ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ  
Ордена Трудового Красного Знамени федеральное государственное   
бюджетное учреждение высшего образования

Московский технический университет связи и информатики

МТУСИ

Кафедра информационной безопасности

Лабораторная работа

Вариант 2

Выполнил студент группы БИН1703 ОТФ-2 \_\_\_\_\_\_\_\_\_\_\_\_\_Бен Режеб Софиэн

Старший преподаватель кафедры ИТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Барков

Москва 2019

**Практикум №1**  
**Компиляция файлов исходного кода и компоновка полученных объектных файлов в исполняемый модуль. Линейные алгоритмы**

**Цель практикума:** Овладеть навыками создания однофайловых и многофайловых проектов в интегрированной среде разработки Microsoft Visual Studio 2015 Community Edition и научится создавать линейные программы на языке C с применением арифметических операций.

***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.

В функции main организовать вычисление задачи вашего варианта дважды:

* для исходных данных, значения которых задать в виде констант в тексте функции main;
* для исходных произвольных данных, значения которых пользователь Вашей программы должен ввести с клавиатуры в процессе выполнения программы.

Проанализировать результаты работы и сделать выводы. Перенести полученные результаты в отчет. Вывод действительных чисел осуществлять с точностью до 0.0001

***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:

double f(double x) // Если функция имеет один параметр

double f(double x, double y) // Если функция имеет два параметра

Другую часть алгоритма оформить как функцию void main(), вызывающую первую функцию нужное количество раз. Записать тексты функций файл с именем task2.с в следующем порядке: функция с параметрами, функция main. Cкомпилировать, скомпоновать и выполнить.

***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.

***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.

Вычислительную часть алгоритма оформить как функцию без параметров и без возвращаемого значения. Прототип функции void f(void)

Другую часть алгоритма оформить как функцию void main(), вызывающую вычислительную функцию. Обмен данными между функциями организовать через глобальные объекты (double x, y – аргументы, double result – результат вычисления). Записать тексты функций в файл с именем task4.с в следующем порядке: функция main, затем функция без возвращаемого значения и без параметров. Cкомпилировать, скомпоновать и выполнить.

***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.

Во второй файл с именем task5\_func.с поместить текст функции с параметрами (double f(double x) или double f(double x, double y)). Скомпилировать только файл task5\_func.c. После раздельной компиляции осуществить совместную компоновку. Полученный исполняемый файл выполнить. Проанализировать результаты работы и сделать выводы.

***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.

После раздельной компиляции осуществить совместную компоновку. Разобраться в проблемах, возникающих при совместной компиляции и компоновки. Полученный исполняемый файл выполнить. Проанализировать результаты работы и сделать выводы.

***Задание 7***

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.

***Задание 8***

Скомпилировать по отдельности файлы из задания 7 task7\_main.c и task7\_func.c. 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.

**Индивидуальные варианты заданий**



Листинг 1 — Исходный файл task\_N.h

#ifndef TaskN\_h

#define TaskN\_h

#define TASK\_1

#endif

Листинг 2 — Исходный файл task\_1.c

#include "TaskN.h"

#ifdef TASK\_1

#include <stdio.h>

#include <math.h>

int main() {

double x = 1.0;

printf("x=%lf\n", x);

printf("f(x)=%lf\n", ((4-pow(x,2))/2));

printf("Input x:\n");

double x2;

scanf("%lf", &x2);

printf("x=%lf\n", x2);

printf("f(x)=%lf\n", ((4-pow(x,2))/2));

}

#endif

Листинг 3 — Исходный файл task\_2.c

#include «TaskN.h"

#ifdef TASK\_2

#include <stdio.h>

#include <math.h>

double f(double x) {

double output;

output = (((4-pow(x,2))/2));

return output;

}

int main() {

double x = 1.0;

printf("x=%lf\n", x);

printf("f(x)=%lf\n", f(x));

printf("Input x:\n");

scanf("%lf", &x);

printf("x=%lf\n", x);

printf("f(x)=%lf\n", f(x));

