……………

Язык **ассемблера –** это машинно-ориентированный язык программирования.

**Ассемблер –** это программа-транслятор, принимающая на входе текст на языке ассемблера, содержащий условные обозначения машинных команд, удобные для человека, и переводящая эти обозначения в последовательность соответствующих кодов машинных команд, понятных процессору.



**Уровень 0** – возможность запуска программ, состоящих из машинных кодов арифметико-логическим блоком (АЛУ) центрального процессора, т.е. реализуется с помощью цифровых электронных схем (VM0).

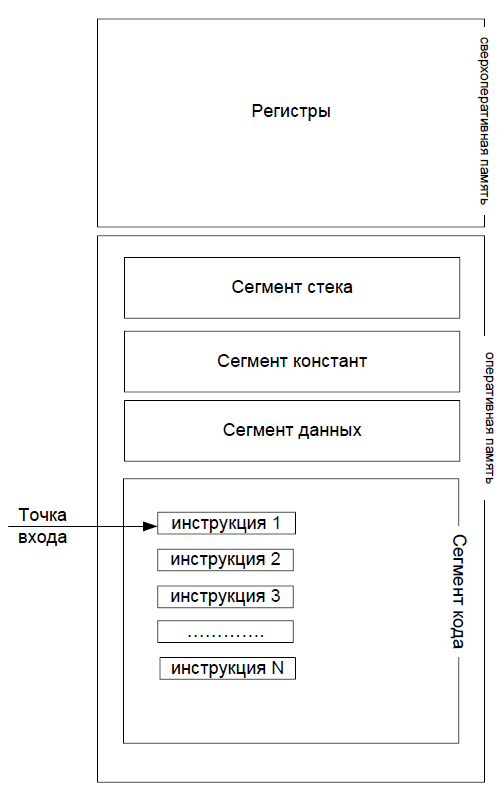
**Уровень 1** – составляет **систему** микрокоманд процессора (VM1), которая выполнена в виде интерпретатора.

**Уровень 2** – система команд процессора. Для выполнения одной команды машинного кода (машинной команды) требуется выполнить, как правило, несколько микрокоманд.

**Уровень 3** – операционная система (комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем).

**Уровни** с 4-го и выше предназначены для прикладных программистов, решающих конкретные задачи.

**Ассемблер:**



**Ответы на вопросы:**

…

5).asm

6) Debug: .exe, .pdb

Папка проекта: .asm, vcxproj…

Папка проекта->Debug: .lst, .obj, .log, .ilk (incremental linker file)

7).obj – находится: папка проекта -> Debug -> … .

8).exe – находится: Debug -> … .

9) .lst (В листинге содержится оттранслированный машинный код, представленный в шестнадцатеричном виде.)

10) Язык ассемблера разрабатывается для **семейства процессоров**: Motorola 680 (MC680), SPARC, IBM370, IA-32.

Для работы с процессорами х86 используются **два типа синтаксиса** ассемблера:

- синтаксис AT&T;

- синтаксис Intel.

**Типы архитектуры:**

**CISC** (Complete Instruction Set Computing) – тип архитектуры процессора с ***полным набором команд*** (IBM с архитектурой IBM/360, процессоры Intel на основе команд x86, процессоры Motorola MC680x0, DEC VAX).

**RISC** (англ. *restricted (reduced) instruction set computer* — «компьютер с ***сокращённым набором команд***») — архитектура процессора, в котором быстродействие увеличивается за счёт упрощения инструкций, для того, чтобы их декодирование было более простым, а время выполнения – меньшим (Sun Ultra SPARC, MIPS, Alpha DEC, PowerPC).

\*\*\*Модель процессора в программе указывается в первой строке кода (.586)

Плоская модель памяти **flat (flat memory model).** Эта модель памяти используется в операционной системе Windows. Адресация любой ячейки памяти будет определяться содержимым одного 32-битного регистра.

\*\*\*В программе указывается .MODEL flat

11) stdcall – используемое соглашение о вызовах процедур

12) .STACK – сегмент стека. Размер стека по умолчанию – 1 Мб.

.CONST – сегмент (или секция) констант.

.DATA – сегмент (или секция) данных.

.CODE – сегмент кода.

Директива EQU определяет константу (подобно #define в языке СИ).

STR1 и STR2 – символьные строки, должны заканчиваться 0 байтом.

HW – неициализированное двойное слово (4 байта = 32-бита).

main PROC – директива PROC определяет начало процедуры.

START – метка.

Директива OFFSET указатель начала строки.

CALL – вызов функции.

main ENDP – конец процедуры main.

END main – последняя инструкция программы, в ней указывается точка входа в программу (определено как имя main).

13) Сегмент кода

14)main PROC – main ENDP

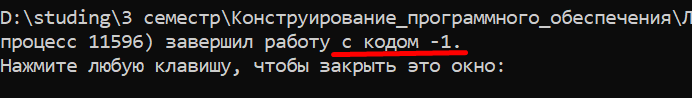
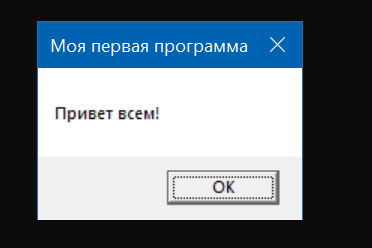
15) Объявление прототипа функции с использованием директивы PROTO

(после символа «:» указывается тип параметра, параметры разделяются символом «,»).

16) call MessageBoxA(?). Параметры WinAPI-функций 32-битные (целые числа). Все WinAPI-функции созданы по соглашению stdcall.

17) START – метка. Начало выполнения.

18) call ExitProcess. С кодом -1 завершается программа.



20) Транслятор языка MASM позволяет упростить вызов функций при помощи INVOKE. Встроенный макрос INVOKE используется для вызова любых функций, прототип которой должен быть задан.

Порядок следования параметров должен точно соответствовать прототипу функции.

20) MessageBox – выводит на экран окно с сообщением и кнопкой выхода.

Параметры функции:

- дескриптор окна, в котором будет появляться окно-сообщение;

- текст, который будет появляться в окне;

- текст заголовка окна;

- тип окна, в частности можно определить количество кнопок выхода.