

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ**

Факультет информационных технологий

Кафедра параллельных вычислений

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ
РАБОТЫ**

Практическая работа №1

Определение времени работы прикладных программ

студента 2 курса, группы 23201

Сорокина Матвея Павловича

Направление 09.03.01 – "Информатика и вычислительная техника"

Преподаватель: А.С. Матвеев

Новосибирск, 2024 г.

Содержание

§ 1	ЦЕЛЬ	2
§ 2	ЗАДАНИЕ	2
§ 3	ОПИСАНИЕ РАБОТЫ	3
§ 4	ЗАКЛЮЧЕНИЕ	3

1 ЦЕЛЬ

Изучение методов измерения времени работы программы, оптимизации этих измерений, анализ влияния различных уровней оптимизации компилятора GCC на время выполнения программы.

2 ЗАДАНИЕ

В ходе работы было необходимо выполнить следующие задачи:

- Написать программу на языке C или C++, которая реализует выбранный алгоритм из задания.
- Проверить правильность работы программы на нескольких тестовых наборах входных данных.
- Выбрать значение параметра N таким, чтобы время работы программы было порядка 15 секунд.
- По приведенной методике определить время работы подпрограммы тестовой программы с относительной погрешностью не более 1%.
- Составить отчет по лабораторной работе.

3 ОПИСАНИЕ РАБОТЫ

- Реализовано задание №4 - алгоритм вычисления синуса с помощью разложения в степенной ряд по первым N членам данного ряда на языке $C++$.
- Для измерения времени работы программы использовалась библиотечная функция *clock_gettime* из библиотеки *time.h*.

Код программы также предоставлен (см. Приложение 1), также предоставлен *bash-скрипт*, компилирующий и запускающий программу с конкретными значениями x и N , записывающий время вычисления синуса угла с различными уровнями оптимизации компилятора *GCC* в *report.csv* файл (см. Приложение 2), после чего запускает *create_table.sh*, визуализирующий информацию из *report.csv*.

Время замерялось перед началом и после окончания работы функции, вычисляющей синус заданного угла. Разность этих двух значений дает общее время выполнения функции. Для проверки точности измерений, код программы запускается несколько раз.

4 ЗАКЛЮЧЕНИЕ

В ходе данной лабораторной работы мы познакомились с различными методами измерения работы программ и научились пользоваться ими на практике.

Приложения

Приложение 1: Исходный код программы

```
#include <iostream>
#include <time.h>
#include <cstdlib> // for atoi and atof
#include <cmath> // for pow

long double DegToRad(long double deg) {
    return deg * M_PI / 180;
}

long double SinCalculation(long double x, long long n) {
    long double sin = 0;
    long double prev = x;
    long double sign = 1.0;

    for (long long i = 1; i < n; i++) {
        sin += sign * prev;
        prev *= (x * x) / ((2 * i) * (2 * i + 1));
        sign = -sign;
    }

    return sin;
}

int main(int argc, char *argv[]) {
    if (argc != 3) {
        std::cerr << "Usage: <angle in degrees> <number of terms>" << std:::
        ↪ endl;
        return 0;
    }

    struct timespec start, end;
    long double x = atoll(argv[1]);
    long long n = atoll(argv[2]);

    std::cout << "x = " << x << ", n = " << n << std::endl;

    int runs = 5;
    double time_total = 0;

    for (long long i = 0; i < runs; i++) {
        clock_gettime(CLOCK_MONOTONIC_RAW, &start);
        double rad_x = DegToRad(x);
        long double sin = SinCalculation(rad_x, n);
        clock_gettime(CLOCK_MONOTONIC_RAW, &end);
```

```

        double taken_time = (end.tv_sec - start.tv_sec) + (end.tv_nsec -
            ↪ start.tv_nsec) / 1e9;
        std::cout << "sin(" << x << ") = " << sin << std::endl;
        std::cout << "Run " << i + 1 << " took " << taken_time << " seconds
            ↪ to complete" << "\n" << std::endl;
        time_total += taken_time;
    }

    std::cout << "Average time: " << time_total / runs << " seconds" << std
        ↪ ::endl;
    return 0;
}

```

Приложение 2: bash-скрипт для компиляции и запуска программы *SinCalculation.cpp*, записи результата в файл report.csv

```

#!/bin/bash

input="SinCalculation.cpp"

optimization_levels=("-O0" "-O1" "-O2" "-O3" "-Os")
n=2300000000
x=90

echo "Optimisation level, N value, X value, Time taken (seconds)" > report.csv

for i in "${optimization_levels[@]"; do
    g++ -o sin $input $i -std=c++11
    output=$(./sin $x $n | grep "Average time:" | awk '{print $3}')
    echo "$i, $n, $x, $output" >> report.csv
done

echo "Successfully generated report.csv"

./create_table.sh

```

Приложение 3: результат работы *bash-скрипта compile_and_run.sh*

```
matveysorokin@16SMatveyS:~/NSU_labs_EVM_3-semester/lab1$ ./compile_and_run.sh
Successfully generated report.csv
Оптимизация      N      X      Время выполнения (сек)
-O0      2300000000      90      15.0147
-O1      2300000000      90      2.47579
-O2      2300000000      90      2.47875
-O3      2300000000      90      2.48826
-Os      2300000000      90      2.7505
```

Приложение 4: результаты измерений в *report.csv* с использованием *bash-скрипта compile_and_run.sh*

```
Optimixation level, N value, X value, Time taken (seconds)
-O0, 2300000000, 90, 15.0147
-O1, 2300000000, 90, 2.47579
-O2, 2300000000, 90, 2.47875
-O3, 2300000000, 90, 2.48826
-Os, 2300000000, 90, 2.7505
```

Приложение 5: результат работы скомпилированной программы *SinCalculation.cpp*

```
matveysorokin@16SMatveyS:~/NSU_labs_EVM_3-semester/lab1$ ./sin 90 2300000000
x = 90, n = 2300000000
sin(90) = 1
Run №1 took 14.8843 seconds to complete

sin(90) = 1
Run №2 took 15.0117 seconds to complete

sin(90) = 1
Run №3 took 14.9104 seconds to complete

sin(90) = 1
Run №4 took 14.9537 seconds to complete

sin(90) = 1
Run №5 took 14.9767 seconds to complete

Average time: 14.9473 seconds
```