

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

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

Студент гр. 1381

Мелькумянц Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

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

Задание.

Подпрограмма должна вызываться из головной программы. При этом должны быть обеспечены заданный способ вызова и обмен параметрами.

Выполнить трансляцию программы с подготовкой ее ассемблерной версии и отладочной информации. Для выбранного контрольного набора исходных данных прогнать программу под управлением отладчика. При этом для каждой команды сопроцессора следует фиксировать содержимое используемых ячеек памяти, регистров ЦП и численных регистров сопроцессора до и после выполнения этой команды. Проверить корректность выполнения вычислений для нескольких наборов исходных данных.

Выполнение работы.

С помощью головной программы производится считывание исходных данных. После ввода данных в переменную `res` запишется результат функции `POLY`, где описан ассемблерный блок. В стек математического сопроцессора записывает `x`, далее в `esi` заносится массив констант, в `edi` заносится количество констант. После чего заносится в стек математического сопроцессора заносится 0.0. Происходит проверка на нулевое количество констант. Далее в цикле `horner` верхний элемент стека домножаем на второй элемент стека, а потом добавляем константу к верхушки стека. После окончания цикла в переменную `y` записывается значение верхушки стека и возвращается значение функции.

Входные данные: $X = 1.432$, $n = 3$, 1.942, 3.421, 0.424

Таблица 1 – Результат прогона программы main в отладчике, начиная с момента вызова ассемблерного блока.

Символический код команды	Содержимое регистров и ячеек памяти	
	До выполнения	После выполнения
Fld x	eip = 00152435 st0 = 0 stat = 0100 tags = ffff	eip = 00152438 st0 = 1.4319999999999999 stat = 3900 tags = 3fff
Mov esi, con	eip = 00152438 esi = 004FFC58	eip = 0015243B esi = 00C93FD8
Mov edi, n	eip = 0015243B edi = 00C93FD8	eip = 0015243E edi = 00000003
fldz	eip = 0015243E st0 = 1.4319999999999999 st1 = 0 stat = 3900 tags = 3fff	eip = 00152440 st0 = 0 st1 = 1.4319999999999999 stat = 3100 tags = 1fff
test edi,edi	eip = 00152440 PE = 1 PL = 0	eip = 00152442 PE = 1 PL = 1
je c_end	eip = 00152442	eip = 00152444
mov ecx,edi	eip = 00152444 ecx = 00000000	eip = 00152446 ecx = 00000003
fmul st,st(1)	eip = 00152446	eip = 0015244C

fadd qword ptr [ecx*8 + esi-8]	eip = 0015244C st0 = 4.2399999999999998 tags = 1fff	eip = 0015244C st0 = 4.2399999999999998 tags = 0fff
loop horner	eip = 0015244C rcx = 00000003	eip = 00152446 ecx = 00000002
fmul st,st(1)	eip = 00152446 st0 = 4.2399999999999998 stat = 3100	eip = 00152448 st0 = 6.0716799999999993 stat = 3120
fadd qword ptr [ecx*8 + esi-8]	eip = 00152448 st0 = 6.0716799999999993 stat = 3120	eip = 0015244C st0 = 4.0281679999999999 stat = 3320
loop horner	eip = 0015244C ecx = 00000002	eip = 00152446 ecx = 00000001
fmul st,st(1)	eip = 00152446 st0 = 4.0281679999999999 stat = 3320	eip = 00152448 st0 = 5.7683365759999993 stat = 3120
fadd qword ptr [ecx*8 + esi-8]	eip = 00152448 st0 = 5.7683365759999993 stat = 3120	eip = 0015244C st0 = 7.7103365759999995 stat = 3320
loop horner	eip = 0015244C ecx = 00000001	eip = 0015244E ecx = 00000000
fstp y	eip = 0015244E st0 = 7.7103365759999995	Eip = 00152451 st0 = 1.4319999999999999

	st1 = 1.4319999999999999 stat = 3320 tags = 0fff	st1 = 0 st7 = 7.7103365759999995 stat = 3920 tags = 3fff
--	--	---

Вывод.

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

Приложение А

Исходный код программы

Название файла: Source.cpp

```
#include <iostream>
#include <fstream>
#include <random>

double POLY(double x, int n, double* con){
    double y;

    _asm {
        fld x;
        mov esi, con
        mov edi, n
        fldz
        test edi, edi

        jz c_end
        mov ecx, edi
        horner:
            fmul st, st(1)
            fadd qword ptr[ecx * 8 + esi - 8]
            loop horner
        c_end :
            fstp y;
    }
    return y;
}

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

    double x;
    std::cout << "Введите x: ";
    std::cin >> x;

    int n;
    std::cout << "Введите количество констант: ";
    std::cin >> n;

    double* con = new double[n];
    double s;

    std::cout << "Введите константы: ";
    for (int i = 0; i < n; i++) {
        std::cin >> con[i];
    }
}
```

```
//double res = POLY(x, n, con);  
double res = POLY(x, n, con);  
  
std::cout << "Результат: " << res;  
  
delete[] con;
```

```
}
```