###### **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**

Работа с высокоуровневыми и низкоуровневыми перефирийными устройствами

студента 2 курса, группы 23201

Сорокина Матвея Павловича

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

А.С. Матвеев

Новосибирск 2024

**СОДЕРЖАНИЕ**

[**ЦЕЛЬ** 3](#_Toc174976371)

[**ЗАДАНИЕ** 3](#_Toc174976372)

[**ОПИСАНИЕ РАБОТЫ** 4](#_Toc174976373)

[**ЗАКЛЮЧЕНИЕ** 4](#_Toc174976374)

[**ПРИЛОЖЕНИЯ** 5](#_Toc174976375)

# **ЦЕЛЬ**

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

# **ЗАДАНИЕ**

1. Написать программу на языке C или C++, которая реализует выбранный алгоритм из задания.
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**

1. Реализовано задание 4 - алгоритм вычисления синуса с помощью разложения в степенной ряд по превым N членам данного ряда на языке C++.
2. 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 знаениями x и N, записывающий время вычисления синуса угла с различными уровнями оптимизации компилятора GCC в report.csv файл (см. Приложение 2, 3), 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.

# **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, записи результата в файл report.csv**

|  |
| --- |
| #!/bin/bash  input="SinCalculation.cpp"  optimization\_levels=("-O0" "-O1" "-O2" "-O3" "-Os")  n=2300000000  x=90  echo "Optimixation 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 (см. Приложение 2)**

A screenshot of a computer

Description automatically generated

**Приложение 4: результаты измерений с использованием bash-скрипта compile\_and\_run.sh (см. Приложение 2) в report.csv**

A screen shot of a computer

Description automatically generated

**Приложение 5: результат работы скомпилированной программы SinCalculation.cpp не используя bash-скрипты**

**A screenshot of a computer

Description automatically generated**