Министерство цифрового развития, связи и массовых коммуникаций

Российской Федерации

Ордена Трудового Красного Знамени

федеральное государственное бюджетное образовательное учреждение

высшего образования

Московский технический университет связи и информатики

(МТУСИ)

Кафедра «Информационная безопасность»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

по дисциплине

««Программирование в системах информационной безопасности»

на тему

«Компиляция файлов исходного кода и компоновка полученных объектных файлов в исполняемый модуль. Линейные алгоритмы»

Вариант №2

Выполнил:

студент группы БСУ1801 Гавриков А.Г.

Проверил:

Старший преподаватель кафедры ИБ

Барков В. В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2021

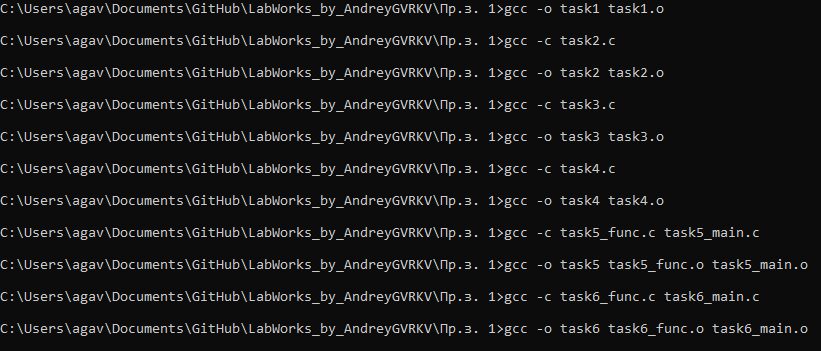
# **Цель**

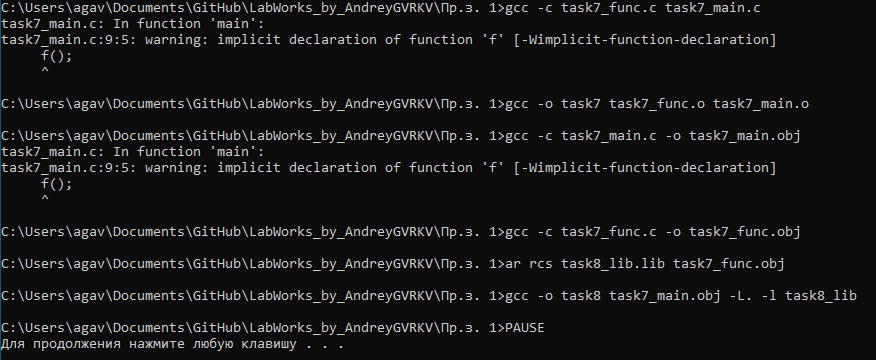
Овладеть навыками создания однофайловых и многофайловых проектов в интегрированной среде разработки Microsoft Visual Studio 2019 Community Edition и научится создавать линейные программы на языке C с применением арифметических операций.

# **Индивидуальные варианты заданий**



**Компиляция всех файлов**





**Задание 1**

Составить программу, имеющую линейный алгоритм и состоящую из одной функции void main(). Программу записать в файл с именем task1.c. Скомпилировать, скомпоновать и выполнить.

#include "stdio.h"

#include "stdlib.h"

#include <math.h>

void main()

{

    double x=5;

    double f;

    f = (cos(x)+sin(x)+cos(3\*x)+sin(3\*x));

    printf ("x = %f\n",x);

    printf ("f = %5.4f\n",f);

    x=0;

    printf("Input x: ");

    scanf("%lf",&x);

    printf("x = %f\n", x);

    f = (cos(x)+sin(x)+cos(3\*x)+sin(3\*x));

    printf ("f = %5.4f\n",f);

    system("pause");

    return;

}

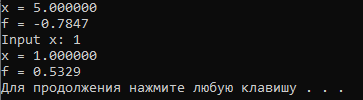


Рисунок 1 – Результат компиляции, компоновки и выполнения task1.exe

**Задание 2**

Линейный алгоритм функции из задания 1 разделить на две процедуры, выделив в одну вычислительные операции этого алгоритма, а в другую все операции ввода-вывода. Каждую процедуру оформить как функцию. Вычислительную часть алгоритма оформить как функцию с параметрами, передаваемыми по значению, и возвращаемым значением. Прототип функции:

