Введение в язык Ассемблера

1. Формат команды ассемблера:

[имя_метки:]	мнемоника_команды	[операнд(ы)]	[;комментарий]

Основные типы операндов:

- непосредственно значение;
- регистр;
- память.

2. Команды пересылки данных

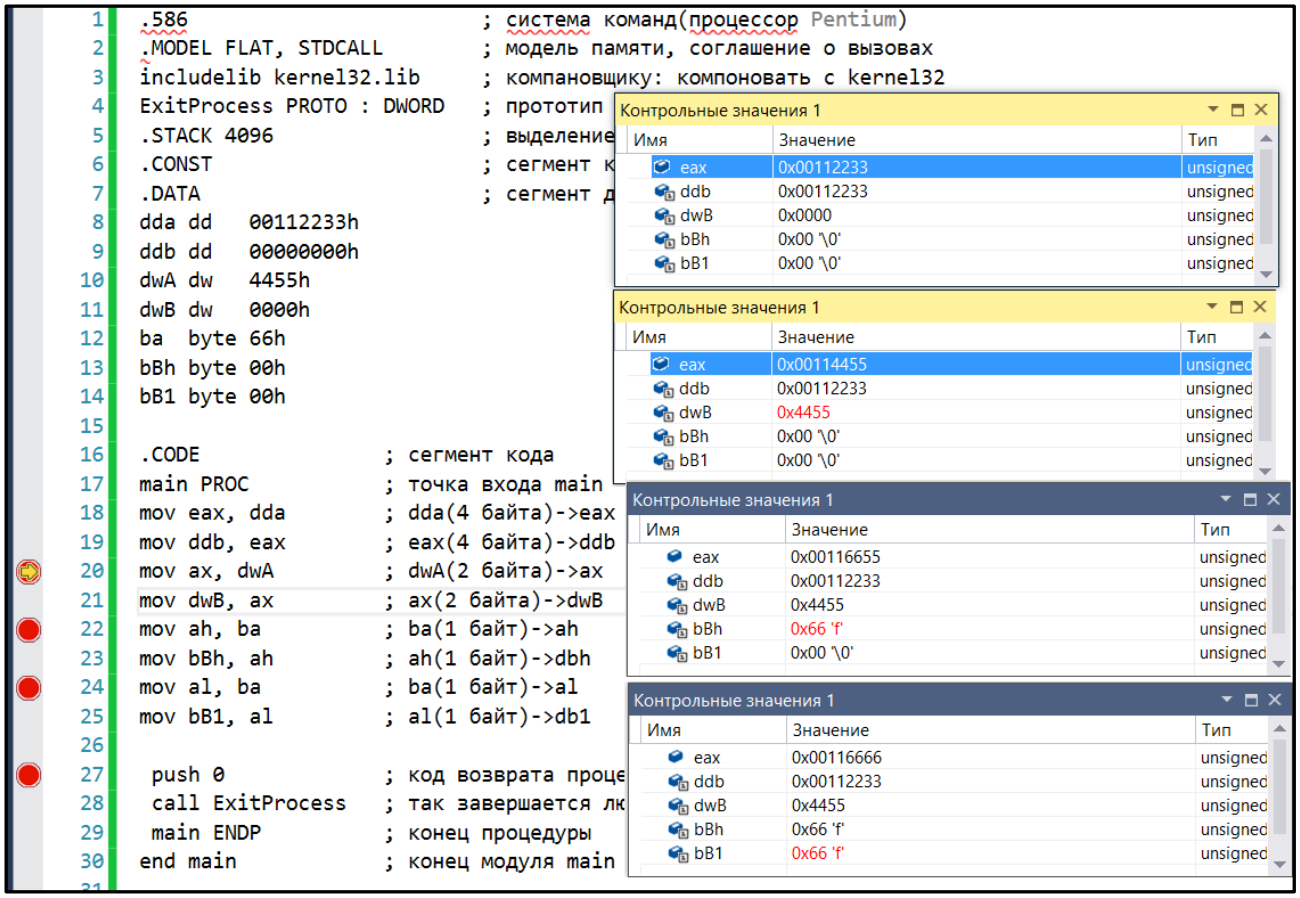
2.1. Команда **MOV** копирует данные из операнда-источника в операнд-получатель.

MOV	получатель	источник

Оба операнда должны быть одного типа и иметь одинаковую длину.
Оба операнда не могут одновременно быть памятью.
В качестве получателя нельзя использовать регистры CS, EIP и IP

MOV - мнемоника команды
Операнды:
«регистр–регистр»;
«регистр–память»;
«память–регистр»;
«регистр–число (непосредственно значение)»;
«память– число».

Использование команды MOV с операндами «память-регистр» и «регистр-память»:



```
.586                    ; система команд(процессор Pentium)
.MODEL FLAT, STDCALL    ; модель памяти, соглашение о вызовах
includelib kernel32.lib ; компановщику: компоновать с kernel32
ExitProcess PROTO : DWORD ; прототип
.STACK 4096             ; выделение
.CONST                  ; сегмент к
.DATA                   ; сегмент д
dda dd   00112233h
ddb dd   00000000h
dwA dw   4455h
dwB dw   0000h
ba  byte 66h
bBh byte 00h
bB1 byte 00h

.CODE                   ; сегмент кода
main PROC               ; точка входа main
mov eax, dda            ; dda(4 байта)->eax
mov ddb, eax            ; eax(4 байта)->ddb
mov ax, dwA             ; dwA(2 байта)->ax
mov dwB, ax             ; ax(2 байта)->dwB
mov ah, ba              ; ba(1 байт)->ah
mov bBh, ah             ; ah(1 байт)->dbh
mov al, ba              ; ba(1 байт)->al
mov bB1, al             ; al(1 байт)->db1

 push 0                 ; код возврата проце
 call ExitProcess       ; так завершается лк
 main ENDP              ; конец процедуры
end main                ; конец модуля main
```

Контрольные значения 1

Имя	Значение	Тип
eax	0x00112233	unsigned
ddb	0x00112233	unsigned
dwB	0x0000	unsigned
bBh	0x00 '\0'	unsigned
bB1	0x00 '\0'	unsigned

Контрольные значения 1

Имя	Значение	Тип
eax	0x00114455	unsigned
ddb	0x00112233	unsigned
dwB	0x4455	unsigned
bBh	0x00 '\0'	unsigned
bB1	0x00 '\0'	unsigned

