

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: М80-210Б-22

Студент: Калиниченко Артём Андреевич

Преподаватель: Соколов Андрей Алексеевич

Оценка: _____

Дата: 15.12.2023

Москва, 2023

Постановка задачи

Вариант 10.

1. Расчет производной функции Float $f'(x) = (f(A + \text{deltaX}) - f(A)) / \text{deltaX}$
2. Подсчёт наибольшего общего делителя для двух натуральных чисел Int GCF(int A, int B) Алгоритм Евклида Наивный алгоритм. Пытаться разделить числа на все числа, что меньше A и B.

Общий метод и алгоритм решения

Использованные системные вызовы:

void *dlopen(const char *filename, int flag) - загружает динамическую библиотеку.

int dlclose(void *handle) - уменьшает на единицу счетчик ссылок на указатель динамической библиотеки handle.

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.

Код программы

realization_1.cpp

```
#include "../include/realization.hpp"
```

```
float Derivative(float A, float deltaX) {
```

```
    return (cosf(A + deltaX) - cosf(A)) / deltaX;
```

```
}
```

```
int GCF(int A, int B) {
```

```
    while (A != 0 and B != 0) {
```

```
        if (A > B) A %= B;
```

```
        else B %= A;
```

```
    }
```

```
    return A + B;
```

```
}
```

realization_2.cpp

```
#include "../include/realization.hpp"
```

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

```
int GCF(int A, int B) {  
    int del = 1, max;  
    while (del < A and del < B) {  
        if (A % del == 0 and B % del == 0) {  
            max = del;  
        }  
        del++;  
    }  
    return max;  
}
```

realization.hpp

```
#pragma once
```

```
#include <cmath>
```

```
extern "C" {  
    float Derivative(float A, float deltaX);  
    int GCF(int A, int B);  
}
```

dyn.cpp

```
#include <iostream>
#include <string>
#include <dlfcn.h>

int main() {
    std::string lib1 = "../build/lib_1.so";
    std::string lib2 = "../build/lib_2.so";

    void *curlib = dlopen(lib1.c_str(), RTLD_LAZY);

    float (*Derivative)(float A, float deltaX);
    int (*GCF)(int A, int B);

    Derivative = (float (*)(float, float)) dlsym(curlib, "Derivative");
    GCF = (int (*)(int, int)) dlsym(curlib, "GCF");

    int com;
    int number_lib = 1;
    std::cout << "Enter 1 or 2 to select a function or -1 to exit or 0 switch
to library" << std::endl;

    while(std::cin >> com) {

        if (com == 0) {
            dlclose(curlib);
            if (number_lib == 1) {
                std::cout << "You switched to library number 2" <<
std::endl;

                curlib = dlopen(lib2.c_str(), RTLD_LAZY);
                number_lib = 2;

            } else {
                std::cout << "You switched to library number 1" <<
std::endl;

                curlib = dlopen(lib1.c_str(), RTLD_LAZY);
                number_lib = 1;
            }
            Derivative = (float (*)(float, float)) dlsym(curlib,
"Derivative");
            GCF = (int (*)(int, int)) dlsym(curlib, "GCF");
        }

        if (com == 1) {
            float A, deltaX, Ans1;
            std::cout << "Enter A and deltaX" << std::endl;
            std::cin >> A >> deltaX;
            Ans1 = Derivative(A, deltaX);
            std::cout << "Derivative value: " << Ans1 << std::endl;
        }

        if (com == 2) {
            int A, B, Ans2;
            std::cout << "Enter A and B" << std::endl;
            std::cin >> A >> B;
            Ans2 = GCF(A, B);
            std::cout << "GCF value: " << Ans2 << std::endl;
        }
    }
}
```

```

        if (com == -1) {
            std::cout << "Exit" << std::endl;
            return 0;
        }

        std::cout << "Enter 1 or 2 to select a function or -1 to exit or 0
switch to library" << std::endl;

    }

}

stat.cpp
#include <iostream>
#include "include/realization.hpp"

int main() {
    int com;

    std::cout << "Enter 1 or 2 to select a function or -1 to exit" << std::endl;

    while (std::cin >> com) {

        if (com == 1) {
            float A, deltaX;
            std::cout << "Enter A and deltaX" << std::endl;
            std::cin >> A >> deltaX;
            std::cout << "Derivative value: " << Derivative(A, deltaX) <<
std::endl;
        }

        if (com == 2) {
            int A, B;
            std::cout << "Enter A and B" << std::endl;
            std::cin >> A >> B;
            std::cout << "GCF value: " << GCF(A, B) << std::endl;
        }

        if (com == -1) {
            std::cout << "Exit" << std::endl;
            return 0;
        }

        std::cout << "Enter 1 or 2 to select a function or -1 to exit" <<
std::endl;
    }
}

```

Вывод

В ходе лабораторной работы я получил опыт разработки динамической библиотеки, узнал о dynamic link и dynamic loading library, их различии; использовал такие системные вызовы, как dlopen, dlsym, dlclose; узнал, как компилируются динамические библиотеки.