Системное программирование

Лекция №2

Основы разработки программы на ассемблере

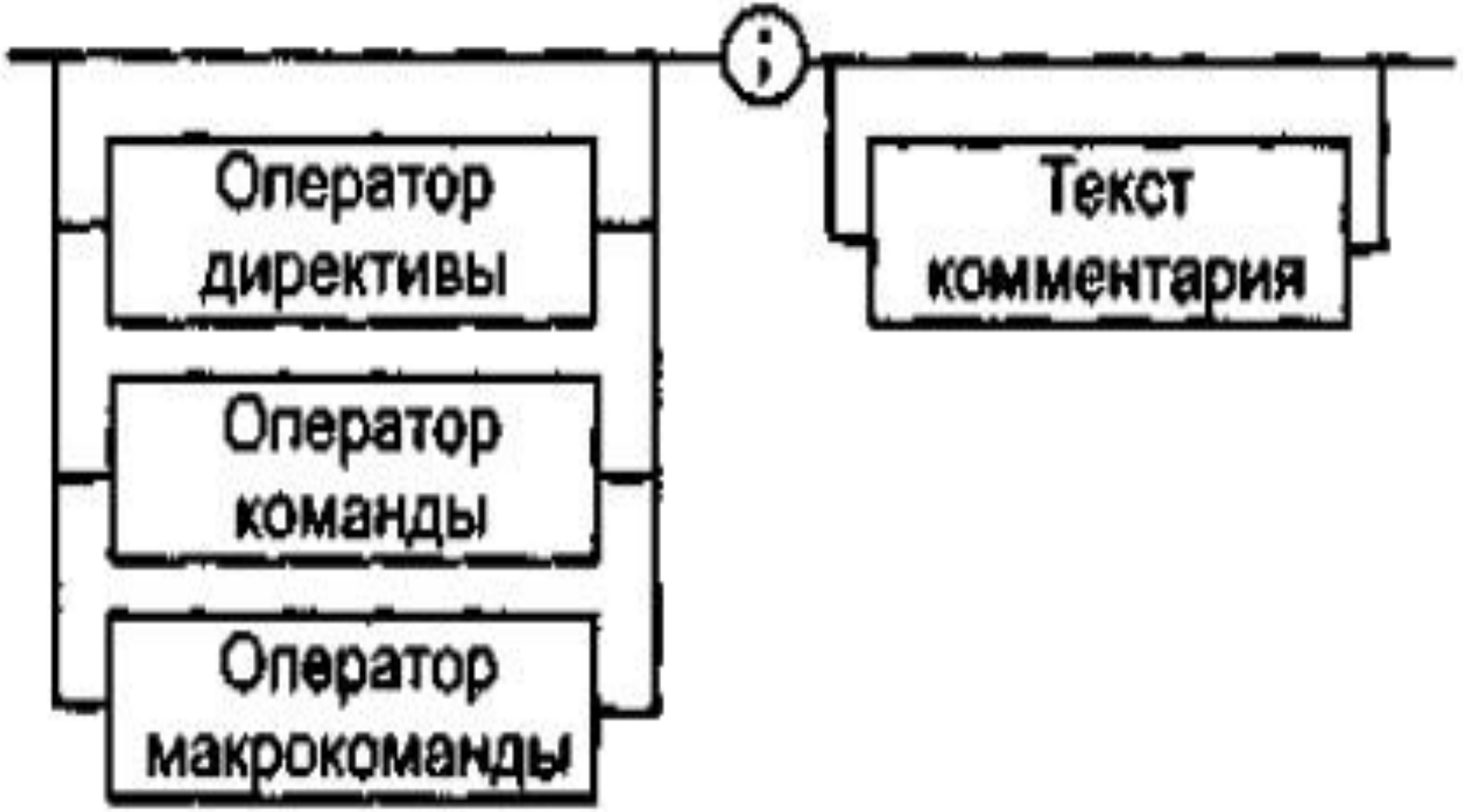
Программа на ассемблере: совокупность блоков памяти, называемых *сегментами*. *Сегменты программы: кода, данных и стека. Каждый сегмент состоит из совокупности отдельных строк, в терминах теории компиляции называемых предложениями языка.*

Системное программирование

Предложения, составляющие программу, могут представлять собой синтаксические конструкции четырех типов:

