**ФГБОУ ВО   
Уфимский университет науки и технологий**

**Кафедра ВМиК**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 90 |  |  |  |  |  |  |  |  |  |  |  |  |
| 80 |  |  |  |  |  |  |  |  |  |  |  |  |
| 70 |  |  |  |  |  |  |  |  |  |  |  |  |
| 60 |  |  |  |  |  |  |  |  |  |  |  |  |
| 50 |  |  |  |  |  |  |  |  |  |  |  |  |
| 40 |  |  |  |  |  |  |  |  |  |  |  |  |
| 30 |  |  |  |  |  |  |  |  |  |  |  |  |
| 20 |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

Параллельное вычисление суммы числового ряда

**ОТЧЕТ**

**по лабораторной работе**

**по** Технологиям параллельного программирования

(*наименование дисциплины*)

|  |
| --- |
| Лабораторная работа 1 (Вар. 25) |
| (обозначение документа) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Группа |  |  | Фамилия, И., О. | Подпись | Дата | Оценка |
| МО-325Б |  |
|  |  |
| Студент | | | Шарыгин М.С. |  |  |  |
| Преподаватель | | | Спеле В.В. |  |  |  |
| Принял | | |  |  |  |  |

**Уфа 2025 г****.**

Содержание

[1 Цель работы 3](#_Toc193367940)

[2 Практическая часть 4](#_Toc193367941)

[2.1 Описание используемых компиляторов 4](#_Toc193367942)

[2.2 Код вычисления суммы ряда 5](#_Toc193367943)

[2.3 Проверка работоспособности 5](#_Toc193367944)

[2.4 Подбор параметра «N» 6](#_Toc193367945)

[2.5 Подбор «Debug/Release» и «x86/x64» 6](#_Toc193367946)

[2.6 Компилятор «Microsoft C/C++» 6](#_Toc193367947)

[2.7 Компилятор «Intel Classic C/C++» 7](#_Toc193367948)

[2.8 Компилятор «Intel Clang/LLVM C/C++» 7](#_Toc193367949)

[2.9 Компилятор «GNU C/C++» 7](#_Toc193367950)

[2.10 «Intel OneAPI Advisor» 8](#_Toc193367951)

[3 Вывод 9](#_Toc193367952)

[4 Приложение 10](#_Toc193367953)

[4.1 Код программы 10](#_Toc193367954)

# Цель работы

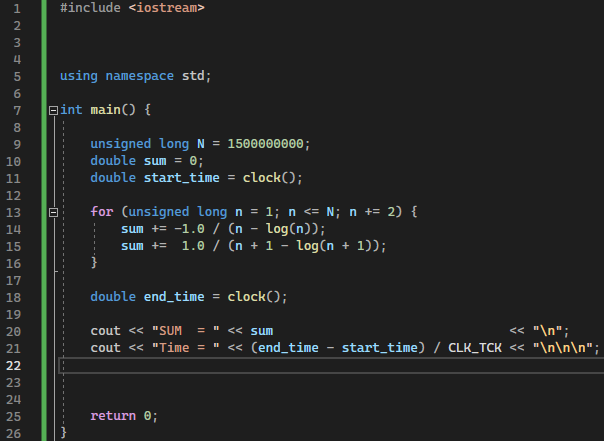
На примере задачи сложения суммы ряда научиться использовать оптимизационные ключи компиляторов в операционных системах Windows (и Linux), а также инструмент для профилирования программ Intel Advisor.

# Практическая часть

## Описание используемых компиляторов

* Компилятор «Microsoft C/C++»: в состав Visual Studio включен компилятор языка «C/C++» позволяющий создавать все, от простых консольных приложений, до универсальных приложений «Windows», приложений «Магазина Windows» и компонентов «.NET»;
* Компилятор «Intel Classic C/C++»: классический оптимизирующий компилятор от «Intel». Входит в состав «Intel OneAPI HPC Toolkit». В среде «Windows» возможна интеграция в «Visual Studio»;
* Компилятор «Intel Clang/LLVM C/C++»: компилятор нового поколения на базе «LLVM» от «Intel». Входит в состав «Intel OneAPI HPC Toolkit». В среде Windows возможна интеграция в Visual Studio;
* Компилятор «GNU C/C++»: «GNU» компилятор, входящий в состав операционной системы «Linux.»

