Федеральное государственное бюджетное образовательное учреждение высшего образования

«Уфимский Государственный Авиационный Технический Университет»

Кафедра ВВТиС

Отчет по дисциплине «Суперкомпьютерные технологии» по лабораторной работе №1

Выполнила студентка группы ПРО

Проверил старший преподаватель

Юлдашев А.В.

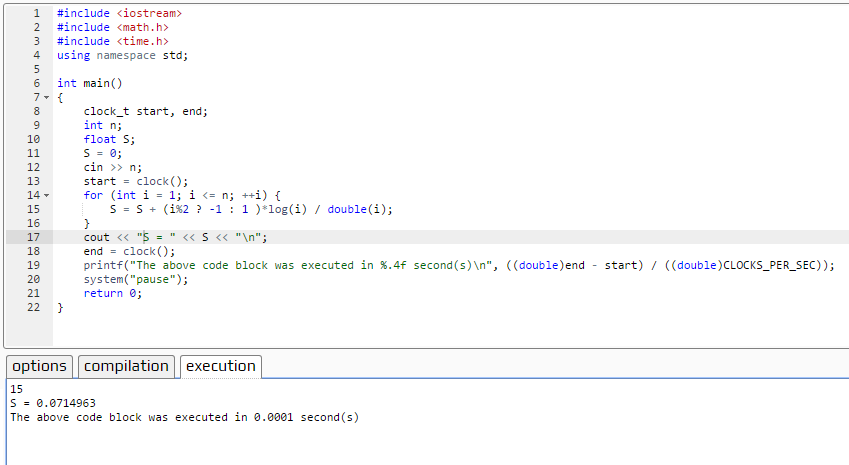
Уфа 2017

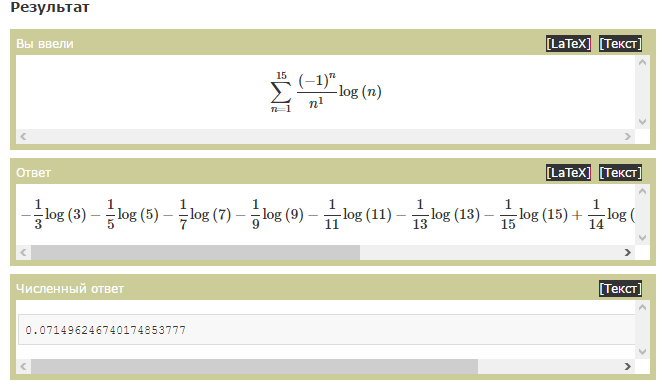
**Задание**

Вариант 5

**Практическая часть**

Проверка





Время работы

* Debug, x86 – 29.7920
* Release, x86 – 13.5560
* Debug, x64 – 22.6250
* **Release, x64 – 4.5230**
* Debug, x64, Intel Compiler – 6.2970
* Release, x64, Intel Compiler – 6.2850

На этом этапе лучший результат был показан на **Release, x64.**

Далее изменяем оптимизацию:

* Debug, x86, Intel Compiler, Disabled (/Od) – 41.3790
* Release, x86, Intel Compiler, Disabled (/Od) – 41.3610
* Debug, x64, Intel Compiler, Disabled (/Od) – 39.5762
* Release, x64, Intel Compiler, Disabled (/Od) – 39.8720
* Debug, x86, Intel Compiler, Minimize Size (/O1) – 39.1590
* Release, x86, Intel Compiler, Minimize Size (/O1) – 38.7760
* Debug, x64, Intel Compiler, Minimize Size (/O1) – 38.4756
* Release, x64, Intel Compiler, Minimize Size (/O1) – 38.9452
* Debug, x86, Intel Compiler, Maximize Speed (/O2) – 38.8400
* Release, x86, Intel Compiler, Maximize Speed (/O2) – 39. 2050
* **Debug, x64, Intel Compiler, Maximize Speed (/O2) – 35. 6275**
* Release, x64, Intel Compiler, Maximize Speed (/O2) – 36.8931
* Debug, x86, Intel Compiler, Highest Optimizations (/O3) – 56.0112
* Release, x86, Intel Compiler, Highest Optimizations (/O3) - 26.4300
* Debug, x64, Intel Compiler, Highest Optimizations (/O3) – 51.2070
* Release, x64, Intel Compiler, Highest Optimizations (/O3) – 50.1579

После изменения оптимизации лучшим результатом стал **Debug, x64, Intel Compiler, Maximize Speed (/O2).**

Далее применяем опцию Parallel.

Вместе с опцией **Intel Processor-Specific Optimisation – 3.9330 c.**

**Заключение**

В результате выполнения лабораторной работы был изучен компилятор Intel, а также были изучены встроенные в компилятор Visual Studio 2015 функции для оптимизации программы. Это такие специальные функции, реализация которых обеспечивается компилятором автоматически. Было выяснено, что компилятор обладает глубоким знанием того, как эффективно расположить последовательность машинных команд так, чтобы код работал максимально быстро на указанной программной архитектуре. Оптимизация программы заключалась в:

* Конфигурация Debug
* Конфигурация Release
* Изменение платформы проекта
* Использование ключей /O1, /O2, /O3 и т.д.

Все методы оптимизации, которые использовались в лабораторной работе, могут значительно улучшить производительность кода на двухзначный процент. Все они поддерживаются в компиляторе Visual Studio 2015. Что делает эти техники действительно важными, так это то, что после их применения компилятор способен делать другие оптимизации.

На такие противоречивые результаты повлияла, на мой взгляд, архитектура, т.е. особенности процессора, на котором и проходили замеры времени работы программы.

Исходный код

#include <iostream>

#include <math.h>

#include <time.h>

using namespace std;

int main()

{

clock\_t start, end;

int n=15;

float S;

S = 0;

cin >> n;

start = clock();

for (int i = 1; i <= n; ++i) {

S = S + (i%2 ? -1 : 1 )\*log(i) / double(i);

}

cout << "S = " << S << "\n";

end = clock();

printf("The above code block was executed in %.4f second(s)\n", ((double)end - start) / ((double)CLOCKS\_PER\_SEC));

system("pause");

return 0;

}