

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Институт №8 "Информационные технологий и прикладная математика"
Кафедра 806 "Вычислительная математика и программирование"

Лабораторная работа №5
по курсу "Операционные системы"
3 семестр

Студент: Леухин М. В.
Группа: М8О-206Б-20
Преподаватель: Соколов А. А.
Дата: 27.11.21
Оценка: 5
Подпись: _____

Москва, 2021

Содержание

1	Постановка задачи	3
2	Основная часть	4
2.1	Листинг файлов библиотек	4
2.2	Статические библиотеки	7
2.3	Динамические библиотеки	8
3	Вывод	10

1 Постановка задачи

Цель работы: приобретение практических навыков в создании динамических библиотек и создании программ, которые используют функции из динамических библиотек.

Задание: требуется создать динамические библиотеки, которые реализуют определённый функционал. Далее использовать эти библиотеки 2 способами:

- Во время компиляции (на этапе линковки).
- 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, заданные вариантом.
- Тестовая программа №1, которая использует одну из библиотек, используя знания, полученные на этапе компиляции.
- Тестовая программа №2, которая загружает библиотеки, используя только их местоположение и контракты.

Пользовательский ввод для обеих программ должен быть организован следующим образом:

- Если пользователь вводит команду «0», то программа переключает одну реализацию контрактов на другую (необходимо только для программы №2). Можно реализовать лабораторную работу без данной функции, но максимальная оценка в этом случае будет «хорошо»;
- «1 arg1 arg2 ... argN», где после «1» идут аргументы для первой функции, предусмотренной контрактами. После ввода команды происходит вызов первой функции, и на экране появляется результат её выполнения;
- «2 arg1 arg2 ... argM», где после «2» идут аргументы для второй функции, предусмотренной контрактами. После ввода команды происходит вызов второй функции, и на экране появляется результат её выполнения.

Вариант 3: расчёт интеграла функции $\sin x$ на отрезке $[A, B]$ с шагом e методом прямоугольников и методом трапеций; подсчёт наибольшего общего делителя двух натуральных чисел (с помощью алгоритма Евклида и с помощью наивного алгоритма).

2 Основная часть

2.1 Листинг файлов библиотек

Файл CmakeLists.txt

```
1 cmake_minimum_required(VERSION 3.20)
2 project(5lab)
3
4 set(CMAKE_C_STANDARD 11)
5
6 # static
7 add_executable(static static.c)
8
9 add_library(static_euclid_gcf STATIC lib/euclid_gcf.c)
10 target_link_libraries(static static_euclid_gcf)
11
12 add_library(static_naive_gcf STATIC lib/naive_gcf.c)
13 # target_link_libraries(static static_naive_gcf)
14
15 add_library(static_rectangle_sin STATIC lib/rectangle_sin.c)
16 target_link_libraries(static static_rectangle_sin)
17
18 add_library(static_trapezoid_sin STATIC lib/trapezoid_sin.c)
19 # target_link_libraries(static static_trapezoid_sin)
20
21 target_link_libraries(static m)
22
23 # dynamic
24 add_executable(dynamic dynamic.c)
25
26 add_library(dynamic_euclid_gcf SHARED lib/euclid_gcf.c)
27 add_library(dynamic_naive_gcf SHARED lib/naive_gcf.c)
28
29 add_library(dynamic_rectangle_sin SHARED lib/rectangle_sin.c)
30 target_link_libraries(dynamic_rectangle_sin m)
31 add_library(dynamic_trapezoid_sin SHARED lib/trapezoid_sin.c)
32 target_link_libraries(dynamic_trapezoid_sin m)
33
34 target_link_libraries(dynamic dl)
```

Файл euclid_gcf.c

```
1 #include "../headers/gcf.h"
2 #include <stdio.h>
3
4 int gcf(int a, int b){
5     if (a % b == 0){
6         return b;
7     }
8 }
```

```

8     return gcf(b, a % b);
9 }
10
11 int GCF(int A, int B){
12
13     printf("|euclid algorithm|\n");
14     return gcf(A, B);
15
16 }

```

Файл naive_gcf.c

```

1 #include "../headers/gcf.h"
2 #include <stdio.h>
3
4 int gcf(int a, int b){
5     if (a < b){
6         int tmp = b;
7         b = a;
8         a = tmp;
9     }
10    for (int i = b; i > 0; --i){
11        if (a % i == 0 && b % i == 0){
12            return i;
13        }
14    }
15    return -1;
16 }
17
18 int GCF(int A, int B){
19
20     printf("|naive algorithm|\n");
21     return gcf(A, B);
22
23 }

```

Файл rectangle_sin.c

```
1 #include "../headers/sin.h"
2 #include <math.h>
3
4 float SinIntegral(float A, float B, float e){
5
6     printf("|rectangle sin|\n");
7     float dx = (B - A) / e;
8     int steps = (B - A) / dx;
9     float current = A;
10    float result = 0;
11    for (int i = 0; i < steps; ++i){
12        result += dx * sin(current + dx / 2);
13        current += dx;
14    }
15    result += (B - current) * sin((B + current) / 2);
16    return result;
17
18 }
```

Файл trapezoid_sin.c

```
1 #include "../headers/sin.h"
2 #include <math.h>
3
4 float SinIntegral(float A, float B, float e){
5
6     printf("|trapezoid sin|\n");
7     float dx = (B - A) / e;
8     int steps = (B - A) / dx;
9     float current = A;
10    float result = 0;
11    for (int i = 0; i < steps; ++i){
12        result += dx * (sin(current) + sin(current + dx)) / 2;
13        current += dx;
14    }
15    result += (B - current) * (sin(B) + sin(current)) / 2;
16    return result;
17
18 }
```

