

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Лабораторная работа №5 по курсу

«Операционные системы»

Тема работы

“Динамические библиотеки”

Студентка: Ивченко Анна Владимировна

Группа: М8О-208Б-20

Вариант: 9

Преподаватель: Миронов Евгений Сергеевич

Оценка: _____

Дата: _____

Подпись: _____

Москва, 2021

Содержание

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

Репозиторий

https://github.com/Anetta123/OS/tree/main/os_lab5

Постановка задачи

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 использует 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 обеих 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).
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 написана на языке Си в UNIX-подобной операционной системе.

Сборка проекта происходит при помощи make-файла

```
g++ programm1.cpp -L. -ld1 -o m1 -Wl,-rpath -Wl,.
```

```
g++ programm2.cpp -ldl -o m2 -Wl,-rpath -Wl,.  
g++ -shared l1.o -o libd1.so  
g++ -shared l2.o -o libd2.so  
g++ -fPIC -c library1.cpp -o l1.o  
g++ -fPIC -c library2.cpp -o l2.o
```

Флаг `-fPIC` означает, что используется относительная переадресация в переходах подпрограмм.

Флаг `-shared` предписывается, что создается динамическая библиотека.

Флаг `-L.` - ищет в каталоге файлы библиотеки.

Флаг `-ldl` используется для работ с функциями `dlopen`, `dlsym`, `dlclose`.

Флаг `-lm/-ldl` связывает с файлом библиотеку `math/lib1`.

Общий метод и алгоритм решения

В файлах `library1.c` и `library2.c` реализовал функции методами, соответствующие условию задания. В тестовой программе №1, которая использует данные библиотеки во время компиляции, прописал цикл `while`, в условии которого идет чтение ключа. Пока ключ может считаться, в зависимости от его значения я выбираю функцию, которая будет обрабатываться и выводить результат в консоль. После выбора функции считываю входные данные для функции, вычисляю её и вывожу результат. В тестовой программе №2 изначально прошу пользователя выбрать стартовую библиотеку. В зависимости от его выбора, загружаю открываю ту или иную библиотеку с помощью функции `dlopen`. Нахожу начало адреса функций `PrimeCount` и `Derivative` с помощью функции `dlsym`. В цикле `while` считываю ключ и в зависимости от его значения обрабатываю ту или иную функцию и вывожу результат на консоль. Если ключ равен 0, с помощью функции `dlclose` закрываю библиотеку и открываю, и загружаю (`dlopen`) другую библиотеку. Также с помощью `dlsym` нахожу начало адреса функций `PrimeCount` и `Derivative` уже другой библиотеки. После окончания ввода, закрываю библиотеку.

Исходный код

library1.cpp

```
#include <cmath>

extern "C" float Derivative(float A, float deltaX);
extern "C" int PrimeCount(int A, int B);
```

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

```
int PrimeCount(int A, int B){
    bool mod = false;
    int quality = 0;
    for (int i = A+1; i < B+1; i++){
        for (int j = i-1; j > 0; j--){
            if (i % j == 0){
                mod = true;
                break;
            }
        }
        if (mod == true){
            mod = false;
            quality += 1;
        }
    }
    return quality;
}
```

library2.cpp

```

#include <cmath>

#include <vector>

using namespace std;

extern "C" float Derivative(float A, float deltaX);

extern "C" int PrimeCount(int A, int B);


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


int PrimeCount(int A, int B){
    vector<bool> numbers(B, true);
    int quality = 0;
    for (int i = 2; i*i <= B; i++){
        if (numbers[i] == true){
            for (int j = i*i; j < B; j+=i){
                numbers[j] = false;
            }
        }
    }
    for (int i = A; i<B; i++){
        if (numbers[i] == true){
            quality +=1;
        }
    }
    return quality;
}

```

programm1.cpp

```

#include <iostream>

using namespace std;

extern "C" float Derivative(float A, float deltaX);

```

```

extern "C" int PrimeCount(int A, int B);

int main()
{
    int flag;
    cout << "Please enter the flag:\n";
    while (scanf("%d",&flag) > 0 ){
        if (flag == 2){
            cout<<"Please enter your dates:\n";
            int A, B;
            cin>>A>>B;
            cout<<"PrimeCount("<< A<<","<<B<<")="<<PrimeCount(A, B)<< endl;
        }
        else if (flag == 1){
            cout<<"Please enter your dates:\n";
            float A, deltaX;
            cin>>A>>deltaX;
            cout<< "Derivative("<<A<<","<<deltaX<<")="<<Derivative(A,deltaX)<<endl;
        }
    }
    return 0;
}

