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 8

Выполнил: студент группы ИНБб-3301-02-00 Ведерников М.П

Проверил: преподаватель Земцов М. А.

Киров 2025

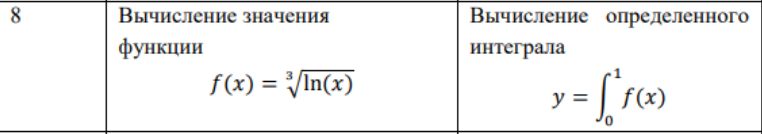
**Цель работы:** знакомство с технологией применения языка ассемблера при разработке программного обеспечения на языках высокого уровня. 

Таблица 1 – Задание, которое надо реализовать

Код программы на ассемблер

.586

.MODEL flat, C

.DATA

step DD ? ; шаг интегрирования h = 1/n

x DD 0.0 ; текущая точка x\_i

sum DD 0.0 ; накапливаемая сумма

.CODE

extern fun\_el:near ; Объявление функции fun\_el

public CalcIntegral ; Экспорт функции

CalcIntegral proc C

push ebp

mov ebp, esp

; Аргумент:

; [ebp+8] = n (количество интервалов)

; Проверка, что n > 0

mov eax, [ebp+8]

test eax, eax

jle error

; Вычисляем шаг h = 1.0 / n

fld1 ; st(0) = 1.0

fidiv dword ptr [ebp+8] ; st(0) = h = 1.0 / n

fstp dword ptr [step] ; сохраняем шаг

; Инициализация x = h (начинаем с x = h, т.к. в 0 функция не определена)

fld dword ptr [step]

fstp dword ptr [x] ; x = h

; Цикл интегрирования (метод прямоугольников)

mov ecx, [ebp+8] ; счетчик цикла = n

fldz ; st(0) = sum = 0.0

@@loop:

; Вычисляем f(x\_i)

push dword ptr [x]

call fun\_el

add esp, 4

; sum += f(x\_i) \* h

fmul dword ptr [step] ; st(0) = f(x\_i) \* h

faddp st(1), st(0) ; sum += f(x\_i) \* h

; x += h

fld dword ptr [x]

fadd dword ptr [step]

fstp dword ptr [x]

loop @@loop

; Возвращаем sum (результат)

fstp dword ptr [sum]

fld dword ptr [sum]

jmp exit

error:

fldz ; Возвращаем 0 в случае ошибки

exit:

pop ebp

ret

CalcIntegral endp

End

Код программы C

#include <iostream>

#include <cmath>

using namespace std;

extern "C" {

float CalcIntegral(int n); // Объявление ассемблерной функции

float fun\_el(float x); // Функция f(x) = кубический корень из ln(x)

}

int main() {

setlocale(LC\_ALL, "Russian");

int n;

cout << "Введите количество интегралов: ";

cin >> n;

if (n <= 0) {

cerr << "Количество должно быть позитивным!" << endl;

return 1;

}

float result = CalcIntegral(n); // Вызов ассемблерной функции

cout << "Значения интегралов: " << result << endl;

return 0;

}

// Функция f(x) = кубический корень из ln(x)

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

if (x <= 0.0f) return 0.0f; // Защита от ln(0) и отрицательных x

return cbrt(log(x)); // cbrt - кубический корень

}

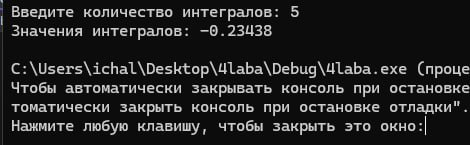


Рисунок 1 – Вывод результатов

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