2.2 Статические библиотеки

Файл static.c

```
1 #include "headers/gcf.h"
2 #include "headers/sin.h"
3 #include <stdio.h>
4
5 int main(){
6
7     int mode;
8     printf("> ");
9     while (scanf("%d", &mode) != EOF){
10         if (mode == 0){
11             printf("this is static program, can't switch
12                 implementation\n> ");
13             continue;
14         }
15         if (mode == 1){
16             float a, b, e;
17             if (scanf("%f %f %f", &a, &b, &e) == EOF){ break; };
18             printf("integral of sin(x) on [%f, %f] is %f\n", a,
19                 b, SinIntegral(a, b, e));
20         } else if (mode == 2){
21             int a, b;
22             if (scanf("%d %d", &a, &b) == EOF){ break; };
23             printf("gcf of %d and %d is %d\n", a, b, GCF(a, b));
24         } else {
25             printf("invalid argument \"%d\"\n", mode);
26         }
27         printf("> ");
28     }
```

Результат работы программы:

```
1 matvey@matvey-Lenovo-IdeaPad-S340-15API:~/labs/2os/5lab/src/build$
2 ./static
3 > 0
4 this is static program, can't switch implementation
5 > 1 0 1 0.05
6 |rectangle sin|
7 integral of sin(x) on [0.000000, 1.000000] is 0.479426
8 > 2 16 28
9 |euclid algorithm|
10 gcf of 16 and 28 is 4
11 >
```

2.3 Динамические библиотеки

Файл dynamic.c

```
1 #include <dlfcn.h>
2 #include <stdio.h>
3
4 int main() {
5
6     int current = 0;
7
8     const char* gcf[2] = { "./libdynamic_euclid_gcf.so",
9                             "./libdynamic_naive_gcf.so" };
10    const char* sin[2] = { "./libdynamic_rectangle_sin.so",
11                           "./libdynamic_trapezoid_sin.so" };
12
13    void* gcf_handler = dlopen(gcf[current], RTLD_LAZY);
14    void* sin_handler = dlopen(sin[current], RTLD_LAZY);
15
16    if (!gcf_handler && !sin_handler) {
17        fprintf(stderr, "dlopen() error: %s\n", dlerror());
18        return -1;
19    }
20
21    int (*gcf_function)(int A, int B);
22    float (*sin_function)(float A, float B, float e);
23
24    gcf_function = dlsym(gcf_handler, "GCF");
25    sin_function = dlsym(sin_handler, "SinIntegral");
26
27    if (!gcf_function && !sin_function) {
28        fprintf(stderr, "dlsym() error: %s\n", dlerror());
29        return -1;
30    }
31
32    int mode;
33    printf("> ");
34    while (scanf("%d", &mode) != EOF) {
35        if (mode == 0) {
36            dlclose(gcf_handler);
37            dlclose(sin_handler);
38
39            gcf_handler = dlopen(gcf[1 - current], RTLD_LAZY);
40            sin_handler = dlopen(sin[1 - current], RTLD_LAZY);
41
42            if (!gcf_handler && !sin_handler) {
43                fprintf(stderr, "dlopen() error: %s\n",
44                        dlerror());
45                return -1;
46            }
47        }
48    }
```



```

45
46     gcf_function = dlsym(gcf_handler, "GCF");
47     sin_function = dlsym(sin_handler, "SinIntegral");
48
49     if (!gcf_function && !sin_function){
50         fprintf(stderr, "dlsym() error: %s\n", dlerror());
51         return -1;
52     }
53
54     current = 1 - current;
55     printf("implementation switched\n> ");
56     continue;
57
58 }
59 if (mode == 1){
60     float a, b, e;
61     if (scanf("%f %f %f", &a, &b, &e) == EOF){ break; };
62     printf("integral of sin(x) on [%f, %f] is %f\n", a,
63           b, (*sin_function)(a, b, e));
64 } else if (mode == 2){
65     int a, b;
66     if (scanf("%d %d", &a, &b) == EOF){ break; };
67     printf("gcf of %d and %d is %d\n", a, b,
68           (*gcf_function)(a, b));
69 } else {
70     printf("invalid argument \"%d\"\n", mode);
71 }
72
73 }

```

Результат работы программы:

```

1  matvey@matvey-Lenovo-IdeaPad-S340-15API:~/labs/2os/5lab/src/build$
   ./dynamic
2  > 1 0 1 0.05
3  |rectangle sin|
4  integral of sin(x) on [0.000000, 1.000000] is 0.479426
5  > 2 16 28
6  |euclid algorithm|
7  gcf of 16 and 28 is 4
8  > 0
9  implementation switched
10 > 1 0 1 0.05
11 |trapezoid sin|
12 integral of sin(x) on [0.000000, 1.000000] is 0.420735
13 > 2 16 28
14 |naive algorithm|
15 gcf of 16 and 28 is 4
16 >

```

3 Вывод

В ходе лабораторной работы я познакомился с тем, что такое динамические библиотеки и в чём преимущества их использования. Плюс использования динамической библиотеки заключается в том, что она, в отличие от статической библиотеки, не привязывается к программе на стадии линковки, а значит в случае изменения кода динамической библиотеки не надо будет перекомпилировать все проекты, где она используется. К тому же при использовании функции из динамической библиотеки вся библиотека в память не загружается, а лишь используемая функция.