

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.

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

ОПИСАНИЕ РАБОТЫ

Вариант задания

Для выполнения мной было выбрано задание №6: Алгоритм вычисления функции $\ln(1+x)$ с помощью разложения в ряд по первым N членам этого ряда.

Реализация алгоритма:

```
#include <stdio.h>

double Power(double number, double pow) {
    double result = 1;
    for (int i = 1; i <= pow; i++) {
        result *= number;
    }
    return result;
}

int main() {
    double n = 50000;
    double x = 0.5;
    double sum = 0;
    for (double i = 1; i <= n; i++) {
        sum += (Power(-1, i + 1) * Power(x, i) / i);
    }
    printf("%f\n", sum);
    return 0;
}
```

Правильность работы программы на нескольких тестовых наборах входных данных была проверена в отчете к предыдущей лабораторной работе.

Графики зависимости времени выполнения программы с уровнями оптимизации -O0, -O1, -O2, -O3, -Os, -Ofast, -Og от параметра N.

Согласно заданию, время работы программы должно занимать от 30 до 60 секунд. Без указания компилятору применить определенный уровень компиляции программа работает 46.107415 секунд при $N = 110000$. Для каждого из вариантов компиляции будем проверять значения времени при $N=110000 \pm 10000$.

```
evm24@comrade:~$ gcc lab1_clock_gettime.c -o lab1.exe
evm24@comrade:~$ ./lab1.exe
0.405465
Time taken: 46.118853 sec.
evm24@comrade:~$ ./lab1.exe
0.405465
Time taken: 46.114592 sec.
evm24@comrade:~$ ./lab1.exe
0.405465
Time taken: 46.107415 sec.
```

Измерение времени для определения значения параметра N

-O0

```
evm24@comrade:~$ gcc -O0 lab2.c -o lab2.exe
evm24@comrade:~$ ./lab2.exe
N=100000
Time taken: 38.736446 sec.
Time taken: 37.939270 sec.
Time taken: 38.520651 sec.
N=110000
Time taken: 46.139652 sec.
Time taken: 45.858860 sec.
Time taken: 45.863036 sec.
N=120000
Time taken: 54.803024 sec.
Time taken: 54.548095 sec.
Time taken: 54.546029 sec.
```

-O1

```
evm24@comrade:~$ gcc -O1 lab2.c -o lab2.exe
evm24@comrade:~$ ./lab2.exe
N=100000
Time taken: 16.234320 sec.
Time taken: 15.974767 sec.
Time taken: 15.979115 sec.
N=110000
Time taken: 19.302868 sec.
Time taken: 20.079771 sec.
Time taken: 19.845262 sec.
N=120000
Time taken: 23.465677 sec.
Time taken: 23.726831 sec.
Time taken: 23.455899 sec.
```

-O2

```
evm24@comrade:~$ gcc -O2 lab2.c -o lab2.exe
evm24@comrade:~$ ./lab2.exe
N=100000
Time taken: 13.101268 sec.
Time taken: 13.277834 sec.
Time taken: 13.204622 sec.
N=110000
Time taken: 16.021244 sec.
Time taken: 15.755328 sec.
Time taken: 15.496810 sec.
N=120000
Time taken: 18.666430 sec.
Time taken: 18.677803 sec.
Time taken: 18.666130 sec.
```

-O3

```
evm24@comrade:~$ gcc -O3 lab2.c -o lab2.exe
evm24@comrade:~$ ./lab2.exe
N=100000
Time taken: 13.156272 sec.
Time taken: 12.923203 sec.
Time taken: 12.837312 sec.
N=110000
Time taken: 15.821030 sec.
Time taken: 15.902838 sec.
Time taken: 16.897258 sec.
N=120000
Time taken: 19.106607 sec.
Time taken: 19.429335 sec.
Time taken: 19.179005 sec.
```

-Os

```
evm24@comrade:~$ gcc -Os lab2.c -o lab2.exe
evm24@comrade:~$ ./lab2.exe
N=100000
Time taken: 16.567251 sec.
Time taken: 15.982669 sec.
Time taken: 15.974745 sec.
N=110000
Time taken: 19.314544 sec.
Time taken: 19.312619 sec.
Time taken: 19.587592 sec.
N=120000
Time taken: 22.957923 sec.
Time taken: 23.302482 sec.
Time taken: 23.492587 sec.
```

