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

Отчет по лабораторной работе №4

Дисциплина «Цифровые устройства и микропроцессоры»

По теме «МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ»

Вариант 10

Выполнил: студент группы ИНБб-3301-02-00                        /И.А.Николаева/

Проверил: доцент кафедры РЭС /М.А.Земцов/

 Киров 2024

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

Задание:

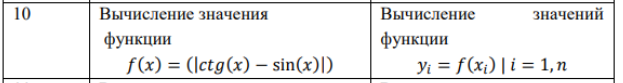


Рисунок 1 – Данные задания

1. Текст программы:

1.1. **Source.cpp**

#include <iostream>

extern "C" float funcASM(unsigned int n, float a[]); //объявл внеш функ

int main() {

unsigned int n;

std::cout << "input n>1: "; //вывод сообщ в конс

std::cin >> n; //счит знач из конс

float\* arr = (float\*)malloc(n \* sizeof(float));

//после вызова функ цел числ помещ в рег еах едх и в память

funcASM(n, arr);

std::cout << "result: ";

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

std::cout << arr[i] << " ";

std::cout << '\n';

return 0;

}

1.2. **Source1.asm**

.586

.model flat,c

.data

tmp real4 0.0

adr dword 0

n dword 0

counter dword 1

.code

extern funcC:near

public funcASM

funcASM proc

push ebp

mov ebp, esp ;сохр указат на адрес вершины стека

mov eax, [ebp+8]

mov n, eax

mov eax, [ebp+12]

mov adr, eax ;сохр в перемен адр адреса нач массива

loop1:

push counter ;запись знач счетчика для вызова функ С

call funcC ;вызов функ С

add esp, 4 ;коррект указат

fstp tmp ;сохр расчит знач

mov ebx, tmp

mov eax, adr

mov ecx, counter

dec ecx

mov [eax+ecx\*4], ebx ;запись нового знач в массив

inc counter

mov eax, n

cmp counter, eax

jbe loop1

mov esp, ebp

pop ebp

ret

funcASM endp

end

1.3. **Source2.c**

#include <math.h>

float funcC(int a) {

return fabs(cos(a) / sin(a) - sin(a));

}

1. Верификация программы:

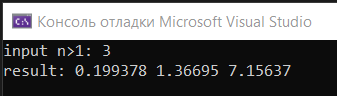


Рисунок 2 – Отладка программы

Вводим число, больше, чем 1 (возьмем 3), проверим ручным расчетом.  
  
ctg(1)=0.642093; sin(1)=0.841471; (|ctg(1)-sin(1)|)=0.199378  
ctg(2)=-0.457658; sin(2)=0.909297; (|ctg(2)-sin(2)|)=1.36695

ctg(3)=-7.01525; sin(3)=0.14112; (|ctg(3)-sin(3)|)=7.15637

Полученные данные сошлись с теми, что на скриншоте консоли.

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