МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ

БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчёт по дисциплине

«Цифровые устройства и микропроцессоры»

Лабораторная работа №4

Вариант 4

Модульное программирование

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил: | студент группы ИНБс-3301 |  | \_\_\_\_\_\_\_\_\_\_\_ |  | А.А.Виноградов |
|  |  |  |  |  |  |
| Проверил: | доцент кафедры РЭС |  | \_\_\_\_\_\_\_\_\_\_\_ |  | М. А. Земцов |

Киров 2023

**Цель работы**: знакомство с технологией применения языка ассемблера при разработке программного обеспечения на языках высокого уровня.

**Исходные данные:**

|  |  |
| --- | --- |
| Функция C | Функция Ассемблера |
|  |  |

**Ход работы:**

Для смешанного программирования на языке высокого уровня и ассемблере используются многомодульные проекты, структура вызова функций которых изображена на рисунке 1.

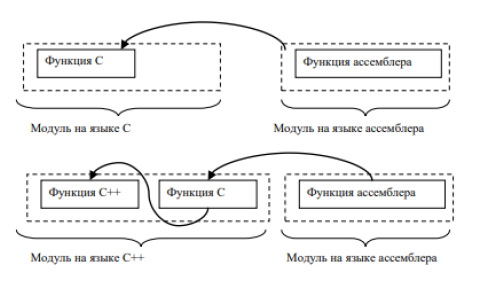


Рисунок 1 - Структура вызова функций

При этом должны выполняться некоторые условия, например, для вызова функций из модуля C на ассемблере они должны быть объявлены как extern, а в модуле ассемблера должна использоваться совместимая модель памяти и функция должна быть объявлена в сегменте .CODE с директивой EXTERN.

На рисунке 2 представлен код для вычисления элемента ряда на языке C.

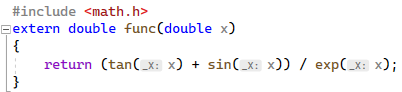


Рисунок 2 - Вычисление значений функции

Функция func объявлена с помощью ключевого слова extern и может быть вызвана в модуле ассемблера. На рисунке 3 представлен фрагмент ассемблерного кода для расчета суммы элементов ряда.

