МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

# Кафедра проектирования информационно-компьютерных систем

Отчет по предмету

Программирование на языке ассемблера

Лабораторная работа №6

«Интерфейс с языками высокого уровня. Работа с математическим сопроцессором.»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группы 130501  Гнездилов А.М. | **Проверил:**  Туровец Н.О. |

Минск 2022

**Цель работы:** Ознакомиться с вариантами внедрения ассемблерной процедуры в программу, написанную на языке программирования C\C++, изучить архитектуру математического сопроцессора и команды работы с ним.

**Задание :** Ввести массив чисел с плавающей точкой на 10 элементов. Вычислить сумму элементов массива.

**Теоретические сведения:**

Написание программы полностью на языке ассемблера допустимо только для небольших программ. На практике используют совмещенные варианты создания программ, которые требуют сочетания ассемблера и более высоких языков программирования:

-- основная часть программы пишется на языке высокого уровня, а на ассемблере пишутся отдельные процедуры, которые должны осуществлять управление нижнего уровня и(или) иметь высокую производительность;

-- ассемблерная программа использует библиотечные средства языков высокого уровня. В данной лабораторной работе выполняется создание основной программы на языке С\С++, а часть связанная с вычислениями на математическом сопроцессоре лежит на ассемблерной процедуре. Для выполнения работы требуется рассмотреть следующие элементы языка ассемблера и операционной системы: 1. Соглашения об объединении программных модулей. Связь ассемблерных модулей с языками высокого уровня требует следующих соглашений, которые сильно зависят от применяемых компиляторов и операционной системы: -- Согласование вызовов. Вызов процедуры и возврат из нее в головную программу должны быть согласованы друг с другом. В DOS вызываемая процедура может находиться: - в том же сегменте, что и команда вызова, при этом вызов называется близким или внутрисегментным (NEAR), адрес возврата занимает слово и возврат из процедуры должен быть тоже близким (RETN), - в другом сегменте, тогда вызов называется дальним или межсегментным (FAR), адрес возврата занимает двойное слово и возврат из процедуры должен быть тоже дальним (RETF). Поэтому при объединении программных модулей, написанных на языках С и ассемблера, эти модули должны использовать одну и ту же модель памяти. В Windows используется односегментная модель памяти FLAT, в которой все вызовы по типу являются близкими и согласование вызовов упрощается. -- Согласование имен.

**Код программы:**

#include <iostream>

#include <cmath>

using namespace std;

float func(float\* arr) {

\_asm {

mov ecx, 10

shr ecx, 1

xor ebx, ebx

lp :

fld dword ptr[esi + 4 \* ebx]

mov eax, 9

sub eax, ebx

fld dword ptr[esi + 4 \* eax]

faddp st(1), st(0)

fst dword ptr[esi + 4 \* ebx]

fstp dword ptr[esi + 4 \* eax]

loop lp

}

}

//Для каждого элемента массива вычислить: Х(i) = X(i) + X(N-i), где N – максимальный индекс элементов массива

int main() {

setlocale(LC\_ALL, "");

float arr[10];

float funcarr[10];

cout << "Введите элементы массива\n";

for (int i = 0; i < 10; i++) {

cout << "array[" << i << "]" << " = ";

cin >> arr[i];

}

cout << "Введенный массив: " << endl;

for (int i = 0; i < 10; i++) {

cout << arr[i] << ' ';

}

func(arr);

cout << endl << "Массив с вычислениями: " << endl;

