###### 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

«Лабораторная работа 2»

Студента 1 курса, 19210 группы

**Пирожков Андрей Константинович**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

(ученая степень, звание)

Д.С.Иванишкин

Новосибирск 2020

**СОДЕРЖАНИЕ**

[ЦЕЛЬ 3](#_Toc51744552)

[ЗАДАНИЕ 3](#_Toc51744553)

[ОПИСАНИЕ РАБОТЫ 4](#_Toc51744554)

[ЗАКЛЮЧЕНИЕ 7](#_Toc51744555)

# ЦЕЛЬ

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

# ЗАДАНИЕ

1. Написать программу на языке C или C++, которая реализует выбранный алгоритм из задания.
2. Проверить правильность работы программы на нескольких тестовых наборах входных данных.
3. Выбрать значение параметра N таким, чтобы время работы программы было порядка 30-60 секунд.
4. Программу скомпилировать компилятором GCC с уровнями оптимизации -O0, -O1, -O2, -O3, -Os, -Ofast, -Og под архитектуру процессора x86.
5. Для каждого из семи вариантов компиляции измерить время работы программы при нескольких значениях N.
6. Составить отчет по лабораторной работе

**Вариант задания: 7**

Алгоритм сортировки методом пузырька. Дан массив случайных чисел длины N. На первой итерации попарно упорядочиваются все соседние элементы; на второй – все элементы, кроме последнего элемента; на третьей – все элементы, кроме последнего элемента и предпоследнего элемента и т.п.

# **ОПИСАНИЕ РАБОТЫ**

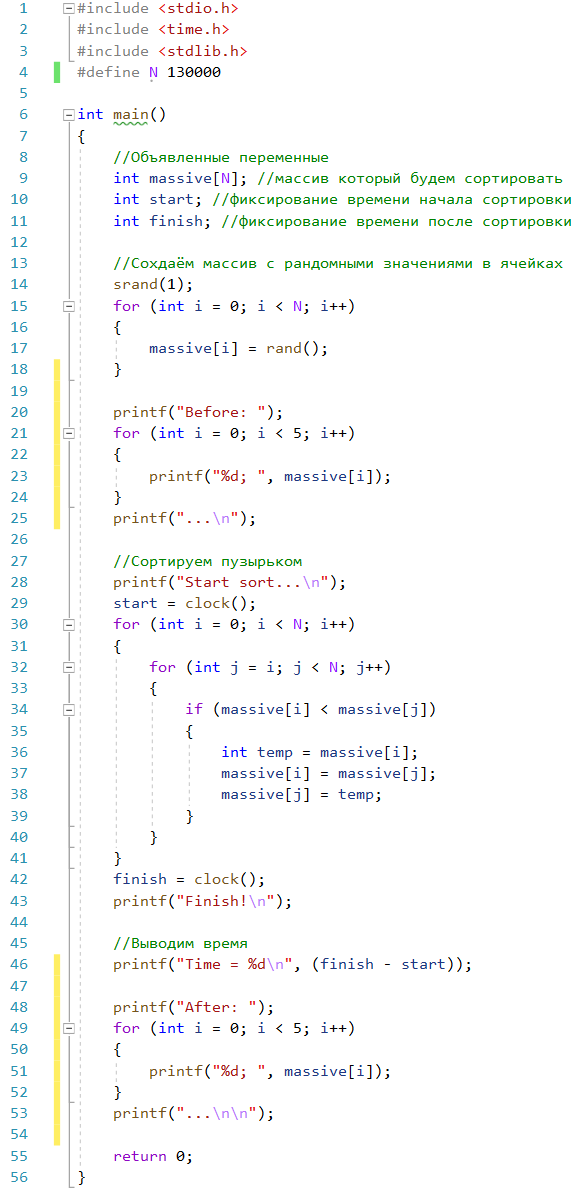
За основу этой лабораторной работы была взята 1-ая лабораторная работа, написанная на языке Си (стандарта 1999г). Был написан алгоритм пузырьковой сортировки массива из N элементов. Входные данные этого массива одинаково генерируются во время работы, а N меняется от 120000 до 160000 с шагом в 10000.

Время фискировалось с помощью функции clock(). Фиксируется начальное время перед сортировкой и конечное время после сортировки. Далее вычитаем из одно другое и получаем приблизительное время сортировки.

В ходе лабораторной работе пришлось внести изменения в код программы из-за особенностей компилятора. Оптимизации O2, O3, Os, Ofast решили, что им незачем сортировать массив если я его не использую, поэтому пришлось выводить на экран хотя бы какой-нибудь элемент.

Провёл 4 эксперимента с N=120000, 130000, 140000, 150000.

Код программы на следующей странице.



Результаты программы с разными входными данными (в секундах):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | O0 | O1 | O2 | O3 | Os | Ofast | Og |
| 120000 | 38,928809 | 23,241352 | 22,742400 | 22,736696 | 22,808972 | 22,704987 | 27,149575 |
| 130000 | 45,758299 | 27,346445 | 26,713850 | 26,703923 | 26,716493 | 26,725797 | 31,705554 |
| 140000 | 53,267429 | 31,748455 | 30,912556 | 31,014478 | 31,070130 | 30,950833 | 37,023092 |
| 150000 | 60,991532 | 36,448184 | 35,563245 | 35,575337 | 35,591740 | 35,607358 | 42,635615 |

Исходя из результатов, можно сделать вывод о том, что Завсисмость от N прямопропорцианальна времени. Такой вывод можно сделать из-за чрезмерной ровности графика.

Помимо этого стоит отметить, что все оптимизации кроме O0 и Og практически одинаково оптимизировали программу. Между O0 в 1,67 раза дольше чем O1 причём во всех тестах с разным N.

# ЗАКЛЮЧЕНИЕ

В ходе этой лабораторной работы я изучил основные функции оптимизирующего компилятора, и некоторые примеры оптимизирующих преобразований и уровней оптимизации. Помимо этого, я получил базовые навыки с работой компилятора gcc, а также исследовал влияние оптимизационных настроек компилятора gcc на время исполнения программы. Результаты моего эксперимента показали, что для сортировки массива пузырьком время прямо пропорционально зависит от количества элементов массива. Об этом говорит равномерные расстояния между графиками.