

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ЭВМ И ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

**ОПРЕДЕЛЕНИЕ ВРЕМЕНИ ПРИКЛАДНЫХ ПРОГРАММ**

Студент:

Овчаренко Дарья Ивановна, группа 23211

Преподаватель:

Ассистент кафедры ПВ ФИТ

Мичуров Михаил Антонович

Новосибирск 2024

# ЦЕЛЬ

1. Изучение методики измерения времени работы подпрограммы.
2. Изучение приемов повышения точности измерения времени работы подпрограммы.
3. Изучение способов измерения времени работы подпрограммы.
4. Измерение времени работы подпрограммы в прикладной программе.

# ЗАДАНИЕ

Вариант задания: 4.

1. Написать программу на языке C или C++, содержащую функцию, которая реализует выбранный алгоритм из задания. Программа должна принимать значение N через параметр в командной строке.
2. Проверить правильность работы программы на нескольких тестовых наборах входных данных.
3. Выбрать значение параметра N таким, чтобы время работы программы было порядка 15 секунд.
4. По приведенной методике определить время работы подпрограммы тестовой программы с относительной погрешностью не более 1%.
5. Составить отчет по лабораторной работе.

# ОПИСАНИЕ РАБОТЫ

В ходе задания использовался компьютер с архитектурой x86\_64, с операционной системой Ubuntu 22.04.5 LTS и процессором AMD A6-6310 APU with AMD Radeon R4 Graphics.

## Пошаговое описание выполненной работы

1. Был создан файл `main.c`
2. Была написана компьютерная программа, которая вычисляет  $\sin(x)$  с помощью разложения в степенной ряд по первым  $N$  членам этого ряда.
3. Нетрудно заметить, что степень  $x$  в числителе увеличивается на 2 в каждом новом слагаемом, и в знаменателе число под знаком факториала также увеличивается на 2 в каждом новом слагаемом. По такому принципу запрограммируем разложение ряда.
4. Код скомпилирован командой `gcc -o sinx main.c`, где `sinx` - исполнительный файл.
5. Запуск программы производится с помощью команды `./sinx argv[1] argv[2]`

На вход программы первый аргумент - число  $x$  (угол в градусах, синус которого необходимо посчитать), второй аргумент - количество членов  $n$ , по которым будет выполнено разложение. Листинг программы представлен в Приложении 1. На выходе ожидается одно число: значение синуса при разложении по ряду Тейлора.

Пусть угол, синус которого нужно посчитать, равен 30 градусам во всех запусках программ.

## Описание методики для определения времени работы программы

Замеры времени работы программы будем проводить с помощью библиотечной функции `clock_gettime`, которая получает значения системного таймера в ОС Linux/UNIX.

Функция `clock_gettime` с параметром `CLOCK_MONOTONIC_RAW` сохраняет значение системного таймера в структуру `struct timespec`. Структура состоит из двух полей: `tv_sec` и `tv_nsec` (их тип `time_t` и `long`, соответственно), задающих количество секунд и наносекунд, прошедших с некоторого неспецифицированного момента времени в прошлом.

В приведённом коде программы (см. Приложение 1) сохраняется значение таймера перед выполнением некоторого кода и после него. Разница показаний преобразуется в секунды и выводится на экран. Реализация функции `clock_gettime` находится в библиотеке `rt`, поэтому при компиляции программы необходимо(можно и без него) добавить ключ компиляции `'-lrt'`.

### **Описание выполненной работы по выбранному методу измерения времени**

Подберем такое значение `n`, при котором время работы программы будет оставлять 15 секунд. Подбранное значение: `n = 900 000 000`.

# ЗАКЛЮЧЕНИЕ

В нашей работе мы узнали базовые команды для компиляции и запуска программы. Познакомились с методами измерения времени программы, узнали что влияет на время программы

## ПРИЛОЖЕНИЕ 1. Листинг программы с библиотечной функцией clock\_gettime

```
#define _POSIX_C_SOURCE 199309L

#include <time.h>

#include <stdio.h>

#include <stdlib.h>


#define PI 3.1415926535897


double CalcSin(double x, long long n){

    double sinx = 0;

    x = x * PI / 180;

    double sum = x;

    for (long long i = 1; i <= 2 * n - 1; i += 2){

        sinx += sum;

        sum = (sum * x * x * (-1)) / ((i + 1) * (i + 2)) ;

    }

    return sinx;

}
```

```

int main(int argc, char **argv){

    struct timespec start, end;

    clock_gettime (CLOCK_MONOTONIC_RAW, &start);

    if (argc == 1){

        printf("Bad input. Enter x and n in command line");

    return 0;

    }

    double x = atoll(argv[1]);

    long long n = atoll(argv[2]);


    double sinx = CalcSin(x, n);

    printf("%lf\n", sinx);


    clock_gettime(CLOCK_MONOTONIC_RAW, &end);

    printf("Time taken: %lf sec.\n", end.tv_sec-start.tv_sec

        + 0.000000001*(end.tv_nsec-start.tv_nsec));

    return 0;

}

```

Строка компиляции: gcc -o sinx main.1.c -lrt

Строка запуска: ./sinx 30 9000000000

Ссылка на репозиторий с кодом: [https://github.com/dadashasha/nsu\\_evm/blob/main/lab1](https://github.com/dadashasha/nsu_evm/blob/main/lab1)