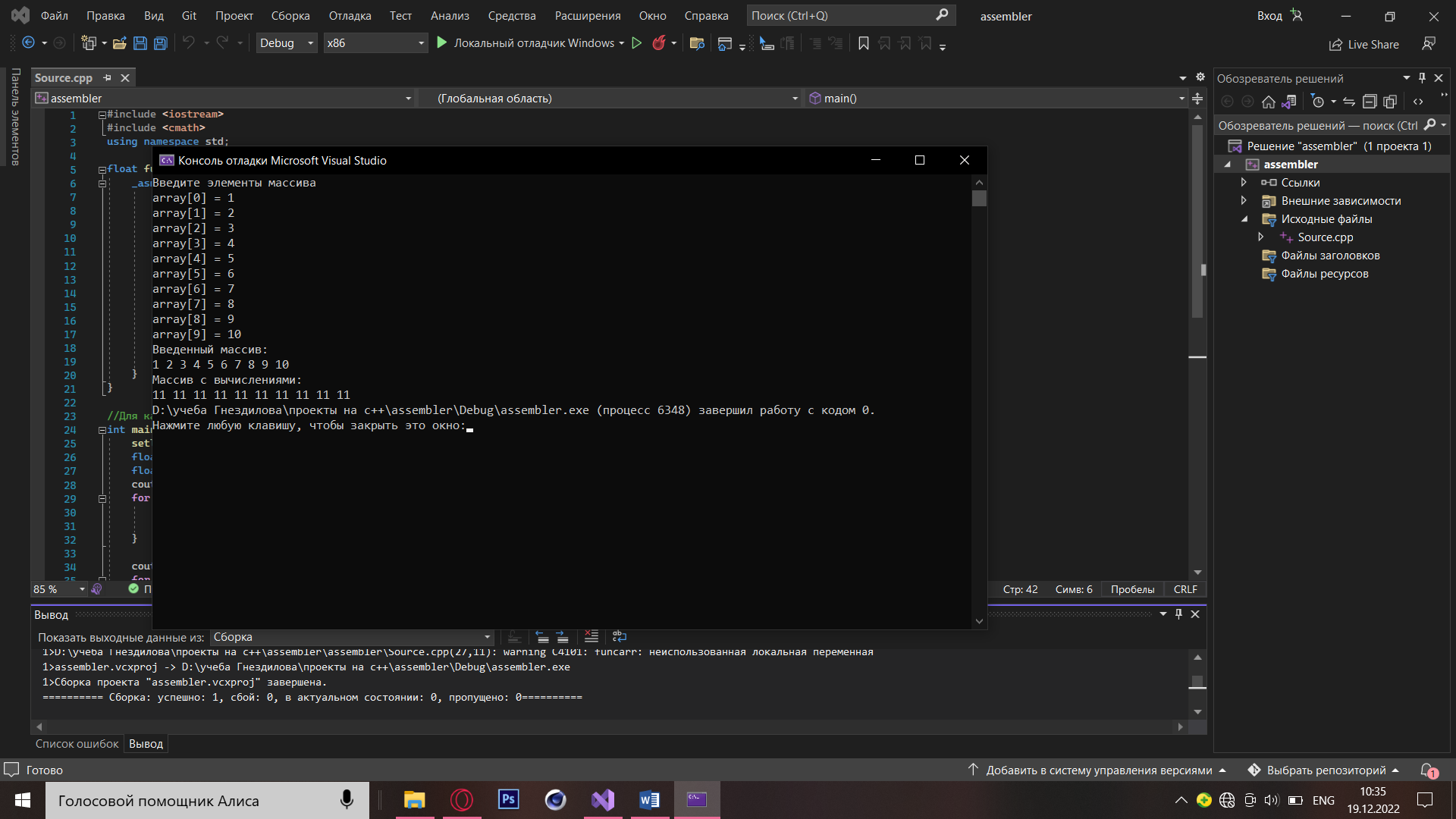
for (int i = 0; i < 10; i++) {

cout << arr[i] + arr[9 - i] << ' ';

}

}

**Результат выполнения задания:**



**Вывод:**

В ходе выполнения лабораторной работы были изучены : варианты внедрения ассемблерной процедуры в программу, написанную на языке программирования C\C++, изучена архитектура математического сопроцессора и команды работы с ним.