**Лабораторная робота №4**

**Структура программы на языке Ассемблера**

**Цель работы**: Ознакомление со структурой программ на языке Ассемблера, получение навыков разработки программ и их настройки.

**Короткие теоретические сведения**

Язык Ассемблера – низкоуровневый язык программирования, который позволяет писать программы на машинном уровне используя условные сокращения – мнемокоды.

Ассемблер - Транслятор исходного текста программы, написанной на языке ассемблера, в программу на машинном языке.

А языке Ассемблера регистр не имеет значения.

Последовательность разработки программ на языке Ассемблера выполняется следующим образом:

1. Создается текстовый файл с расширением \*.asm
2. Пишеться код программы на языке Ассемблера
3. С помощью транслятора Ассемблера код программы преобразовывается в машинный (создается файл с расширением \*.obj)
4. С помощью линковщика преобразовать машинный код программы в исполняемый.

Процесс компиляции программы на языке Ассемблера отличается у разных Ассемблеров.

Для примера, в ассемблере masm32 для компиляции программы нужно ввести в терминале:

Для программ для windows:

> ml /c /coff \*.asm

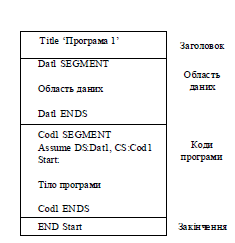
> link /subsystem:console(windows) \*.obj

Для 16-битных программ:

> ml /c \*.asm

> link16 /TINY \*.obj

Структура программы на языке Ассемблера (в EMU8086, в других Ассемблерах структура может отличатся).



Элементы языка Ассемблера:

Переменные данные. Для представления переменных используются Директивы Данных db, dw, dd, которые указывают транслятору на выделение определенного объема памяти: байт, слово, двойное слово (1, 2 и 4 байта), а также для представление символьной строки из одного или более символов.

Директивы данных обычно представляют в сегменте данных. Синтаксис этих директив имеет такой вид:

[Имя] db Выражение [, выражение...]

[Имя] dw Выражение [, выражение...]

[Имя] dd Выражение [, выражение...]

Сегмент памяти. Сегменты памяти предназначены для выделения памяти для данных и кода программы. Они имеют синтаксис:

**Имя\_сегмента Segment**

Тело сегмента

**Имя\_сегмента ENDS**

Определение адреса:

**Seg вираз**

**offset вираз**

**Задания**

**Задание 1.1.** Набрать текст программы вывода на экран символьной информации.

1. Код программы:

title word

dat segment

a db 'Hello world1',10,13

b db 'Hello world2',10,13,'$'

c db '34'

dat ends

cod segment

assume cs:cod, ds:dat

start:

mov ax, dat

mov ds, ax

xor ax, ax

mov ah, 09h

mov dx, offset a

int 21h

mov ah, 02h

mov dx, 'c'

int 21h

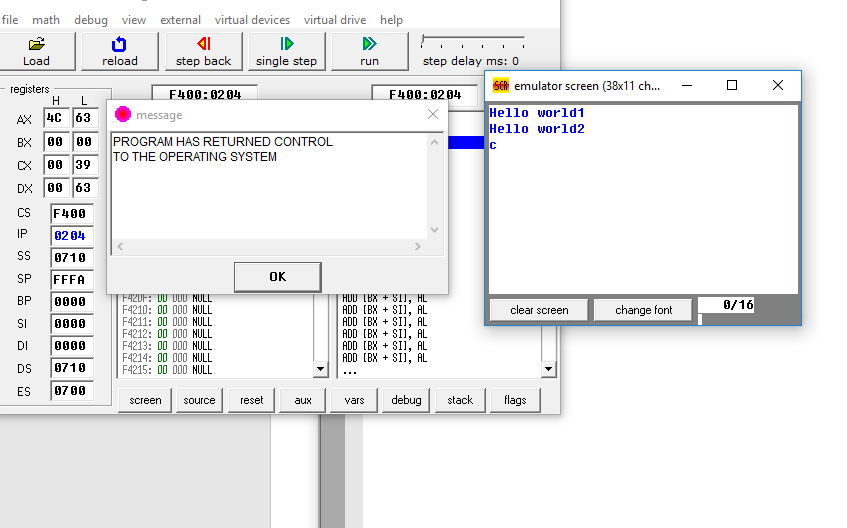
mov ah, 4Ch

int 21h

cod ends

end start

1. Результат выполнения программы



**Вывод:** на этой лабораторной работе я ознакомился со структурой программ на языке Ассемблер, получил навыки разработки программ и их настройки.