  
  
# Компиляция main2 (если требуется динамическая загрузка)  
#main2: src/main2.c  
#    $(CC) -o $@ $< $(LDFLAGS)  
  
# Очистка  
clean:  
    rm -f main1 main2 lib/*.dylib  
  
# Очистка
```

```
#clean:
#  rm -f main1 main2 lib/*.so
```


main1.c

```
root@ctf4c85db2e:/workspace# strace ./main1
execve("./main1", ["/./main1"], 0xffffd34baad0 /* 9 vars */) = 0
brk(NULL)                                = 0xaaab0098d000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xfffffb352e000
fcntl(AT_FDCWD, "etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/libfunc1.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\x177ELF\x2\x11\x00\x00\x00\x00\x03\x0267\x01\x00\x00\x00\x00\x00"... , 832) = 832
fststat(3, {st_mode=S_IFREG|0755, st_size=69344, ...}) = 0
getcwd("/workspace", 128)              = 11
mmap(NULL, 196656, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_DENYWRITE, -1, 0) = 0xfffffb34c5000
mmap(0xfffffb34d000, 131120, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xfffffb34d000
munmap(0xfffffb34c5000, 45056)         = 0
munmap(0xfffffb34f1000, 16432)        = 0
mprotect(0xfffffb34d1000, 122880, PROT_NONE) = 0
mmap(0xfffffb34ef000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0xfffffb34ef000
close(3)                               = 0
openat(AT_FDCWD, "/lib/libm.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fststat(3, {st_mode=S_IFREG|0644, st_size=9167, ...}) = 0
mmap(NULL, 9167, PROT_READ, MAP_PRIVATE, 3, 0) = 0xfffffb352b000
close(3)                               = 0
openat(AT_FDCWD, "/lib/aarch64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\x177ELF\x2\x11\x00\x00\x00\x00\x03\x0267\x01\x00\x00\x00\x00\x00"... , 832) = 832
fststat(3, {st_mode=S_IFREG|0644, st_size=591800, ...}) = 0
mmap(NULL, 720920, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_DENYWRITE, -1, 0) = 0xfffffb341f000
mmap(0xfffffb3420000, 655384, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xfffffb3420000
munmap(0xfffffb341f000, 4096)         = 0
munmap(0xfffffb34c1000, 57368)        = 0
mprotect(0xfffffb34a5000, 106496, PROT_NONE) = 0
mmap(0xfffffb34bf000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0xfffffb34bf000
close(3)                               = 0
openat(AT_FDCWD, "/lib/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/aarch64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\x177ELF\x2\x11\x00\x00\x00\x00\x03\x0267\x01\x00\x00\x0360\x206\x20\x00\x00"... , 832) = 832
fststat(3, {st_mode=S_IFREG|0755, st_size=1722920, ...}) = 0
mmap(NULL, 1892240, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_DENYWRITE, -1, 0) = 0xfffffb3252000
mmap(0xfffffb3260000, 1826704, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xfffffb3260000
munmap(0xfffffb3252000, 57344)       = 0
munmap(0xfffffb341e000, 8080)        = 0
mprotect(0xfffffb33fa000, 77824, PROT_NONE) = 0
mmap(0xfffffb340d000, 20480, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0xfffffb340d000
mmap(0xfffffb3412000, 49040, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0xfffffb3412000
close(3)                               = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xfffffb3529000
set_tid_address(0xfffffb35290f0)      = 1750
set_robust_list(0xfffffb3529100, 24)  = 0
```