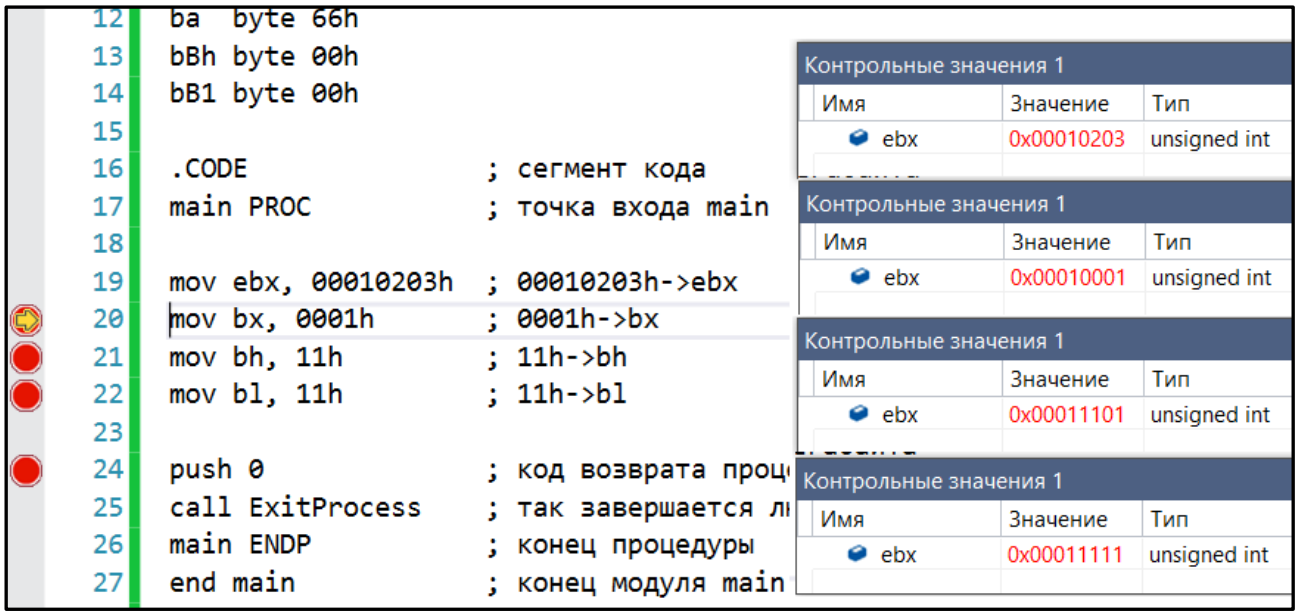
Контрольные значения 1

Имя	Значение	Тип
eax	0x00116655	unsigned
ddb	0x00112233	unsigned
dwB	0x4455	unsigned
bBh	0x66 'f'	unsigned
bB1	0x00 '\0'	unsigned

Контрольные значения 1

Имя	Значение	Тип
eax	0x00116666	unsigned
ddb	0x00112233	unsigned
dwB	0x4455	unsigned
bBh	0x66 'f'	unsigned
bB1	0x66 'f'	unsigned

Использование команды MOV с операндами «регистр-число»:



```
ba  byte 66h
bBh byte 00h
bB1 byte 00h

.CODE                   ; сегмент кода
main PROC               ; точка входа main

mov ebx, 00010203h      ; 00010203h->ebx
mov bx, 0001h           ; 0001h->bx
mov bh, 11h             ; 11h->bh
mov bl, 11h             ; 11h->bl

push 0                  ; код возврата проц
call ExitProcess        ; так завершается л
main ENDP               ; конец процедуры
end main                ; конец модуля main
```

Контрольные значения 1

Имя	Значение	Тип
ebx	0x00010203	unsigned int

Контрольные значения 1

Имя	Значение	Тип
ebx	0x00010001	unsigned int

Контрольные значения 1

Имя	Значение	Тип
ebx	0x00011101	unsigned int

Контрольные значения 1

Имя	Значение	Тип
ebx	0x00011111	unsigned int

Использование команды MOV с операндами «память-число»:

```
.DATA; сегмент данных
dda dd   00112233h
ddb dd   00000000h
dwA dw   4455h
dwB dw   0000h
ba  byte 66h
bBh byte 00h
bBl byte 00h

.CODE                ; сегмент кода
main PROC            ; точка входа main

mov ddb, 00010203h   ; 00010203h->ddb
mov dwB, 0001h       ; 0001h->dwB
mov bBh, 11h         ; 11h->bBh
mov bBl, 11h         ; 11h->bBl

push 0               ; код возврата процесса Windows(параметр ExitProcess)
call ExitProcess     ; так завершается любой процесс Windows
main ENDP            ; конец процедуры
end main             ; конец модуля main
```

Контрольные значения 1

Имя	Значение	Тип
ddb	0x00010203	unsigned long
dwB	0x0001	unsigned short
bBh	0x11 '\x11'	unsigned char
bBl	0x11 '\x11'	unsigned char

2.2. Команда расширения целых беззнаковых чисел MOVZX (***move with zero-extend***) копирует содержимое исходного операнда в больший по размеру регистр получателя данных. При этом оставшиеся ***неопределенными биты*** регистра-получателя (это старшие 16 или 24 бита) сбрасываются в ноль.

```
.586
.MODEL flat, stdcall
includelib kernel32.lib
ExitProcess PROTO :DWORD
.stack 4096
.const
.data                    ; сегмент данных
  ddb       dd      00000000h
  dwA       dw      4455h
  dwb       dw      0000h
  bA1       byte    66h
  bA2       byte    77h
  bBh       byte    00h
  bB1       byte    00h

.CODE                    ; сегмент кода
  main PROC              ; начало процедуры

 mov    ecx,    0ffffffh
 movzx  ecx,    dwA          ; dwA -> ecx
 mov    ecx,    0ffffffh
 movzx  ecx,    bA1          ; bA1 -> ecx
 mov    ecx,    0ffffffh
 movzx  cx, bA2              ; bA2 -> ch

  push  0                ; код возврата (параметр
  call  ExitProcess      ; завершение процесса Wi
 main ENDP               ; конец процедуры
end main                 ; конец модуля, точка вх
```