Листинг 2 – Исходный код файла task2.c

#include "stdio.h"

#include "stdlib.h"

#include <math.h>

double f(double x)

{

    return (cos(x)+sin(x)+cos(3\*x)+sin(3\*x));

}

void main()

{

    double x=5;

    printf ("x = %f\n",x);

    printf ("f = %5.4f\n",f(x));

    x=0;

    printf("Input x: ");

    scanf("%lf",&x);

    printf("x = %f\n", x);

    printf ("f = %5.4f\n",f(x));

    system("pause");

    return;

}

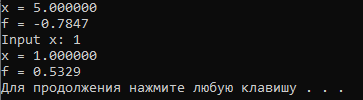


Рисунок 2 – Результат компиляции, компоновки и выполнения task2.exe

**Задание 3**

Создать файл task3.с, в котором изменить порядок записи текстов функций, созданных в задании 2. Функции записать в следующем порядке: функция main, функция с параметрами, организующая вычисления (double f(double x) или double f(double x, double y)). Внести требуемые дополнения, добиться успешной компиляции, скомпоновать и выполнить.

Листинг 3 – Исходный код файла task3.c

#include "stdio.h"

#include "stdlib.h"

#include <math.h>

double f(double x);

void main()

{

    double x=5;

    printf ("x = %f\n",x);

    printf ("f = %5.4f\n",f(x));

    printf("Input x: ");

    scanf("%lf",&x);

    printf("x = %f\n", x);

    printf ("f = %5.4f\n",f(x));

    system("pause");

    return;

}

double f(double x)

{

    return (cos(x)+sin(x)+cos(3\*x)+sin(3\*x));

}

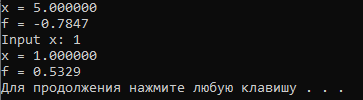


Рисунок 3 – Результат компиляции, компоновки и выполнения task3.exe

**Задание 4**

По-прежнему линейный алгоритм функции из задания 1 разделить на две процедуры, выделив в одну вычислительные операции этого алгоритма, а в другую все операции ввода-вывода.

Листинг 4 – Исходный код файла task4.c

#include "stdio.h"

#include "stdlib.h"

#include <math.h>

void f(void);

double x = 5;

double result;

void main()

{

    printf ("x = %f\n",x);

    f();

    printf ("f = %5.4f\n",result);

    printf("Input x: ");

    scanf("%lf",&x);

    f();

    printf("x = %f\n", x);

    printf ("f = %5.4f\n",result);

    system("pause");

}

void f(void)

{

    result = (cos(x)+sin(x)+cos(3\*x)+sin(3\*x));

}

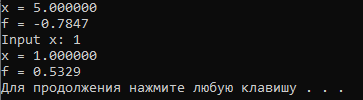


Рисунок 4 – Результат компиляции, компоновки и выполнения task4.exe

**Задание 5**

В этом задании необходимо разделить текст файла task3.с на два файла. В первый файл с именем task5\_main.c поместить текст функции main. Скомпилировать только файл task5\_main.c.

Листинг 5 – Исходный код файла task5\_main.c

#include "stdio.h"

#include "stdlib.h"

extern double f(double x);

void main(void)

{

    double x=5;

    printf ("x = %f\n",x);

    printf ("f = %5.4f\n",f(x));

    printf("Input x: ");

    scanf("%lf",&x);

    printf("x = %f\n", x);

    printf ("f = %5.4f\n",f(x));

    system("pause");

    return;

}

Листинг 6 – Исходный код файла task5\_func.c

#include <math.h>

double f(double x)

{

    return (cos(x)+sin(x)+cos(3\*x)+sin(3\*x));

}

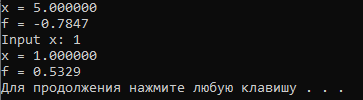


Рисунок 5 – Результат компиляции, компоновки и выполнения task5.exe

**Задание 6**

В этом задании необходимо разделить текст файла task4.с на два файла.