```
req(0xfffffb3529740, 0x20, 0, 0xd428bc00) = 0
mprotect(0xfffffb340d000, 12288, PROT_READ) = 0
mprotect(0xfffffb34bf000, 4096, PROT_READ) = 0
mprotect(0xfffffb34ef000, 4096, PROT_READ) = 0
mprotect(0xaaadbaf000, 4096, PROT_READ) = 0
mprotect(0xfffffb3534000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0xfffffb352b000, 9167) = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
getrandom("\x33\x7e\x17\xb3\x55\x72\xa1\x37", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0xaaab0098d000
brk(0xaaab009ae000) = 0xaaab009ae000
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\262\321\213\321\205\320\276\320\264' \320\264\320\273"
..., 49Введите 'выход' для выхода.
) = 49
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\272\320\276\320\274\320\260\320\275\320\264\321\203: ",
31Введите команду: ) = 31
read(0, 1
"1\n", 1024) = 2
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\277\320\260\321\200\320\260\320\274\320\265\321\202\321
\200\321"..., 46Введите параметры (A deltaX): ) = 46
read(0, 0.5 3
"0.5 3\n", 1024) = 6
write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202 \320\277\321\200\320\276\320\270\320\267\320
\262\320"..., 53Результат производной: -0.604680
) = 53
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\272\320\276\320\274\320\260\320\275\320\264\321\203: ",
31Введите команду: ) = 31
read(0, 2
"2\n", 1024) = 2
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\277\320\260\321\200\320\260\320\274\320\265\321\202\321
\200 "..., 37Введите параметр (x): ) = 37
read(0, 1
"1\n", 1024) = 2
write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202 E: 2.000000\n", 31Результат E: 2.000000
) = 31
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\272\320\276\320\274\320\260\320\275\320\264\321\203: ",
31Введите команду: ) = 31
read(0, выход
"\320\262\321\213\321\205\320\276\320\264\n", 1024) = 11
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)
exit_group(0) = ?
+++ exited with 0 +++
root@6fff4c85dhe2: /workspace#
```

main2.c

```
root@cf6ff4c85dbe2:/workspace# strace ./main2
execve("./main2", ["/main2"], 0xffffcb533d60 /* 9 vars */) = 0
brk(NULL) = 0xaaaa9279000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xffff92dc3000
faccessat(AT_FDCWD, "/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=9167, ...}) = 0
mmap(NULL, 9167, PROT_READ, MAP_PRIVATE, 3, 0) = 0xffff92dc0000
close(3) = 0
openat(AT_FDCWD, "/lib/aarch64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\360\206\2\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1722920, ...}) = 0
mmap(NULL, 1892240, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_DENYWRITE, -1, 0) = 0xffff92bbd000
mmap(0xffff92bc0000, 1826704, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xffff92bc0000
munmap(0xffff92bbd000, 12288) = 0
munmap(0xffff92d7e000, 53136) = 0
mprotect(0xffff92d5a000, 77824, PROT_NONE) = 0
mmap(0xffff92d6d000, 20480, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0xffff92d6d000
mmap(0xffff92d72000, 49040, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0xffff92d72000
close(3) = 0
set_tid_address(0xffff92dc3fb0) = 1877
set_robust_list(0xffff92dc3fc0, 24) = 0
rseq(0xffff92dc4600, 0x20, 0, 0xd428bc00) = 0
mprotect(0xffff92d6d000, 12288, PROT_READ) = 0
mprotect(0xaaaa9237f000, 4096, PROT_READ) = 0
mprotect(0xffff92dc9000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0xffff92dc0000, 9167) = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
getrandom("\x8a\x15\xe0\xfc\xae\x5a\x63\x27", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0xaaaa9279000
brk(0xaaaa929a000) = 0xaaaa929a000
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 '\320\262\321\213\321\205\320\276\320\264' \320\264\320\273"... , 49Введите 'выход' для выхода.
) = 49
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 '0', \321\207\321\202\320\276\320\261\321\213 \320"... , 95Введите '0', чтобы переключиться между библиотеками.
) = 95
openat(AT_FDCWD, "./lib/libfunc1.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=69568, ...}) = 0
getcwd("/workspace", 128) = 11
mmap(NULL, 196656, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_DENYWRITE, -1, 0) = 0xffff92b8f000
mmap(0xffff92b90000, 131120, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xffff92b90000
munmap(0xffff92b8f000, 4096) = 0
munmap(0xffff92bb1000, 57392) = 0
```

