

Новосибирский Государственный Университет

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

Курс “ЭВМ и периферийные устройства”

Лабораторная работа №1

«ОПРЕДЕЛЕНИЕ ВРЕМЕНИ РАБОТЫ ПРИКЛАДНЫХ ПРОГРАММ»

Вариант №5

Выполнил:

Пятаев Егор, гр. 15206

Преподаватель:

Городничев Максим Александрович

Новосибирск 2016

Цели работы

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.

Вариант задания

Алгоритм вычисления функции e^x с помощью разложения в ряд Маклорена по первым N членам этого ряда.

Методика определения времени работы программы

Для определения времени работы программы была использована библиотечная функция *clock_gettime* т. к. степень загрузки процессора была не высока и точность системного таймера в ОС GNU Linux: 1 нс.

Результат измерения времени работы программы

Время работы: 15.106 секунд ($N = 3.7 \cdot 10^9$).

Точность: 1 нс.

Относительная погрешность: $(1 \text{ нс} / 15.106) \cdot 100 \ll 1\%$.

Листинг реализованной программы

header.h:

```
#ifndef H_1
#define H_1
#define BAD_ARGS 1
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
```

```
double calc_ex(double, double);
```

```
#endif
```

calc_ex.c:

```
#include "header.h"
double calc_ex(double n, double x) {

    double ex = 1;
    double i;
    double j = 1;

    /*Calculate ex*/
    if (x != 0) {
        for (i = 1; i < n; i++){
            j*=(x/i);
            ex+=j;
        }
    } else return ex;

    return ex;
}
```

main.c:

```
#include "header.h"
```

```
int main(int argc, char *argv[]) {  
    double ex;  
    double n;  
    double x;  
    struct timespec start, end;
```

```
    if (argc == 3 ) {
```

```
        n = atof(argv[1]);  
        x = atof(argv[2]);
```

```
        if(n <= 0){  
            printf("Enter N > 0");  
            return BAD_ARGS;  
        }
```

```
        clock_gettime(CLOCK_MONOTONIC_RAW, &start);
```

```
        ex = calc_ex(n, x);
```

```
        clock_gettime(CLOCK_MONOTONIC_RAW, &end);
```

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

```
        printf("e^x = %.10f\n", ex);
```

```
    } else printf("Bad arguments");
```

```
    return 0;
```

```
}
```

Команда компиляции: gcc main.c calc.c -o calc [-lrt]

Команда запуска: ./calc [значение N] [значение X]

Выводы

Для достижения поставленных целей написана программа с алгоритмом вычисления функции e^x с помощью разложения в ряд Маклорена по первым N членам этого ряда.

Были изучены методики, приемы повышения точности и способы измерения времени работы подпрограммы и выбрана подходящая методика. Для изменения времени работы программы, использована библиотечная функция *clock_gettime*, получающая значения системного таймера.

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