}

#endif

Листинг 4 — Исходный файл task\_3.c

#include «TaskN.h"

#ifdef TASK\_3

#include <stdio.h>

#include <math.h>

void main() {

double x = 1.0;

printf("x=%lf\n", x);

printf("f(x)=%lf\n", func(x));

printf("Input x:\n");

scanf("%lf", &x);

printf("x=%lf\n", x);

printf("f(x)=%lf\n", func(x));

}

double f(double x) {

double output;

output = (((4-pow(x,2))/2));

return output;

}

#endif

Листинг 5 — Исходный файл task\_4.c

#include "TaskN.h"

#ifdef TASK\_4

#include <stdio.h>

#include <math.h>

double x, result;

void main() {

x = 1.0;

printf("x=%lf\n", x);

printf("f(x)=%lf\n", f());

printf("Input x:\n");

scanf("%lf", &x);

printf("x=%lf\n", x);

printf("f(x)=%lf\n", f());

}

void f(void) {

output = (((4-pow(x,2))/2));

}

#endif

Листинг 6 — Исходный файл task\_5\_f.c

#include "TaskN.h"

#ifdef TASK\_5\_f

#include <stdio.h>

#include <math.h>

double f(double x) {

double output;

output=((4-pow(x,2))/2));

return output;

}

Листинг 7 — Исходный файл task\_5\_m.c

#include "TaskN.h"

#ifdef TASK\_5\_m

#include <stdio.h>

#include <math.h>

#include "task\_5\_f.c"

void main() {

double x = 1.0;

printf("x=%lf\n", x);

printf("f(x)=%lf\n", f(x));

printf("Input x:\n");

scanf("%lf", &x);

printf("x=%lf\n", x);

printf("f(x)=%lf\n", f(x));

}

#endif

Листинг 8 — Исходный файл task\_6\_f.c

#include "TaskN.h"

#ifdef TASK\_6\_f

#include <stdio.h>

#include <math.h>

double x, result;

void f(void) {

output = ((4-pow(x,2))/2));

}

Листинг 9 — Исходный файл task\_6\_m.c

#include "TaskN.h"

#ifdef TASK\_6\_m

#include <stdio.h>

#include <math.h>

#include «task\_6\_f.c"

void main() {

x = 1.0;

printf("x=%lf\n", x);

f();

printf("f(x)=%lf\n", result);

printf("Input x:\n");

scanf("%lf", &x);

printf("x=%lf\n", x);

f();

printf("f(x)=%lf\n", result);

}

#endif

Листинг 10 — Исходный файл LB1\_header.h

#ifndef LB1\_header

#define LB1\_header

double x, result;

void f(void);

#endif

Листинг 11 — Исходный файл task\_7\_f.c

#include "LB1\_header.h"

#include <stdio.h>

#include <math.h>

void f(void) {

result = ((4-pow(x,2))/2));

}

Листинг 12 — Исходный файл task\_7\_m.c

#include "LB1\_header.h"

#include <stdio.h>

#include <math.h>

void main() {

x = 1.0;

printf("x=%lf\n", x);

f();

printf("f(x)=%lf\n", result);

printf("Input x:\n");

scanf("%lf", &x);

printf("x=%lf\n", x);

f();

printf("f(x)=%lf\n", result);

}

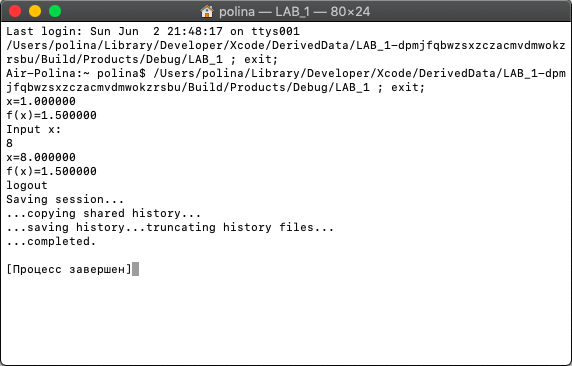


Рисунок 1