

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №8
по дисциплине «Организация ЭВМ и систем»
Тема: Обработка вещественных чисел. Программирование
математического сопроцессора.
Вариант 2

Студент гр. 1303

Кузнецов Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Составить программу для освоения навыков работы с математическим сопроцессором на языке ассемблера, вычисляющая значение гиперболического косинуса.

Задание.

Разработать подпрограмму на языке Ассемблера, обеспечивающую вычисление заданной математической функции с использованием математического сопроцессора. Подпрограмма должна вызываться из головной программы, разработанной на языке С. При этом должны быть обеспечены заданный способ вызова и обмен параметрами. Альтернативный вариант реализации: разработать на языке Ассемблера фрагмент программы, обеспечивающий вычисление заданной математической функции с использованием математического сопроцессора, который включается по принципу inline в программу, разработанную на языке С.

ВАРИАНТ 2.

* function

Name cosh - hyperbolic function:

Usage double cosh(double x);

Prototype in math.h

Description cosh computes the hyperbolic cosine of the input value.

$$\cosh(x) = (\exp^x + \exp^{-x}) / 2$$

Выполнение работы.

Программа начинается с ввода значения x , для которого нужно рассчитать значение функции $\cosh(x)$, а так же создается переменная в которую будет записано значение экспоненты. Для возведения экспоненты в степень x , применяется следующее математическое преобразование, которое будет применено в ассемблерной вставке: $a^b == 2^{(b \cdot \log_2(a))}$. В соответствии с этим ассемблерный блок разбивается на подзадачи.

Возведение экспоненты в степень. Получение из этого $e^{(-x)}$ путем работы с регистрами. Суммирование полученных e^x и $e^{(-x)}$. И деление суммы на два. Полученный результат выводится в консоль, а так же выводится абсолютная погрешность вычислений.

Тестирование.

№ Теста	Ввод	Вывод	Результат
1	3.14	Вычисленное значение $\cosh(x)$: 11.573575 Абсолютная погрешность вычисления: 0.000000000000000177636	Верно
2	-2,71	Вычисленное значение $\cosh(x)$: 7.547906 Абсолютная погрешность вычисления: 0.000000000000000177636	Верно
3	0.691311	Вычисленное значение $\cosh(x)$: 1.248625 Абсолютная погрешность вычисления: 0.00000000000000022204	Верно

Таблица фиксации изменений стека и регистров при $x=3$.

Команда	До выполнения	После выполнения
fld qword ptr[x]	ST0 = +0.0000000000000000e+0000	ST0 = +3.0000000000000000e+0000
fld qword ptr[e]	ST0 = +3.0000000000000000e+0000 ST1 = +0.0000000000000000e+0000	ST0 = +2.7182818284590450e+0000 ST1 = +3.0000000000000000e+0000
fyl2x	ST0 = +2.7182818284590450e+0000	ST0 = +4.3280851226668899e+0000
fld st	ST0 = +4.3280851226668899e+0000 ST1 = +0.0000000000000000e+0000	ST0 = +4.3280851226668899e+0000 ST1 = +4.3280851226668899e+0000
frndint	ST0 = +4.3280851226668899e+0000	ST0 = +4.0000000000000000e+0000
fsub st(1), st	ST1 = +4.3280851226668899e+0000	ST1 = +3.2808512266688999e-0001
fxch st(1)	ST0 = +4.0000000000000000e+0000 ST1 = +3.2808512266688999e-0001	ST0 = +3.2808512266688999e-0001 ST1 = +4.0000000000000000e+0000
f2xm1	ST0 = +3.2808512266688999e-0001	ST0 = +2.5534605769922903e-0001
fldl	ST0 = +2.5534605769922903e-0001 ST1 = +4.0000000000000000e+0000	ST0 = +1.0000000000000000e+0000 ST1 = +2.5534605769922903e-0001

	ST2 = +0.0000000000000000e+0000	ST2 = +4.0000000000000000e+0000
faddp st(1), st	ST0 = +1.0000000000000000e+0000 ST1 = +2.5534605769922903e-0001 ST2 = +4.0000000000000000e+0000	ST0 = +1.2553460576992290e+0000 ST1 = +4.0000000000000000e+0000 ST2 = +0.0000000000000000e+0000
fscale	ST0 = +1.2553460576992290e+0000	ST0 = +2.0085536923187664e+0001
fstp st(1)	ST1 = +4.0000000000000000e+0000	ST1 = +0.0000000000000000e+0000
fst qword ptr[res]	-	-
fldl	ST0 = +2.0085536923187664e+0001 ST1 = +0.0000000000000000e+0000	ST0 = +1.0000000000000000e+0000 ST1 = +2.0085536923187664e+0001
fddiv qword ptr[res]	ST0 = +1.0000000000000000e+0000	ST0 = +4.9787068367863951e-0002
fadd st, st(1)	ST0 = +4.9787068367863951e-0002	ST0 = +2.0135323991555527e+0001
fddiv qword ptr[two]	ST0 = +2.0135323991555527e+0001	ST0 = +1.0067661995777763e+0001
fstp qword ptr[res]	ST0 = +1.0067661995777763e+0001 ST1 = +2.0085536923187664e+0001	ST0 = +2.0085536923187664e+0001 ST1 = +0.0000000000000000e+0000

Вывод.

Составлена программа для вычисления $\cosh(x)$, а так же освоены навыки работы с математическим сопроцессором ассемблера.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab8.cpp

```
#include <math.h>
#include <iostream>

long double x;
long double two = 2;
long double e = exp(1);
long double res;

int main() {
    system("chcp 1251 > nul");
    setlocale(LC_CTYPE, "rus");

    std::cout << "Введите значение x...\n";
    std::cin >> x;
    __asm{

        ; Возведение e^x
        fld qword ptr[x]
        fld qword ptr[e]
        fyl2x
        fld st
        frndint
        fsub st(1), st
        fxch st(1)
        f2xm1
        fld1
        faddp st(1), st
        fscale
        fstp st(1)
        fst qword ptr[res]

        ; Деление 1 / e^x
        fld1
        fdiv qword ptr[res]

        ; Получение e^x + 1 / (e^x)
        fadd st, st(1)

        ; Получение cosh(x)
        fdiv qword ptr[two]

        ; Запись ответа
        fstp qword ptr[res]
    }
    printf("Вычисленное значение cosh(x): %lf\n", res);
    printf("Абсолютная погрешность вычисления: %.20lf\n", abs(res -
cosh(x)));
    return 0;
}
```