

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Организация ЭВМ и систем»
Тема: Изучение режимов адресации и
формирования исполнительного адреса.

Студент гр. 9383

Орлов Д.С.

Преподаватель

Ефремов М.А.

Цель работы.

Найти и закомментировать ошибки в представленной программе на Ассемблере. Изучить режимы адресации и формирования исполнительного адреса в Ассемблере.

Задание

Вариант 4.

Лабораторная работа 2 предназначена для изучения режимов адресации, использует готовую программу **lr2_comp.asm** на Ассемблере, которая в автоматическом режиме выполняться не должна, так как не имеет самостоятельного функционального назначения, а только тестирует режимы адресации. Поэтому ее выполнение должно производиться под управлением отладчика в пошаговом режиме.

В программу введен ряд ошибок, которые необходимо объяснить в отчете по работе, а соответствующие команды закомментировать для прохождения трансляции.

Необходимо составить протокол выполнения программы в пошаговом режиме отладчика по типу таблицы 1 предыдущей лабораторной работы и подписать его у преподавателя.

На защите студенты должны уметь объяснить результат выполнения каждой команды с учетом используемого вида адресации. Результаты, полученные с помощью отладчика, не являются объяснением, а только должны подтверждать ваши объяснения.

Содержимое файла **lr2_comp.asm** см. в приложении А.

Ошибки в коде.

Сообщение об ошибке:

lr2_comp.asm(45): error A2052: Improper operand type

Строчка кода:

```
mov  
mem3,[bx]
```

Пояснение:

Нельзя читать из памяти и писать в память одной командой. (делать `mov` из памяти в память).

Сообщение об ошибке:

lr2_comp.asm(60): error A2055: Illegal register value

Строчка кода:

mov ax,matr[bx*4][di]

Пояснение:

Нельзя умножать 16-битные регистры (BX).

Сообщение об ошибке:

lr2_comp.asm(83): error A2046: Multiple base registers

Строчка кода:

mov ax,matr[bp+bx]

Пояснение:

Нельзя использовать более одного базового регистра.

Сообщение об ошибке:

lr2_comp.asm(84): error A2047: Multiple index registers

Строчка кода:

mov ax,matr[bp+di+si]

Пояснение:

Нельзя использовать более одного индексного регистра.

Исправление ошибок.

Содержимое сегментных регистров до старта

программы: CS : 1A0A DS : 19F5 ES : 19F5 HS : 19F5

SS : 1A0A

FS : 19F5

CX: 00B8

Адрес Команды	Символический код команды	16- ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения .	После выполнения
0000	PUSH DS	1E	(SP) = 0018 (DS) = 19F5 (IP) = 0000 Stack:+0 0000	(SP) = 0016 (DS) = 19F5 (IP) = 0001 Stack: +0 19F5
0001	SUB AX, AX	2BC0	(AX) = 0000 (IP) = 0001	(AX) = 0000 (IP) = 0003
0003	PUSH AX	50	(SP) = 0016 (AX) = 0000 (IP) = 0003 Stack: +0 19F5 +2 0000	(SP) = 0014 (IP) = 0004 (AX) = 0000 Stack: +0 0000 +0 19F5

0004	MOV AX, 1A07	B8071 A	(AX) = 0000 (IP) = 0004	(AX) = 1A07 (IP) = 0007
0007	MOV DS, AX	8ED8	(DS) = 19F5 (IP) = 0007	(DS) = 1A07 (IP) = 0009
0009	MOV AX, 01F4	B8F40 1	(AX) = 1A07 (IP) = 0009	(AX) = 01F4 (IP) = 000C
000C	MOV CX, AX	8BC8	(CX) = 00B8 (IP) = 000C	(CX) = 01F4 (IP) = 000E
000E	MOV BL, 24	B324	(BX) = 0000 (IP) = 000E	(BX) = 0024 (IP) = 0010
0010	MOV BH, CE	B7CE	(BX) = 0024 (IP) = 0010	(BX) = CE24 (IP) = 0012
0012	MOV [0002], FFCE	C7060200CE FF	(IP) = 0010	(IP) = 0018
0018	MOV BX, 0006	BB060 0	(BX) = CE24 (IP) = 0018	(BX) = 0006 (IP) = 001B
001B	MOV [0000], AX	A3000 0	(IP) = 001B	(IP) = 001E
001E	MOV AL, [BX]	8A07	(AX) = 01F4 (IP) = 001E	(AX) = 010C (IP) = 0020
0020	MOV AL, [BX+03]	8A470 3	(AX) = 010C (IP) = 0020	(AX) = 0109 (IP) = 0023
0023	MOV CX, [BX+03]	8B4F0 3	(CX) = 01F4 (IP) = 0023	(CX) = 0509 (IP) = 0026
0026	MOV DI, 0002	BF020 0	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	MOV AL, [000E+DI]	8A850 E00	(AX) = 0109 (IP) = 0029	(AX) = 0128 (IP) = 002D
002D	MOV CX, [000E + DI]	8B8D0 E00	(CX) = 0509 (IP) = 002D	(CX) = 3228 (IP) = 0031
0031	MOV BX, 0003	BB030 0	(BX) = 0006 (IP) = 0034	(BX) = 0003 (IP) = 0034
0034	MOV AL, [0016+BX+DI]	8A811 600	(AX) = 0128 (IP) = 0038	(AX) = 01F9 (IP) = 0038
0038	MOV CX, [0016+BX+DI]	8B891 60	(CX) = 3228 (IP) = 0038	(CX) = FAF9 (IP) = 003C

