



Тема урока: **Основные команды  
языка ассемблер**

# Синтаксис языка ассемблера определяется системой команд конкретного процессора

## Assembly Programming Language

Типичными командами языка ассемблера являются (большинство примеров даны для Intel-синтаксиса архитектуры x86):

- Команды пересылки данных (*mov* и др.);
- Арифметические команды (*add, sub, mul* и др.);
- Логические и побитовые операции (*or, and, xor, shr* и др.);
- Команды управления ходом выполнения программы (*jmp, loop, ret* и др.);
- Команды вызова прерываний (иногда относят к командам управления): *int*;
- Команды ввода/вывода в порты (*in, out*).

## Типичный формат записи команд:

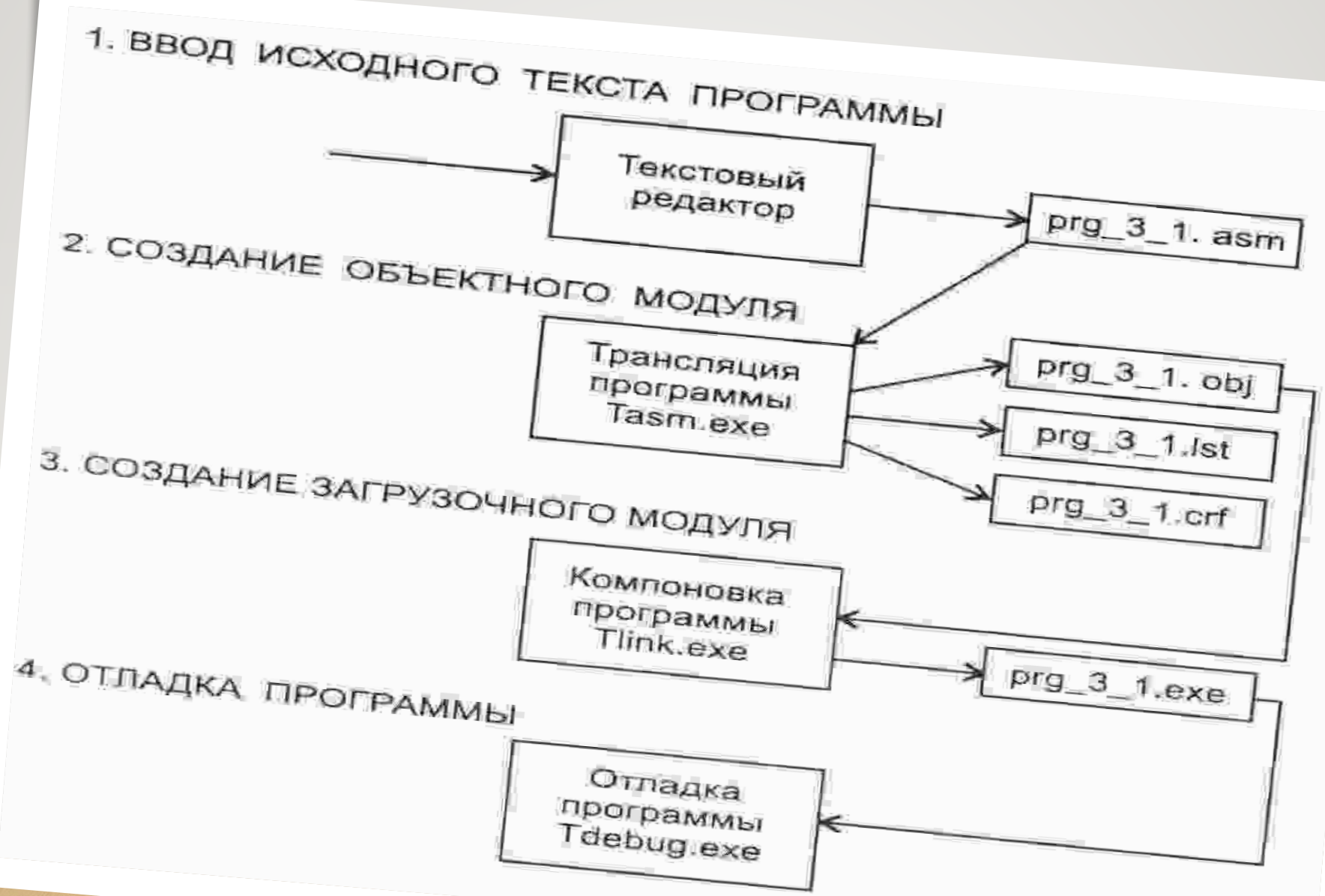
[метка:] **мнемокод** [операнды] [;комментарий]

