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: М8О-209БВ-24

Студент: Лисов Д.С.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 4.12.25

Москва, 2025

Постановка задачи

Вариант 22.

Требуется создать динамические библиотеки, которые реализуют заданный вариантом функционал. 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:

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

Провести анализ двух типов использования библиотек.

Пользовательский ввод для обоих 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). Можно реализовать лабораторную работу без данной функции, но максимальная оценка в этом случае будет «хорошо»;
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.

Функция 1: вычисление GCD для двух чисел (алгоритмом Евклида и наивным алгоритмом)
Функция 2: рассчёт значения числа пи при заданной длине ряда (формула Лейбница, формула Валлиса)

Общий метод и алгоритм решения

Использованные системные вызовы:

- dlopen() - загрузка библиотеки;
- dlsym() - поиск символа;
- dlclose() - выгрузка библиотеки;
- dlerror() - обработка ошибок;

Алгоритм заданы в условии задачи.

Код программы

contract.h

```
#ifndef CONTRACT_H
#define CONTRACT_H


typedef int (*gcd_func)(int, int);


typedef double(*pi_func)(int);


typedef struct {
```

```
	gcd_func gcd;
	pi_func pi;
	const char* name;
} LibraryFunctions;

#endif
```

libeuclid_leibniz.c

```
#include <stdio.h>

int gcd_euclid(int a, int b) {
	while (b != 0) {
		int t = b;
		b = a % b;
		a = t;
	}
	if (a < 0)
		return -a;
	return a;
}

double pi_leibniz(int iterations) {
	if (iterations <= 0) return 0.0;

	double pi = 0.0;
	for (int i = 0; i < iterations; ++i) {
		double term = 1.0 / (2 * i + 1);
		if (i % 2 == 0) {
			pi += term;
		} else {
			pi -= term;
		}
	}
	return 4.0 * pi;
}
```

libnaive_wallis.c

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

int gcd_naive(int a, int b) {
	if (a < 0) a = -a;
	if (b < 0) b = -b;

	int min = (a < b) ? a : b;
	int res = 1;
	for (int i = 2; i <= min; ++i) {
		if (a % i == 0 && b % i == 0) {
			res = i;
```

```
            }
      }

      return res;
}

double pi_wallis(int iterations) {
      if (iterations <= 0) return 0.0;

      double pi = 1.0;
      for (int i = 1; i <= iterations; ++i) {
            double n = (double)i;
            double term = (4.0 * n * n) / (4.0 * n * n - 1.0);
            pi *= term;
      }
      return 2.0 * pi;
}
```

program1.c

```
#include <stdio.h>
#include <stdlib.h>
#include "contract.h"

extern int gcd_euclid(int a, int b);
extern double pi_leibniz(int iterations);

int main() {
      LibraryFunctions lib = {
            .gcd = gcd_euclid,
            .pi = pi_leibniz,
            .name="Euclid+Leibniz (static link)"
      };

      printf("Программа 1: Использование библиотеки на этапе компиляции\n");
      printf("Текущая библиотека: %s\n\n", lib.name);
      printf("Доступные команды:\n");
      printf("\t0 - информация о библиотеке\n");
      printf("\t1 a b - вычислить НОД чисел a и b\n");
      printf("\t2 n - вычислить Пи с n итерациями\n");
      printf("\tq - выход\n\n");

      char command[100];
      while (1) {
            printf("> ");
            if (fgets(command, sizeof(command), stdin) == NULL) {
                  break;
            }
```

```
            if (command[0] == 'q' || command[0] == 'Q') {
                  break;
            }

            if (command[0] == '0') {
                  printf("Текущая библиотека: %s\n", lib.name);
                  printf("Алгоритм НОД: Евклида\n");
                  printf("Алгоритм Пи: Формула Лейбница\n");
            } else if (command[0] == '1') {
                  int a, b;
                  if (sscanf(command + 1, "%d %d", &a, &b) == 2) {
                        int result = lib.gcd(a, b);
                        printf("НОД(%d, %d) = %d\n", a, b, result);
                  } else {
                        printf("Ошибка: требуется 2 числа\n");
                  }
            } else if (command[0] == '2') {
                  int n;
                  if (sscanf(command + 1, "%d", &n) == 1) {
                        if (n > 0) {
                              double result = lib.pi(n);
                              printf("Число Пи:(%d итераций)= %.15f\n", n,
result);
                        } else {
                              printf("Ошибка: количество итераций должно быть
положительным числом!\n");
                        }
                  } else {
                        printf("Ошибка: требуется одно число!\n");
                  }
            } else if (command[0] != '\n') {
                  printf("Неизвестная команда\n");
            }
      }