Контрольные значения 1

Имя	Значение	Тип
ecx	0x0fffffff	unsigned int

Контрольные значения 1

Имя	Значение	Тип
ecx	0x00004455	unsigned int

Контрольные значения 1

Имя	Значение	Тип
ecx	0x0fffffff	unsigned int

Контрольные значения 1

Имя	Значение	Тип
ecx	0x00000066	unsigned int

Контрольные значения 1

Имя	Значение	Тип
ecx	0x0fffffff	unsigned int

Контрольные значения 1

Имя	Значение	Тип
ecx	0x0fff0077	unsigned int

```
    bA1  byte 66h
    bA2  byte 77h
    bBh  byte 00h
    bBl  byte 00h
  .code
   main PROC

     mov     eax, 00112233h
     mov     ebx, 44556677h
     mov     ecx, 8899AABBh
     mov     eax, ebx
     mov     ax,  cx
     mov     ah,  bh
     mov     al,  bl

     push  0
     call ExitProcess
   main ENDP
  end main
```

Контрольные значения 1

Имя	Значение	Тип
eax	0x00112233	unsigned int
ebx	0x44556677	unsigned int
ecx	0x8899aabb	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x44556677	unsigned int
ebx	0x44556677	unsigned int
ecx	0x8899aabb	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x4455aabb	unsigned int
ebx	0x44556677	unsigned int
ecx	0x8899aabb	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x44556677	unsigned int
ebx	0x44556677	unsigned int
ecx	0x8899aabb	unsigned int

```
.586
.MODEL flat, stdcall
includelib kernel32.lib
ExitProcess PROTO :DWORD
.stack 4096
.const
.data                   ; сегмент данных
  ddb       dd      00000000h
  dwA       dw      4455h
  dwb       dw      0000h
  bA1       byte    66h
  bA2       byte    77h
  bBh       byte    00h
  bB1       byte    00h

.CODE                   ; сегмент кода
  main PROC             ; начало процедуры

 mov    eax,    00112233h
 mov    ebx,    44556677h
 movzx  eax,    bx
 movzx  eax,    bh
 movzx  eax,    bl
 mov    eax,    00112233h

  push  0               ; код возврата (па
  call  ExitProcess     ; завершение процесса Windows
 main ENDP              ; конец процедуры
end main                ; конец модуля, точка входа main
```

Контрольные значения 1

Имя	Значение	Тип
eax	0x00112233	unsigned int
ebx	0x44556677	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x00006677	unsigned int
ebx	0x44556677	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x00000066	unsigned int
ebx	0x44556677	unsigned int

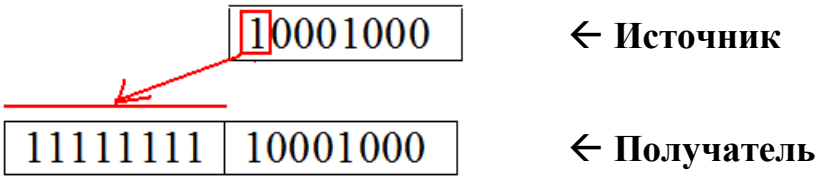
Контрольные значения 1

Имя	Значение	Тип
eax	0x00000077	unsigned int
ebx	0x44556677	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x00112233	unsigned int
ebx	0x44556677	unsigned int

2.3. Команда расширения целых беззнаковых чисел MOVSX (*move with sing-extend* – переместить и дополнить знаком) копирует содержимое исходного операнда в больший по размеру регистр получателя данных. При этом оставшиеся неопределенными биты регистра-получателя (это старшие 16 или 24 бита) дополняет значением ***знакового бита*** исходного операнда.



← **Источник**

← **Получатель**

2.4. Команда XCHG (***exchange data*** – обмен данными) обменивает содержимое двух операндов (! длины операндов должны совпадать !).

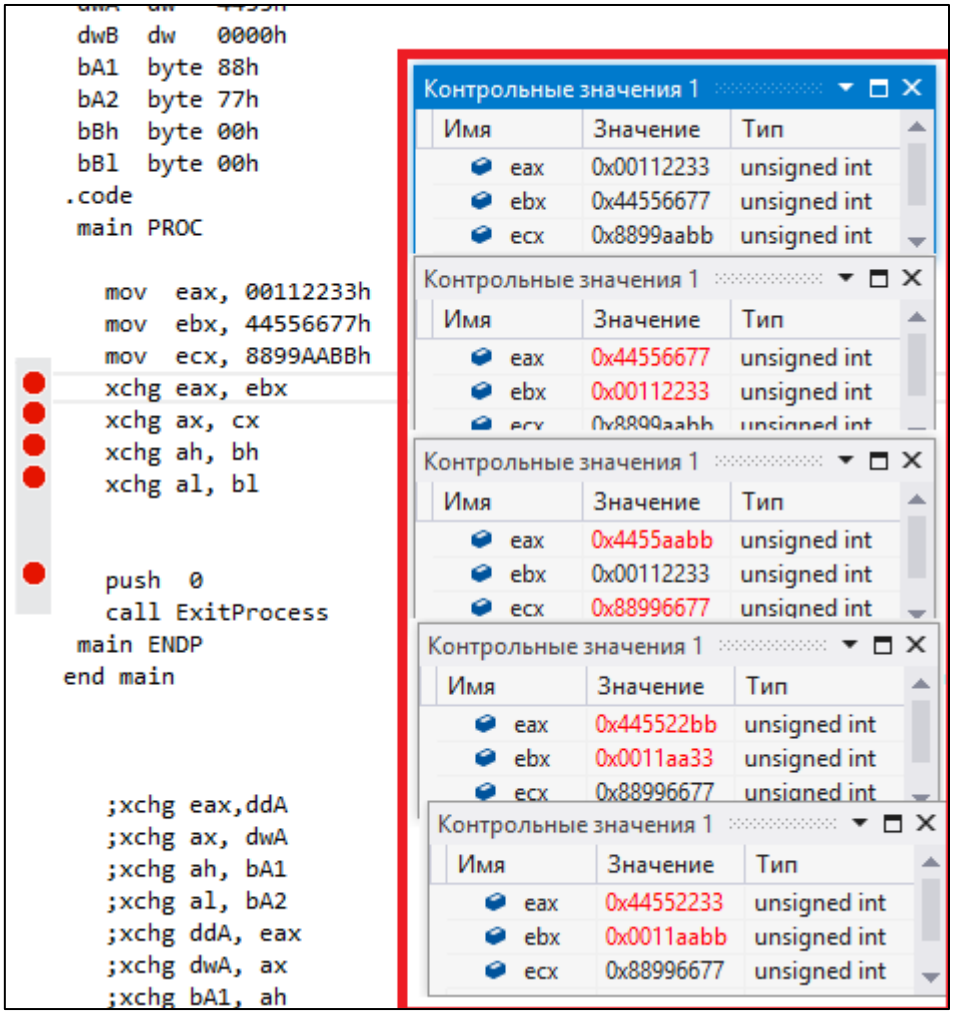
Допустимые операнды:

