Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №5 по курсу

«Операционные системы»

ДИНАМИЧЕСКИЕ БИБЛИОТЕКИ

Студент: Лошманов Юрий Андреевич
Группа: М8О–206Б–20
Вариант: 1
Преподаватель: Соколов Андрей Алексеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2020.

Постановка задачи

Цель работы

Целью является приобретение практических навыков в:

- Создание динамических библиотек
- Создание программ, которые используют функции динамических библиотек

Задание

Требуется создать динамические библиотеки, которые реализуют определенный функционал. Далее использовать данные библиотеки 2-мя способами:

1. Во время компиляции (на этапе «линковки»/linking)
2. Во время исполнения программы, загрузив библиотеки в память с помощью системных вызовов

В конечном итоге, в лабораторной работе необходимо получить следующие части:

- Динамические библиотеки, реализующие контракты, которые заданы вариантом;
- Тестовая программа (программа No1), которая используют одну из библиотек, используя знания полученные на этапе компиляции;
- Тестовая программа (программа No2), которая загружает библиотеки, используя только их местоположение 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 No2). 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.

Контракты и реализации функций

№	Описание	Сигнатура	Реализаци 1	Реализаци 2
1	Рассчет интеграла функции sin(x) на отрезке [A, B] с шагом e	Float Sin-Integral(float A, float B, float e)	Подсчет интеграла методом прямоугольников.	Подсчет интеграла методом трапеций.
2	Рассчет производной функции cos(x) в точке A с приращением deltaX	Float Derivative(float A, float deltaX)	f'(x) = (f(A + deltaX) – f(A))/ deltaX	f'(x) = (f(A + deltaX) – f(A-deltaX))/ (2*deltaX)

Общие сведения о программах

Всего код состоит из двух программ и четырёх библиотек. Библиотеки представляют из себя две реализации и по две на каждую функцию. Библиотеки являются динамическими, с позиционно-независимым кодом. Первая программа загружает библиотеки исходя из информации, полученной на этапе линковки, а вторая с помощью функции dlopen.

Общий метод и алгоритм решения

Для реализации поставленной задачи необходимо:

1. Понять этапы и флаги компиляции динамических библиотек.
2. Изучить принципы работы функций dlopen, dlclose, dlsym и dlerror.
3. Подключить библиотеки dlfcn, необходимые для работы с вышеперечисленными функциями и их аргументами.
4. Написать исходные файлы библиотек.
5. Написать файлы программ
6. Изучить принципы компиляции библиотек и программ, использующих их
7. Проверить работоспособность
8. Написать Makefile и собрать проект

Анализ

По моему мнению, преимущество загрузки библиотеки с помощью системных вызовов заключается в том, что можно легко обработать ошибки, выгрузить библиотеку и загрузить новую в любой момент работы программы. В то время как преимущество загрузки библиотеки в начале работы программы, по знаниям на этапе «линковки», является простота работы с функциями и типами, так как не нужно постоянно извлекать нужный символ из библиотеки.

Основные файлы программы

sin_integral.h (Представление 1):

```
#ifndef LAB5_SIN_INTEGRAL_H
#define LAB5_SIN_INTEGRAL_H


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


#endif //LAB5_SIN_INTEGRAL_H
```

sin_integral.c (Представление 1):

```
#include <math.h>


#include "../sin_integral.h"


float SinIntegral(float A, float B, float e) {
    if (A == B || e <= 0) {
        return 0;
    }

    if (A > B) {
        float tmp = B;
        B = A;
        A = tmp;
    }

    if (B - A <= e) {
        return sinf(A) * (B - A);
    }

    float integral = 0;
    while (A + e <= B) {
        integral += sinf(A) * e;
        A += e;
    }

    if (A + e != B) {
        integral += sinf(A) * (B - A);
    }

    return integral;
}
```

derivative.h (Представление 1):

```
#ifndef LAB5_DERIVATIVE_H
```

```
#define LAB5_DERIVATIVE_H


float Derivative(float A, float deltaX);


#endif //LAB5_DERIVATIVE_H
```

derivative.c (Представление 1):

```
#include <math.h>


#include "../derivative.h"


float Derivative(float A, float deltaX) {
    return (cosf(A + deltaX) - cosf(A)) / deltaX;
}
```

sin_integral.h (Представление 2):

```
#ifndef LAB5_SIN_INTEGRAL_H
#define LAB5_SIN_INTEGRAL_H


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


#endif //LAB5_SIN_INTEGRAL_H
```