003C	MOV AX, 1A07	B8071 A	(AX) = 01F9 (IP) = 003C	(AX) = 1A07 (IP) = 003F
003F	MOV ES, AX	8EC0	(ES) = 19F5 (IP) = 003F	(ES) = 1A07 (IP) = 0041
0041	MOV AX, ES:[BX]	268B0 7	(AX) = 1A07 (IP) = 0041	(AX) = 00FF (IP) = 0044
0044	MOV AX, 0000	B8000 0	(AX) = 00FF (IP) = 0044	(AX) = 0000 (IP) = 0047
0047	MOV ES, AX	8EC0	(ES) = 1A07 (IP) = 0047	(ES) = 0000 (IP) = 0049
0049	PUSH DS	1E	(DS) = 1A07 (SP) = 0014 (IP) = 0049 Stack: +0 0000 +2 19F5 +6 0000	(DS) = 1A07 (SP) = 0012 (IP) = 004A Stack: +0 1A07 +2 0000 +6 19F5
004A	POP ES	07	(ES) = 0000 (SP) = 0012 (IP) = 004A Stack: +0 1A07 +2 0000 +6 19F5	(ES) = 1A07 (SP) = 0014 (IP) = 004B Stack: +0 0000 +2 19F5 +6 0000
004B	MOV CX, ES:[BX-01]	268B4 FFF	(CX) = 0509 (IP) = 004B	(CX) = FFCE (IP) = 004F
004F	XCHG AX, CX	91	(AX) = 0000 (CX) = FFCE (IP) = 004A	(AX) = FFCE (CX) = 0000 (IP) = 0050
0050	MOV DI, 0002	BF020 0	(DI) = 0002 (IP) = 0050	(DI) = 0002 (IP) = 0053
0053	OV ES:[BX+DI], AX	268901	(IP) = 0053	(IP) = 0056

0056	MOV BP, SP	8BE7	(BP) = 0000 (IP) = 0056	(BP) = 0014 (IP) = 0058
0058	PUSH [0000]	FF360000	(SP) = 0014 (IP) = 0058 Stack: +0 0000 +2 19F5 +4 0000 +6 0000	(SP) = 0012 (IP) = 005C Stack: +0 01F4 +2 0000 +4 19F5 +6 0000
005C	PUSH [0002]	FF360200	(SP) = 0012 (IP) = 005C Stack: +0 01F4 +2 0000 +4 19F5 +6 0000	(SP) = 0010 (IP) = 0060 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5
0060	MOV BP, SP	8BEC	(BP) = 0014 (IP) = 0060	(BP) = 0010 (IP) = 0062
0062	MOV DX, [BP+02]	8B5602	(DX) = 0000 (IP) = 0062	(DX) = 01F4 (IP) = 0065
0065	RET Far 0002	CA0200	(SP) = 0010 (CS) = (IP) = 0065 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5	(SP) = 0016 (CS) = 01F4 (IP) = FFCE Stack: +0 19F5 +2 0000 +4 0000 +6 0000

Таблица 1. Результат пошагового выполнения lr2_comp.exe

Выводы.

В процессе выполнения лабораторной работы были найдены, проанализированы и исправлены ошибки в представленной программе. Изучены режимы адресации и формирования исполнительного адреса на языке Ассемблер.

ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Текст файла `lr2_comp.asm`

```
; Программа изучения режимов адресации процессора IntelX86
EOL EQU '$'

    ind
    EQ
    U
    n1
    EQ
    U
    n2
    EQ
    U

; Стек программы
AStack SEGMENT STACK DW
    12 DUP(?)
AStack ENDS

; Данные программы
DATA    SEGMENT
; Директивы описания данных
mem1    DW 0
mem2    DW 0
mem3    DW 0
vec1    DB 12,11,10,9,5,6,7,8
vec2    DB -40,-50,40,50,-20,-30,20,30
matr    DB 5,6,7,8,-8,-7,-6,-5,1,2,3,4,-4,-3,-2,-1
DATA    ENDS

; Код программы
CODE    SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack
; Головная процедура
Main    PROC FAR
    push    DS
    sub     AX,AX
    push    AX
    mov     AX,DATA
    mov     DS,AX
    ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ СМЕЩЕНИЙ
; Регистровая адресация mov
    ax,n1
```

```

mov      cx,ax
mov      bl,EOL
mov      bh,n2
; Прямая адресация mov
mov      mem2,n2
mov      bx,OFFSET vec1
mov      mem1,ax
; Косвенная адресация mov
mov      al,[bx]
mov      mem3,[bx]
; Базированная адресация mov
mov      al,[bx]+3
mov      cx,3[bx]
; Индексная адресация mov
mov      di,ind
mov      al,vec2[di]
mov      cx,vec2[di]
; Адресация с базированием и индексированием mov
mov      bx,3
mov      al,matr[bx][di]
mov      cx,matr[bx][di]
mov      ax,matr[bx*4][di]
; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ
; Переопределение сегмента
;----- вариант 1
mov      ax, SEG vec2
mov      es, ax
mov      ax, es:[bx]
mov      ax, 0
;----- вариант 2
mov      es, ax
push     ds
pop      es
mov      cx, es:[bx-1]
xchg     cx,ax
;----- вариант 3
mov      di,ind
mov      es:[bx+di],ax
; вариант 4
mov      bp,sp
mov      ax,matr[bp+bx] mov
ax,matr[bp+di+si]
; Использование сегмента стека
push     mem1 push
mem2 mov bp,sp

```



```

mov      dx,[bp]+2
ret      2
Main     ENDP
CODE     ENDS

END      Main

```

Текст файла lr2_comp.lst

Microsoft (R) Macro Assembler Version 5.10 10/28/20 23:55:4

Page 1-1

; Программа изучения режимов адресации процессора IntelX86

= 0024 EOL EQU '\$'

= 0002 ind EQU

= 01F4 n1 EQU

=-0032 n2 EQU

; Стек программы

0000 AStack SEGMENT STACK

0000 000C[DW 12 DUP(?)

????

]

0018 AStack ENDS

; Данные программы

0000 DATA SEGMENT

; Директивы описания данных

(0000 m DW 0

(e

(m

(l

(0000 m DW 0

(e

(m

(2

(0000 m DW 0

(e

(m

(3

(0C 0B 0A 09 ve DB

(05 06 cl 12,11,10,9,5,6,7,

(8

(

07 08

000E D8 CE 28 32 EC E2 vec2 DB -40,-50,40,50,-20,-30,20,30

14 1E

```

0016 05 06 07 08 F8 F9      matr  DB 5,6,7,8,-8,-7,-6,-5,1,2,3,4,-4,-3,-
2,-1
FA FB 01 02 03 04 FC FD
FE FF
0026                          DATA  ENDS

```

; Код программы

```

0000                          CODE  SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack

```

; Головная процедура

```

0000                          Main  PROC FAR
0000 1E                        push  DS
0001 2B C0                     sub   AX,AX
0003 50                         push  AX
0004 B8 ---- R                 mov   AX,DATA
0007 8E D8                     mov   DS,AX
; ПРОВЕРКА РЕЖИМОВ АДРЕСА НА УРОВНЕ СМЕЩЕНИЙ

```

; Регистровая адресация

```

0009 B8 01F4                    mov
000C 8B C8                      m      ax,n1
                                o      cx,ax
                                v
000E B3 24                      m      bl,EOL
                                o
                                v
0010 B7 CE                      m      bh,n2
                                o
                                v

```

; Прямая адресация

```

0012 C7 06 0002 R              m      mem2,n2
FFCE                          o
                                v
0018 BB 0006 R                 m      bx,OFFSET
                                o      vec1
                                v
001B A3 0000 R                 m      mem1,ax
                                o
                                v