«регистр-регистр»;

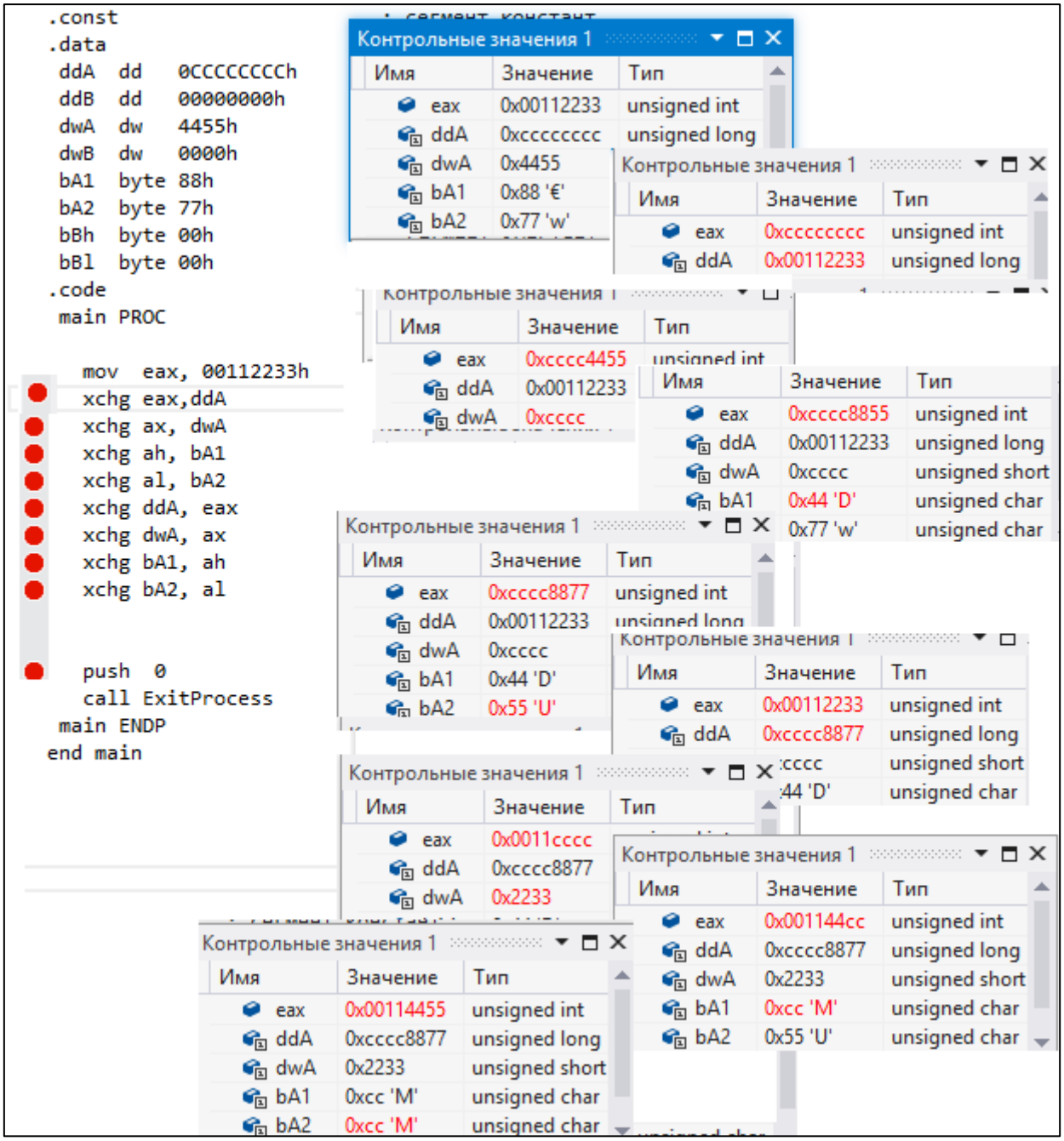
«регистр-память»;

«память-регистр».

Применение для операндов «регистр - регистр»:



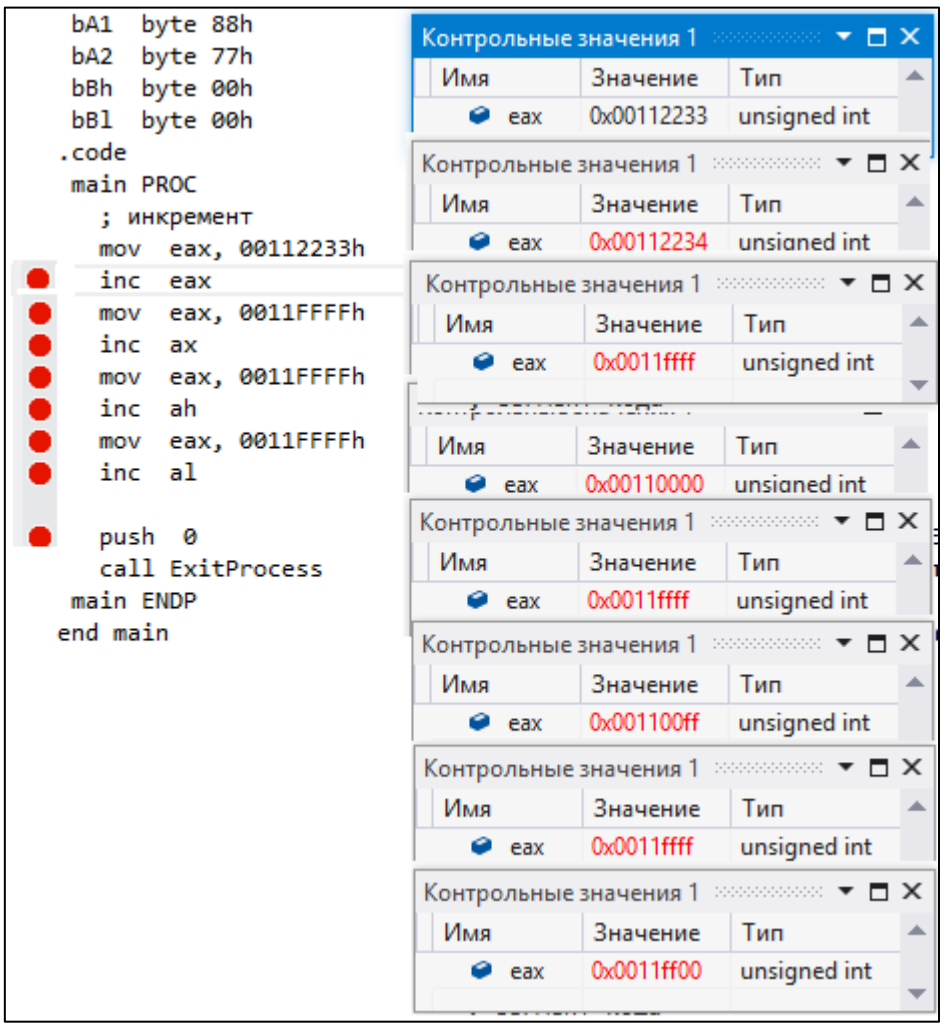
Применение для операндов «регистр-память», «память регистр»



