Министерство высшего образования и науки РФ

Вятский государственный университет

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчёт по предмету «Цифровые устройства и микропроцессоры» Лабораторная работа № 4

«Модульное программирование»

Вариант 9

Выполнил: студент группы ИНБб-3301-02-00 Чаланов И.С.

Проверил: доцент кафедры РЭС Земцов М. А.

Киров 2025

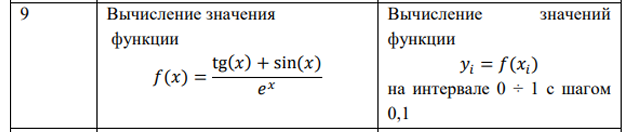
**Цель работы:** знакомство с технологией применения языка ассемблера при разработке программного обеспечения на языках высокого уровня. 

Таблица 1 – Задание, которое надо реализовать

Код программы на ассемблер

.586 ; Указываем процессор (Pentium)

.MODEL flat, C ; Используем плоскую модель памяти и соглашения языка C

.DATA ; Начало сегмента данных

; Глобальная переменная для хранения текущего значения x

current\_x DD 0.0 ; Выделяем 4 байта (float) и инициализируем нулем

.CODE ; Начало сегмента кода

; Объявляем внешнюю функцию fun\_el, которая реализована в C

extern fun\_el:near

; Объявляем функцию CalcFuncValues как общедоступную (видимую из C)

public CalcFuncValues

; Начало процедуры CalcFuncValues с соглашениями языка C

CalcFuncValues proc C

; Пролог функции - сохраняем ebp и устанавливаем новый стековый фрейм

push ebp

mov ebp, esp

; Аргументы функции (по соглашению cdecl):

; [ebp+8] - n (количество значений)

; [ebp+12] - указатель на массив результатов y

; [ebp+16] - начальное значение x (start)

; [ebp+20] - шаг (step)

; Загружаем параметры в регистры

mov ecx, [ebp+8] ; ecx = n (счетчик цикла)

mov eax, [ebp+12] ; eax = указатель на массив y

; Инициализируем current\_x начальным значением start

fld dword ptr [ebp+16] ; Загружаем start в стек FPU

fstp dword ptr [current\_x] ; Сохраняем start в current\_x

; Начало цикла вычисления значений функции

@@for\_i:

; Сохраняем регистры, которые будут использоваться в функции fun\_el

push ecx ; сохраняем счетчик цикла

push eax ; сохраняем указатель на массив y

; Подготавливаем аргумент для функции fun\_el (текущее значение x)

push dword ptr [current\_x] ; Помещаем current\_x в стек

call fun\_el ; Вызываем функцию fun\_el

add esp, 4 ; Очищаем стек от аргумента (соглашение cdecl)

; Восстанавливаем регистры

pop eax ; восстанавливаем указатель на массив y

pop ecx ; восстанавливаем счетчик цикла

; Сохраняем результат (возвращаемое значение в ST(0)) в массив y

mov edx, [ebp+8] ; edx = n

sub edx, ecx ; edx = n - ecx (текущий индекс 0-based)

fstp dword ptr [eax + edx\*4] ; Сохраняем результат в y[i]

; Увеличиваем current\_x на шаг

fld dword ptr [current\_x] ; Загружаем current\_x в стек FPU

fadd dword ptr [ebp+20] ; Прибавляем step

fstp dword ptr [current\_x] ; Сохраняем новое значение обратно в current\_x

loop @@for\_i ; Повторяем цикл, пока ecx != 0

; Эпилог функции - восстанавливаем указатель стека и ebp

mov esp, ebp

pop ebp

ret ; Возврат из функции

CalcFuncValues endp

End ; Конец модуляКод программы C

#include <iostream>

#include <cmath>

using namespace std;

// Объявляем внешние функции, которые будут реализованы в ассемблере и C

extern "C" {

void CalcFuncValues(int n, float\* y, float start, float step);

float fun\_el(float x);

}

int main(int argc, char\*\* argv) {

// Параметры для вычисления функции

const float start = 0.0f; // Начало интервала (0.0)

const float step = 0.1f; // Шаг вычисления (0.1)

const int n = 11; // Количество точек (0.0, 0.1, ..., 1.0)

// Выделяем память для массива результатов

float\* y = new float[n];

// Вызываем ассемблерную функцию для вычисления значений

CalcFuncValues(n, y, start, step);

// Выводим результаты вычислений

cout << "Function values on interval [0, 1] with step 0.1:" << endl;

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

float x = start + i \* step;

cout << "f(" << x << ") = " << y[i] << endl;

}

// Освобождаем выделенную память

delete[] y;

return 0;

}

// Функция для вычисления значения элемента по варианту 9: f(x) = (tg(x) + sin(x)) / e^x

extern "C" float fun\_el(float x) {

float tg = tanf(x); // Вычисляем тангенс x

float sin\_x = sinf(x); // Вычисляем синус x

float exp\_x = expf(x); // Вычисляем e^x

return (tg + sin\_x) / exp\_x; // Возвращаем результат по формуле варианта 9

}

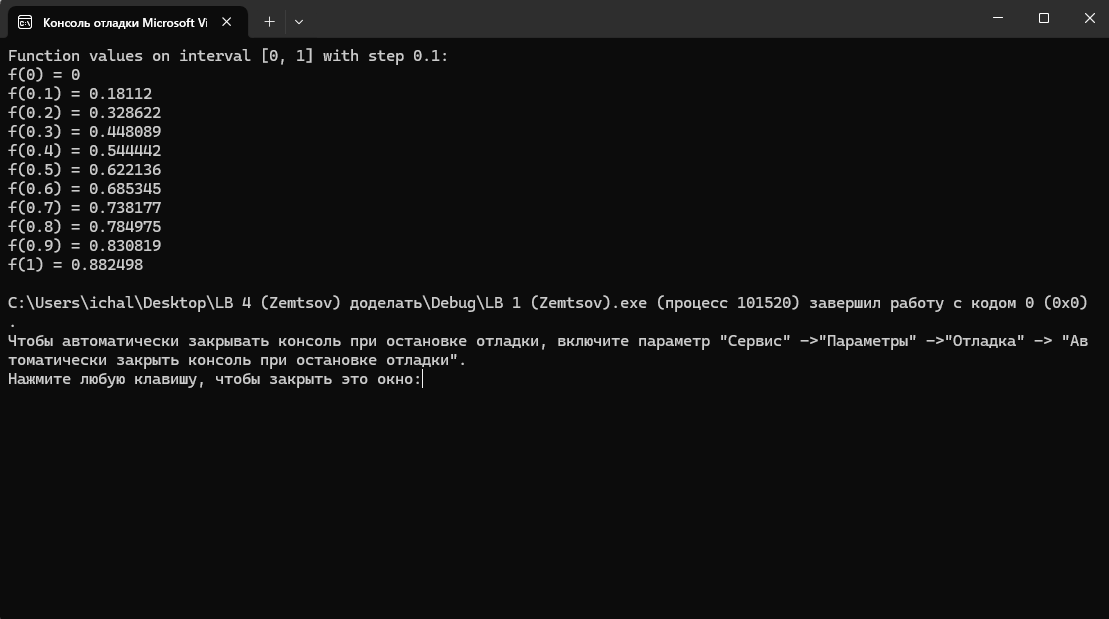


Рисунок 1 –Результаты вычисления

**Вывод:** познакомились с технологией применения языка ассемблера при разработке программного обеспечения на языках высокого уровня.