      return 0;
}
```

program2.c

```
#include <stdlib.h>
#include <stdio.h>
#include <dlfcn.h>
#include <string.h>
#include "contract.h"

const char* LIBRARY_NAMES[] = {
      "./libeuclid_leibniz.so",
      "./libnaive_wallis.so"
};
```

```
const int NUM_LIBS = 2;

void* current_lib_handle = NULL;
LibraryFunctions current_lib;
int current_lib_index = 0;

int load_library(int index) {
	if (index < 0 || index >= NUM_LIBS) {
		return 0;
	}

	if (current_lib_handle != NULL) {
		dlclose(current_lib_handle);
	}

	current_lib_handle = dlopen(LIBRARY_NAMES[index], RTLD_LAZY);
	if (!current_lib_handle) {
		fprintf(stderr, "Ошибка загрузки библиотеки: %s\n", dlerror());
		return 0;
	}

	gcd_func gcd_ptr = (gcd_func)dlsym(current_lib_handle, index == 0 ?
"gcd_euclid" : "gcd_naive");
	pi_func pi_ptr = (pi_func)dlsym(current_lib_handle, index == 0 ?
"pi_leibniz" : "pi_wallis");

	if (!gcd_ptr || !pi_ptr) {
		fprintf(stderr, "Ошибка загрузки функции: %s\n", dlerror());
		dlclose(current_lib_handle);
		return 0;
	}

	current_lib.gcd = gcd_ptr;
	current_lib.pi = pi_ptr;
	current_lib.name = index == 0 ? "Euclid+Leibniz" : "Naive+Wallis";
	current_lib_index = index;

	return 1;
}

void print_library_info() {
	printf("Доступные библиотеки:\n");
	for (int i = 0; i < NUM_LIBS; ++i) {
		printf("    %d: %s", i, LIBRARY_NAMES[i]);
		if (i == current_lib_index)
			printf(" (текущая)");
```

```
            printf("\n");
      }
      printf("\n");
}

int main() {
      printf("Программа 2: Динамическая загрузка библиотек\n\n");
      if (!load_library(0)) {
            fprintf(stderr, "Не удалось загрузить библиотеку по умолчанию\n");
            return 1;
      }

      print_library_info();

      printf("Доступные команды:\n");
      printf("    0 - переключить библиотеку\n");
      printf("    1 a b - Вычислить НОД 2 чисел a и b\n");
      printf("    2 n - Вычислить Пи с n итерациями\n");
      printf("    i - информация о текущей библиотеке\n");
      printf("    q - выход\n\n");

      char command[100];
      while (1) {
            printf("> ");
            if (fgets(command, sizeof(command), stdin) == NULL) {
                  break;
            }

            command[strcspn(command, "\n")] == 0;
            if(command[0] == 'q' || command[0] == 'Q') {
                  break;
            }

            if (command[0] == '0') {
                  int new_index = (current_lib_index + 1) % NUM_LIBS;
                  if (load_library(new_index)) {

current_lib.name);      printf("Переключено на библиотеку: %s\n",
                  } else {
                        printf("Ошибка переключения библиотеки\n");
                  }
            }

            else if (command[0] == 'i' || command[0] == 'I') {
```

```
                printf("Текущая библиотека: %s\n", current_lib.name);
                printf("Индекс: %d\n", current_lib_index);
                printf("Путь: %s\n", LIBRARY_NAMES[current_lib_index]);
            }
            else if (command[0] == '1') {
                int a, b;
                if (sscanf(command + 1, "%d %d", &a, &b) == 2) {
                    int result = current_lib.gcd(a, b);
                    printf("НОД(%d, %d) = %d\n", a, b, result);
                } else {
                    printf("Ошибка: требуется 2 числа\n");
                }
            }
            else if (command[0] == '2') {
                int n;
                if (sscanf(command + 1, "%d", &n) == 1) {
                    if (n > 0) {
                        double result = current_lib.pi(n);
                        printf("Пи(%d итераций) = %.15f\n", n, result);
                    } else {
                        printf("Ошибка: количество итераций должно быть
положительным числом\n");
                    }
                } else {
                    printf("Ошибка: требуется одно число\n");
                }
            }
            else if (command[0] != '\0') {
                printf("Неизвестная команда\n");
            }
        }