3. Целочисленное сложение и вычитание

3.1. Команды **INC** и **DEC** прибавляют и вычитают единицу из указанного операнда.

INC: операнд регистр:



INC: операнд память:

```
.model flat,stdcall          ; модель памяти, соглашение
includelib kernel32.lib      ; компановщику: компоновать
ExitProcess PROTO   :DWORD   ; прототип функции
.stack 4096                  ; сегмент стека объемом  409
.const                       ; сегмент констант
.data                        ; сегмент данных
 ddA  dd   0CCCCCCCCh
 ddB  dd   00000000h
 dwA  dw   4455h
 dwB  dw   0000h
 bA1  byte 88h
 bA2  byte 77h
 bBh  byte 00h
 bBl  byte 00h
.code
 main PROC
   ; инкремент
   inc  ddA
   inc  dwA
   inc  bA1

   push  0
   call ExitProcess
```

Контрольные значения 1

Имя	Значение	Тип
ddA	0xcccccccc	unsigned long
dwA	0x4455	unsigned short
bA1	0x88 '€'	unsigned char

Контрольные значения 1

Имя	Значение	Тип
ddA	0xcccccccd	unsigned long
dwA	0x4456	unsigned short
bA1	0x89 '‰'	unsigned char

```
.data                        ; сегмент данных
 ddA  dd   0CCCCCCCCh
 ddB  dd   00000000h
 dwA  dw   4455h
 dwB  dw   0000h
 bA1  byte 88h
 bA2  byte 77h
 bBh  byte 00h
 bBl  byte 00h
.code
 main PROC
   ; декремент
   mov eax, 00112233h
   mov ebx, 44550000h
   mov ecx, 77880000h
   dec eax
   dec bx
   dec ch
   dec cl

   push  0                   ; код возрата процесса (параметр
   call ExitProcess          ; так должен заканчиваться любой
 main ENDP                   ; конец процедуры
end main                     ; конец модуля, main - точка вход
```

Контрольные значения 1

Имя	Значение	Тип
eax	0x00112233	unsigned int
ebx	0x44550000	unsigned int
ecx	0x77880000	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x00112232	unsigned int
ebx	0x4455ffff	unsigned int
ecx	0x7788ffff	unsigned int

```
.stack 4096                    ; сегмент стека объемом  4096
.const                         ; сегмент констант
.data                          ; сегмент данных
 ddA  dd   0CCCCCCCCh
 ddB  dd   00000000h
 dwA  dw   4455h
 dwB  dw   0000h
 bA1  byte 88h
 bA2  byte 77h
 bBh  byte 00h
 bBl  byte 00h
.code
 main PROC
   ; декремент
   dec  ddA
   dec  dwA
   dec  bA1

   push  0
   call ExitProcess            ; так должен заканчиваться лк
 main ENDP                     ; конец процедуры
end main                       ; конец модуля, main - точка
```

Контрольные значения 1

Имя	Значение	Тип
ddA	0xcccccccc	unsigned long
dwA	0x4455	unsigned short
bA1	0x88 '€'	unsigned char

Контрольные значения 1

Имя	Значение	Тип
ddA	0xcccccccb	unsigned long
dwA	0x4454	unsigned short
bA1	0x87 '‡'	unsigned char

3.2. Команда **ADD** прибавляет операнд-источник к операнду получателю. Команда **ADD** изменяет значения флагов переноса, переполнения, знака и др. (о флагах позже).

```
ExitProcess PROTO   :DWORD  : прототип функции
.stack 4096
.const
.data
 ddA  dd   0CCCCCCCCh
 ddB  dd   00000000h
 dwA  dw   4455h
 dwB  dw   0000h
 bA1  byte 88h
 bA2  byte 77h
 bBh  byte 00h
 bBl  byte 00h
.code
 main PROC
   ; декремент
   mov eax, 00112233h
   mov ebx, 44556677h
   mov ecx, 8899AABBh
   add eax, ebx
   add bx,  cx
   add ch,  bl
   add cl,  bh


   push  0
   call ExitProcess
 main ENDP
end main
```

Контрольные значения 1

Имя	Значение	Тип
eax	0x00112233	unsigned int
ebx	0x44556677	unsigned int
ecx	0x8899aabb	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x446688aa	unsigned int
ebx	0x44556677	unsigned int
ecx	0x8899aabb	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x446688aa	unsigned int
ebx	0x44551132	unsigned int
ecx	0x8899aabb	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x446688aa	unsigned int
ebx	0x44551132	unsigned int
ecx	0x8899dcbb	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x446688aa	unsigned int
ebx	0x44551132	unsigned int
ecx	0x8899dccc	unsigned int

```
.const                          ; сегмент констант
.data                           ; сегмент данных
 ddA  dd   11111111h
 ddB  dd   00000000h
 dwA  dw   2222h
 dwB  dw   0000h
 bA1  byte 33h
 bA2  byte 44h
 bBh  byte 00h
 bBl  byte 00h
.code
 main PROC
   ; декремент
   mov eax, 00112233h
   mov ebx, 44556677h
   mov ecx, 8899AABBh
   add eax, ddA
   add bx,  dwa
   add ch,  bA1
   add cl,  bA2

   push  0                      ; код возрата процесса (параметр
   call ExitProcess             ; так должен заканчиваться любой
```