sin_integral.c (Представление 2):

```
#include <math.h>


#include "../sin_integral.h"


float SinIntegral(float A, float B, float e) {
    if (A == B || e <= 0) {
        return 0;
    }

    if (A > B) {
        float tmp = B;
        B = A;
        A = tmp;
    }

    if (B - A <= e) {
        float y1 = sinf(A);
        float y2 = sinf(B);
        return (y1 + y2) / 2 * (B - A);
    }

    float integral = 0;
    while (A + e <= B) {
        float y1 = sinf(A);
        float y2 = sinf(B);
        integral += (y1 + y2) / 2 * e;
        A += e;
    }

    if (A + e != B) {
```

```
        float y1 = sinf(A);
        float y2 = sinf(B);
        integral += (y1 + y2) / 2 * (B - A);
    }

    return integral;
}
```

derivative.h (Представление 2):

```
#ifndef LAB5_DERIVATIVE_H
#define LAB5_DERIVATIVE_H


float Derivative(float A, float deltaX);


#endif //LAB5_DERIVATIVE_H
```

derivative.c (Представление 2):

```
#include <math.h>


#include "../derivative.h"


float Derivative(float A, float deltaX) {
    return (cosf(A + deltaX) - cosf(A - deltaX)) / (2 * deltaX);
}
```

main1.c:

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


#include "lib/imp1/sin_integral.h"
#include "lib/imp1/derivative.h"


int main() {
 int n;
 float par1, par2, par3;
 while (scanf("%d", &n) != EOF) {
        if (n == 1) {
                if (scanf("%f %f %f", &par1, &par2, &par3) < 3) {
                        fprintf(stderr, "Invalid parameters\n");
                        exit(1);
                }
                printf("%f\n", SinIntegral(par1, par2, par3));
        } else if (n == 2) {
                if (scanf("%f %f", &par1, &par2) < 2) {
                        fprintf(stderr, "Invalid parameters\n");
                        exit(2);
                }
                printf("%f\n", Derivative(par1, par2));
        } else {
                fprintf(stderr, "Invalid command\n");
                exit(3);
        }
 }
 return 0;
}
```

main2.c:

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


#define LIBINT1 "./lib/imp1/libint.so"
#define LIBDER1 "./lib/imp1/libder.so"
#define LIBINT2 "./lib/imp2/libint.so"
#define LIBDER2 "./lib/imp2/libder.so"


int main() {
 int lib = 1;
 void *handle1 = dlopen(LIBINT1, RTLD_LAZY);
 void *handle2 = dlopen(LIBDER1, RTLD_LAZY);

 if (dlerror()) {
        fprintf(stderr, "dlopen error\n");
        exit(1);
 }

 int n;
 float par1, par2, par3;

 while (scanf("%d", &n) == 1) {
        if (n == 0) {
                if (lib == 1) {
                        lib = 2;
                        handle1 = dlopen(LIBINT2, RTLD_LAZY);
                        handle2 = dlopen(LIBDER2, RTLD_LAZY);
                } else {
                        lib = 1;
                        handle1 = dlopen(LIBINT1, RTLD_LAZY);
                        handle2 = dlopen(LIBDER1, RTLD_LAZY);
                }

                if (dlerror()) {
                        fprintf(stderr, "dlopen");
                        exit(2);
                }

                printf("Library has successfully changed\n");
        } else if (n == 1) {
                if (scanf("%f %f %f", &par1, &par2, &par3) != 3) {
                        fprintf(stderr, "Invalid parameters");
                        exit(3);
                }

                float (*SinIntegral) (float, float, float);
                SinIntegral = dlsym(handle1, "SinIntegral");
                if (SinIntegral == NULL) {
                        fprintf(stderr, "dlsym error\n");
                        exit(4);
                }

                printf("Result = %f\n", SinIntegral(par1, par2, par3));
        } else if (n == 2) {
                if (scanf("%f %f", &par1, &par2) != 2) {
                        fprintf(stderr, "Invalid parameters");
                        exit(5);
                }
```

```
                float (*Derivative) (float, float);
                Derivative = dlsym(handle2, "Derivative");
                if (Derivative == NULL) {
                        fprintf(stderr, "dlsym error\n");
                        exit(6);
                }

                printf("Result = %f\n", Derivative(par1, par2));
                } else {
                        fprintf(stderr, "Invalid command\n");
                }
 }

 dlclose(handle1);
 dlclose(handle2);

 if (dlerror()) {
        fprintf(stderr, "dlclose error\n");
        exit(7);
 }
 return 0;
}
```

