Министерство образования и науки Российской Федерации

Новосибирский национальный исследовательский государственный университет

ЭВМ и периферийные устройства

Отчёт по лабораторной работе № 1

«Определение времени работы прикладных программ»

Студент: Добровлянина Елизавета Максимовна, 24205

Преподаватель: Мичуров Михаил Антонович

Новосибирск, 2025г.

**ЦЕЛИ**

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.

**ЗАДАЧИ**

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

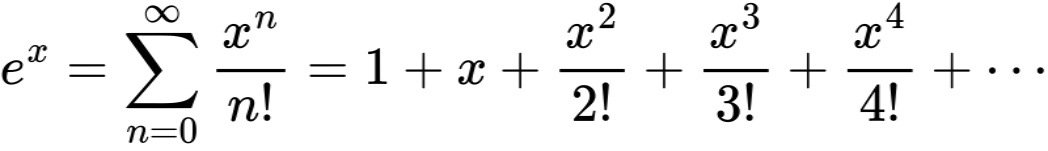
тестовой программы с относительной погрешностью не более 1%.

1. Составить отчет по лабораторной работе.

**ОПИСАНИЕ РАБОТЫ**

В ходе выполнения лабораторной работы был выбран 5 вариант задания:

*Алгоритм вычисления функции ex с помощью разложения в ряд Маклорена по первым N членам этого ряда:*

**

Программа, реализующая выбранный алгоритм, была написана на языке программирования С (см. Приложение).

Была проверена правильность работы программы: алгоритм отработал работал корректно. Уже при N=100 программа вывела корректно 15 знаков после запятой. Ниже прилагается таблица тестовых данных и вывода программы.

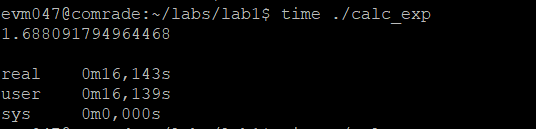
|  |  |
| --- | --- |
| Ввод | Вывод |
| N=10 | 1.688091794516430 |
| N=100 | 1.688091794964468 |
| N=1000 | 1.688091794964468 |

Для измерения времени работы программы была использована утилита time.

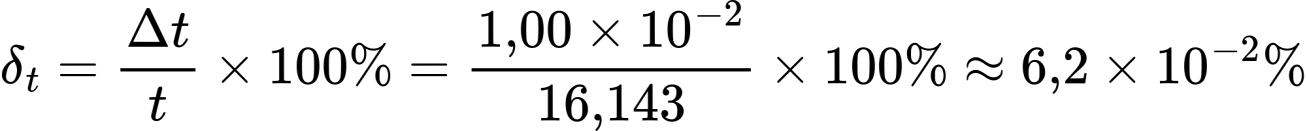
Утилита time измеряет время работы приложения во многих конфигурациях ОС GNU Linux/UNIX. Утилита time выдаёт следующие временные характеристики работы программы: real – общее время работы программы согласно системному таймеру, user – время, которое работал пользовательский процесс (кроме времени работы других процессов) и sys – время, затраченное на выполнение системных вызовов программы.

Путем перебора входных данных было выбрано значение N такое, чтобы время работы программы было более 15с.: N = 2000000000.

Ниже прилагается снимок экрана:



Относительная погрешность была получена по формуле:



Полученный результат удовлетворяет требованиям задания.

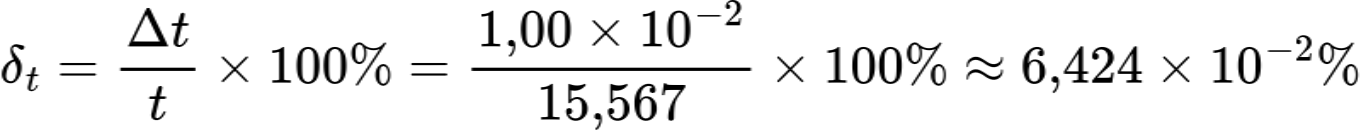
**Для сокращения времени работы программы были опробованы различные методы: м**ногократное измерение, уменьшение влияния посторонних процессов, сброс буфера отложенной записи на диск.

Ниже прилагается таблица полученных результатов:

|  |  |
| --- | --- |
| Прием для уточнения времени работы программы | Результат измерения |
| Многократное измерение (наименьшее значение) | real 0m16,067s  user 0m16,059s  sys 0m0,004s |
| Уменьшение влияния посторонних процессов | real 0m15,616s  user 0m15,612s  sys 0m0,000s |
| Сброс буфера отложенной записи на диск | real 0m15,567s  user 0m15,563s  sys 0m0,000s |

Из полученных измерений наиболее точным будет минимальное. Это связано с тем, что сторонние факторы вносят различный вклад в измеряемый интервал при каждом запуске. При этом время работы самой программы остается неизменным.

Новая относительная погрешность:



**ПРИЛОЖЕНИЕ**

Ссылка на репозиторий с кодом: <https://github.com/dem24205/evm_labs/tree/master/lab1>

Команда компиляции:

gcc -o calc\_exp main.c -lm -O0 -Wall -Wextra -Wpedantic -Werror

*Код программы:*

#include "math.h"

#include <stdio.h>

#define ARG M\_PI/6.0

double calc\_exp(double arg, int accur) {

    double result = 1.0;

    double term = 1.0;

    for (int n = 0; n < accur-1; ++n) {

        term \*= arg/(n+1);

        result += term;

    }

    return result;

}

int main(void)

{

    int N = 2000000000;

    double result = calc\_exp(ARG, N);

    printf("%.15f\n", result);

    return 0;

}