Контрольные значения 1

Имя	Значение	Тип
eax	0x00112233	unsigned int
ebx	0x44556677	unsigned int
ecx	0x8899aabb	unsigned int

Контрольные значения 1

Имя	Значение	Тип
eax	0x11223344	unsigned int
ebx	0x44558899	unsigned int
ecx	0x8899ddff	unsigned int

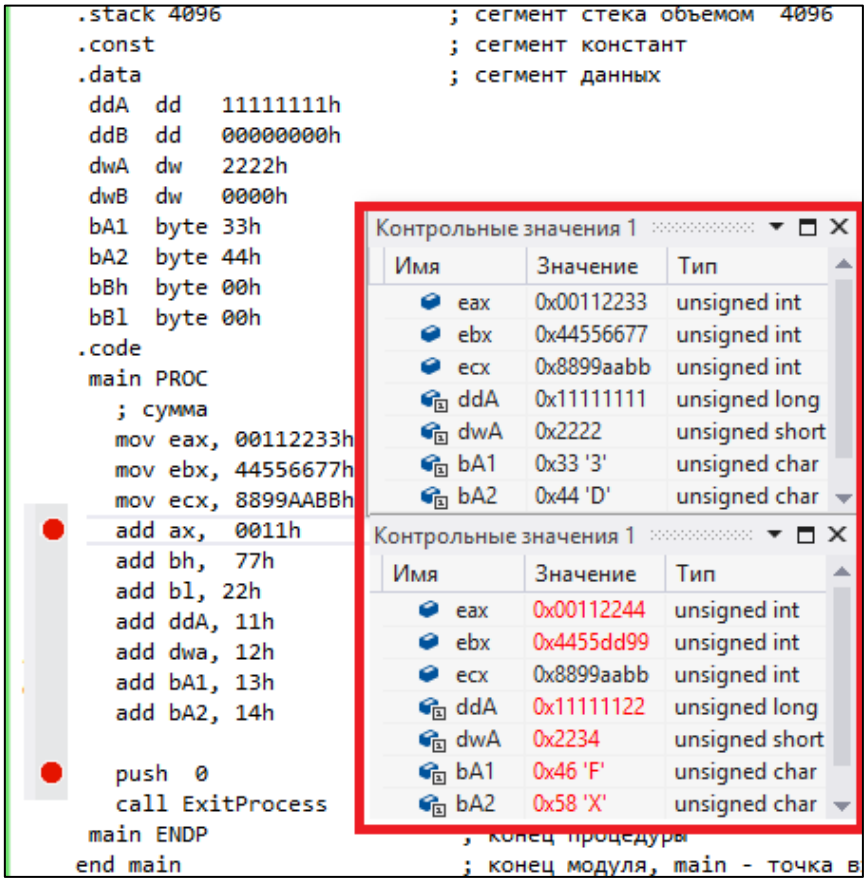
```
 ddA  dd   11111111h
 ddB  dd   00000000h
 dwA  dw   2222h
 dwB  dw   0000h
 bA1  byte 33h
 bA2  byte 44h
 bBh  byte 00h
 bBl  byte 00h
.code
 main PROC
   ; декремент
   mov eax, 00112233h
   mov ebx, 44556677h
   mov ecx, 8899AABBh
   add ddA, eax
   add dwa, bx
   add bA1, ch
   add bA2, cl

   push  0
   call ExitProcess
 main ENDP
end main
```

Контрольные значения 1

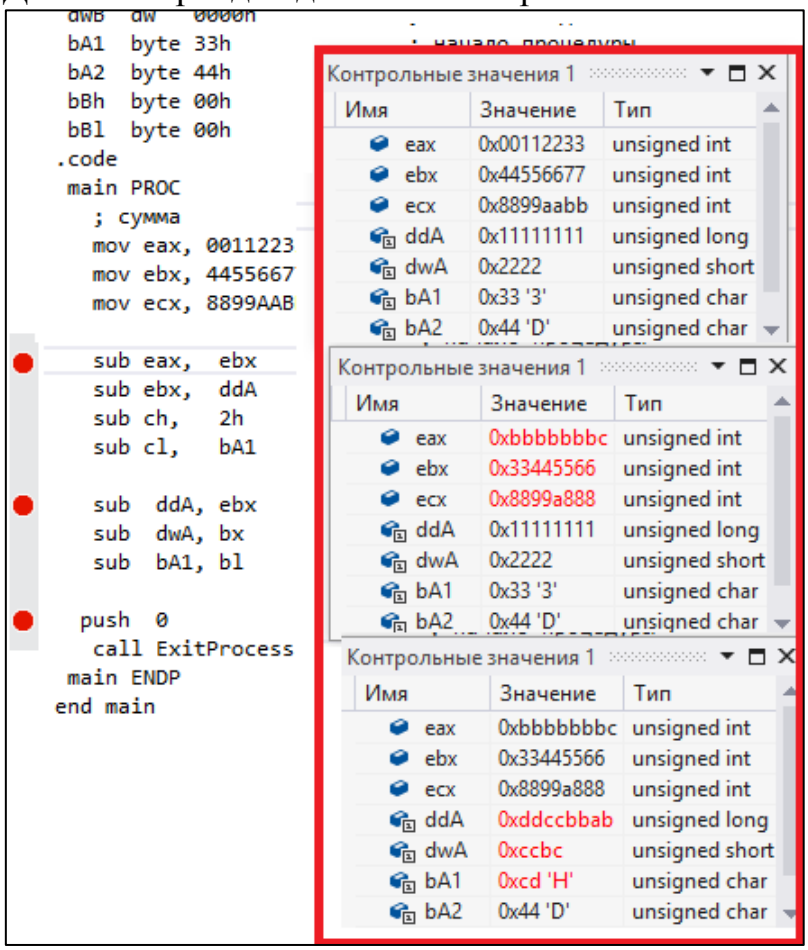
Имя	Значение	Тип
eax	0x00112233	unsigned int
ebx	0x44556677	unsigned int
ecx	0x8899aabb	unsigned int
ddA	0x11111111	unsigned long
dwA	0x2222	unsigned short
bA1	0x33 '3'	unsigned char
bA2	0x44 'D'	unsigned char