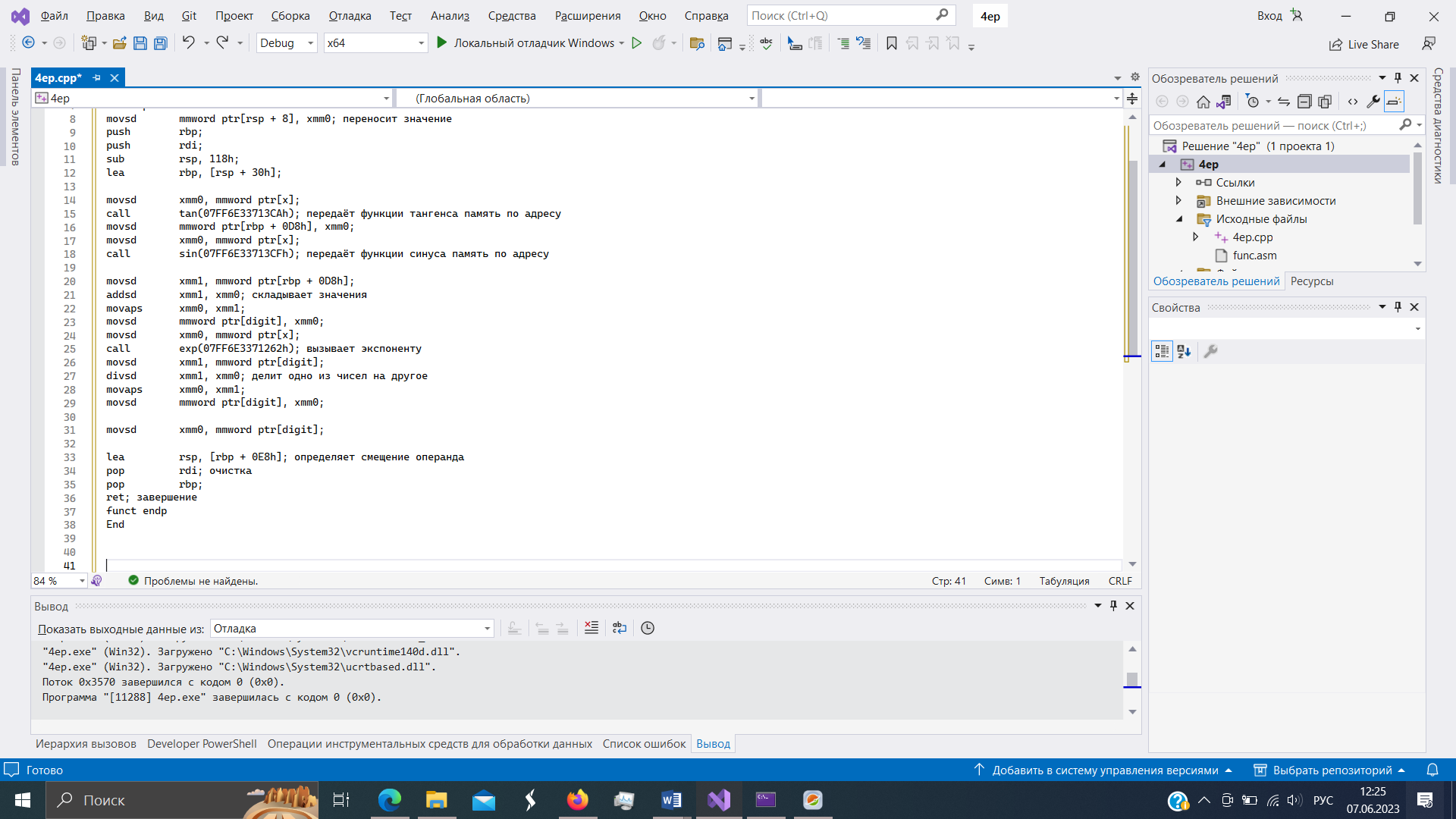


Рисунок 3 - Ассемблерный модуль

Как видно, в сегменте .CODE функция funct объявлена с директивой extern. Данная функция для вычисления значений функции объявлена с помощью директивы public для того, чтобы ее можно было вызвать из модуля на C++. Код основного модуля на С++ представлен на рисунке 4.

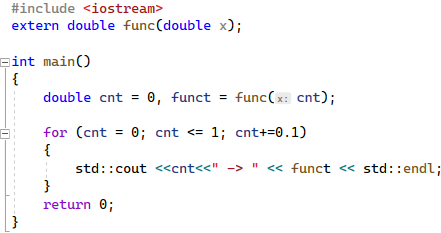


Рисунок 4 - Модуль С++

На рисунке 5 показан вывод результатов при вычислении значений функции.

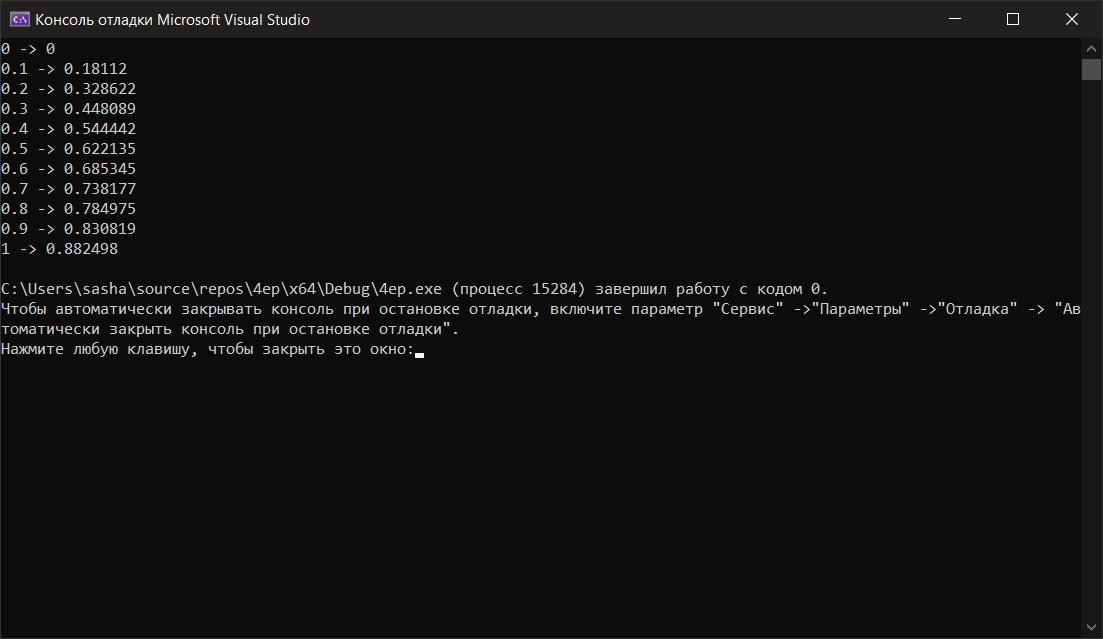


Рисунок 5 - Пример

**Вывод:** в ходе выполнения лабораторной работы были изучены принципы организации многомодульных проектов на языках высокого уровня с применением ассемблерных модулей.

**Код программы:**

Source.cpp

#include <iostream>

extern "C" double func(double x);

int main()

{

for (double cnt = 0; cnt <= 1; cnt+=0.1)

{

std::cout <<cnt<<" -> " << func(cnt) << std::endl;

}

return 0;

}

file1.c

#include <math.h>

extern double func(double x)

{

return (tan(x) + sin(x)) / exp(x);

}

file2.asm

.586

.MODEL flat, C

.DATA

.CODE

extern func:near ; объявление внешней функции func

public funct

funct proc C

movsd mmword ptr [rsp+8],xmm0 ; переносит значение

push rbp ;

push rdi ;

sub rsp,118h ;

lea rbp,[rsp+30h] ;

movsd xmm0,mmword ptr [x] ;

movsd mmword ptr [rbp+0D8h],xmm0 ;

movsd xmm0,mmword ptr [x] ;

movsd xmm1,mmword ptr [rbp+0D8h] ;

addsd xmm1,xmm0 ; складывает значения

movaps xmm0,xmm1 ;

movsd mmword ptr [digit],xmm0 ;

movsd xmm0,mmword ptr [x] ;

movsd xmm1,mmword ptr [digit] ;

divsd xmm1,xmm0 ; делит одно из чисел на другое

movaps xmm0,xmm1 ;

movsd mmword ptr [digit],xmm0 ;

movsd xmm0,mmword ptr [digit] ;

lea rsp,[rbp+0E8h] ; определяет смещение операнда

pop rdi ; очистка

pop rbp ;

ret ; завершение

funct endp

End