```

Microsoft (R) Macro Assembler Version 5.10

10/28/20 23:55:4

Page 1-2

; Косвенная адресация

E 8A 07

```

                                mov   al,[bx]
mov                             mem3,[bx]
LR2_COMP.ASM(45): error A2052: Improper operand type

```

; Базированная адресация

```

0020 8A 47 03                  mov   al,[bx]+3
0023 8B 4F 03                  mov   cx,3[bx]

```

; Индексная адресация

```

0026 BF 0002                  mov   di,ind

```

```

0029 8A 85 000E R      mov    al,vec2[di]
002D 8B 8D 000E R      mov    cx,vec2[di]
LR2_COMP.ASM(52): warning A4031: Operand types must match
; Адресация с базирование индексированием 0031
BB 0003                mov    bx,3
(      8A 81 0016 R      mov    al,matr[bx
(                               ][di]
.
.
(      8B 89 0016 R      mov    cx,matr[bx
(                               ][di]
.
.
LR2_COMP.ASM(56): warning A4031: Operand types must match
003C 8B 85 0022 R      mov    ax,matr[bx*4][di]
LR2_COMP.ASM(57): error A2055: Illegal register value
; ПРОВЕРКА РЕЖИМОВ АДРЕСА С УЧЕТОМ СЕГМЕНТОВ
; Переопределение сегмента
;-----вариант 1
(      B8      R      mov    ax, SEG
(                               vec2
.
(      8E C0                mov    es, ax
(
.
.
(      26: 8B 07                mov    ax,
(                               es:[bx]
.
.
(      B8 0000                mov
(                               ax, 0
.
.
; вариант 2
004B 8E C0                m      es, ax
o
v
004D 1E                pu      ds
sh
004E 07                p      es
o
p
004F 26: 8B 4F FF                m      cx, es:[bx-1]
o
v
0053 91                xc      cx,ax
hg
; вариант 3
0054 BF 0002                mov    di,ind
0057 26: 89 01                m      es:[bx+di],ax
o
v

```

```

; вариант 4
005A 8B EC          m      bp,sp
                   o
                   v

005C 3E: 8B 86 0016      mov
R                    ax,matr[bp+bx]
LR2_COMP.ASM(76): error A2046: Multiple base registers
0061 3E: 8B 83 0016 R      mov  ax,matr[bp+di+si]
LR2_COMP.ASM(77): error A2047: Multiple index registers
; Использование сегмента стека
0066 FF 36 0000 R      push  mem1
006A FF 36 0002 R      push  mem2
006E 8B EC          mov  bp,sp
0070 8B 56 02          mov  dx,[bp]+2
0073 CA 0002          ret  2
0076                  Main  ENDP
LR2_COMP.ASM(84): error A2006: Phase error between passes
0076                  CODE  ENDS

```

```

                                END    Main
Microsoft (R) Macro Assembler Version 5.10                10/28/20 23:55:4
                                Symbols-1

```

Segments and Groups:

<i>N a m e</i>	<i>Length</i>	<i>Align</i>	<i>Combine</i>	<i>Class</i>
ASTACK.....	0018		PARA	STACK
CODE.....	0076		PARA	NONE
DATA.....	0026		PARA	NONE

Symbols:

<i>N a m e</i>	<i>Type</i>	<i>Value</i>	<i>Attr</i>
EOL.....	NUMBE	0	
.	R	0	
		2	
		4	
IND.....	NUMBE	0	
.	R	0	
		0	
		2	
MAIN.....	F PROC	0	CODE Length =
..		0	0076
		0	
		0	

<i>MATR</i>	<i>L BYTE</i>	<i>0016</i>	<i>DATA</i>
<i>MEM1</i>	<i>L WORD</i>	<i>0000</i>	<i>DATA</i>
<i>MEM2</i>	<i>L WORD</i>	<i>0002</i>	<i>DATA</i>
<i>MEM3</i>	<i>L WORD</i>	<i>0004</i>	<i>DATA</i>
<i>N1</i>	<i>NUMBER</i>	<i>01F4</i>	
<i>N2</i>	<i>NUMBER</i>	<i>-0032</i>	
<i>VEC1</i>	<i>L BYTE</i>	<i>0006</i>	<i>DATA</i>
<i>VEC2</i>	<i>L BYTE</i>	<i>000E</i>	<i>DATA</i>
<i>@CPU</i>	<i>TEXT</i>	<i>0101h</i>	
<i>@FILENAME</i>	<i>TEXT</i>	<i>LR2_COMP</i>	
<i>@VERSION</i>	<i>TEXT</i>	<i>510</i>	