        if (current_lib_handle != NULL) {
            dlclose(current_lib_handle);
        }

        return 0;
}
```

Протокол работы программы

Тестирование:

```
$ ./program1
Программа 1: Использование библиотеки на этапе компиляции
Текущая библиотека: Euclid+Leibniz (static link)

Доступные команды:
      0 - информация о библиотеке
      1 a b - вычислить НОД чисел a и b
```

```
      2 n - вычислить Пи с n итерациями
      q - выход


> 0
Текущая библиотека: Euclid+Leibniz (static link)
Алгоритм НОД: Евклида
Алгоритм Пи: Формула Лейбница
> 1 1000 2251592
НОД(1000, 2251592) = 8
> 2 16
Число Пи:(16 итераций)= 3.079153394197428
> 2 10000
Число Пи:(10000 итераций)= 3.141492653590034
> q


$ ./program2
Программа 2: Динамическая загрузка библиотек


Доступные библиотеки:
      0: ./libeuclid_leibniz.so (текущая)
      1: ./libnaive_wallis.so


Доступные команды:
      0 - переключить библиотеку
      1 a b - Вычислить НОД 2 чисел a и b
      2 n - Вычислить Пи с n итерациями
      i - информация о текущей библиотеке
      q - выход


> i
Текущая библиотека: Euclid+Leibniz
Индекс: 0
Путь: ./libeuclid_leibniz.so
> 0
Переключено на библиотеку: Naive+Wallis
> i
Текущая библиотека: Naive+Wallis
Индекс: 1
Путь: ./libnaive_wallis.so
> 1 1000 16
НОД(1000, 16) = 8
> 2 12985792
Пи(12985792 итераций) = 3.141592593282112
> 0
Переключено на библиотеку: Euclid+Leibniz
> 1 67127 15217
НОД(67127, 15217) = 1
```

```
> 1 2434 12564
НОД(2434, 12564) = 2
> 2 100
Пи(100 итераций) = 3.131592903558554
> q
```

Strace:

```
$ strace -f ./program1
execve("./program1", ["./program1"], 0x7ffecca77bb8 /* 63 vars */) = 0
brk(NULL)                                       = 0x5c223ae3d000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x75b6cec27000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Нет такого файла или
каталога)
openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v3/libeuclid_leibniz.so", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v2/libeuclid_leibniz.so", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "./libeuclid_leibniz.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
fstat(3, {st_mode=S_IFREG|0775, st_size=15232, ...}) = 0
getcwd("/home/demetrio/MyCodesZZZZZ/OS_LABS/Lab4", 128) = 41
mmap(NULL, 16400, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x75b6cec22000
mmap(0x75b6cec23000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1000) = 0x75b6cec23000
mmap(0x75b6cec24000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x2000) = 0x75b6cec24000
mmap(0x75b6cec25000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2000) = 0x75b6cec25000
close(3)                                        = 0
openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v3/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v2/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "./libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (Нет такого
файла или каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=153815, ...}) = 0
mmap(NULL, 153815, PROT_READ, MAP_PRIVATE, 3, 0) = 0x75b6cebfc000
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\0\220\243\2\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
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x75b6ce800000
mmap(0x75b6ce828000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x28000) = 0x75b6ce828000
mmap(0x75b6ce9b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1b0000) = 0x75b6ce9b0000
mmap(0x75b6ce9ff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1fe000) = 0x75b6ce9ff000
mmap(0x75b6cea05000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x75b6cea05000
close(3)                                        = 0
```