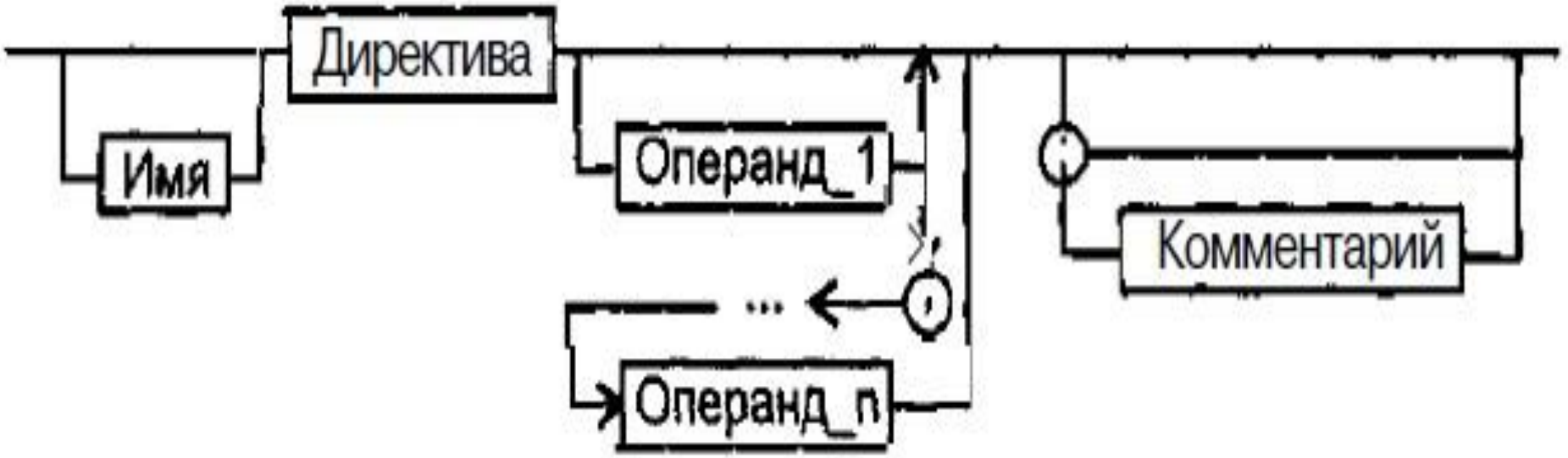
- *Команды* (инструкции) - символические аналоги машинных команд. В процессе трансляции инструкции ассемблера преобразуются в соответствующие команды системы команд процессора.
- *Макрокоманды* — это оформляемые определенным образом предложения текста программы, замещаемые во время трансляции другими предложениями.
- *Директивы* являются указанием транслятору ассемблера на выполнение некоторых действий. У директив нет аналогов в машинном представлении.
- *Комментарии* содержат любые символы, в том числе и буквы русского алфавита. Комментарии игнорируются транслятором.