87 Source Lines

87 Total Lines

19 Symbols

47784 + 459473 Bytes symbol space free 2

Warning Errors

5 Severe Errors

Текст файла lr2_comp.asm

; Программа изучения режимов адресации процессора IntelX86

EOL EQU '\$'

ind

EQ

U

n1

EQ

U

n2

EQ

U

; Стек программы

AStack SEGMENT STACK DW

12 DUP(?)

AStack ENDS

; Данные программы

DATA SEGMENT

; Директивы описания данных

mem1 DW 0

mem2 DW 0

```

mem3      DW 0
vec1      DB 12,11,10,9,5,6,7,8
vec2      DB -40,-50,40,50,-20,-30,20,30
matr      DB 5,6,7,8,-8,-7,-6,-5,1,2,3,4,-4,-3,-2,-1
DATA      ENDS

; Код программы
CODE      SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack
; Головная процедура
Main      PROC FAR
push      DS
sub        AX,AX
push      AX
mov        AX,DATA
mov        DS,AX
; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ СМЕЩЕНИЙ
; Регистровая адресация mov
          ax,n1
mov        cx,ax
mov        bl,EOL
mov        bh,n2
; Прямая адресация mov
          mem2,n2
mov        bx,OFFSET vec1
mov        mem1,ax
; Косвенная адресация mov
          al,[bx]
;      mov    mem3,[bx]
; Базированная адресация mov
          al,[bx]+3
mov        cx,3[bx]
; Индексная адресация mov
          di,ind
mov        al,vec2[di]
mov        cx,vec2[di]
; Адресация с базированием и индексированием mov
          bx,3
mov        al,matr[bx][di]
mov        cx,matr[bx][di]
;      mov    ax,matr[bx*4][di]
; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ
; Переопределение сегмента
;----- вариант 1
mov        ax, SEG vec2
mov        es, ax
mov        ax, es:[bx]
mov        ax, 0

```

```

; вариант 2
mov     es, ax
push    ds
pop      es
mov     cx, es:[bx-1]
xchg    cx, ax

; вариант 3
mov     di, ind
mov     es:[bx+di], ax

; вариант 4
m      bp, sp
o
v
m      ax, matr[bp+bx
o      ]
v
m      ax, matr[bp+di
o      +si]
v
; Использование сегмента стека push
mem1
push    mem2
mov     bp, sp
mov     dx, [bp]+2
ret     2
Main    ENDP
CODE    ENDS

END     Main

```

Текст файла lr2_comp.lst

Microsoft (R) Macro Assembler Version 5.10

12/16/20 19:03:0

Page 1-1

изучения режии

; Программа

процессора I

¼ов адресации

= 0024

ntelX86

= 0002

EOL EQU '\$'

= 01F4

ind EQU 2

= -0032

n1 EQU 500

n2 EQU -50

0000
STACK

; Стек программы
AStack SEGMENT

0000 000C[DW 12 DUP(?)
	????
]
0018	AStack ENDS
	; Данные
программы	
0000	DATA SEGMENT
	; Директивы
описания даннэ	
	□x
0000 0000	mem1 DW 0
0002 0000	mem2 DW 0
0004 0000	mem3 DW 0
0006 0C 0B 0A 09 05 06	vec1 DB 12,11,10,9,5,6,7,8
07 08	
000E D8 CE 28 32 EC E2	vec2 DB -40,-50,40,50,-20,-
30,20,30	
14 1E	
0016 05 06 07 08 F8 F9	matr DB 5,6,7,8,-8,-7,-6,-5,1,2,3,4,-
4,-3,-2,-1	
FA FB 01 02 03 04	
FC FD FE FF	
0026	DATA ENDS
	; Код программы
0000	CODE SEGMENT
	ASSUME
CS:CODE, DS:DATA, SS:AStack	
	; Головная
процедура	
0000	Main PROC FAR
0000 1E	push DS
0001 2B C0	sub AX,AX
0003 50	push AX
0004 B8 ---- R	mov AX,DATA
0007 8E D8	mov DS,AX
	; ПРОВЕРКА
РЕЖИМОВ АДРЕСАЙ	
СМЕЩЕНИЙ	{ИИ НА УРОВНЕ
	; Регистровая
адресация	
0009 B8 01F4	mov ax,n1
000C 8B C8	mov cx,ax
000E B3 24	mov bl,EOL
0010 B7 CE	mov bh,n2
	; Прямая
адресация	

