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 дисциплине

«Цифровые устройства и микропроцессоры»

Лабораторная работа №4

«МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ»

Вариант №11

|  |  |  |
| --- | --- | --- |
| Выполнила: студентка гр. ИКТб-3301 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Н.А. Мусихина |
| Проверил: преподаватель кафедры ФПМТ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | М.А. Земцов |

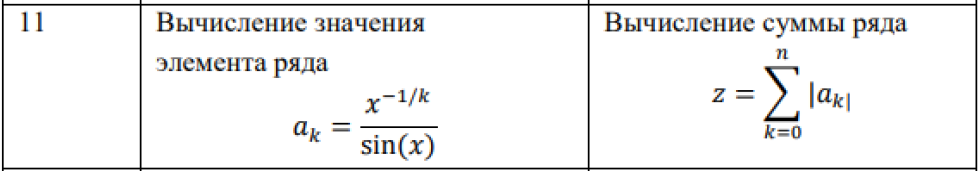
Киров 2023

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

**Ход работы:**

Условие варианта представлено в таблице 1.

Таблица 1 – Условие варианта



**Текст программы представлен ниже:**

***Source.cpp***

#include <iostream>

using namespace std;

extern "C" float SumR(int k, float x);

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

{ setlocale(LC\_ALL, "Russian");

int n;

float x;

cout << "Введите x " << endl;

cin >> x;

cout << "Введите количество элементов ряда " << endl;

cin >> n;

double R = 0.0;

R = SumR(n, x);

cout <<"Сумма "<< n <<" элементов ряда = " << R << endl;

return 0;

}

***Header.h***

#include <math.h>

extern "C" float fun\_el(int k, float x)

{

float f;

f = pow(x, -(1.0 / k)) / sin(x);

return f;

}

***S1.asm***

.586

.MODEL flat,C

.DATA

SUM DD 0.0

i\_local DD 0

.CODE

extern fun\_el:near ; объявление внешней функции fun\_el

public SumR

SumR proc C

push ebp

mov ebp,esp

mov i\_local,1

mov ecx, dword ptr [ebp+8]

@@for\_i: ; начало цикла

push ecx

push dword ptr [ebp + 12]

push i\_local

call fun\_el

fld SUM

add esp,8

fadd

pop ecx

inc i\_local

FABS;берем модуль числа

fstp SUM

loop @@for\_i ; конец цикла

fld SUM

mov esp,ebp

pop ebp

ret

SumR endp

End

**Расчет вручную:**

Произведем ручной расчет при x=3 и количестве элементов ряда = 4:

**Расчет в программе:**

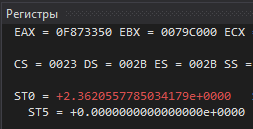


Рисунок 1 – Подсчет 1 слагаемого

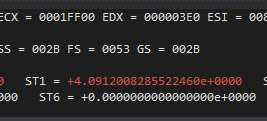


Рисунок 2 – Подсчет 2 слагаемого

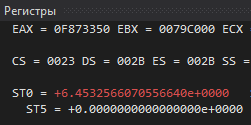


Рисунок 3 – Нахождение суммы 2 слагаемых

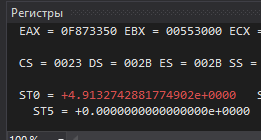


Рисунок 4 – Подсчет 3 слагаемого

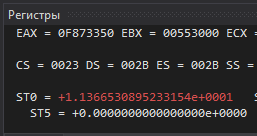


Рисунок 5 – Нахождение суммы 3 слагаемых

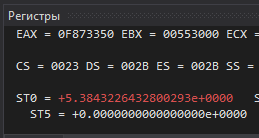


Рисунок 6 – Подсчет 4 слагаемого

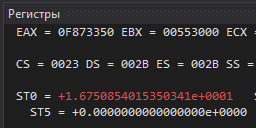


Рисунок 7 – Сумма всех слагаемых

**Вывод:** в ходе данной лабораторной работы познакомились с модульным программированием, синтаксисом EXTERN, PUBLIC, а также с организацией вызова функций с использованием языка Си/С++ и ассемблера.