В первый файл с именем task6\_main.c поместить текст функции main. Скомпилировать только файл task6\_main.c.

Во второй файл с именем task6\_func.с поместить определения глобальных объектов (double x, y – аргументы, double result – результат вычисления) и текст функции без параметров (void f(void)). Скомпилировать только файл task6\_func.c.

Листинг 7 – Исходный код файла task6\_main.c

#include "stdio.h"

#include "stdlib.h"

#include <math.h>

void f (void);

double x;

double result;

void main()

{

    printf ("x = %f\n",x);

    f();

    printf ("f = %5.4f\n",result);

    printf("Input x: ");

    scanf("%lf",&x);

    f();

    printf("x = %f\n", x);

    printf ("f = %5.4f\n",result);

    system("pause");

}

Листинг 8 – Исходный код файла task6\_func.c

#include "stdio.h"

#include <math.h>

double x = 5.0;

double result;

void f (void)

{

    result =  (cos(x)+sin(x)+cos(3\*x)+sin(3\*x));

}

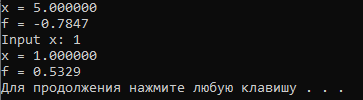


Рисунок 6 – Результат компиляции, компоновки и выполнения task6.exe

**Задание 7**

В этом задании необходимо модифицировать тексты файлов из задания 6.

Описание функции и внешних переменных выделить в отдельный заголовочный файл func.h, включить его содержимое в файлы task7\_main.c и

task7\_func.c. Определение внешних переменных произвести в файле task7\_func.c.

Скомпилировать по отдельности файлы task7\_main.c и task7\_func.c. Произвести компоновку, выполнить полученный исполняемый файл.

Листинг 9 – Исходный код файла task7\_main.c

#include "stdio.h"

#include "stdlib.h"

#include <math.h>

#include "func.h"

void main()

{

    printf ("x = %f\n",x);

    f();

    printf ("f = %5.4f\n",result);

    printf("Input x: ");

    scanf("%lf",&x);

    f();

    printf("x = %f\n", x);

    printf ("f = %5.4f\n",result);

    system("pause");

}

Листинг 10 – Исходный код файла task7\_func.c

#include "stdio.h"

#include <math.h>

#include "func.h"

double x = 5;

double result;

void f (void)

{

    result =  (cos(x)+sin(x)+cos(3\*x)+sin(3\*x));

}

Листинг 11 – Исходный код файла func.h

void f2(void);

extern double x;

extern double result;

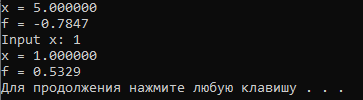
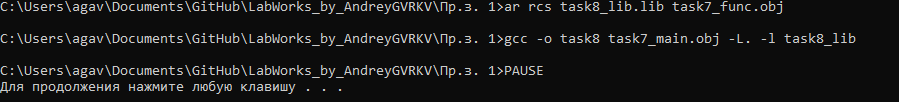


Рисунок 7 – Результат компиляции, компоновки и выполнения task7.exe

**Задание 8**

Скомпилировать по отдельности файлы из задания 7 task7\_main. obj и task7\_func. obj. Создать статическую библиотеку task8\_lib.lib, включающую объектный файл task7\_func.obj. Полученную статическую библиотеку скомпоновать с файлом task7\_main.obj. Выполнить полученный исполняемый файл.

Компонуем task7\_func.c и func.h в файл библиотеки task8\_lib.lib. Создаем консольное приложение с task8\_lib.lib и task7\_main.c, компилируем и компонуем.



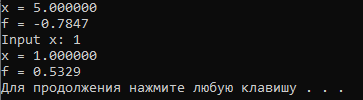


Рисунок 8 - Результат компиляции, компоновки и выполнения task8.exe

# **Вывод**

В данной работе были разобраны процесс создания .c файлов, компиляции их в .obj файлы и процесс их компоновки в .exe файлы. Так же рассмотрели особенности работы с заголовочными файлами, объявление глобальных переменных и различные типы переменных и функций.