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

«Определение времени работы прикладных программ»

студента 2 курса, группы 22206

***Тропина Никиты Васильевича***

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

доцент, к.т.н.

А.Ю.Власенко

Новосибирск 2023

# СОДЕРЖАНИЕ

[ЦЕЛИ 3](#_Toc146825537)

[ЗАДАНИЕ 4](#_Toc146825538)

[ОПИСАНИЕ РАБОТЫ 5](#_Toc146825539)

[ЗАКЛЮЧЕНИЕ 7](#_Toc146825540)

[Приложение 1. Реализованная программа на C++ 8](#_Toc146825541)

# ЦЕЛИ

1. Изучение процедуры 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.
5. Изучение основных функций оптимизирующего компилятора и некоторых примеров оптимизирующих преобразований и уровней оптимизации.
6. Получение базовых навыков работы с компилятором GCC.
7. Исследование влияния оптимизационных настроек компилятора GCC на время исполнения программы.

# ЗАДАНИЕ

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 таким, чтобы время работы функции было от 30 до 60 секунд.
4. Программу скомпилировать компилятором GCC с уровнями оптимизации – **-O0, -O1, -O2, -O3, -Os, -Ofast, -Og** под архитектуру процессора x86(x86-64).
5. Для каждого из семи вариантов компиляции измерить время работы программы для нескольких значений N(0.5\*, ,1.5\*).
6. Составить отчет по лабораторной работе.

Вариант задания: Алгоритм вычисления определенного интеграла

сложной функции методом трапеций:

где f(x) = sin(x), a = 0, b = , N – число интервалов.

# ОПИСАНИЕ РАБОТЫ

В ходе практической работы была написана программа на языке C++, содержащая функцию **calculateTrapezoidIntegral**, которая реализует алгоритм вычисления определенного интеграла методом трапеций

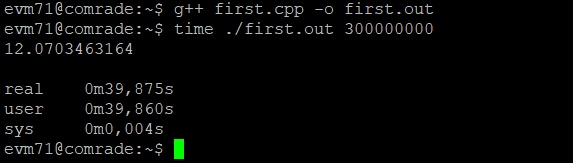
где f(x) = sin(x), a = 0, b = , N – число интервалов. Реализованная программа приведена в Приложении 1.

Была проверена правильность работы программы на нескольких входных данных:

* Значение данного интеграла равно , что в приближении: 12.070346316.
* При N = 10000, получен результат 12.070346068.
* При N = 1000000, получен результат 12.07346316.

Следовательно, программа работает корректно; при больших значениях N будет достигаться большая точность.

В частности, было выбрано значение N = 300000000, при котором программа работает 39,875 секунд. Программа была скомпилирована компилятором g++, время ее работы измерялось с помощью утилиты **time**.



Описание выбранной методики:

Утилита time выдаёт следующие временные характеристики работы программы:

real - общее время работы программы согласно системному таймеру,

user - время, которое работал пользовательский процесс (кроме времени работы других процессов) и

sys - время, затраченное на выполнение системных вызовов программы.

После этого программа была скомпилирована с уровнями оптимизации **-O0, -O1, -O2, -O3, -Os, -Ofast, -Og** компилятора g++.

Были проведены измерения времени работы программы при значениях N = 300000000, = 0,5 \* N (1500000000) и = 1,5 \* N (4500000000). Результаты измерения представлены в виде таблицы, составлены графики зависимости времени работы программы от параметра N с различными уровнями оптимизации.

Табл. 1. Результаты измерения времени работы программы с различными значениями параметра N с различными уровнями оптимизации:

|  |  |  |  |
| --- | --- | --- | --- |
| **Уровень оптимизации** | **= 0,5 \* N** | **N** | **= 1,5 \* N** |
| **-O0** | 20,812s | 40,092s | 60,258s |
| **-O1** | 20,799s | 39,855s | 59,598s |
| **-O2** | 20,981s | 40,621s | 61,462s |
| **-O3** | 16,586s | 32,879s | 51,872s |
| **-Os** | 20,160s | 40,109s | 60,881s |
| **-Ofast** | 6,846s | 13,439s | 20,307s |
| **-Og** | 20,166s | 40,932s | 61,547s |

Графики зависимости времени выполнения программы с уровнями оптимизации -O0, -O1, -O2, -O3, -Os, -Ofast, -Og от параметра N.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения практической работы была написана программа на языке C++, скомпилирована с помощью компилятора g++ с различными уровнями оптимизации (-O0, -O1, -O2, -O3, -Os, -Ofast, -Og). При помощи утилиты time было измерено время работы программы с различными уровнями оптимизации компилятора с различными значениями параметра N (0,5 \* N = 150000000, N = 300000000, 1,5 \* N = 450000000). Была получена зависимость времени работы программы от входного значения N. Следует отметить, что на представленных в отчете графиках некоторые данные сливаются, т.к. сложность выполнения программы заключается в вычислении математических функций и их количестве. Следовательно, выбиваются графики уровней оптимизаций, направленных на ускорение данных вычислений. Также можно отметить, что сложность выполнения программы на всех уровнях оптимизации растет линейно.

# Приложение 1. Реализованная программа на C++

