

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

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

Студентка гр. 9382

Пя С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Научиться работать с математическим сопроцессором. Разработать программу на языке Ассемблера, обеспечивающую вычисление заданной математической функции.

Основные теоретические положения.

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.

Задание:

function

Name Acos - compute acos

Usage double Acos (double *xP);

Prototype in math.h

Description Computes acos of the number pointed to by xP.

Arguments to acos must be in the range -1 to 1, acos returns a value in the range 0 to pi.

Use the trig identities $\text{acos}(x) = \text{atan}(\sqrt{1-x^2} / x) *$

Ход работы:

В качестве ЯВУ используется C/C++. Была реализована функция acos, содержащая в себе ассемблерную вставку, которая отвечает за вычисление арккосинуса. Данные вычисления производились с помощью таких операций, как сложение, вычитание, деление, умножение, взятие корня, взятие арктангенса. Также был предусмотрен случай с отрицательным значением.

Тестирование.

№	Входные данные	Выходные данные
1.	0	acos(x) of math.h: 1.5707963267949 acos(x) of assembler: 1.5707963267949
2.	1	acos(x) of math.h: 0 acos(x) of assembler: 0
3.	-1	acos(x) of math.h: 3.14159265358979 acos(x) of assembler: 3.14159265358979
4.	-0.5	acos(x) of math.h: 2.0943951023932 acos(x) of assembler: 2.0943951023932

Выводы.

В результате выполнения лабораторной работы был разработан код, состоящий из ассемблерного модуля и остальной части на ЯВУ C++, который вычисляет арккосинус.

Приложение.

Текст файла Source.cpp

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

using namespace std;
/* var6
function
Name Acos - compute acos
Usage double Acos (double *xP);
Prototype in math.h
Description Computes acos of the number pointed to by xP.
Arguments to acos must be in the range -1 to 1, acos returns a value in the range 0 to
pi.
Use the trig identities  $\text{acos}(x) = \text{atan}(\sqrt{1-x^2} / x) *$  */

double acosAsm(double* xP) {
    double x = *xP;
    double y = -1;
    _asm {
        fld x; // кладем в стек x
        fld x;
        fmul; // переумножаем два первых значения в стеке
        fld1; // кладем единицу в стек
        fxch st(1); // меняем местами элементы стека st(0) и st(1)
        fsub; // вычитаем из 1ого 0ый элемент стека и кладем в 0 элемент
        fsqrt; // вычисляем корень 0ого элемента стека
        fld x;
        fdiv; // делим 1ый элемент стека на 0ой и кладем в стек
        fld1;
        fpatan; // вычисляем арктангенс от числа, образованного делением 1 элемента на 0
элемент стека
        fstp y; // кладем элемент 0 в переменную y
        // определяем дальнейшие действия в случае отрицательного или положительного
значения
        fldz; // кладем в стек число 0
        fld x;
        fcom; // сравниваем два элемента из стека
        fstsw ax; // берем результат из сопроцессора
        sahf; // загружаем ax в регистр флагов
        jae finish; // в случае, если x >= 0
        fld y;
        fldpi; // кладем в стек константу пи
        fadd; // складываем два элемента из стека и кладем в 0 элемент
        fstp y;
    finish:
    }
    return y;
}

int main()
{
    double x;
    cout << "Enter x:\n";
    cin >> x;
    while ((x < -1) || (x > 1)) {
        cout << "Data must be at range of -1 to 1.\n" << endl;
        cout << "Enter x again:\n";
    }
}
```

```
        cin >> x;
    }
    cout << "acos(x) of math.h: " << setprecision(15) << acos(x) << endl;
    cout << "acos(x) of assembler: " << setprecision(15) << acosAsm(&x) << endl;
    return 0;
}
```