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 Бородина Артёма Максимовича 2 курса, 19205 группы

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:
к.т.н, доцент
А.Ю. Власенко

Новосибирск 2020

СОДЕРЖАНИЕ

ЦЕЛЬ 3

ЗАДАНИЕ 3

ОПИСАНИЕ РАБОТЫ 4

ЗАКЛЮЧЕНИЕ 5

Приложение *1*. *Скрипт* 6

Приложение *2*. *График* 7

Приложение *3*. *Хеш-суммы* 8

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 вариант:

Алгоритм вычисления числа π с помощью разложения в ряд (ряд Грегори-Лейбница) по формуле Лейбница N первых членов ряда.

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.

ОПИСАНИЕ РАБОТЫ

1. Изучено понятие компилятора, его характеристики и основные функции, а также уровни компиляции, примеры оптимизирующих преобразований в GCC и их способы указания.

2. Для замеров времени была использована ранее написанная программа для вычисления числа π с помощью формулы Лейбница.

3. Был написан bash скрипт (Приложение 1) способный выдавать таблицу формата .csv с результатами замеров.

4. Мною были проведены по 11 замеров времени работы программы (Приложение 2) на каждый уровень оптимизации GCC:
   при N от $3*10^8$ до $4*10^9$ (с шагом в $3 * 10^8$).

5. Определение времени работы программы осуществлялось с помощью unix-утилиты time.

6. Был построен график зависимости времени от N для каждого уровня оптимизации GCC.

7. Из полученных данных видно, что время всех уровней оптимизации растет линейно с увеличение N. O2 и O3 имеют почти одинаковое время так как их исходные файлы одинаковые. Это можно увидеть, проверив их хеш-сумму (Приложение 3). Самым быстрым по времени, как и самым большим по размеру, уровнем оптимизации оказался Ofast. Время O1 и Og почти равно, потому что они оба компилируются почти без каких-либо оптимизаций. Уровень Os по времени сравним с O2 и O3, но при этом имеет наименьший размер.

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы были изучены основы компилятора, уровни оптимизации и примеры оптимизации. Было исследовано влияние уровней оптимизации на время исполнения программы.

Приложение 1. Скрипт

```
/home/evmpu/19205/Borodin/Lab2/measureTimeCSV   [----]  0 L:[  1+40  41/ 41] *(920 / 920b) <EOF>       UTF-8
#!/bin/bash
YELLOW='\033[0;33m'
MAGENTA='\033[0;35m'
CYAN='\033[0;36m'
NORMAL='\033[0m'

Name=$1
Num=$2
Step=$3
NumberOfSteps=$4
Output=$5
DataFile=column.temp

>$Output

echo -e >> $Output
for var in 0 1 2 3 s f g; do
    echo -e "\"-O$var\"" >> $Output
done


for (( i=1; i <= $NumberOfSteps; i++ )); do
    >$DataFile
    echo -e "$Num" >> temp
    echo -en "${MAGENTA}  N=$Num${NORMAL}: "
        for suffix in 0 1 2 3 s f g; do
            { time ./$Name$suffix $Num ; } 2> $DataFile > output
            sed -n '3p' $DataFile | awk '{print $2}' | sed -r 's/[sm]/ /g' | awk '{print $1*60 + $2}' | sed -r 's/[.]/,/g' >> temp
            echo -en "${YELLOW}$suffix "
        done
    mv temp $DataFile.
    paste -d';' $Output $DataFile >> $Output
    tail -n +9 "$Output" > "$Output.tmp" && mv "$Output.tmp" "$Output"
    Num=$((Num + Step))
    echo -e
done

rm output $DataFile
echo -e "  ${YELLOW}DONE${NORMAL}"
```