Программы на ассемблере



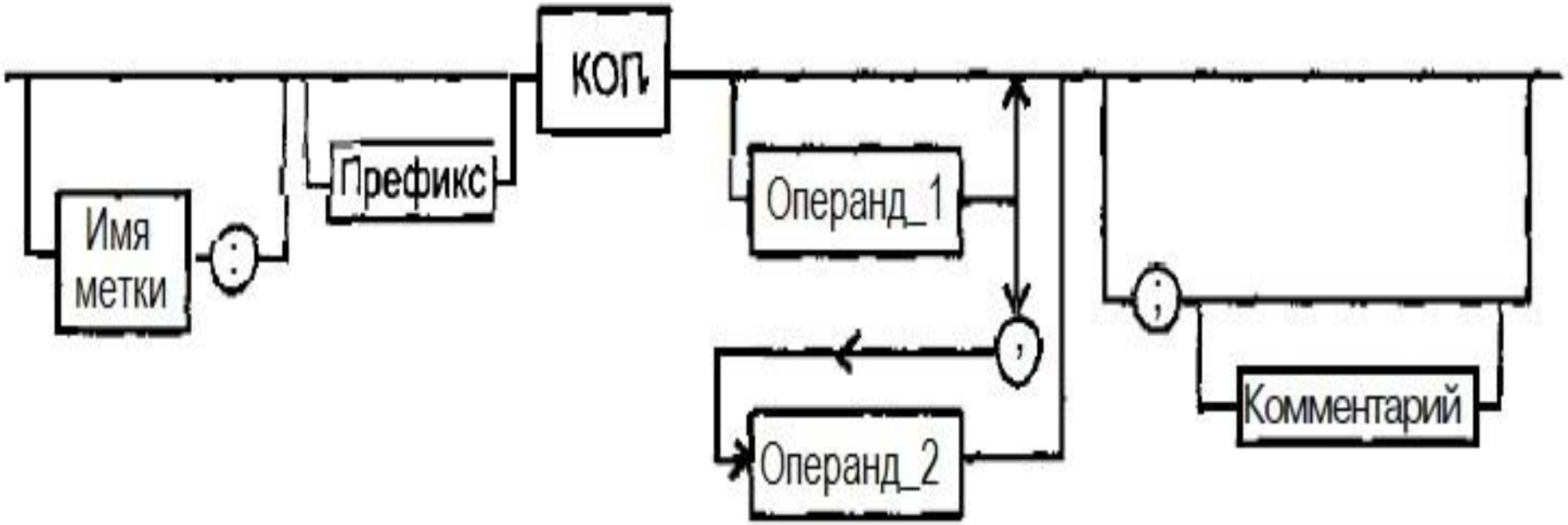
Формат предложений ассемблера

Программы на ассемблере



Формат директив

Программы на ассемблере



Формат команд и макрокоманд

Программы на ассемблере

Имя метки — символьный идентификатор. Значение - адрес первого байта этого предложения.

Префикс — символическое обозначение элемента машинной команды, предназначенного для изменения стандартного действия следующей за ним команды ассемблера.

Имя — идентификатор, отличающий данную директиву от других одноименных директив.

Код операции (КОП) и *директива* — это мнемонические обозначения соответствующей машинной команды, макрокоманды или директивы транслятора.

Операнды — части команды, макрокоманды или директивы ассемблера, обозначающие объекты, над которыми производятся действия.

```
<1>     ; Prg_6_l.asm
<2>     ;Программа преобразования двузначного шестнадцатеричного числа
<3>     ;в символьном виде в двоичное представление.
<4>     ;Вход: исходное шестнадцатеричное число из двух цифр,
<5>     ;вводится с клавиатуры.
<б>     ;Выход: результат преобразования помещается в регистр dl.
<9>     data segment para public "data"       ;сегмент данных
<10>    message db "Введите две шестнадцатеричные цифры,$"
<11>    data ends
<12>    stk segment stack
<13>    db 256 dup ("?")                      ;сегмент стека
<14>    stk ends
<15>    code segment para public "code" ;начало сегмента кода
<16>    main proc                       ;начало процедуры main
<17>    assume cs:code,ds:data,ss:stk
<18>        mov ax,data                 ;адрес сегмента данных в регистр ax
<19>        mov ds,ax                   ;ax в ds
<20>        mov ah,9
<21>        mov dx,offset message
<22>        int 21h
<23>        xor ax,ax                   ;очистить регистр ax
<24>        mov ah,1h                   ;1h в регистр ah
<25>        int 21h                     ;генерация прерывания с номером 21h
```

```
<2б>        mov dl,al           ;содержимое регистра al в регистр dl
<27>        sub dl,30h          ;вычитание: (dl)=(dl)-30h
<28>        cmp dl,9h           ;сравнить (dl) с 9h
<29>        jle M1              ;перейти на метку M1, если dl<9h или dl=9h
<30>        sub dl,7h           ;вычитание: (dl)=(dl)—7h
<31>    M1:                     ;определение метки M1
<32>        mov cl,4h           ;пересылка 4h в регистр cl
<33>        shl dl,cl           ;сдвиг содержимого dl на 4 разряда влево
<34>        int 21h             ;вызов прерывания с номером 21h
<35>        sub al,30h          ;вычитание: (dl)=(dl)—30h
<36>        cmp al,9h           ;сравнить (al) с 9h
<37>        jle M2              ;перейти на метку M2, если al<9h или al=9h
<38>        sub al,7h           ;вычитание: (al)=(al)-7h
<39>    M2:                     ;определение метки M2
<40>        add dl,7h           ;сложение: (dl)=(dl)+(al)
<41>        mov ax,4c00h        ;пересылка 4c00h в регистр ax
<42>        int 21h             ;вызов прерывания с номером 21h
<43>    main endp               ;конец процедуры main
<44>    code ends               ;конец сегмента кода
<45>    end main                ;конец программы с точкой входа main
```

Программы на ассемблере

Алфавит ассемблера:

ASCII_символ_буква — все латинские буквы A - Z, a - z, (прописные и строчные буквы считаются эквивалентными);

decdigit — цифры от 0 до 9;

специальные знаки _ ? @ $ &;

