###### 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 ЛАБОРАТОРНОЙ РАБОТЫ

«ИЗУЧЕНИЕ ОПТИМИЗИРУЮЩЕГО КОМПИЛЯТОРА»

студента 2 курса, 18209 группы

**Большим Максима Антоновича**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

Власенко Андрей Юрьевич

Новосибирск 2019

1. **Цель**

1. Изучение основных функций оптимизирующего компилятора, и некоторых примеров оптимизирующих преобразований и уровней оптимизации.

2. Получение базовых навыков работы с компилятором GCC.

3. Исследование влияния оптимизационных настроек компилятора GCC на время исполнения программы.

1. **Задание**

Написать программу на языке C++, которая реализует вычисления числа Пи с помощью разложения в ряд (ряд Грегори-Лейбница) по формуле Лейбница N первых членов ряда. Реализовав код алгоритма, проверить правильность работы программы. Выбрать значение параметра N таким, чтобы время работы программы было порядка 30-60 секунд. Программу скомпилировать компилятором GCC с уровнями оптимизации **-O0, -O1, -O2, -O3, -Os, -Ofast, -Og** под архитектуру процессора **x86\_64**. Для каждого из семи вариантов компиляции измерить время работы программы.

1. **Описание работы**

Компилятор, помимо компиляции, т.е. трансляции кода высокоуровневого языка программирования в нативный код, способен оптимизировать написанный программистом код. В компиляторе GCC, которым предполагается компиляция программы, существует несколько уровней компиляции:

* На уровне **O0** почти все оптимизации отключены. Предпочтителен для отладки кода на уровне ассемблера;
* На уровне **O1** включены оптимизации для уменьшения размера бинарного исполняемого файла и такие оптимизации, уменьшающие время работы программы, которые не сильно замедляют работу компилятора;
* На уровне **O2** включены практически все доступные оптимизации, кроме тех, что ускоряют вычисления за счет увеличения размера кода;
* В уровне **O3** включены все оптимизации из уровня O2, к ним добавлены оптимизации времени работы программы, которые могут приводить к увеличению размера бинарного исполняемого файла;
* Уровень **Os** служит для оптимизации размера программ, в него включено подмножество оптимизаций из уровня O2.
* Уровень **Ofast** включает все оптимизации уровня O3, а также ряд других, таких как использование более быстрых и менее точных математических функций.
* Уровень **Og** производит все оптимизации, которые сохраняют возможность просмотра стека вызовов, фрагментов исходного текста программы, относящихся к разным уровням этого стека, и возможность приостановки программы для каждой строки исходного текста, содержащей операторы.

Стоит отметить, что уровни оптимизации выше O2 способны привести

компилируемый код программы в нерабочее состояние, так как могут

использовать не безопасные методы оптимизации и ускорения

вычислений программы, или же способны уменьшить точность

вычислений программы, так как используют более быстрые и менее

точные математические функции.

1. **Исследования**

Был проведён анализ времени выполнения вычисления числа Пи с помощью разложения в ряд (ряд Грегори-Лейбница) по формуле Лейбница N первых членов ряда. В ходе работы при компиляции программы использовались все возможные уровни оптимизации компилятора. В ходе исследования была достигнуто необходимое время выполнения алгоритма: ~30-60 секунд. Скомпилированные на разных уровнях программы выдавали разные результаты. Вычисления происходили на машине с архитектурой X86\_64.

1. **Результаты работы**

В таблице ниже приведены результаты работы программы и время,

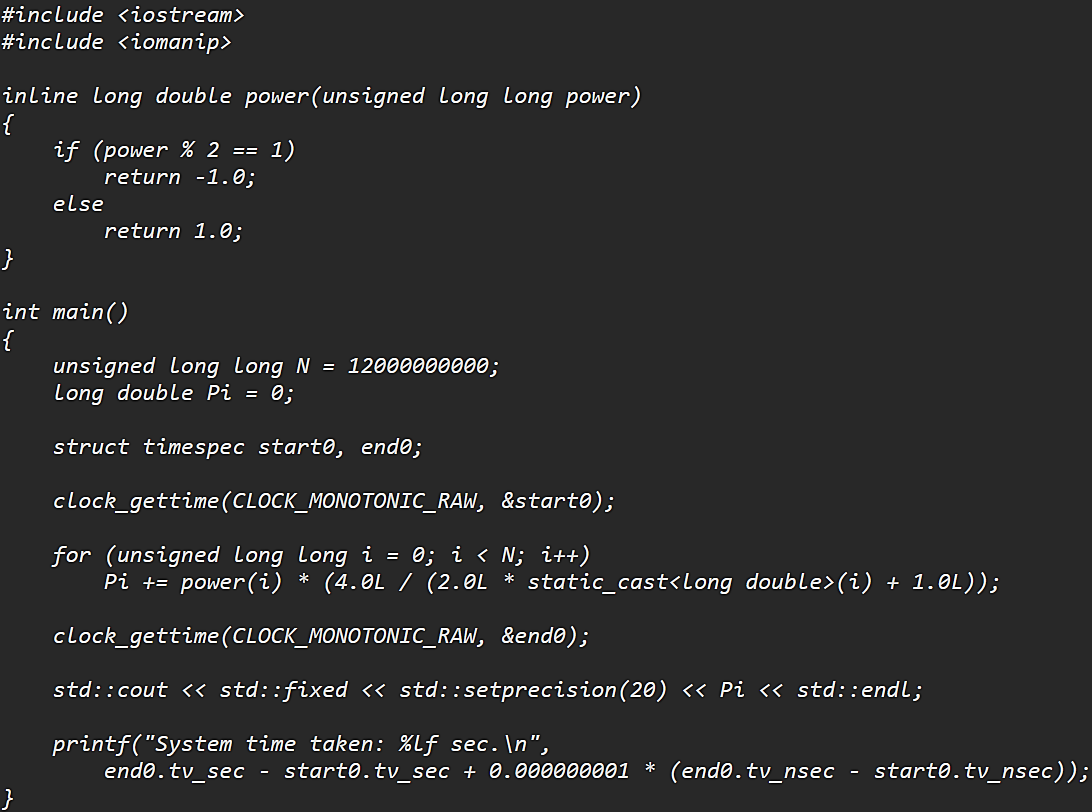
потраченное на исполнение алгоритма при разных ключах компиляции:

|  |  |  |  |
| --- | --- | --- | --- |
| Оптимизация\N | 6000000000 | 9000000000 | 12000000000 |
| O0 | 33.331558 | 49.904620 | 66.473201 |
| O1 | 31.386494 | 46.059941 | 62.236673 |
| O2 | 31.209416 | 46.018633 | 61.084755 |
| O3 | 30.356019 | 45.395034 | 60.507886 |
| Os | 30.303866 | 45.411631 | 60.640926 |
| Ofast | 30.273229 | 45.420054 | 60.487572 |
| Og | 30.235626 | 45.305600 | 60.431758 |

Стоит отметить, что при значении N=3000000000 некоторые уровни

оптимизации замедляли вычисления алгоритма на 3-4 секунды в отличие от программы, скомпилированной на уровне O0.

График производительности программы приведен ниже:

1. **Листинг программы**
2. **Выводы**

По результатам проведённых исследований было подтверждено, что уровни оптимизации кода алгоритма способны уменьшить время работы на несколько секунд. Вычисленные значения числа Пи до 20 знаков почти ничем не различались между программами, скомпилированными с разными уровнями оптимизации. Ошибки в оптимизированных программах не были обнаружены, все скомпилированные версии алгоритма работали исправно и выдавали хороший результат.