## Код вычисления суммы ряда

Рисунок 2.1 – Код вычисления суммы ряда

## Проверка работоспособности

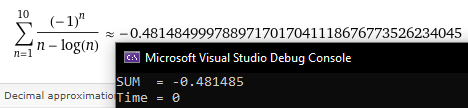


Рисунок 2.2 – Пример при N = 10

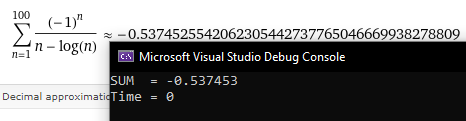


Рисунок 2.3 – Пример при N = 100

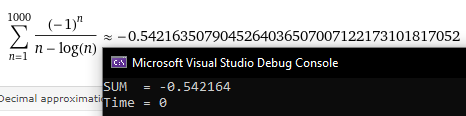


Рисунок 2.4 – Пример при N = 1000

## Подбор параметра «N»

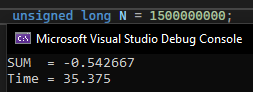


Рисунок 2.5 – Подобранный параметр

## Подбор «Debug/Release» и «x86/x64»



Рисунок 2.6 – Подбор «Debug/Release» и «x86/x64»

## Компилятор «Microsoft C/C++»

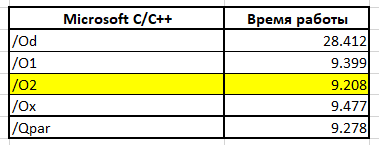


Рисунок 2.7 – Компилятор «Microsoft C/C++»

## Компилятор «Intel Classic C/C++»

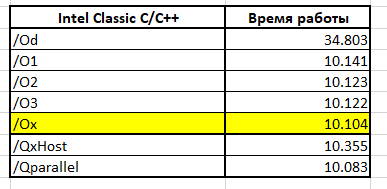


Рисунок 2.8 – Компилятор «Intel Classic C/C++»

## Компилятор «Intel Clang/LLVM C/C++»

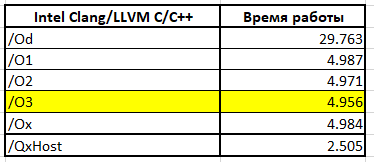


Рисунок 2.9 – Компилятор «Intel Clang/LLVM C/C++»

## Компилятор «GNU C/C++»

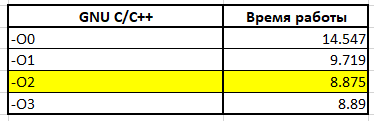


Рисунок 2.10 – Компилятор «GNU C/C++»

## «Intel OneAPI Advisor»

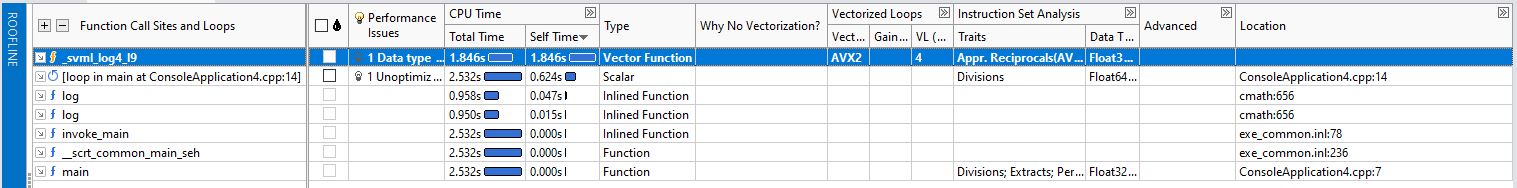


Рисунок 2.11 – Векторизация кода

# Вывод

В ходе лабораторной работы мы на примере задачи сложения суммы ряда научились использовать оптимизационные ключи компиляторов в операционных системах Windows (и Linux), а также инструмент для профилирования программ Intel Advisor.

# Приложение

## Код программы

#include <iostream>

using namespace std;

int main() {

unsigned long N = 1500000000;

double sum = 0;

double start\_time = clock();

for (unsigned long n = 1; n <= N; n += 2) {

sum += -1.0 / (n - log(n));

sum += 1.0 / (n + 1 - log(n + 1));

}

double end\_time = clock();

cout << "SUM = " << sum << "\n";

cout << "Time = " << (end\_time - start\_time) / CLK\_TCK << "\n\n\n";

return 0;

}