Makefile:

```
CC=gcc
CFLAGS=-O3 -Wall -Wextra

BIN=./bin
SRC=./src
IMP1=$(BIN)/lib/imp1
IMP2=$(BIN)/lib/imp2

MAIN1=$(BIN)/main1
MAIN2=$(BIN)/main2
LIBINT1=$(IMP1)/libint.so
LIBINT2=$(IMP2)/libint.so
LIBDER1=$(IMP1)/libder.so
LIBDER2=$(IMP2)/libder.so


all: bin $(LIBINT1) $(LIBINT2) $(LIBDER1) $(LIBDER2) $(MAIN1) $(MAIN2)

bin:
 mkdir -p ./bin ./bin/lib/ ./bin/lib/imp1 ./bin/lib/imp2

$(BIN)/main1: $(SRC)/main1.c
 $(CC) $(CFLAGS) $^ -o $@ -L./bin/lib/imp1 -lder -lint -Wl,-rpath,./lib/imp1

$(BIN)/main2: $(SRC)/main2.c
 $(CC) $(CFLAGS) $^ -o $@ -ldl

$(LIBINT1): $(SRC)/lib/imp1/src/sin_integral.c $(SRC)/lib/imp1/sin_integral.h
 $(CC) $(CFLAGS) -fPIC -shared $< -o $@ -lm

$(LIBINT2): $(SRC)/lib/imp2/src/sin_integral.c $(SRC)/lib/imp2/sin_integral.h
 $(CC) $(CFLAGS) -fPIC -shared $< -o $@ -lm

$(LIBDER1): $(SRC)/lib/imp1/src/derivative.c $(SRC)/lib/imp1/derivative.h
 $(CC) $(CFLAGS) -fPIC -shared $< -o $@ -lm

$(LIBDER2): $(SRC)/lib/imp2/src/derivative.c $(SRC)/lib/imp2/derivative.h
 $(CC) $(CFLAGS) -fPIC -shared $< -o $@ -lm
```

Пример работы

```
yuryloshmanov@ubuntu:~/Desktop$ git clone https://github.com/yuryloshmanov/os_lab_5
Cloning into 'os_lab_5'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 21 (delta 4), reused 18 (delta 4), pack-reused 0
Unpacking objects: 100% (21/21), 3.13 KiB | 400.00 KiB/s, done.
yuryloshmanov@ubuntu:~/Desktop$ cd os_lab_5/
yuryloshmanov@ubuntu:~/Desktop/os_lab_5$ make
make: *** No targets specified and no makefile found.  Stop.
yuryloshmanov@ubuntu:~/Desktop/os_lab_5$ ls
README.md  src
yuryloshmanov@ubuntu:~/Desktop/os_lab_5$ cd src
yuryloshmanov@ubuntu:~/Desktop/os_lab_5/src$ make
mkdir -p ./bin ./bin/lib/ ./bin/lib/imp1 ./bin/lib/imp2
gcc -O3 -Wall -Wextra -fPIC -shared src/lib/imp1/src/sin_integral.c -o bin/lib/imp1/libint.so -lm
gcc -O3 -Wall -Wextra -fPIC -shared src/lib/imp2/src/sin_integral.c -o bin/lib/imp2/libint.so -lm
gcc -O3 -Wall -Wextra -fPIC -shared src/lib/imp1/src/derivative.c -o bin/lib/imp1/libder.so -lm
gcc -O3 -Wall -Wextra -fPIC -shared src/lib/imp2/src/derivative.c -o bin/lib/imp2/libder.so -lm
gcc -O3 -Wall -Wextra src/main1.c -o bin/main1 -L./bin/lib/imp1 -lder -lint -Wl,-rpath,./lib/imp1
gcc -O3 -Wall -Wextra src/main2.c -o bin/main2 -ldl
yuryloshmanov@ubuntu:~/Desktop/os_lab_5/src$ cd bin/
yuryloshmanov@ubuntu:~/Desktop/os_lab_5/src/bin$ ./main1
1 1 2 0.001
0.956370
2 3.14 0.0003
-0.001391
yuryloshmanov@ubuntu:~/Desktop/os_lab_5/src/bin$ ./main2
1 1 2 0.001
Result = 0.956370
0
Library has successfully changed
1 1 2 0.001
Result = 0.932813
2 3.14 0.0003
Result = -0.001589
0
Library has successfully changed
2 3.14 0.0003
Result = -0.001391
yuryloshmanov@ubuntu:~/Desktop/os_lab_5/src/bin$
```

Вывод

Я часто видел, пользуясь операционной системой Windows, файлы dll, для меня было загадкой то, для чего они нужны. Благодаря этой лабораторной работе, я понял что эти файлы представляет собой динамические библиотеки, узнал об отличиях статических и динамических библиотек, создал свои собственные динамические библиотеки и попробовал разные способы их загрузки.