```
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x75b6cebf9000
arch_prctl(ARCH_SET_FS, 0x75b6cebf9740) = 0
set_tid_address(0x75b6cebf9a10)         = 11019
set_robust_list(0x75b6cebf9a20, 24)     = 0
rseq(0x75b6cebfa060, 0x20, 0, 0x53053053) = 0
mprotect(0x75b6ce9ff000, 16384, PROT_READ) = 0
mprotect(0x75b6cec25000, 4096, PROT_READ) = 0
mprotect(0x5c22238f5000, 4096, PROT_READ) = 0
mprotect(0x75b6cec5f000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x75b6cebfc000, 153815)          = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
getrandom("\x6d\xa8\x29\xde\xc7\x24\x89\xff", 8, GRND_NONBLOCK) = 8
brk(NULL)                               = 0x5c223ae3d000
brk(0x5c223ae5e000)                     = 0x5c223ae5e000
write(1, "\
320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260 1: \
320\230\321\201\320\277\320\276\320\273"..., 107Программа 1: Использование
библиотеки на этапе компиляции
) = 107
write(1, "\320\242\320\265\320\272\321\203\321\211\320\260\321\217 \
320\261\320\270\320\261\320\273\320\270\320\276\321\202\320\265\320"..., 67Текущая
библиотека: Euclid+Leibniz (static link)


) = 67
write(1, "\
320\224\320\276\321\201\321\202\321\203\320\277\320\275\321\213\320\265 \
320\272\320\276\320\274\320\260\320\275\320\264\321"..., 35Доступные команды:
) = 35
write(1, "\t0 - \
320\270\320\275\321\204\320\276\321\200\320\274\320\260\321\206\320\270\321\217 \
320\276 \320\261\320"..., 50	0 - информация о библиотеке
) = 50
write(1, "\t1 a b - \
320\262\321\213\321\207\320\270\321\201\320\273\320\270\321\202\321\214 \
320\235\320\236"..., 53	1 a b - вычислить НОД чисел a и b
) = 53
write(1, "\t2 n - \
320\262\321\213\321\207\320\270\321\201\320\273\320\270\321\202\321\214 \
320\237\320\270 \321"..., 57	2 n - вычислить Пи с n итерациями
) = 57
write(1, "\tq - \320\262\321\213\321\205\320\276\320\264\n", 16   q - выход
) = 16
write(1, "\n", 1
)                       = 1
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "> ", 2> )                       = 2
read(0, 0
"0\n", 1024)                    = 2
write(1, "\320\242\320\265\320\272\321\203\321\211\320\260\321\217 \
320\261\320\270\320\261\320\273\320\270\320\276\321\202\320\265\320"..., 66Текущая
библиотека: Euclid+Leibniz (static link)
) = 66
write(1, "\320\220\320\273\320\263\320\276\321\200\320\270\321\202\320\274 \
320\235\320\236\320\224: \320\225\320\262\320\272\320"..., 40Алгоритм НОД: Евклида
) = 40
write(1, "\320\220\320\273\320\263\320\276\321\200\320\270\321\202\320\274 \
320\237\320\270: \320\244\320\276\321\200\320\274\321"..., 55Алгоритм Пи: Формула
Лейбница
```

```
) = 55
write(1, "> ", 2> )                          = 2
read(0, 1 1000 1252
"1 1000 1252\n", 1024)          = 12
write(1, "\320\235\320\236\320\224(1000, 1252) = 4\n", 23НОД(1000, 1252) = 4
) = 23
write(1, "> ", 2> )                          = 2
read(0, 2 1000
"2 1000\n", 1024)               = 7
write(1, "\320\247\320\270\321\201\320\273\320\276 \320\237\320\270:(1000 \
320\220\321\202\320\265\321\200\320\260 ..., 59число Пи:(1000 итерации)=
3.140592653839794
) = 59
write(1, "> ", 2> )                          = 2
read(0, q
"q\n", 1024)                    = 2
exit_group(0)                             = ?
+++ exited with 0 +++
```

Strace:

