

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

Студент: Волков А.Д.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 10.12.2023

Москва, 2023

Постановка задачи

Вариант 5.

№	Описание	Сигнатура	Реализация 1	Реализация 2
1	Рассчет интеграла функции $\sin(x)$ на отрезке $[A, B]$ с шагом e	Float SinIntegral(float A, float B, float e)	Подсчет интеграла методом прямоугольников.	Подсчет интеграла методом трапеций.
5	Рассчет значения числа Пи при заданной длине ряда (K)	float Pi(int K)	Ряд Лейбница	Формула Валлиса

Общий метод и алгоритм решения

Использованные системные вызовы:

- `void *dlopen(const char *filename, int flag);` – загружает динамическую библиотеку, имя которой указано в строке `filename`, и возвращает прямой указатель на начало динамической библиотеки;
- `void *dlsym(void *handle, char *symbol);` – использует указатель на динамическую библиотеку, возвращаемую `dlopen`, и оканчивающееся нулем символьное имя, а затем возвращает адрес, указывающий, откуда загружается этот символ;
- `int dlclose(void *handle);` – уменьшает на единицу счетчик ссылок на указатель динамической библиотеки `handle`.

В случае с программой, которая реализует методы библиотеки на этапе линковки, мы просто подключаем заголовочный файл (т.е. программа уже на этапе компиляции имеет контракты необходимых методов), и далее начинается бесконечный пользовательский ввод, в котором пользователь может ввести две команды: 1 – использование метода для вычисления интеграла синуса, 2 – использование метода для вычисления числа Пи. Для того чтобы выйти из программы нужно подать на вход символ EOF (Ctrl+d). В Makefile я продемонстрировал 3 этапа компилирования: компиляция, ассемблирование и линковка. В случае с программой, которая реализует функционал библиотек во время исполнения программы, мы уже не подключаем заголовочный файл, но перед тем как компилировать основную программу, мы компилируем программы с реализациями функций, с определенными флагами, для создания динамических библиотек, а именно с ключами `-fPIC` – `positional independent code`, флаг необходимый для нужного формата динамических библиотек, `-shared` – позволяет создать `shared_object` и `-o` – флаг, компилирующий объектные файлы. Далее мы компилируем основную программу, только в отличие от прошлой программы, мы ее компилируем с флагом `-ldl` – то есть с помощью флага `-l` мы подключаем библиотеку `libdl.so`, которая позволяет динамически подгружать библиотеки. В этой программе у нас тоже есть бесконечный пользовательский ввод, но в нем есть одна дополнительная команда 0 – команда, позволяющая сменить библиотеку. Также необходимо отметить флаг `-lm` – флаг, позволяющий использовать все функции математического пакета.

Код программы

foos.h

```
#pragma once
```

```
float SinIntegral(float A, float B, float e);
```

```
float Pi(int K);
```

realization1.c

```
#include "foos.h"
```

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

```
// Using method of left rectangles
```

```
float SinIntegral(float A, float B, float e) {
```

```
    float result = 0;
```

```
    for (float i = A; i <= B; i += e) {
```

```
        result += sin(i) * e;
```

```
    }
```

```
    return result;
```

```
}
```

```
// Using Leibniz row
```

```
float Pi(int K) {
```

```
    float result = 0;
```

```
    for (int i = 0; i <= K; i++) {
```

```
        result += (i % 2 == 0 ? 1.0 : -1.0) / (2 * i + 1);
```

```
    }
```

```
    return result * 4;
```

```
}
```

realization2.c

```
#include "foos.h"
```

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

```
// Using method of trapezoids
```

```
float SinIntegral(float A, float B, float e) {
```

```
    float result = (sin(A) + sin(B)) / 2;
```

```
    for (float i = A + e; i <= B - e; i += e) {
```

```
        result += sin(i);
```

```
    }
```

```
    return e * result;
```

```
}
```

```
// Using Vallis formula
```

```
float Pi(int K) {
```

```
    float result = 1;
```

```
    for (int i = 1; i <= K; i++) {
```

```
        result *= (4.0 * i * i) / (4 * i * i - 1);
```

```
    }
```

```
    return result * 2;
```

```
}
```

linking_main.c

```
#include "foos.h"
#include <stdio.h>

int main() {
    int cmd;
    while (printf("Enter the command (1 - calculating integral of sin, 2 - calculating
Pi number): ") > 0 && scanf("%d", &cmd) != EOF) {
        if (cmd == 1) {
            float A, B, e;
step): "); printf("Enter the arguments (begin of the segment, end of the segment,
scanf("%f %f %f", &A, &B, &e);
printf("Result: %f\n", SinIntegral(A, B, e));
        } else if (cmd == 2) {
            int K;
            printf("Enter the quantity of elements in the row: ");
            scanf("%d", &K);
            printf("Result: %f\n", Pi(K));
        } else {
            printf("Wrong command! Try again!\n");
        }
    }
}
```

