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: Армишев Кирилл Константинович
Группа: М8О–208Б–21
Вариант: 1
Преподаватель: Соколов Андрей Алексеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2022.

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. 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 знания полученные на этапе компиляции;

• Тестовая программа (программа №2), 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 №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.

Общие сведения о программе

Программа компилируется из файла main.c. Также я дополнительно реализовал 2 динамических библиотеки, для расчета производной и для расчета интеграла.

Системные вызовы

1. **dlopen** – загружает динамический общий объект (общую библиотеку) из файла, имя которого указано в строке filename (завершается null) и возвращает непрозрачный описатель на загруженный объект.
2. **dlsym** – функция возвращает адрес, по которому символ расположен в памяти(указывается одним из аргументов).
3. **dlclose** – уменьшает счётчик ссылок на динамически загружаемый общий объект, на который ссылается handle. Если счётчик ссылок достигает нуля, то объект выгружается. Все общие объекты, которые были автоматически загружены при вызове dlopen() для объекта, на который ссылается handle, рекурсивно закрываются таким же способом.

Общий метод и алгоритм решения.

Для реализации поставленной задачи необходимо:

1. Изучить принципы работы dlsym, dlopen, dlclose.
2. Написать библиотеку для вычисления производной
3. Написать библиотеку для вычисления интеграла
4. В файле main1.c подключить библиотеку на этапе компиляции.
5. В файле main2.c загрузить библиотечные функции в runtime, с помощью dlsym, dlopen, dlclose.

Основные файлы программы

Math1.c:

```
#include "stdio.h"
#include "math.h"
```

```
float SinIntegral1(float a, float b, float n){
   int i;
   double h,x,sum=0,integral;
   h=fabs(b-a)/n;
   for(i=1;i<n;i++){
      x=a+i*h;
      sum=sum+sin(x);
   }
   integral=(h/2)*(sin(a)+sin(b)+2*sum);
   return integral;
}
```

Math12.c

```
#include "stdio.h"
#include "math.h"

float SinIntegral2(float a, float b, float n){
   double h,S=0,x;
   int i;
   h=fabs(b-a)/n;
   for(i=0;i<n-1;i++)
   {
      x=a+i*h;
      S=S+sin(x);
   }
   S=h*S;
   return S;
}
```

Math2.c

```
#include "stdio.h"
#include "math.h"
#include "stdlib.h"
#include "string.h"

float derivative1(float A, float deltaX){
   float der = (cos(A+ deltaX)-cos(A))/deltaX;
   return der;
}
```

Math1.h

```
#ifndef MATH1_H
#define MATH1_H

float SinIntegral1(float a, float b, float n);
float SinIntegral2(float a, float b, float n);

#endif
```

Math2.h

```
#ifndef MATH2_H
#define MATH2_H

float derivative1(float A, float deltaX);
float derivative2(float A, float deltaX);

#endif
```

program1.c

```
#include "stdio.h"
#include "stdlib.h"
#include "math1.h"
#include "math2.h"

int main(int argc, char *argv[]){
   // Argument error check
   if (argc == 1){
      perror("INCORRECT INPUT KEYS\n");
      exit(0);
   }
   else {
      if ((atoi(argv[1]) == 1)) {
         printf("%f\n", SinIntegral1(atof(argv[2]), atof(argv[3]), atof(argv[4])));
      }
      else if((atoi(argv[1]) == 2)){
         printf("%f\n", derivative1(atof(argv[2]), atof(argv[3])));
      }
      return 0;
   }
}
```

program2.c

```
#include <stdio.h>
#include <dlfcn.h>
#include <math.h>
#include <stdlib.h>
int main(int argc, char * argv[]) {
    void * handler1;
    void * handler2;
    float(*Integral)(float, float, float);
    float(*Derivative)(float, float);
    if (argc > 2) {
        if(atoi(argv[1]) == 0) {
            if(atoi(argv[2]) == 1) {
                handler1 = dlopen("/Users/kirillarmishev/Downloads/project/cmake-
```

```
build-debug/src/libraries/libmath22.dylib", RTLD_LAZY);
            if (!handler1) {
                fprintf(stderr, "Open of dynamic library math22 with error: %s\n",
dlerror());
                exit(-1);
            }
            Derivative = dlsym(handler1, "derivative2");
            printf("result of derivative(2 realisation)  is %f\n",
(*Derivative)(atof(argv[3]), atof(argv[4])));
            dlclose(handler1);
          } else {
            handler2 = dlopen("/Users/kirillarmishev/Downloads/project/cmake-
build-debug/src/libraries/libmath12.dylib", RTLD_LAZY);
            if (!handler2) {
                fprintf(stderr, "Open of dynamic library math12 with error: %s\n",
dlerror());
                exit(-2);
            }
            Integral = dlsym(handler2, "SinIntegral2");
            printf("result of integral is %f\n",
(*Integral)(atof(argv[3]),atof(argv[4]),atof(argv[5])));
            dlclose(handler2);
          }
       }
       else if(atoi(argv[1]) == 1) {
          handler1 = dlopen("/Users/kirillarmishev/Downloads/project/cmake-build-
debug/src/libraries/libmath2.dylib", RTLD_LAZY);
          if (!handler1) {
              fprintf(stderr, "Open of dynamic library math2 with error: %s\n",
dlerror());
              exit(-1);
          }
          Derivative = dlsym(handler1, "derivative1");
          printf("result of derivative(1 realisation) is %f\n",
(*Derivative)(atof(argv[2]), atof(argv[3])));
          dlclose(handler1);
       }
       else if(atoi(argv[1]) == 2) {
          handler2 = dlopen("/Users/kirillarmishev/Downloads/project/cmake-build-
debug/src/libraries/libmath1.dylib", RTLD_LAZY);
          if (!handler2) {
              fprintf(stderr, "Open of dynamic library math1 with error: %s\n",
```

```
dlerror());
                exit(-2);
            }
            Integral = dlsym(handler2, "SinIntegral1");
            printf("result of integral is %f\n",
(*Integral)(atof(argv[2]),atof(argv[3]),atof(argv[4])));
            dlclose(handler2);
        }
    }
}
```

Вывод

В данной лабораторной работе я познакомился с принципами создания и использования динамических библиотек, реализовал две программы которые используют 2 разных подхода к использованию функций динамической библиотеки: подключение библиотеки на этапе линковки или подключение библиотеки в рантайме, с помощью системных вызовов. Работа с библиотеками является очень важным навыком, необходимым программисту.