Контрольные значения 1

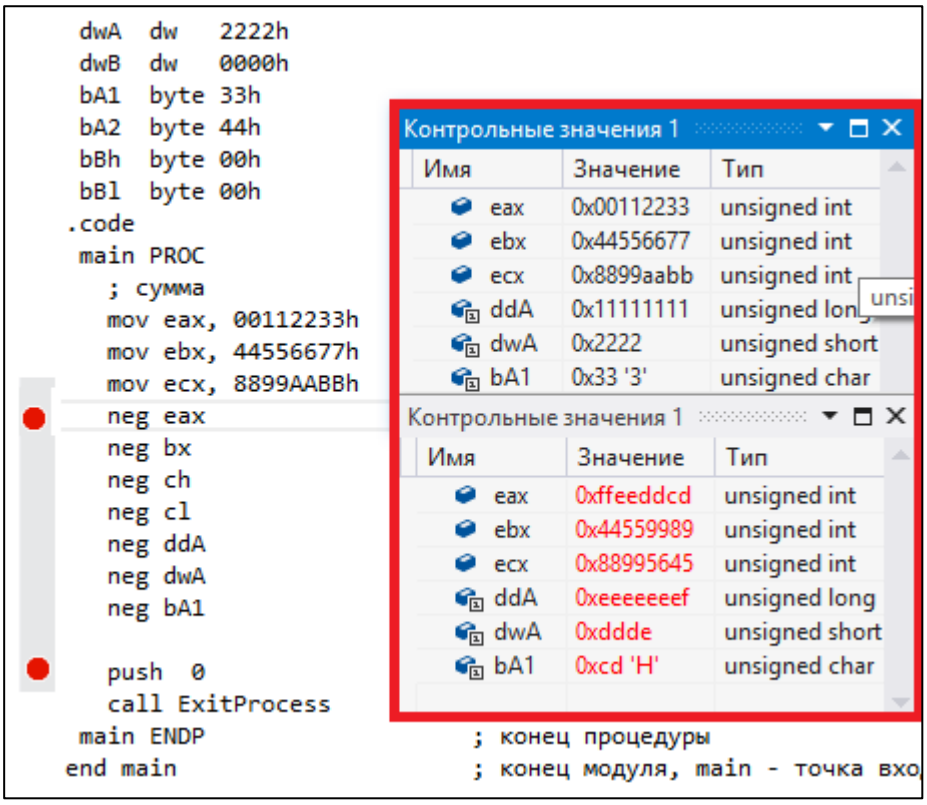
Имя	Значение	Тип
eax	0x00112233	unsigned int
ebx	0x44556677	unsigned int
ecx	0x8899aabb	unsigned int
ddA	0x11223344	unsigned long
dwA	0x8899	unsigned short
bA1	0xdd 'Э'	unsigned char
bA2	0xff 'я'	unsigned char


.stack 4096 ; сегмент стека объемом 4096
.const ; сегмент констант
.data ; сегмент данных
ddA dd 11111111h
ddB dd 00000000h
dwA dw 2222h
dwB dw 0000h
bA1 byte 33h
bA2 byte 44h
bBh byte 00h
bBl byte 00h
.code
main PROC
; сумма
mov eax, 00112233h
mov ebx, 44556677h
mov ecx, 8899AABBh
add ax, 0011h
add bh, 77h
add bl, 22h
add ddA, 11h
add dwa, 12h
add bA1, 13h
add bA2, 14h
push 0
call ExitProcess
main ENDP
end main
; конец процедуры
; конец модуля, main - точка в
Контрольные значения 1
Имя Значение Тип
eax 0x00112233 unsigned int
ebx 0x44556677 unsigned int
ecx 0x8899aabb unsigned int
ddA 0x11111111 unsigned long
dwA 0x2222 unsigned short
bA1 0x33 '3' unsigned char
bA2 0x44 'D' unsigned char
Контрольные значения 1
Имя Значение Тип
eax 0x00112244 unsigned int
ebx 0x4455dd99 unsigned int
ecx 0x8899aabb unsigned int
ddA 0x11111122 unsigned long
dwA 0x2234 unsigned short
bA1 0x46 'F' unsigned char
bA2 0x58 'X' unsigned char

3.3. Команда **SUB** вычитает ***операнд-источник*** из ***операнда получателя*** данных. Процессор заменяет ее на команду сложения с отрицательным числом (отрицательные числа представляются в дополнительном коде).
Длины операндов должны быть равны.

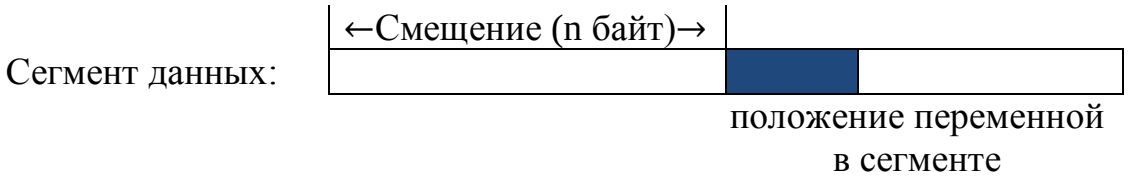


3.4. Команда **NEG** изменяет знак числа на противоположный.



4. Смещение в сегменте данных

Опреатор **OFFSET** – возвращает смещение метки данных относительно начала сегмента. Под смещением понимается то количество байтов, которое отделяет метку данных относительно начала сегмента:




```
ddA  dd   11111111h
ddB  dd   00000000h
dwA  dw   2222h
dwB  dw   0000h
bA1  byte 33h
bA2  byte 44h
bBh  byte 00h
bBl  byte 00h
.code
main PROC
  ; смещение
  mov eax, OFFSET ddA
  mov ebx, OFFSET dwA
  mov ecx, OFFSET bA2


  push  0
  call ExitProcess
```

Контрольные значения 1

Имя	Значение	Тип
eax	0x001b4000	unsigned int
ebx	0x001b4008	unsigned int
ecx	0x001b400d	unsigned int