dynamic_main.c

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

int main() {
    void *handler = dlopen("./librealization1.so", RTLD_LAZY);
    if (!handler) {
        fputs(dlerror(), stderr);
        exit(1);
    }
    int realization = 1;
    int cmd;
    char *error;
    while (printf("Enter the command (0 - changing realization (library), 1 -
calculating integral of sin, 2 - calculating Pi number): ") > 0 && scanf("%d", &cmd) !=
EOF) {
        if (cmd == 0) {
            if (dlclose(handler) != 0) {
                perror("dlclose");
                exit(1);
            }
            realization = (realization == 1 ? 2 : 1);
            handler = (realization == 1 ? dlopen("./librealization1.so", RTLD_LAZY) :
dlopen("./librealization2.so", RTLD_LAZY));
        }
    }
}
```

```

        if (!handler) {
            fputs(dlerror(), stderr);
            exit(1);
        }
        printf("Realization was changed from realization%d to realization%d.\n",
(realization == 1 ? 2 : 1), realization);
    } else if (cmd == 1) {
        float (*SinIntegral)(float, float, float);
        *(float **) (&SinIntegral) = dlsym(handler, "SinIntegral");
        if ((error = dlerror()) != NULL) {
            fprintf(stderr, "%s\n", error);
            exit(1);
        }
        float A, B, e;
step): "); printf("Enter the arguments (begin of the segment, end of the segment,
        scanf("%f %f %f", &A, &B, &e);
        printf("Result: %f\n", SinIntegral(A, B, e));
    } else if (cmd == 2) {
        float (*Pi)(int);
        *(float **) (&Pi) = dlsym(handler, "Pi");
        if ((error = dlerror()) != NULL) {
            fprintf(stderr, "%s\n", error);
            exit(1);
        }
        int K;
        printf("Enter the quantity of elements in the row: ");
        scanf("%d", &K);
        printf("Result: %f\n", Pi(K));
    } else {
        printf("Wrong command! Try again!\n");
    }
}
if (dlclose(handler) != 0) {
    perror("dlclose");
    exit(1);
}
exit(0);
}

```

Makefile

```

COMPILER = gcc
COMPILER_FLAGS = -std=c99 -Wall -pedantic
COMPILE_FLAG = -c
OBJECT_FLAG = -o
MATH_FLAG = -lm
PIC_FLAG = -fPIC
SHARED_FLAG = -shared
DYNAMIC_LOADING_FLAG = -ldl

```

```
all : linking_realization1 linking_realization2 dynamic

linking_realization2 : linking_main realization2
    $(COMPILER) $(COMPILER_FLAGS) linking_main.o realization2.o $(OBJECT_FLAG)
linking_realization2 $(MATH_FLAG)

linking_realization1 : linking_main realization1
    $(COMPILER) $(COMPILER_FLAGS) linking_main.o realization1.o $(OBJECT_FLAG)
linking_realization1 $(MATH_FLAG)

linking_main: linking_main.c
    $(COMPILER) $(COMPILER_FLAGS) $(COMPILE_FLAG) linking_main.c

realization2 : realization2.c foos.h
    $(COMPILER) $(COMPILER_FLAGS) $(COMPILE_FLAG) realization2.c

realization1 : realization1.c foos.h
    $(COMPILER) $(COMPILER_FLAGS) $(COMPILE_FLAG) realization1.c

dynamic : dynamic_main dynamic_realization1 dynamic_realization2
    $(COMPILER) $(COMPILER_FLAGS) dynamic_main.o $(OBJECT_FLAG) dynamic
$(DYNAMIC_LOADING_FLAG)

dynamic_main : dynamic_main.c
    $(COMPILER) $(COMPILER_FLAGS) $(COMPILE_FLAG) dynamic_main.c
$(DYNAMIC_LOADING_FLAG)

dynamic_realization2 : realization2.c foos.h
    $(COMPILER) $(COMPILER_FLAGS) $(PIC_FLAG) $(SHARED_FLAG) $(OBJECT_FLAG)
librealization2.so realization2.c $(MATH_FLAG)

dynamic_realization1 : realization1.c foos.h
    $(COMPILER) $(COMPILER_FLAGS) $(PIC_FLAG) $(SHARED_FLAG) $(OBJECT_FLAG)
librealization1.so realization1.c $(MATH_FLAG)

clean :
    rm *.o *.so linking_realization1 linking_realization2 dynamic
```

Протокол работы программы

Тестирование:

