

# Лабораторная работа №1

## «Знакомство с языком ассемблер»

### **Цель работы:**

знакомство на примере ассемблера TASM с языком программирования низкого уровня ассемблер: с его синтаксисом, структурой исходного кода программы, способами трансляции, линковки и отладки.

### **Задание:**

Разработать программу, решающую следующую задачу:

Пользователь вводит с клавиатуры начальное и конечное значения диапазона десятичных беззнаковых однобайтных чисел. Программа выводит на экран последовательность всех чисел этого диапазона. Если начальное значение больше конечного, и если вводятся неправильные символы, то программа должна выдать ошибку.

### **Отчет по работе должен включать:**

Титульный лист, цель работы и задание, исходные тексты программ с необходимыми комментариями.

### **Рекомендуемая последовательность решения задачи:**

1. Ознакомиться с жизненным циклом ассемблерной программы в средстве TASM.

Последовательность процедур формирования программы следующая:

- Составление исходного кода программы. Этот этап может осуществляться в любом текстовом редакторе, не форматировующим текст (например, Notepad).
- Трансляция (ассемблирование) программы. Для осуществления этого действия в TASM необходимо запустить файл `tasm.exe` и в качестве параметра указать имя файла, где содержится исходный текст программы.

Например:

`tasm.exe prg1` – транслирование программы, исходный текст которой находится в файле `prg1.asm`

В результате сформируется объектный файл `prg1.obj` который будет содержать последовательность машинных кодов данной программы.

Помимо имени файла в качестве параметра можно указывать дополнительные опции. Мы будем использовать следующие:

`tasm.exe /zi prg1` – транслирование программы, с занесением в объектный файл информации для отладки.

`tasm.exe /zi prg1 , ,` - помимо объектного файла с информацией для отладки также создается файл листинга, содержащий полезную информацию о программе.

- Компоновка (линковка) объектного модуля (загрузка и редактирование связей). Для ее осуществления необходимо запустить файл tlink.exe и в качестве параметра указать имя объектного файла.

Например:

tlink.exe prg1 – линковка объектного файла prg1.obj. В результате будет создан исполняемый файл prg1.exe.

Здесь можно через командную строку также задавать дополнительные параметры, мы будем использовать следующие:

tlink.exe /v prg1 – в исполняемый файл будет занесена информация для отладки.

- отладка программы

Осуществляется путем запуска отладчика td.exe с указанием в качестве параметра имени исполняемого файла.

Например

td.exe prg1

- выполнение программы.

Осуществляется путем запуска исполняемого файла на выполнение.

2. Ознакомиться со структурой программы на ассемблере.

Примерный шаблон программы на ассемблере выглядит следующим образом:

MASM

MODEL SMALL

.DATA ;здесь описывается сегмент данных

.STACK 256 ;здесь описывается сегмент стека

.CODE

MAIN PROC ; здесь представлен исходный код процедуры PROC

MOV AX, @DATA ;инициализация сегмента данных

MOV DS, AX

; здесь непосредственно расположен код программы

MOV AX, 4C00H ;корректный выход из программы

INT 21H

MAIN ENDP ; конец процедуры MAIN

END MAIN ; конец программы с точкой входа MAIN

3. Изучить способы ввода информации с клавиатуры и вывода информации на экран, а также возможности отладчика Turbo Debugger.

В операционной системе DOS существуют готовые обработчики прерываний вывода на экран символьной информации и для ввода информации с клавиатурой.

Вывод символа на экран осуществляется следующим образом (на примере символа "С":

MOV AH, 6

```
MOV DL, 43H ;43H – код ASCII-код символа “С”  
INT 21H
```

Вывод строки символов на экран осуществляется следующим образом:

```
.DATA  
text DB ‘Вывод строки символов $’
```

```
.CODE  
MOV AH, 9  
MOV DX, OFFSET text ;в регистр DX поместить смещение переменной text  
INT 21H
```

Чтение символа с клавиатуры и одновременное отображение его на экране осуществляется следующим образом:

```
MOV AH, 1  
INT 21H
```

Считанный код символа располагается в регистре AL

Для чтения строки символов необходимо сначала осуществить следующую инициализацию:

В сегменте данных задать буфер, указав в нем число символов строки, подлежащих вводу

```
.DATA  
buf DB n, 0, 0, ..., 0, 0
```

где n – максимальное число вводимых символов. Количество нулей в директиве должно быть на 2 большее, чем n. На место следующего нуля после ввода будет записано количество фактически введенных символов, на место последующих нулей – ASCII коды введенных символов. Последний нуль используется для кода клавиши Enter – 13.

Затем нужно занести в регистр DX адрес буфера

Пример

```
.DATA  
buf DB 3, 0, 0, 0, 0, 0  
MOV AH, 0AH;  
MOV DX, OFFSET buf  
INT 21H
```

4. Изучить способы представления символьных данных в формате ASCII.

Для этого следует организовать ввод числа с одним десятичным знаком, и вывод его удвоенного значения.

5. Реализовать возможность ввода в программу численных беззнаковых данных максимального размера в один байт. Подтверждение ввода каждого числа осуществляется клавишей Enter.

6. Реализовать алгоритм вывода всех чисел заданного диапазона.