```
mprotect(0xffff92b91000, 122880, PROT_NONE) = 0
mmap(0xffff92baf000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0xffff92baf000
close(3) = 0
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=9167, ...}) = 0
mmap(NULL, 9167, PROT_READ, MAP_PRIVATE, 3, 0) = 0xffff92dc0000
close(3) = 0
openat(AT_FDCWD, "/lib/aarch64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=591800, ...}) = 0
mmap(NULL, 720920, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_DENYWRITE, -1, 0) = 0xffff92adf000
mmap(0xffff92ae0000, 655384, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xffff92ae0000
munmap(0xffff92adf000, 4096) = 0
munmap(0xffff92b81000, 57368) = 0
mprotect(0xffff92b65000, 106496, PROT_NONE) = 0
mmap(0xffff92b7f000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8f000) = 0xffff92b7f000
close(3) = 0
mprotect(0xffff92b7f000, 4096, PROT_READ) = 0
mprotect(0xffff92baf000, 4096, PROT_READ) = 0
munmap(0xffff92dc0000, 9167) = 0
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\272\320\276\320\274\320\260\320\275\320\264\321\203: ", 31Введите команду: ) = 31
read(0, 1
"1\n", 1024) = 2
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\277\320\260\321\200\320\260\320\274\320\265\321\202\321\200\321"... , 46Введите параметры (A deltaX): ) = 46
read(0, 0.5 3
"0.5 3\n", 1024) = 6
write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202 \320\277\321\200\320\276\320\270\320\267\320\262\320"... , 53Результат производной: -0.604680
) = 53
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\272\320\276\320\274\320\260\320\275\320\264\321\203: ", 31Введите команду: ) = 31
read(0, 2
"2\n", 1024) = 2
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\277\320\260\321\200\320\260\320\274\320\265\321\202\321\200"... , 37Введите параметр (x): ) = 37
read(0, 20
"20\n", 1024) = 3
write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202 E: 2.653298\n", 31Результат E: 2.653298
) = 31
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\272\320\276\320\274\320\260\320\275\320\264\321\203: ", 31Введите команду: ) = 31
read(0, 0
"0\n", 1024) = 2
```

```

munmap(0xffff92b90000, 131120) = 0
munmap(0xffff92ae0000, 655384) = 0
openat(AT_FDCWD, "./lib/libfunc2.so", 0_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=69520, ...}) = 0
getcwd("/workspace", 128) = 11
mmap(NULL, 196648, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_DENYWRITE, -1, 0) = 0xffff92b8f000
mmap(0xffff92b90000, 131112, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xffff92b90000
munmap(0xffff92b8f000, 4096) = 0
munmap(0xffff92bb1000, 57384) = 0
mprotect(0xffff92b91000, 122880, PROT_NONE) = 0
mmap(0xffff92baf000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0xffff92baf000
close(3) = 0
openat(AT_FDCWD, "/etc/ld.so.cache", 0_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=9167, ...}) = 0
mmap(NULL, 9167, PROT_READ, MAP_PRIVATE, 3, 0) = 0xffff92dc0000
close(3) = 0
openat(AT_FDCWD, "/lib/aarch64-linux-gnu/libm.so.6", 0_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=591800, ...}) = 0
mmap(NULL, 720920, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_DENYWRITE, -1, 0) = 0xffff92adf000
mmap(0xffff92ae0000, 655384, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xffff92ae0000
munmap(0xffff92adf000, 4096) = 0
munmap(0xffff92bb1000, 57368) = 0
mprotect(0xffff92b65000, 106496, PROT_NONE) = 0
mmap(0xffff92b7f000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8f000) = 0xffff92b7f000
close(3) = 0
mprotect(0xffff92b7f000, 4096, PROT_READ) = 0
mprotect(0xffff92baf000, 4096, PROT_READ) = 0
munmap(0xffff92dc0000, 9167) = 0
write(1, "\320\277\320\265\321\200\320\265\320\272\320\273\321\216\321\207\320\270\321\202\321\214\321\201\321\217 \320\262 \320\261"... , 70переключиться в библиотеку: ./lib/libfunc2.so
) = 70
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\272\320\276\320\274\320\260\320\275\320\264\321\203: ", 31Введите команду: ) = 31
read(0, 1
"1\n", 1024) = 2
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\277\320\260\321\200\320\260\320\274\320\265\321\202\321\200\321"... , 46Введите параметры (A deltaX): ) = 46
read(0, 0.5 3
"0.5 3\n", 1024) = 6
write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202 \320\277\321\200\320\276\320\270\320\267\320\262\320"... , 53Результат производной: -0.022552
) = 53
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\272\320\276\320\274\320\260\320\275\320\264\321\203: ", 31Введите команду: ) = 31

```

```

read(0, выход
"\320\262\321\213\321\205\320\276\320\264\n", 1024) = 11
munmap(0xffff92b90000, 131112) = 0
munmap(0xffff92ae0000, 655384) = 0
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)
exit_group(0) = ?
+++ exited with 0 +++
root@cf4c85dbe2:/workspace#

```