```
$ ./linking_realization1
```

```
Enter the command (1 - calculating integral of sin, 2 - calculating Pi number): 1
```

```
Enter the arguments (begin of the segment, end of the segment, step): 3 4 1
```

```
Result: -0.615682
```

```
Enter the command (1 - calculating integral of sin, 2 - calculating Pi number): 2
```

```
Enter the quantity of elements in the row: 5
```

```
Result: 2.976046
```

```
$ ./linking_realization2
```

```
Enter the command (1 - calculating integral of sin, 2 - calculating Pi number): 1
```

```
Enter the arguments (begin of the segment, end of the segment, step): 3 4 1
```

```
Result: -0.307841
```

```
Enter the command (1 - calculating integral of sin, 2 - calculating Pi number): 2
```

```
Enter the quantity of elements in the row: 5
```

```
Result: 3.002176
```

```
$ ./dynamic
```

```
Enter the command (0 - changing realization (library), 1 - calculating integral of sin,  
2 - calculating Pi number): 1
```

```
Enter the arguments (begin of the segment, end of the segment, step): 3 4 1
```

```
Result: -0.615682
```

```
Enter the command (0 - changing realization (library), 1 - calculating integral of sin,  
2 - calculating Pi number): 2
```

```
Enter the quantity of elements in the row: 5
```

```
Result: 2.976046
```

```
Enter the command (0 - changing realization (library), 1 - calculating integral of sin,  
2 - calculating Pi number): 0
```

```
Realization was changed from realization1 to realization2.
```

```
Enter the command (0 - changing realization (library), 1 - calculating integral of sin,  
2 - calculating Pi number): 1
```

```
Enter the arguments (begin of the segment, end of the segment, step): 3 4 1
```

```
Result: -0.307841
```

```
Enter the command (0 - changing realization (library), 1 - calculating integral of sin,  
2 - calculating Pi number): 2
```

```
Enter the quantity of elements in the row: 5
```

```
Result: 3.002176
```

Strace:

```
$ strace -f ./dynamic
execve("./dynamic", [ "./dynamic" ], 0x7ffec778aab8 /* 76 vars */) = 0
brk(NULL)                                = 0x5587cc12d000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe4a59e300) = -1 EINVAL (Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f447ab4d000
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=69647, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 69647, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f447ab3b000
close(3)                                 = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
832 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0"... , 832) =
= 784 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)
pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"... , 48,
848) = 48
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0
=\340\2563\265?\356\25x\261\27\313A#\350"... , 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2216304, ...}, AT_EMPTY_PATH) = 0
= 784 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)
mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f447a800000
mmap(0x7f447a828000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x28000) = 0x7f447a828000
mmap(0x7f447a9bd000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1bd000) = 0x7f447a9bd000
mmap(0x7f447aa15000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x214000) = 0x7f447aa15000
mmap(0x7f447aa1b000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f447aa1b000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f447ab38000
arch_prctl(ARCH_SET_FS, 0x7f447ab38740) = 0
set_tid_address(0x7f447ab38a10)          = 52421
set_robust_list(0x7f447ab38a20, 24)      = 0
rseq(0x7f447ab390e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f447aa15000, 16384, PROT_READ) = 0
mprotect(0x5587ca616000, 4096, PROT_READ) = 0
mprotect(0x7f447ab87000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f447ab3b000, 69647)            = 0
getrandom("\xe5\x5c\x4a\x41\x3d\x88\x42\x64", 8, GRND_NONBLOCK) = 8
brk(NULL)                                = 0x5587cc12d000
brk(0x5587cc14e000)                      = 0x5587cc14e000
openat(AT_FDCWD, "./librealization1.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\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=15600, ...}, AT_EMPTY_PATH) = 0
```



