###### 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

«Определение времени работы прикладных программ»

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 3

ЗАДАНИЕ 3

ОПИСАНИЕ РАБОТЫ 4

ЗАКЛЮЧЕНИЕ 5

Приложение 1. Код программы epower 6

# ЦЕЛЬ

1. Изучение методики измерения времени работы подпрограммы.
2. Изучение приемов повышения точности измерения времени работы   
   подпрограммы.
3. Изучение способов измерения времени работы подпрограммы.
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 15 секунд.

4. По приведенной методике определить время работы подпрограммы тестовой программы с относительной погрешностью не более 1%.

**Вариант задания: №5**

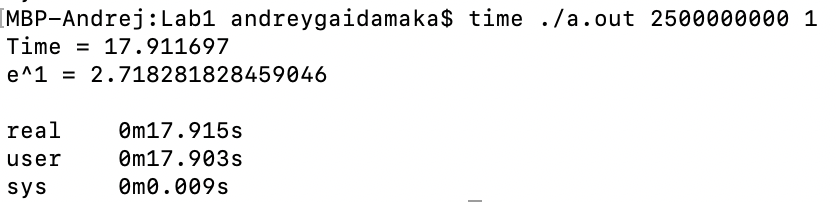
# ОПИСАНИЕ РАБОТЫ

Алгоритм написанный на языке С был проверен на входных данных:

n=5, x = 1; n = 5, x = 2; n = 5, x = 3; После был проведен подбор параметра n для того чтобы время работы программы составляло от 15 до 30 секунд, на моем компьютере данное значение равно 2500000000, при нем время работы программы составило17.915 секунд при измерении при помощи утилиты time и 17.911 при измерении при помощи функции clock\_gettime()

**Выбранная методика измерения:** Утилита time и функция clock\_gettime()

**Строки компиляции:** gcc main.c -o epower



# ЗАКЛЮЧЕНИЕ

Измерение времени выполнения программы необходимо при написании программ, так как анализ времени выполнения помогает в оптимизации программы.

Так же анализ времени выполнения программы помогает во временной оценке работоспособности алгоритма

# 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 = 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;*

*}*