5. Выравнивание на границу 2, 4, 8, 16

```
.586                            ; система команд (процессор Pentium)
.model flat,stdcall             ; модель памяти, соглашение о вызова
includelib kernel32.lib         ; компановщику: компоновать с kernel
ExitProcess PROTO   :DWORD      ; прототип функции
.stack 4096                     ; сегмент стека объемом  4096
.const                          ; сегмент констант
.data                           ; сегмент данных
 ddA  dd   11111111h
 ddB  dd   00000000h
 dwA  dw   2222h
 dwB  dw   0000h
 bA1  byte 33h
     ; align 2
 bA2  byte 44h
 bBh  byte 00h
    ; align 16
 bBl  byte 00h
.code
 main PROC
   ; смещение
   mov eax, OFFSET bA2
   mov ebx, OFFSET bBl

   push  0                      ; код возрата процесса (параметр Ex
   call ExitProcess             ; так должен заканчиваться любой пр
 main ENDP                      ; конец процедуры
end main                        ; конец модуля, main - точка входа
```

Контрольные значения 1

Имя	Значение	Тип
eax	0x00c2400d	unsigned int
ebx	0x00c2400f	unsigned int

Директива **ALIGN** используется для выравнивания адреса переменной на границу байта, слова (2 байта), двойного слова (4 байта), учетверенного слова или параграфа (16 байтов).

```
.model flat,stdcall             ; модель памяти, соглашение о вызовах
includelib kernel32.lib         ; компановщику: компоновать с kernel32.lib
ExitProcess PROTO   :DWORD      ; прототип функции
.stack 4096                     ; сегмент стека объемом  4096
.const                          ; сегмент констант
.data                           ; сегмент данных
 ddA  dd   11111111h
 ddB  dd   00000000h
 dwA  dw   2222h
 dwB  dw   0000h
 bA1  byte 33h
 align 2       ; выравнивани
 bA2  byte 44h
 bBh  byte 00h
 align 16      ; выравнивани
 bBl  byte 00h
.code                           ; сегмент кода
 main PROC                      ; начало процедуры
   ; смещение
   mov eax, OFFSET bA2
   mov ebx, OFFSET bBl

   push  0                      ; код возрата процесса (параметр ExitProcess )
   call ExitProcess             ; так должен заканчиваться любой процесс Windo
 main ENDP                      ; конец процедуры
end main                        ; конец модуля, main - точка входа
```

Контрольные значения 1

Имя	Значение	Тип
eax	0x00c7400e	unsigned int
ebx	0x00c74010	unsigned int

6. **Директива LABEL** (определение дополнительных имен)

```
.const                          ; сегмент констант
.data                           ; сегмент данных
 lab3 label dword
 ddA  dd   11111111h
 ddB  dd   00000000h
 dwA  dw   2222h
 dwB  dw   0000h
 lab1 label  byte
 lab2 label  dword
 bA1  byte 33h
 bA2  byte 44h
 bBh  byte 00h
 bBl  byte 00h
.code
 main PROC
   ; LABEL
   movzx eax, lab1
   mov   ebx, lab2
   mov   ecx, lab3 + 8


   push  0                      ; код возрата проце
   call ExitProcess             ; так должен заканч
 main ENDP                      ; конец процедуры
end main                        ; конец модуля, mai
```

Контрольные значения 1

Имя	Значение
eax	0x00000033
ebx	0x00004433
ecx	0x00002222

7. **Оператор TYPE** (возвращает размер указанной переменной в байтах)

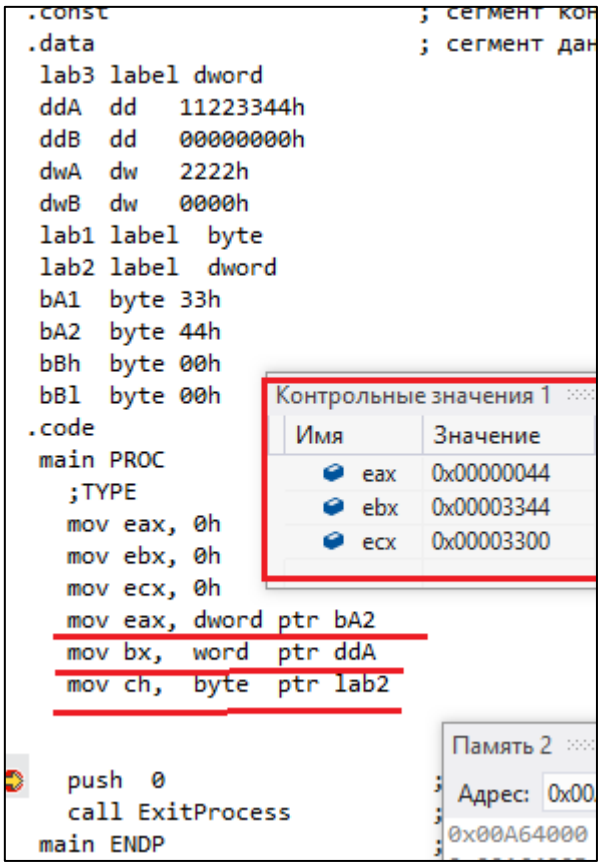
```
.586                            ; система команд (пр
.model flat,stdcall             ; модель памяти, сог
includelib kernel32.lib         ; компановщику: комп
ExitProcess PROTO   :DWORD      ; прототип функции
.stack 4096                     ; сегмент стека объе
.const                          ; сегмент констант
.data                           ; сегмент данных
 lab3 label dword
 ddA  dd   11111111h
 ddB  dd   00000000h
 dwA  dw   2222h
 dwB  dw   0000h
 lab1 label  byte
 lab2 label  dword
 bA1  byte 33h
 bA2  byte 44h
 bBh  byte 00h
 bBl  byte 00h
.code
 main PROC
   ;TYPE
   mov eax, type ddA
   mov ebx, type lab1
   mov ecx, type bBl


   push  0                      ; код возрата проце
   call ExitProcess             ; так должен заканч
```

Контрольные значения 1

Имя	Значение
eax	0x00000004
ebx	0x00000001
ecx	0x00000001

8. Оператор PTR

Если размеры операндов не совпадают, то можно переопределить размер операнда. Например, для загрузки коротких переменных в больший регистр. Для такого уточнения можно использовать оператор переопределения типа **PTR**. Типы могут быть **byte**, **word**, **dword**.



9. Операторы SIZEOF и LENGTHOF

Опреатор LENGTHOF определяет количество элементов в массиве, перечисленного в одной строке.

Опреатор SIZEOF возвращает значение равное произведению значений, возвращаемых операторами LENGTHOF и TYPE.