```

getcwd("/home/lexasy/\320\240\320\260\320\261\320\276\321\207\320\270\320\271
\321\201\321\202\320\276\320\273/PROJECTS/OSlabs-3rd-sem/lab4/src", 128) = 70
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f447ab48000
mmap(0x7f447ab49000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1000) = 0x7f447ab49000
mmap(0x7f447ab4a000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f447ab4a000
mmap(0x7f447ab4b000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x2000) = 0x7f447ab4b000
close(3) = 0
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=69647, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 69647, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f447ab26000
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\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"... , 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f447aa3f000
mmap(0x7f447aa4d000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xe000) = 0x7f447aa4d000
mmap(0x7f447aac9000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x8a000) = 0x7f447aac9000
mmap(0x7f447ab24000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xe4000) = 0x7f447ab24000
close(3) = 0
mprotect(0x7f447ab24000, 4096, PROT_READ) = 0
mprotect(0x7f447ab4b000, 4096, PROT_READ) = 0
munmap(0x7f447ab26000, 69647) = 0
= 0 newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH)
= 0 newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH)
write(1, "Enter the command (0 - changing "..., 115Enter the command (0 - changing
realization (library), 1 - calculating integral of sin, 2 - calculating Pi number): ) =
115 read(0, 1
"1\n", 1024) = 2
write(1, "Enter the arguments (begin of th"... , 70Enter the arguments (begin of the
segment, end of the segment, step): ) = 70
read(0, 3 4 1
"3 4 1\n", 1024) = 6
write(1, "Result: -0.615682\n", 18Result: -0.615682
) = 18
write(1, "Enter the command (0 - changing "..., 115Enter the command (0 - changing
realization (library), 1 - calculating integral of sin, 2 - calculating Pi number): ) =
115 read(0, 2
"2\n", 1024) = 2
write(1, "Enter the quantity of elements i"... , 43Enter the quantity of elements in the
row: ) = 43
read(0, 5

```

```

"5\n", 1024)                = 2
write(1, "Result: 2.976046\n", 17Result: 2.976046
)                = 17
write(1, "Enter the command (0 - changing "..., 115Enter the command (0 - changing
realization (library), 1 - calculating integral of sin, 2 - calulating Pi number): ) =
115 read(0, 0
"0\n", 1024)                = 2
munmap(0x7f447ab48000, 16432)        = 0
munmap(0x7f447aa3f000, 942344)      = 0
openat(AT_FDCWD, "./librealization2.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
newfstatat(3, "", {st_mode=S_IFREG|0775, st_size=15600, ...}, AT_EMPTY_PATH) = 0
getcwd("/home/lexasy/\320\240\320\260\320\261\320\276\321\207\320\270\320\271
\321\201\321\202\320\276\320\273/PROJECTS/OSlabs-3rd-sem/lab4/src", 128) = 70
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f447ab48000
mmap(0x7f447ab49000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1000) = 0x7f447ab49000
mmap(0x7f447ab4a000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f447ab4a000
mmap(0x7f447ab4b000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x2000) = 0x7f447ab4b000
close(3)                = 0
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=69647, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 69647, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f447ab26000
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"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f447aa3f000
mmap(0x7f447aa4d000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xe000) = 0x7f447aa4d000
mmap(0x7f447aac9000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x8a000) = 0x7f447aac9000
mmap(0x7f447ab24000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xe4000) = 0x7f447ab24000
close(3)                = 0
mprotect(0x7f447ab24000, 4096, PROT_READ) = 0
mprotect(0x7f447ab4b000, 4096, PROT_READ) = 0
munmap(0x7f447ab26000, 69647)        = 0
write(1, "Realization was changed from rea"..., 59Realization was changed from
realization1 to realization2.
) = 59
write(1, "Enter the command (0 - changing "..., 115Enter the command (0 - changing
realization (library), 1 - calculating integral of sin, 2 - calulating Pi number): ) =
115 read(0, 1
"1\n", 1024)                = 2
write(1, "Enter the arguments (begin of th"..., 70Enter the arguments (begin of the

```

```

segment, end of the segment, step): ) = 70
read(0, 3 4 1
"3 4 1\n", 1024)          = 6
write(1, "Result: -0.307841\n", 18Result: -0.307841
)          = 18
write(1, "Enter the command (0 - changing "..., 115Enter the command (0 - changing
115 realization (library), 1 - calculating integral of sin, 2 - calculating Pi number): ) =
read(0, 2
"2\n", 1024)          = 2
write(1, "Enter the quantity of elements i"..., 43Enter the quantity of elements in the
row: ) = 43
read(0, 5
"5\n", 1024)          = 2
write(1, "Result: 3.002176\n", 17Result: 3.002176
)          = 17
write(1, "Enter the command (0 - changing "..., 115Enter the command (0 - changing
115 realization (library), 1 - calculating integral of sin, 2 - calculating Pi number): ) =
read(0, "", 1024)          = 0
munmap(0x7f447ab48000, 16432)          = 0
munmap(0x7f447aa3f000, 942344)          = 0
exit_group(0)          = ?
+++ exited with 0 +++

```

Вывод

В ходе выполнения лабораторной работы, я вспомнил все этапы компилирования программы, а также научился работать с динамическими библиотеками, что позволит в будущем писать свои библиотеки и в нужный момент, в какой либо программе ее использовать. Также я узнал больше флагов gcc, о которых даже не догадывался.