

**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 2 курса, 23209 группы
Инокова Семёна Шухратовича

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:
А. Ю. Кудинов

Новосибирск 2024

СОДЕРЖАНИЕ

<u>ЦЕЛЬ.....</u>	<u>2</u>
<u>ЗАДАНИЕ.....</u>	<u>3</u>
<u>ОПИСАНИЕ РАБОТЫ.....</u>	<u>3</u>
<u>ЗАКЛЮЧЕНИЕ.....</u>	<u>4</u>
<u>Приложение 1. Код программы для работы.....</u>	<u>5</u>
<u>Приложение 2. Время работы программы при $n = 2150000000$.....</u>	<u>6</u>
<u>Приложение 3. Использование команды time для измерения.....</u>	<u>7</u>
<u>Приложение 4. Измерение времени работы при различных n.....</u>	<u>8</u>

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

Написать на языке С или С++ алгоритм вычисления числа Π с помощью разложения в ряд (ряд Грегори-Лейбница) по формуле Лейбница N первых членов ряда:

ОПИСАНИЕ РАБОТЫ

1. Для написания кода, который реализует алгоритм вычисления числа π был выбран язык C.
2. Для измерения времени работы программы была выбрана Библиотечная функция `clock_gettime`, так как этот метод доступный и стабильный в наших условиях.
3. Произвёл замеры времени при различных n , нашёл n , при котором время составляет около 15 секунд (2.150.000.000).
4. Использовал команду `time` для измерения времени работы программы.

ЗАКЛЮЧЕНИЕ

В ходе данной работы я познакомился с различными методами измерения времени работы программы и использовал их на практике. Я понял, что для точного замера нужно учитывать множество факторов.

Приложение 1. Код программы для работы

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <time.h>
#include <math.h>

double piCalculation(long long n) {
    double pi = 4;

    for (long long i = 1; i <= n; ++i) {
        double x = 4. / (2*i + 1);
        if (i % 2) {
            pi -= x;
        } else {
            pi += x;
        }
    }

    return pi;
}

int main(int argc, char *argv[]) {
    struct timespec start, end;
    long long n = strtoll(argv[1], NULL, 10);

    clock_gettime(CLOCK_MONOTONIC_RAW, &start);
    double pi = piCalculation(n);
    clock_gettime(CLOCK_MONOTONIC_RAW, &end);

    printf("Pi number: %.10lf\n", pi);
    printf("Time taken: %lf sec.\n", end.tv_sec-start.tv_sec + 0.000000001*(end.tv_nsec-start.tv_nsec));

    return 0;
}
```

Приложение 2. Время работы программы при $n = 2150000000$

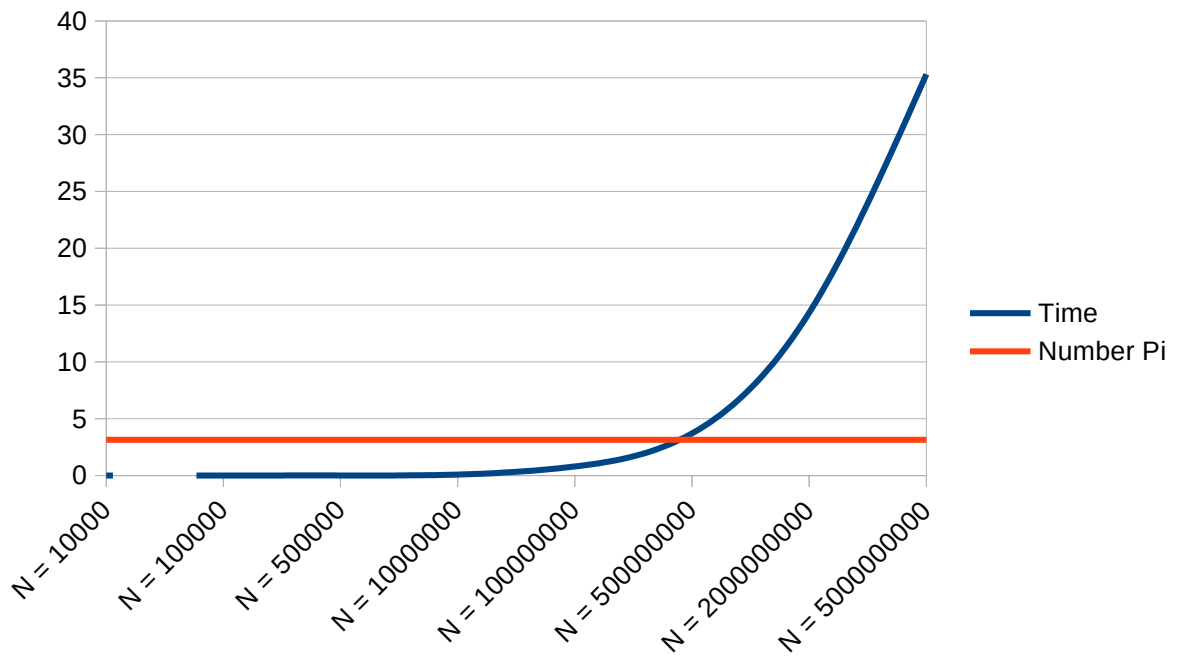
```
evm116@comrade:~/lab2$ ./main 2150000000
Pi number: 3.1415926541
Time taken: 15.117425 sec.
evm116@comrade:~/lab2$ ./main 2150000000
Pi number: 3.1415926541
Time taken: 15.355990 sec.
```

Приложение 3. *Использование команды `time` для измерения*

```
evm116@comrade:~/lab2$ time ./main 2150000000
Pi number: 3.1415926541
Time taken: 15.098429 sec.

real    0m15,101s
user    0m15,093s
sys     0m0,004s
```

Приложение 4. *Измерение времени работы при различных n*



N	Time	Number Pi
N = 10000	0.000129	3.1416926436
N = 100000	0.001381	3.1416026535
N = 500000	0.006300	3.1415946536
N = 10000000	0.084540	3.1415927536
N = 100000000	0.793765	3.1415926636
N = 500000000	3.707707	3.1415926556
N = 2000000000	14.317186	3.1415926541
N = 5000000000	35.302306	3.1415926538