

**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**

Студент гр. 9382

\_\_\_\_\_

Субботин М.О.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Научиться обрабатывать вещественные числа. Также научиться программировать математический сопроцессор.

### **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 in-line в программу, разработанную на языке С.

### **Ход выполнения:**

#### **ВАРИАНТ 6.**

\* function

Name Usage Prototype in Description

Acos - compute acos double Acos (double \*xP); math.h

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

В программе реализована функция Acos, в которой находится ассемблерная inline вставка, отвечающая за вычисление арккосинуса. Работая со стеком и применяя операции умножения, сложения, вычитания, деления, взятия корня, получается значение  $\sqrt{1-x^2}/x$ . Затем с помощью оператора fpatan берется арктангенс  $\arctan(\sqrt{1-x^2}/x)$ , то, что и требуется по заданию. Только если изначально введенное значение было отрицательное, то надо к получившемуся значению еще прибавить  $\pi$ .

### Тестирование.

№	Входные данные	Выходные данные
1	-1	acos(x) из библиотеки math.h: 3.14159265358979 Acos(x) ассемблерный: 3.14159265358979
2	1	acos(x) из библиотеки math.h: 0 Acos(x) ассемблерный: 0
3	0.1435	acos(x) из библиотеки math.h: 1.42679920834195 Acos(x) ассемблерный: 1.42679920834195
4	0.68923	acos(x) из библиотеки math.h: 0.810370551392384 Acos(x) ассемблерный: 0.810370551392384
5	-0.86723	acos(x) из библиотеки math.h: 2.62040812045222 Acos(x) ассемблерный: 2.62040812045222
6	-0.111	acos(x) из библиотеки math.h: 1.68202553843552 Acos(x) ассемблерный: 1.68202553843552

### Выводы.

Была изучена обработка вещественных чисел. Также было изучено программирование математического сопроцессора.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД

```
#include <math.h>
#include <iostream>
#include <iomanip>
using namespace std;

/* 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 Acos(double* xP)
{
    double x = *xP;
    double y = -1;
    _asm {
        fld x;          // вставляем в стек x, stack: 0)x
        fld x;          // вставляем в стек x, stack: 0)x , 1)x
        fmul;           // перемножает два первых значения в стеке, stack:
0)x^2
        fld 1;          // вставляем единицу в стек, stack: 0)1 , 1)x^2
        fxch st(1);     // меняем местами элементы стека 0 и 1, stack: 0)x^2, 1)1
        fsub;           // вычитаем из 1, 0 элемент стека и вставляем в 0
элемент, stack: 0) 1-x^2
        fsqrt;          // вычисляем корень у 0 элемента стека и вставляем в 0
элемент, stack: 0) sqrt(1-x^2)
        fld x;          // вставляем в стек x, stack: 0)x , 1)sqrt(1-x^2)
        fdiv;           // делим 1 элемент стека на нулевой и вставляем в стек,
stack: 0) sqrt(1-x^2)/x
        fld 1;          // вставляем 1 в стек, stack: 0)1 , 1)sqrt(1-x^2)/x
        fpatan;         // считаем арктангенс от числа, образованного
                        // делением 1 элемента на 0 элемент стека, stack:
0) arctan(sqrt(1-x^2)/x)
        fstp y;         // достаем из стека элемент 0 в переменную y.

        //надо рассмотреть случай с отрицательным значением

        fldz;          //вставляем в стек число 0, stack: 0) 0
        fld x;          //вставляем в стек переменную x, stack: 0)x, 1)0
    }
```

```

        fcom;          //сравниваем два элемента из стека
        fstsw ax;      //забираем результат из сопроцессора
        sahf           //загружаем ax в регистр флагов
                jae to_end; //если x >= 0
        fld y;         //записываем в стек переменную y, stack 0)y
        fldpi;         //записываем в стек константу пи, stack: 0)pi, 1)y
        fadd;          //складываем два элемента из стека и кладем в 0
элемент, stack: 0) pi+y
        fstp y;        //кладем элемент из стека в y
to_end:
    }
    return y;
}

```

```

int main()
{
    system("chcp 1251 > nul");
    double x;
    cout << "Введите x: ";
    cin >> x;
    while (x < -1 || x>1) {
        cout << "Аргумент должен быть в пределах от -1 до 1!" << endl;
        cout << "Введите x: ";
        cin >> x;
    }
    cout << "acos(x) из библиотеки math.h: " << setprecision(15) << acos(x) <<
endl;
    cout << "Acos(x) ассемблерный: " << setprecision(15) << Acos(&x) << endl;
    return 0;
}

```