разделители : , . [ ] ( ) < > { } + / * % ! " " ? \ = # ^

Программы на ассемблере

Ключевые слова — служебные символы языка ассемблера:

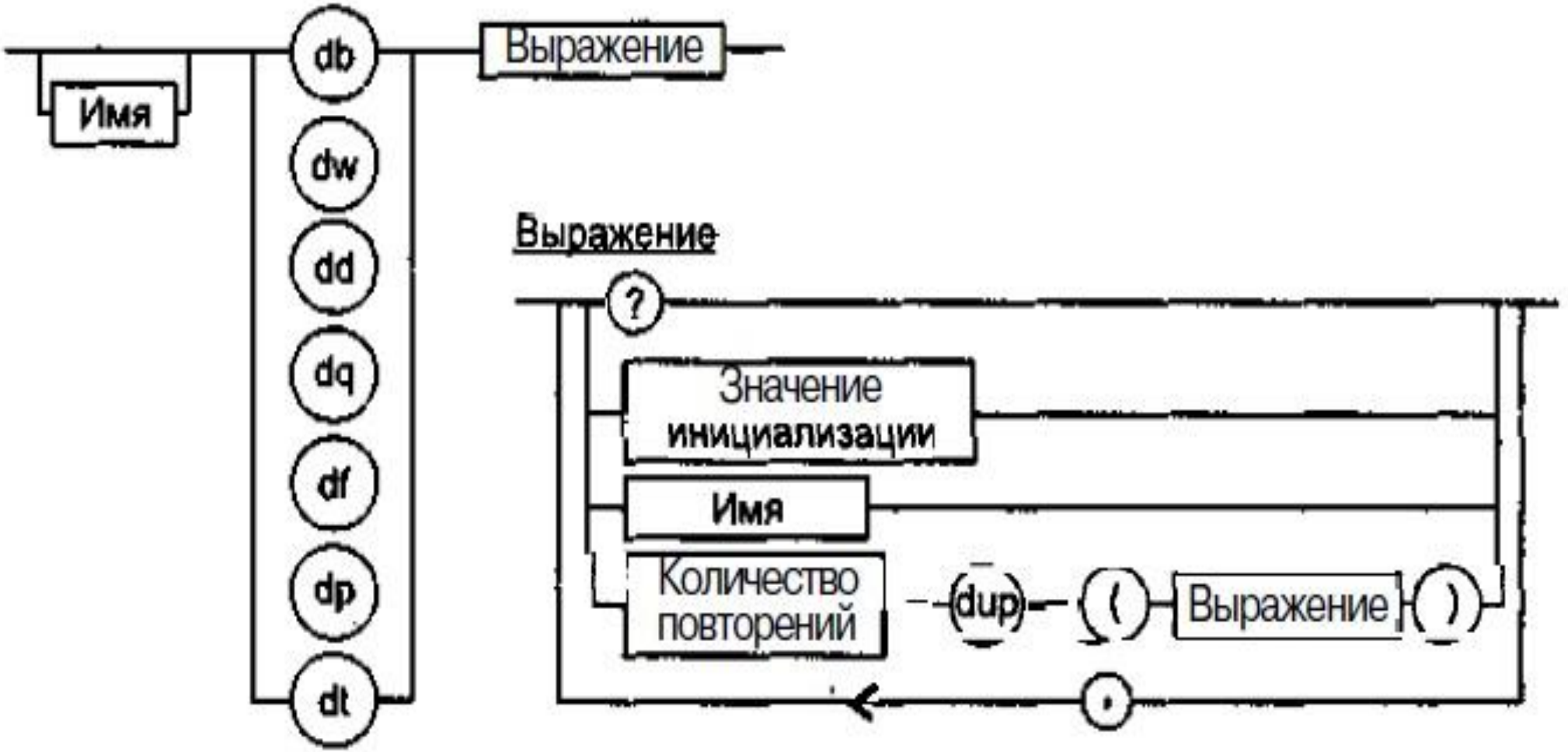
- названия регистров (AL, AH, BL, BH, CL, CH, DL, OH, AX, EAX, BX, EBX, CX, ECX, DX, EDX, BP, EBP, SP, ESP, DI, EDI, SI, ESI, CS, DS, ES, FS, GS, SS, CRO, CR2, CR3, DRO, DRI, DR2, DR3, DR6, DR7);
- операторы (BYTE, SBYTE, WORD, SWORD, DWORD, SDWORD, FWORD, QWORD, TBYTE, REAL4, REALS, REAL10, NEAR16, NEAR32, FAR16, FAR32, AND, NOT, HIGH, LOW, HIGHWORD, LOWWORD, OFFSET, SEG, LROFFSET, TYPE, THIS, PTR, WIDTH, MASK, SIZE, SIZEOF, LENGTH, LENGTHOF, ST, SHORT, TYPE, OPATTR, MOD, NEAR, FAR, OR, XOR, EQ, NE, LT, LE, GT, GE, SHR, SHL);
- названия команд (КОП) ассемблера, префиксов.