где **мнемокод** — непосредственно мнемоника инструкции процессору. К ней могут быть добавлены префиксы (повторения, изменения типа адресации и пр.).

В качестве операндов могут выступать константы, адреса регистров, адреса в оперативной памяти и пр. Различия между синтаксисом **Intel** и **AT&T** касаются в основном порядка перечисления операндов и указания различных методов адресации.

```
Date: Assembler Editor Suchen Block Einfügen Undo Hilfe
7: 6568 50 0 E:00000000/00000000/00000000/00000000/00000000/00000000/00000000/00000000
>PART 'convert_2d'
OPT 0-
>PART 'play rout'
OPT 0+
>PART 'setup'
>PART 'setup_other'
>PART 'setup_sprites'
PART 'setup_shadesprites'
setup_shadesprites:
    lea    shade_sprite_buf,A0
    lea    shade_sprite_data,A1
    moveq #15,D0
shade_next_sprite:
    movea.l A0,A2
    moveq #15,D2
shade_next_copy:
    move.l (A1)+,(A0)+
    dbra  D2,shade_next_copy
    movea.l A2,A1
    cmpi.w #15,D0
    cwb!  n #12'D0
    wop69' | 05'H1
    qpl9  05' 21996~06K1~cob1
    wop6' | (A1)+(A0)+
    21996~06K1~cob1:
    wop60  #12'D0
```

# 1. Схема процесса разработки программ на *ассемблере*:



## 2. Программное обеспечение для разработки программ на ассемблере.

**Turbo Assembler (TASM)** — программный пакет компании Borland, предназначенный для разработки программ на языке ассемблера для архитектуры x86.

**1. Создание файла с расширением asm и помещением туда программного кода.**

**2. Трансляция** - процесс формирования *объектного модуля (с расширением obj)* из текстового файла с расширением *.asm*. Выполняет трансляцию программа *tasm.exe (tasm32.exe)*

*Пример: tasm hello.asm*

В случае если не обнаружено синтаксических ошибок, будет создан объектный файл *hello.obj*

**3. Компоновка** - процесс формирования *исполняемого файла* из объектного модуля. Выполняет компоновку программа *tlink.exe (tlink32.exe)*

*Пример: tlink hello.obj*

*Результат: hello.exe*

**4. Отладка** - процесс выполнения разработанной программы с целью проверки её работы. Выполняется отладка программой *td.exe (td32.exe)*

*Td hello.exe*

Название команды	Класс команды	Выполняемое действие
<b>Mul</b>	Арифметические	<b>Mul – команда умножения.</b> Она умножает регистр <b>AX</b> на то, что стоит после нее. Результат заносится в регистр <b>AX</b> .
<b>Div</b>		
<b>Add</b>		
<b>Sub</b>		
<b>Mov</b>		
<b>Inc</b>		
<b>Dec</b>		
<b>Neg</b>		
<b>Push</b>		
<b>Nop</b>		
<b>Cmp</b>		
<b>Jmp</b>		
<b>Pop</b>		
<b>Call</b>		
<b>Loop</b>		
<b>Ret</b>		

# Контрольные вопросы:

1. Что такое регистр процессора?
2. Что такое регистры общего назначения (РОН)?
3. Что такое счётчик команд?