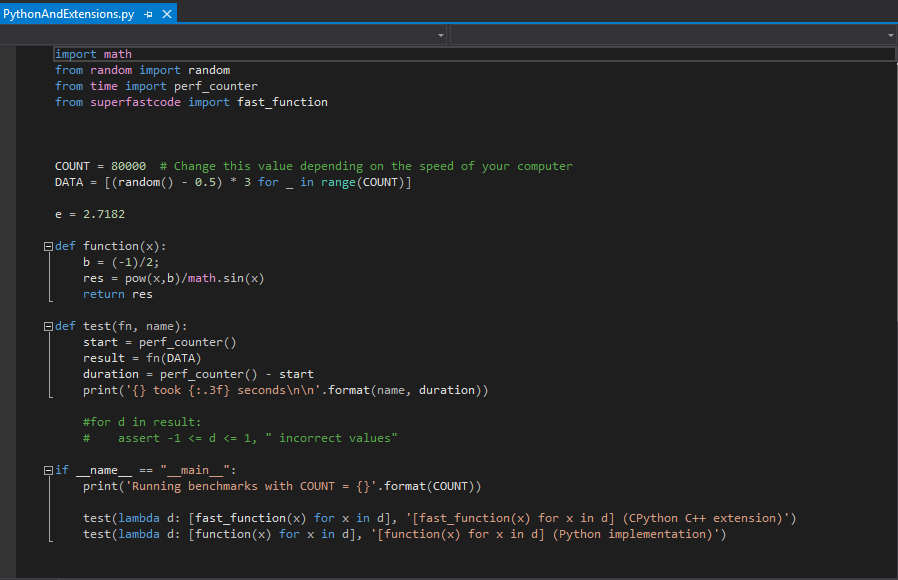


Рисунок 8 – Код на Python

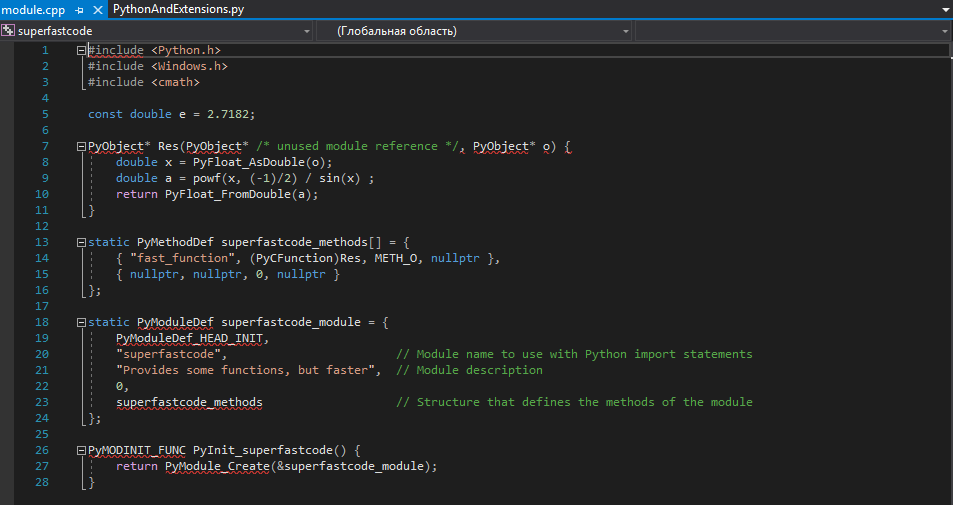


Рисунок 9 – Модуль на С++ для Python

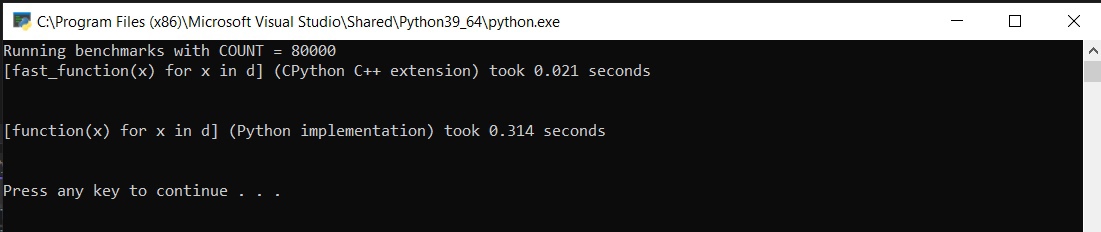


Рисунок 10 – Результат работы python-кода и python-кода с расширением на С++

Вывод: python-код с расширением на С++ отработал в 15 раз быстрее, чем код на python без расширения. Результаты можно обосновать тем, что написанная на С++ DLL, является модулем ускорения для проекта на Python.