Программы на ассемблере

Константы ассемблера :

1. Двоичная – последовательность из цифр 0 и 1, заканчивающаяся буквой B; например 10011001B.
2. Десятичная – последовательность из цифр от 0 до 9, которая может заканчиваться буквой D; например 129D или 129.
3. Десятичная – последовательность из цифр от 0 до 9 и букв от A до F, заканчивающаяся буквой H. Первым символом должна быть одна из цифр от 0 до 9; например 0E23H.
4. Литерал – строка букв, цифр и других символов, заключенная в кавычки или апострофы.

Программы на ассемблере



Директивы описания данных простых типов

Программы на ассемблере

DB — резервирование памяти для данных размером 1 байт. Директивой DB можно задавать следующие значения:

- выражение или константу, принимающую значение из диапазона -128...+127 (для чисел со знаком) или 0...255 (для чисел без знака);
- 8-разрядное относительное выражение, использующее операции HIGH и LOW;
- символьную строку из одного или более символов, которая заключается в кавычки (в этом случае определяется столько байтов, сколько символов в строке).

Программы на ассемблере

DW — резервирование памяти для данных размером два байта. Директивой DW можно задавать следующие значения:

- выражение или константу, принимающую значение из диапазона -32 768...32 767 (для чисел со знаком) или 0...65 535 (для чисел без знака);
- выражение, занимающее 16 или менее битов, в качестве которого может выступать смещение в 16-битовом сегменте или адрес сегмента;
- 1-или 2-байтовая строка, заключенная в кавычки.

Программы на ассемблере

DD — резервирование памяти для данных размером четыре байта. Директивой DD можно задавать следующие значения:

- выражение или константу, принимающую значение из диапазона -32 768...+32 767 (для чисел со знаком и процессора i8086), 0...65 535 (для чисел без знака и процессора i8086), -2 147 483 648...+2 147 483 647 (для чисел со знаком и процессора i386 и выше) или 0...4 294 967 295 (для чисел без знака и процессора i386 и выше);
- относительное или адресное выражение, состоящее из 16-разрядного адреса сегмента и 16-разрядного смещения;
- строку длиной до 4 символов, заключенную в кавычки.

Программы на ассемблере

DQ — резервирование памяти для данных размером 8 байтов. Директивой DQ можно задавать следующие значения:

- выражение или константу, принимающую значение из диапазона -32 768...+32 767 (для чисел со знаком и процессора 18086), 0...65 535 (для чисел без знака и процессора i8086), -2 147 483 648...+2 147 483 647 (для чисел со знаком и процессора i386 и выше) или 0...4 294 967 295 (для чисел без знака и процессора i386 и выше);
- относительное или адресное выражение, состоящее из 32 или менее битов (для i80386) или 16 или менее битов (для первых моделей процессоров Intel);
- константу со знаком из диапазона $-2^{63}...2^{63}- 1$;
- константу без знака из диапазона $0...2^{64}- 1$;
- строку длиной до 8 байтов, заключенную в кавычки.

Программы на ассемблере

DT — резервирование памяти для данных размером 10 байтов. Директивой DT можно задавать следующие значения:

- выражение или константу, принимающую значение из диапазона -32 768...+32 767 (для чисел со знаком и процессора i8086), 0...65 535 (для чисел без знака и процессора i8086), -2 147 483 648...+2 147 483 647 (для чисел со знаком и процессора i386 и выше) или 0...4 294 967 295 (для чисел без знака и процессора 1386 и выше);
- относительное или адресное выражение, состоящее из 32 или менее *битов* (для i80386) или 16 или менее битов (для первых моделей);
- адресное выражение, состоящее из 16-разрядного сегмента и 32-разрядного смещения;
- константу со знаком из диапазона $-2^{79}...2^{79}-1$;
- константу без знака из диапазона $0...2^{80}-1$;
- строку длиной до 10 байтов, заключенную в кавычки;
- упакованную десятичную константу в диапазоне 0...99 999 999 999 999 999 999.