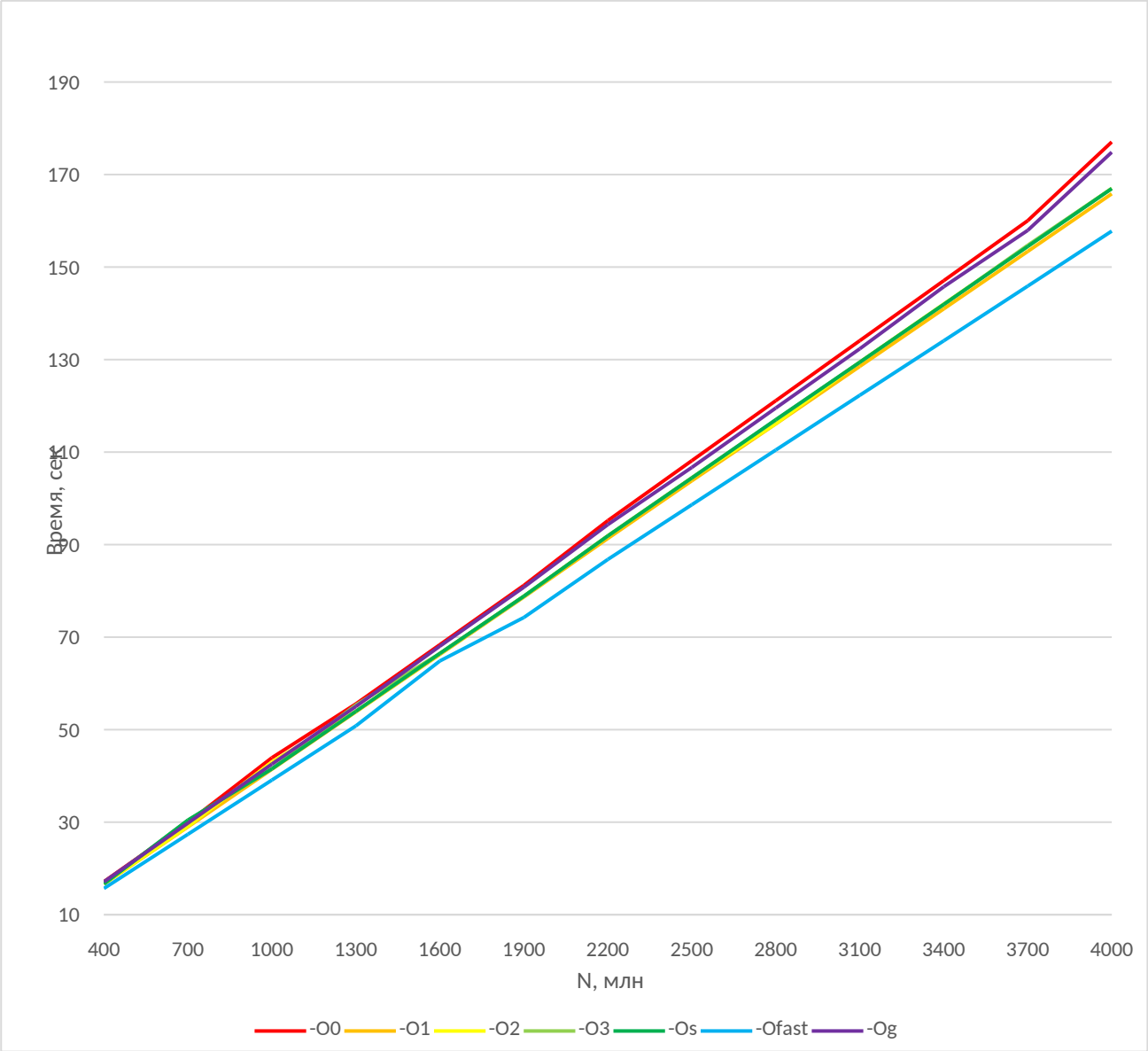
Скрипт принимает на вход 5 параметров: название файла, начальное число, шаг, кол-во шагов, файл вывода

Пример вывода:

```
evmpu@fitnsu-1135-2:~/19205/Borodin/Lab2$ ./measureTimeCSV main_ 25000000 25000000 7 output.csv
  N=25000000: 0 1 2 3 s f g
  N=50000000: 0 1 2 3 s f g
  N=75000000: 0 1 2 3 s f g
  N=100000000: 0 1 2 3 s f g
  N=125000000: 0 1 2 3 s f g
  N=150000000: 0 1 2 3 s f g
  N=175000000: 0 1 2 3 s f g
  DONE
evmpu@fitnsu-1135-2:~/19205/Borodin/Lab2$ cat output.csv
;25000000;50000000;75000000;100000000;125000000;150000000;175000000
'-O0;1,104;2,169;3,245;4,315;5,384;6,461;7,543
'-O1;1,069;2,107;3,15;4,186;5,263;6,259;7,295
'-O2;1,073;2,116;3,156;4,201;5,23;6,272;7,319
'-O3;1,073;2,116;3,16;4,559;5,23;6,273;7,311
'-Os;1,075;2,118;3,159;4,564;5,237;6,271;7,315
'-Of;1,014;1,995;2,975;3,954;4,932;5,906;7,251
'-Og;1,091;2,159;3,229;4,466;5,363;6,422;7,484
evmpu@fitnsu-1135-2:~/19205/Borodin/Lab2$
```

Приложение 2. График

N, млн	-O0	-O1	-O2	-O3	-Os	-Ofast	-Og
400	17,139	16,613	16,655	16,737	16,65	15,677	17,107
700	29,945	28,981	29,089	30,394	30,448	27,389	29,802
1000	43,93	41,44	43,019	41,53	41,535	39,107	42,554
1300	55,568	53,851	53,969	55,342	53,971	50,819	55,013
1600	68,373	66,262	66,445	66,455	66,481	64,852	68,052
1900	81,185	78,669	78,844	78,806	78,858	74,248	80,797
2200	95,189	91,338	91,895	91,895	91,882	86,801	94,332
2500	108,148	103,801	104,41	104,462	104,404	98,628	106,69
2800	121,118	116,099	116,156	116,926	117,012	110,46	119,534
3100	134,091	128,534	129,44	129,435	129,421	122,288	132,336
3400	147,064	140,968	142,018	141,885	141,935	134,067	145,751
3700	160,037	153,388	154,473	154,713	154,455	145,894	157,941
4000	177,045	165,838	166,997	166,985	166,963	157,774	174,829



Приложение 3. Хеш-суммы

```
evmpu@fitnsu-1135-2:~/19205/Borodin/Lab2$ ./compile main.cpp main_
  main_0 compiled with md5: d1101c6cb95bf982e47839bcee33b1ab and file size: 13616
  main_1 compiled with md5: 230f8c705b187de421787cfbf638edfc and file size: 13624
  main_2 compiled with md5: 1f73a2dce7be8cbddc1907c82dba1b7c and file size: 13624
  main_3 compiled with md5: 1f73a2dce7be8cbddc1907c82dba1b7c and file size: 13624
  main_s compiled with md5: 8828781413c049792547bacf9852b50a and file size: 9400
  main_f compiled with md5: 5fd8cb73c714c7d6fbab07b49039ec18 and file size: 15112
  main_g compiled with md5: 70f6533527e1efd3ad0921494857a1d1 and file size: 13696
evmpu@fitnsu-1135-2:~/19205/Borodin/Lab2$
```