```

Programm2.cpp

```

#include<iostream>
#include <stdio.h>
#include<stdlib.h>
#include<dlfcn.h>
using namespace std;

int main(){
    void* adres = NULL;//адресс для доступа к библиотеке

    int (*PrimeCount)(int A, int B);

```

```

float (*Derivative)(float A, float deltaX);

const char* libeary_mas[]={ "libd1.so", "libd2.so" };
int curlib;
int statl;
cout<<"Enter the start library:"<<endl;
cout << '\t' << "1 for first library" <<endl;
cout << '\t' << "2 for second library" <<endl;
cin>>statl;
bool flag = 1;
while(flag){
    if (statl==1){
        curlib = 0;
        flag = 0;
    }
    else if(statl==2){
        curlib=1;
        flag=0;
    }
    else{
        cout<<"Please, enter again!"<<endl;
        cin>>statl;
    }
}

adres = dlopen(libeary_mas[curlib],RTLD_LAZY);//RTLD_LAZY выполняется поиск
только тех символов, на которые есть ссылки из кода
if (!adres){
    cout<<"Error";
    exit(EXIT_FAILURE);
}

PrimeCount = (int (*)(int , int))dlsym(adres,"PrimeCount");//возвращаем адрес функции из
памяти библиотеки /dlsym присваивает указателю на функцию, объявленному в начале, ее
адрес в библиотеке

```



```

Derivative = (float (*)(float, float))dlsym(adres, "Derivative");
int command;

cout<<"Please read the nexst rules? before you enter the command";

cout << '\t' << "0 for changing the contract;" << std:: endl;
cout << '\t' << "1 for calculating Derivative;" << std:: endl;
cout << '\t' << "2 for calculating PrimeCount;" << std:: endl;
while (printf("Please enter your command: ") && (scanf("%d", &command)) != EOF){
    if (command==0){
        dlclose(adres); //освобождает указатель на библиотеку и программа перестает ей
        пользоваться
        if (curlib==0){
            curlib=1;

            adres=dlopen(libeary_mas[curlib],RTLD_LAZY);

            if(!adres){
                cout<<"Error";

                exit(EXIT_FAILURE);

            }

            PrimeCount = (int (*)(int , int))dlsym(adres,"PrimeCount");//возвращаем адрес
            функции из памяти библиотеки /dlsym присваивает указателю на функцию, объявленному
            в начале, ее адрес в библиотеке

            Derivative = (float (*)(float, float))dlsym(adres, "Derivative");

        }
        else if(curlib==1){
            curlib=0;

            adres = dlopen(libeary_mas[curlib],RTLD_LAZY);

            if(!adres){
                cout<<"Error";

                exit(EXIT_FAILURE);

            }

            PrimeCount = (int (*)(int, int))dlsym(adres,"PrimeCount");//возвращаем адрес
            функции из памяти библиотеки /dlsym присваивает указателю на функцию, объявленному
            в начале, ее адрес в библиотеке

```

```

        Derivative = (float(*) (float, float)) dlsym(adres, "Derivative");

    }

    cout << "You have changed contracts!" << std:: endl;
}

else if (command==2){
    int A, B;

    cin>>A>>B;

    int primeCount =PrimeCount(A,B);

    if (primeCount==-1){
        cout<<"please enter again\n";
    }
    else{
        cout<<"PrimeCount("<< A<<","<<B<<")="<<PrimeCount(A, B)<< endl;
    }
}

else if (command==1){
    float A, deltaX;

    cin>>A>>deltaX;

    float derivative =Derivative(A, deltaX);

    if (derivative==-1){
        cout<<"please enter again\n";
    }
    else{
        cout<< "Derivative("<<A<<","<<deltaX<<")="<<Derivative(A,deltaX)<<endl;
    }
}

else{
    cout<<"You had to enter only 0, 1 or 2!" << std:: endl;
}

}

dlclose(adres);

```

```
    return 0;  
}
```

Демонстрация работы программы

anetta@anetta-VirtualBox:~/Рабочий стол/lab5\$./m2

Enter the start library:

1 for first library

2 for second library

1

Please read the next rules? before you enter the command 0 for changing the contract;

1 for calculating Derivative;

2 for calculating PrimeCount;

Please enter your command: 1

3 5

Derivative(3,5)=0.168898

Please enter your command: 2

4 8

PrimeCount(4,8)=4

Please enter your command: 0

You have changed contracts!

Выводы

Данная лабораторная работа научила работать с динамическими библиотеками. Я создала динамические библиотеки функций, а также я поработала с теорией, поняла основной принцип динамических библиотек, их отличие от статических библиотек.