0012 C7 06 0002 R FFCE
 0018 BB 0006 R
 001B A3 0000 R

mov mem2,n2
 mov bx,OFFSET vec1
 mov mem1,ax

; Косвенная

адресация

Microsoft (R) Macro Assembler Version 5.10

12/16/20 19:03:0

Page 1-2

001E 8A 07

mov al,[bx]
 ;mov mem3,[bx]

; Базированная

адресация

0020 8A 47 03
 0023 8B 4F 03

mov al,[bx]+3
 mov cx,3[bx]

; Индексная

адресация

0026 BF 0002
 0029 8A 85 000E R
 002D 8B 8D 000E R

mov di,ind
 mov al,vec2[di]
 mov cx,vec2[di]

lr2_comp.asm(54): warning A4031: Operand types must match

; Адресация с

базированиеИ

¼ u

индексированием

0031 BB 0003
 0034 8A 81 0016 R
 0038 8B 89 0016 R

mov bx,3
 mov al,matr[bx][di]
 mov cx,matr[bx][di]

lr2_comp.asm(59): warning A4031: Operand types must match

;mov

ax,matr[bx*4][di]

; ПРОВЕРКА

РЕЖИМОВ АДРЕСАИ

ИИ С УЧЕТОМ

СЕГМЕНТОВ

; Переопределение

сегмент

a

; ----- вариант 1

003C B8 ---- R
 003F 8E C0
 0041 26: 8B 07
 0044 B8 0000

mov ax, SEG vec2
 mov es, ax
 mov ax, es:[bx]
 mov ax, 0

; ----- вариант 2

0047 8E C0	mov es, ax
0049 1E	push ds
004A 07	pop es
004B 26: 8B 4F FF	mov cx, es:[bx-1]
004F 91	xchg cx,ax
	; ----- вариант 3
0050 BF 0002	mov di,ind
0053 26: 89 01	mov es:[bx+di],ax
	; ----- вариант 4
0056 8B EC	mov bp,sp
	;mov
ax,matr[bp+bx]	
	;mov
ax,matr[bp+di+si]	
	; Использование
сегмента э	
	• тека
0058 FF 36 0000 R	push mem1
005C FF 36 0002 R	push mem2
0060 8B EC	mov bp,sp
0062 8B 56 02	mov dx,[bp]+2
0065 CA 0002	ret 2
0068	Main ENDP
0068	CODE ENDS
Microsoft (R) Macro Assembler Version 5.10	12/16/20 19:03:0
Page 1-3	

	END Main
Microsoft (R) Macro Assembler Version 5.10	12/16/20 19:03:0
Symbols-1	

Segments and Groups:

<i>N a m e</i>	<i>Length</i>	<i>Align</i>	<i>Combine</i>	<i>Class</i>
ASTACK.....	0018	PARA	STACK	
CODE.....	0068	PARA	NONE	
DATA.....	0026	PARA	NONE	

Symbols:

<i>N a m e</i>	<i>Type</i>	<i>Value</i>	<i>Attr</i>
EOL	NUMBER	0024	

<i>IND</i>	<i>NUMBER</i>	<i>0002</i>	
<i>MAIN</i>	<i>F PROC</i>	<i>0000</i>	<i>CODE</i>
	<i>Length =</i>	<i>0068</i>	
<i>MATR</i>	<i>L BYTE</i>	<i>0016</i>	<i>DATA</i>
<i>MEM1</i>	<i>L WORD</i>	<i>0000</i>	<i>DATA</i>
<i>MEM2</i>	<i>L WORD</i>	<i>0002</i>	<i>DATA</i>
<i>MEM3</i>	<i>L WORD</i>	<i>0004</i>	<i>DATA</i>
<i>N1</i>	<i>NUMBER</i>	<i>01F4</i>	
<i>N2</i>	<i>NUMBER</i>	<i>-0032</i>	
<i>VEC1</i>	<i>L BYTE</i>	<i>0006</i>	<i>DATA</i>
<i>VEC2</i>	<i>L BYTE</i>	<i>000E</i>	<i>DATA</i>
<i>@CPU</i>	<i>TEXT</i>	<i>0101h</i>	
<i>@FILENAME</i>	<i>TEXT</i>	<i>lr2_comp</i>	
<i>@VERSION</i>	<i>TEXT</i>	<i>510</i>	

94 Source Lines

94 Total Lines

19 Symbols

47800 + 459460 Bytes symbol space free

2 Warning Errors

0 Severe Errors