

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

Студентка гр. 1381
Преподаватель

Новак П.И.
Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Разработать подпрограмму на языке Ассемблера, обеспечивающую вычисление заданной математической функции с использованием математического сопроцессора.

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.

Пользователю предлагается ввод значения x , для которого нужно рассчитать значение функции $\cosh(x)$, создаются глобальные переменные для записи результата, значения экспоненты и константа 2 для работы с функцией. Для совершения операции существует функция \cosH , подсчитывающая значение гиперболического косинуса. Для возведения экспоненты в степень x , применяется 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. Полученный результат выводится в консоль.

Тестирование.

№ Теста	Ввод	Вывод	Результат
1	12	Вычисленное значение $\cosh(x)$: 81377.395713	Верно
2	-8,6	Вычисленное значение $\cosh(x)$: 2715.829888	Верно
3	0.412	Вычисленное значение $\cosh(x)$: 1.086079	Верно

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

Команда	До выполнения	После выполнения
fld x	ST0 = +0.0000000000000000e+0000	ST0 = +3.0000000000000000e+0000
fld 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
fld1	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
fdiv res	ST0 = +1.0000000000000000e+0000	ST0 = +4.9787068367863951e-0002
fadd st, st(1)	ST0 = +4.9787068367863951e-0002	ST0 = +2.0135323991555527e+0001
fdiv two	ST0 = +2.0135323991555527e+0001	ST0 = +1.0067661995777763e+0001
fstp res	ST0 = +1.0067661995777763e+0001 ST1 = +2.0085536923187664e+0001	ST0 = +2.0085536923187664e+0001 ST1 = +0.0000000000000000e+0000

Вывод.

Получены навыки работы со специальными инструкциями Ассемблера для чисел с плавающей запятой. Разработана программа на ЯВУ, которая вычисляет значение гиперболического косинуса.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab8.cpp

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

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

long double cosH(long double result1) {
    __asm {

; Возведение e^x
fld x                //загрузка переменных в стек st
fld e
fyl2x               //ST(1) = ST(1) * log2(ST(0))
fld st              //Загрузить ST(0) в регистровый стек
frndint             //Округлить ST(0) к целому
fsub st(1), st      //ST(1) = ST(1) - ST(0)
fxch st(1)          //Обмен значений ST(0) и ST(1)
f2xm1               //ST(0) = 2^ST(0) - 1
fld1                //Загрузить +1.0 в регистровый стек
faddp st(1), st     //ST(1) = ST(0) + ST(1)
fscale              //ST(0) = ST(0) * 2^ST(1)
fstp st(1)          //Сохранить ST(0) в ST(1)
fst result1

; Деление 1 / e^x
fld1                //Загрузить +1.0 в регистровый стек
fdiv result1

fadd st, st(1)      //Получение e^x + 1 / (e^x)
fdiv two            //Получение cosh(x)
```

```

fstp result1                //ответ
}

return result1;
}

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

std::cout << "Вычисление cosh(x). Введите значение x:\n";
std::cin >> x;

result = cosH(result);

printf("Вычисленное значение cosh(x): %lf\n", result);
return 0;
}

```