```
$ strace -f ./program2
execve("./program2", ["./program2"], 0x7fffddeeea98 /* 63 vars */) = 0
brk(NULL)                                 = 0x5e4d85557000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7bc75e922000
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
fstat(3, {st_mode=S_IFREG|0644, st_size=153815, ...}) = 0
mmap(NULL, 153815, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7bc75e8fc000
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\0\220\243\2\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
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7bc75e600000
mmap(0x7bc75e628000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x28000) = 0x7bc75e628000
mmap(0x7bc75e7b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1b0000) = 0x7bc75e7b0000
mmap(0x7bc75e7ff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1fe000) = 0x7bc75e7ff000
mmap(0x7bc75e805000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7bc75e805000
close(3)                                  = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7bc75e8f9000
arch_prctl(ARCH_SET_FS, 0x7bc75e8f9740) = 0
set_tid_address(0x7bc75e8f9a10)         = 11288
set_robust_list(0x7bc75e8f9a20, 24)     = 0
rseq(0x7bc75e8fa060, 0x20, 0, 0x53053053) = 0
mprotect(0x7bc75e7ff000, 16384, PROT_READ) = 0
```

```
     mprotect(0x5e4d796f7000, 4096, PROT_READ) = 0
     mprotect(0x7bc75e95a000, 8192, PROT_READ) = 0
     prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
     munmap(0x7bc75e8fc000, 153815)          = 0
     fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
     getrandom("\xf2\x65\x58\x99\x9e\xc5\x42\x14", 8, GRND_NONBLOCK) = 8
     brk(NULL)                               = 0x5e4d85557000
     brk(0x5e4d85578000)                     = 0x5e4d85578000
     write(1, "\
320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260 2: \
320\224\320\270\320\275\320\260\320\274"..., 83Программа 2: Динамическая загрузка
библиотек
     ) = 83
     write(1, "\n", 1
     )                          = 1
     openat(AT_FDCWD, "./libeuclid_leibniz.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
     fstat(3, {st_mode=S_IFREG|0775, st_size=15232, ...}) = 0
     getcwd("/home/demetrio/MyCodesZZZZZ/OS_LABS/Lab4", 128) = 41
     mmap(NULL, 16400, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7bc75e91d000
     mmap(0x7bc75e91e000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1000) = 0x7bc75e91e000
     mmap(0x7bc75e91f000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x2000) = 0x7bc75e91f000
     mmap(0x7bc75e920000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2000) = 0x7bc75e920000
     close(3)                                = 0
     mprotect(0x7bc75e920000, 4096, PROT_READ) = 0
     write(1, "\
320\224\320\276\321\201\321\202\321\203\320\277\320\275\321\213\320\265 \
320\261\320\270\320\261\320\273\320\270\320\276\321"..., 41Доступные библиотеки:
     ) = 41
     write(1, "\t0: ./libeuclid_leibniz.so (\321\202\320\265"..., 44   0:
./libeuclid_leibniz.so (текущая)
     ) = 44
     write(1, "\t1: ./libnaive_wallis.so\n", 25     1: ./libnaive_wallis.so
     ) = 25
     write(1, "\n", 1
     )                          = 1
     write(1, "\
320\224\320\276\321\201\321\202\321\203\320\277\320\275\321\213\320\265 \
320\272\320\276\320\274\320\260\320\275\320\264\321"..., 35Доступные команды:
     ) = 35
     write(1, "\t0 - \
320\277\320\265\321\200\320\265\320\272\320\273\321\216\321\207\320\270\321\202\321\
214 \320\261\320\270"..., 49 0 - переключить библиотеку
     ) = 49
     write(1, "\t1 a b - \
320\222\321\213\321\207\320\270\321\201\320\273\320\270\321\202\321\214 \
320\235\320\236"..., 55      1 a b - Вычислить НОД 2 чисел a и b
     ) = 55
     write(1, "\t2 n - \
320\222\321\213\321\207\320\270\321\201\320\273\320\270\321\202\321\214 \
320\237\320\270 \321"..., 57 2 n - Вычислить Пи с n итерациями
     ) = 57
     write(1, "\ti - \
320\220\320\275\321\204\320\276\321\200\320\274\320\260\321\206\320\270\321\217 \
320\276 \321\202\320"..., 65 i - информация о текущей библиотеке
     ) = 65
     write(1, "\tq - \320\262\321\213\321\205\320\276\320\264\n", 16   q - выход
```