-Ofast

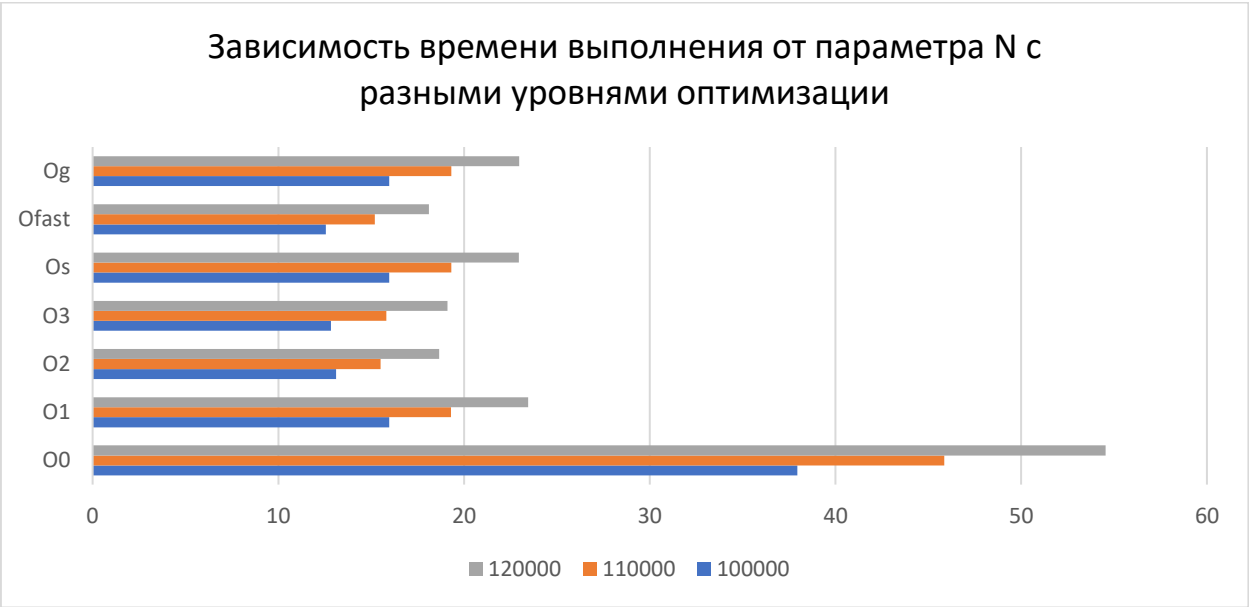
```
evm24@comrade:~$ gcc -Ofast lab2.c -o lab2.exe
evm24@comrade:~$ ./lab2.exe
N=100000
Time taken: 12.805647 sec.
Time taken: 12.551000 sec.
Time taken: 12.557744 sec.
N=110000
Time taken: 15.231129 sec.
Time taken: 15.190035 sec.
Time taken: 15.193453 sec.
N=120000
Time taken: 18.105826 sec.
Time taken: 18.592385 sec.
Time taken: 18.334847 sec.
```

-Og

```
evm24@comrade:~$ gcc -Og lab2.c -o lab2.exe
evm24@comrade:~$ ./lab2.exe
N=100000
Time taken: 16.601182 sec.
Time taken: 15.976051 sec.
Time taken: 16.050449 sec.
N=110000
Time taken: 19.848594 sec.
Time taken: 19.317843 sec.
Time taken: 19.325963 sec.
N=120000
Time taken: 22.973060 sec.
Time taken: 23.227312 sec.
Time taken: 22.983436 sec.
```

Сводная таблица результатов измерений (в секундах):

	O0	O1	O2	O3	Os	Ofast	Og
100000	37,93927	15,974767	13,101268	12,837312	15,974745	12,551	15,976051
110000	45,85886	19,302868	15,49681	15,82103	19,312619	15,190035	19,317843
120000	54,546029	23,455899	18,66613	19,106607	22,957923	18,105826	22,97306



ЗАКЛЮЧЕНИЕ

Я изучила уровни оптимизации компилятора GCC, получила основные навыки работы с ним, а также исследовала влияние оптимизационных настроек на время исполнения программы.

По графику видим, что любой из уровней оптимизации заметно уменьшил время исполнения программы по сравнению с -O0.

По графику зависимости времени выполнения программы видно, что для выбранного мной алгоритма быстрее всего работают уровни оптимизации -O2, -O3 и -Ofast.