###### 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

«Анализ времени работы программ при разных режимах компиляции»

студента 2 курса, группы 20204

**Дронова Дениса Юрьевича**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

Кандидат технических наук

А.Ю. Власенко

Новосибирск 2021

**СОДЕРЖАНИЕ**

[ЦЕЛЬ 3](#_Toc18443921)

[ЗАДАНИЕ 3](#_Toc18443922)

[ОПИСАНИЕ РАБОТЫ 4](#_Toc18443923)

[ЗАКЛЮЧЕНИЕ 5](#_Toc18443924)

[Приложение 1. Листинг программы sin(x) 6](#_Toc18443925)

**ЦЕЛИ РАБОТЫ**

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.
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 N0 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 при нескольких значениях N (0.5\* N0, N0, 1.5\* N0).

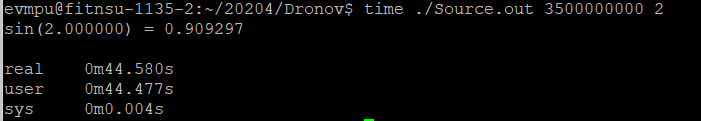
**Вариант задания: №4**

# ОПИСАНИЕ РАБОТЫ

Алгоритм подсчета синуса был написан на языке программирования Си, а затем скомпилирован. Готовая программа была запущена и успешно прошла тесты на правильность работы алгоритма вычисления синуса. После был проведен подбор параметра n для того, чтобы время работы составляло от 30 до 60 секунд. Значение подобранного параметра N равно **3500000000**. При данном параметре время работы программы составило **44,580** секунды (real time) при использовании утилиты time.

Команда компиляции: **g++ Source.c -o Source.out**

Команда запуска: **time ./Source.out 3500000000 2**



Затем были произведены замеры работы программы при разных значениях n, и при разных режимах компиляции. На основе полученных данных были построены графики и таблица.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | O0 | O1 | O2 | O3 | Os | Ofast | Og |
| 0.5\*N | 22,245 | 18,339 | 18,341 | 18,334 | 18,285 | 9,464 | 18,212 |
| N | 44,58 | 36,738 | 36,601 | 36,704 | 36,571 | 18,921 | 36,487 |
| 1.5\*N | 89,03 | 73,401 | 73,316 | 73,187 | 73,001 | 37,818 | 72,958 |

Команды компиляции: **g++ Source.c -O0 -o Source.out**

**g++ Source.c -O1 -o Source.out**

**g++ Source.c -O2 -o Source.out**

**g++ Source.c -O3 -o Source.out**

**g++ Source.c -Os -o Source.out**

**g++ Source.c -Ofast -o Source.out**

**g++ Source.c -Og -o Source.out**

Команды запуска: **time ./Source.out 3500000000 2**

**time ./Source.out 1750000000** **2**

**time ./Source.out 7000000000 2**

# ЗАКЛЮЧЕНИЕ

Время работы программы является одним из самых важных критериев, учитываемых при разработке алгоритмов и т. п. Анализ времени помогает при оптимизации программ, поэтому очень важно уметь пользоваться инструментарием, приспособленным для его измерения. Помимо ручных оптимизаций программ, существенный вклад может внести использование различных уровней компиляции. Так, на примере алгоритма определения синуса аргумента с разложением в ряд Тейлора выяснилось, что использование любого из предложенных ключей компиляции (-O1, -O2, -O3, -Os, -Ofast, -Og) может привести к сокращению времени работы программы почти в 2 раза по сравнению с уровнем компиляции, не использующего оптимизации. Основываясь на полученных в результате выполнения задания данных, можно сделать вывод, что для данного алгоритма наименьшее время работы достигалось в среднем при использовании режима Оfast.

Приложение 1. *Листинг программы sin(x)*

#include <stdio.h>

#include <stdlib.h>

#include <string>

#define PI 3.14159265358979323846

double sine(double x, long long N) {

x = x - (long)(x / 2 / PI) \* PI \* 2;

double current = x, sum = x;

for (long long i = 3; i < 2 \* N; i += 2) {

current = -current / i / (i - 1) \* x \* x;

sum += current;

}

return sum;

}

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

if (argc != 3)

exit(0);

long long n = atoll(argv[1]);

double x = atof(argv[2]);

printf("sin(%lf) = %lf\n", x, sine(x, n));

return 0;

}