```
) = 16
write(1, "\n", 1
)                       = 1
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "> ", 2> )                       = 2
read(0, i
"i\n", 1024)                    = 2
write(1, "\320\242\320\265\320\272\321\203\321\211\320\260\321\217 \
320\261\320\270\320\261\320\273\320\270\320\276\321\202\320\265\320..., 52Текущая
библиотека: Euclid+Leibniz
) = 52
write(1, "\320\230\320\275\320\264\320\265\320\272\321\201: 0\n", 16Индекс: 0
) = 16
write(1, "\320\237\321\203\321\202\321\214: ./libeuclid_leibniz.so"..., 33Путь:
./libeuclid_leibniz.so
) = 33
write(1, "> ", 2> )                       = 2
read(0, 1 1000 23
"1 1000 23\n", 1024)            = 10
write(1, "\320\235\320\236\320\224(1000, 23) = 1\n", 21НОД(1000, 23) = 1
) = 21
write(1, "> ", 2> )                       = 2
read(0, 2 1125
"2 1125\n", 1024)               = 7
write(1, "\320\237\320\270(1125 \
320\270\321\202\320\265\321\200\320\260\321\206\320\270\320\271) = 3."..., 48Пи(1125
итераций) = 3.142481542303102
) = 48
write(1, "> ", 2> )                       = 2
read(0, 0
"0\n", 1024)                    = 2
munmap(0x7bc75e91d000, 16400)           = 0
openat(AT_FDCWD, "./libnaive_wallis.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"...,
832) = 832
fstat(3, {st_mode=S_IFREG|0775, st_size=15224, ...}) = 0
getcwd("/home/demetrio/MyCodesZZZZZ/OS_LABS/Lab4", 128) = 41
mmap(NULL, 16400, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7bc75e91d000
mmap(0x7bc75e91e000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1000) = 0x7bc75e91e000
mmap(0x7bc75e91f000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x2000) = 0x7bc75e91f000
mmap(0x7bc75e920000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2000) = 0x7bc75e920000
close(3)                                = 0
mprotect(0x7bc75e920000, 4096, PROT_READ) = 0
write(1, "\
320\237\320\265\321\200\320\265\320\272\320\273\321\216\321\207\320\265\320\275\320\
276 \320\275\320\260 \320\261\320\270\320"..., 63Переключено на библиотеку: Naive+Wallis
) = 63
write(1, "> ", 2> )                       = 2
read(0, 1 245323 215123
"1 245323 215123\n", 1024)      = 16
write(1, "\320\235\320\236\320\224(245323, 215123) = 1\n", 27НОД(245323,
215123) = 1
) = 27
```

```
write(1, "> ", 2> )                         = 2
read(0, 2 125
"2 125\n", 1024)                 = 6
write(1, "\320\237\320\270(125 \
320\270\321\202\320\265\321\200\320\260\321\206\320\270\320\271) = 3.1"..., 47Пи(125
итераций) = 3.13534074649878 8
) = 47
write(1, "> ", 2> )                         = 2
read(0, i
"i\n", 1024)                     = 2
write(1, "\320\242\320\265\320\272\321\203\321\211\320\260\321\217 \
320\261\320\270\320\261\320\273\320\270\320\276\321\202\320\265\320 ..., 50Текущая
библиотека: Naive+Wallis
) = 50
write(1, "\320\230\320\275\320\264\320\265\320\272\321\201: 1\n", 16Индекс: 1
) = 16
write(1, "\320\237\321\203\321\202\321\214: ./libnaive_wallis.so\n",
31Путь: ./libnaive_wallis.so
) = 31
write(1, "> ", 2> )                         = 2
read(0, q
"q\n", 1024)                     = 2
munmap(0x7bc75e91d000, 16400)           = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```

Вывод

В ходе выполнения лабораторной работы были успешно реализованы и проанализированы два принципиально разных подхода к использованию динамических библиотек в операционной системе Linux. Работа продемонстрировала глубокое понимание механизмов связывания и загрузки исполняемого кода в современных операционных системах.

Созданы две специализированные математические библиотеки с идентичным интерфейсом, разработан единый контракт (contract.h), обеспечивающий совместимость реализаций.

Разработаны 2 подхода к интеграции библиотек:

Программа 1: Статическая линковка на этапе компиляции

Программа 2: Динамическая загрузка во время выполнения с поддержкой "горячей" замены библиотек