###### 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 (ки) 2 курса, 18205 группы

Гайдамака Андрея Владиславовича

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

(ученая степень, звание)

А. Ю. Власенко

Новосибирск 2019

**СОДЕРЖАНИЕ**

ЦЕЛЬ 4

ЗАДАНИЕ 4

ОПИСАНИЕ РАБОТЫ 5

ЗАКЛЮЧЕНИЕ 7

Приложение 1. Код программы epower 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.
4. Измерение времени работы подпрограммы в прикладной программе.

# 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.

**Вариант задания: №5**

# ОПИСАНИЕ РАБОТЫ

Были проведены замеры времени работы программы при разных параметрах оптимизации и значениях n, по этим данным были построены графики

Строки компиляции: gcc main.c -O0 -o opt0.out gcc main.c -O1 -o opt1.out gcc main.c -O2 -o opt2.out gcc main.c -O3 -o opt3.out gcc main.c -Os -o optS.out gcc main.c -Ofast -o optFast.out gcc main.c -Og -o optg.out

# ЗАКЛЮЧЕНИЕ

Время работы программы с флагом компиляции O0 существенно отличается от времени программ с включенной оптимизацией. Для данной программы существенного различия времени выполнения при флагах: O1, O2, O3, Os, Ofast, Og не наблюдалось. По собранным данным можно сказать, что программа скомпилированная с флагом Ofast отрабатывает за наименьшее время на всех n на которых проводились замеры

# Приложение 1. Код программы epower

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <time.h>*

*int factorial(const int n){*

*int res = 1;*

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

*res \*= i;*

*}*

*return res;*

*}*

*int power(int a, int n){*

*if (n == 0)*

*return 1;*

*if (n % 2 == 0)*

*return power(a \* a, n / 2);*

*return a \* power(a, n - 1);*

*}*

*double epower(const long long int n, const long long int x){*

*double epow = 0;*

*double last = 0;*

*double tmp = 0;*

*for(long int i = 0; i <= n; ++i){*

*if (i <= 7){*

*last = (double)power(x, i) / factorial(i);*

*epow += last;*

*}*

*else{*

*last \*= (double)((double)x / i);*

*epow += last;*

*}*

*}*

*return epow;*

*}*

*int main(int argc, char \*\* argv) {*

*if (argc != 3){*

*return 1;*

*}*

*long long int n = atoll(argv[1]);*

*long long int x = atoll(argv[2]);*

*struct timespec start, end;*

*clock\_gettime(CLOCK\_MONOTONIC\_RAW, &start);*

*double epow = epower(n, x);*

*clock\_gettime(CLOCK\_MONOTONIC\_RAW, &end);*

*printf("Time = %lf \n", end.tv\_sec - start.tv\_sec + 0.000000001 \* (end.tv\_nsec - start.tv\_nsec));*

*printf("e^%lld = %.15lf\n", x, epow);*

*return 0;*

*}*