

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 режима адресации и формирования исполнительного
адреса.

Студент гр. 00383

Самара Р. Д.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2021

Цель работы.

Изучить режимы адресации в языке Ассемблера с помощью готовой программы. Выполнить программу в отладчике, найти ошибки и составить протокол работы в отладчике.

Задание.

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.

Порядок выполнения работы:

1. Получить у преподавателя вариант набора значений исходных данных (массивов) `vec1`, `vec2` и `matr` из файла `lr2.dat`, приведенного в каталоге Задания и занести свои данные вместо значений, указанных в приведенной ниже программе.
2. Протранслировать программу с созданием файла диагностических сообщений; объяснить обнаруженные ошибки и закомментировать соответствующие операторы в тексте программы.
3. Снова протранслировать программу и скомпоновать загрузочный модуль.

4. Выполнить программу в пошаговом режиме под управлением отладчика с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды.
5. Результаты прогона программы под управлением отладчика должны быть подписаны

Ход работы. Вариант 3.

Создан файл lab2.asm. При попытке трансляции MASM выдаёт ошибки, представленные ниже.

- 1) `mov mem3,[bx]` - lab2.asm(48): error A2052: Improper operand type
Попытка положить ячейку памяти в ячейку памяти, что не допустимо.
- 2) `mov cx,vec2[di]` - lab2.asm(55): warning A4031: Operand types must match
Несовпадение размеров операндов: нельзя положить элементы `vec2` размером 1 байт в регистр `CX` размером в 2 байта.
- 3) `mov cx,matr[bx][di]` lab2.asm(59): warning A4031: Operand types must match
Несовпадение размеров операндов: нельзя положить в регистр `CX` размером 2 байта элементы матрицы `matr` длиной 1 байт.
- 4) `mov ax,matr[bx*4][di]`

lab2.asm(60): error A2055: Illegal register value

Недопустимое значение регистра
- 5) `mov ax,matr[bp+bx]`

lab2.asm(80): error A2046: Multiple base registers

Для адресации нельзя использовать несколько базовых регистров одновременно.
- 6) `mov ax,matr[bp+di+si]`

lab2.asm(81): error A2047: Multiple index registers

Нельзя использовать несколько индексных регистров (`DI`, `SI`) одновременно.

Строки с ошибками были закомментированы, новый файл lab2_fix.asm протранслирован без ошибок. Был слинкован файл lab2_fix.obj, получен исполняемый файл lab2_fix.exe, который был запущен в отладчике. Результат в табл 1.

(AX) = 0000, (BX) = 0000

(CX) = 00B0, (DX) = 0000

(SI) = 0000, (DI) = 0000

(BP) = 0000, (SP) = 0018

(CS) = 1A0A, (DS) = 19F5

(ES) = 19F5 (SS) = 1A05

Табл. 1

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	push ds	1E	(DS) = 19F5 (SP) = 0018 (IP) = 0000 Stack+0 0000	(DS) = 19F5 (SP) = 0016 (IP) = 0001 Stack+0 19F5
0001	sub ax, ax	2BC0	(AX) = 0000 (IP) = 0001	(AX) = 0000 (IP) = 0003
0003	push ax	50	(AX) = 0000 (IP) = 0003 (SP) = 0016 Stack+0 19F5	(AX) = 0000 (IP) = 0004 (SP) = 0014 Stack+0 0000

			Stack+2 0000	Stack+2 19F5
0004	mov ax, 1A07	B8071A	(AX) = 0000 (IP) = 0004	(AX) = 1A07 (IP) = 0007
0007	mov ds, ax	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 0007	(AX) = 1A07 (DS) = 1A07 (IP) = 0009
0009	mov ax, 01F4	B8F401	(AX) = 1A07 (IP) = 0009	(AX) = 01F4 (IP) = 000C
000C	mov cx, ax	8BC8	(AX) = 01F4 (CX) = 00B0 (IP) = 000C	(AX) = 01F4 (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	C7060200CEFF	(IP) = 0012	(IP) = 0018
0018	mov bx, 0006	BB0600	(BX) = CE24 (IP) = 0018	(BX) = 0006 (IP) = 001B
001B	mov [0000], ax	A30000	(AX) = 01F4 (IP) = 001B	(AX) = 01F4 (IP) = 001E
001E	mov al, [bx]	8A07	(AX) = 01F4 (BX) = 0006 (IP) = 0001E	(AX) = 0101 (BX) = 0006 (IP) = 0020
0020	mov al, [bx+03]	8A4703	(AX) = 0101 (BX) = 0006 (IP) = 0020	(AX) = 0104 (BX) = 0006 (IP) = 0023
0023	mov cx, [bx+03]	8B4F03	(BX) = 0006 (CX) = 01F4	(BX) = 0006 (CX) = 0804

			(IP) = 0023	(IP) = 0026
0026	mov di, 0002	BF0200	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	mov al, [000E+di]	8A850E00	(AX) = 0104 (DI) = 0002 (IP) = 0029	(AX) = 010A (DI) = 0002 (IP) = 002D
002D	mov bx, 0003	BB0300	(BX) = 0006 (IP) = 002D	(BX) = 0003 (IP) = 0030
0030	mov al, [0016+bx+di]	8A811600	(AX) = 010A (BX) = 0003 (DI) = 0002 (IP) = 0030	(AX) = 01FD (BX) = 0003 (DI) = 0002 (IP) = 0034
0034	mov ax, 1A07	B8071A	(AX) = 01FD (IP) = 0034	(AX) = 1A07 (IP) = 0037
0037	move es, ax	8EC0	(AX) = 1A07 (ES) = 19F5 (IP) = 0037	(AX) = 1A07 (ES) = 1A07 (IP) = 0039
0039	mov ax: es:[bx]	268B07	(AX) = 1A07 (BX) = 0003 (ES) = 1A07 (IP) = 0039	(AX) = 00FF (BX) = 0003 (ES) = 1A07 (IP) = 003C
003C	mov ax, 0000	B80000	(AX) = 1A07 (IP) = 003C	(AX) = 0000 (IP) = 003F
003F	8EC0	move es, ax	(AX) = 0000 (ES) = 1A07 (IP) = 003F	(AX) = 0000 (ES) = 0000 (IP) = 0041
0041	push ds	1E	(DS) = 1A07 (SP) = 0014 (IP) = 0041	(DS) = 1A07 (SP) = 0012 (IP) = 0042

			Stack+0 0000 Stack+2 19F5 Stack+4 0000	Stack+0 1A07 Stack+2 0000 Stack+4 19F5
0042	pop es	07	(ES) = 0000 (IP) = 0042 (SP) = 0012 Stack+0 1A07 Stack+2 0000 Stack+4 19F5	(ES) = 1A07 (IP) = 0043 (SP) = 0014 Stack+0 0000 Stack+2 19F5 Stack+4 0000
0043	mov cx, es:[bx-01]	268B4FFF	(CX) = 0004 (ES) = 1A07 (BX) = 0003 (IP) = 0043	(CX) = FFCE (ES) = 1A07 (BX) = 0003 (IP) = 0047
0047	xchg ax, cx	91	(AX) = 0000 (CX) = FFCE (IP) = 0047	(AX) = FFCE (CX) = 0000 (IP) = 0048
0048	mov di, 0002	BF0200	(DI) = 0002 (IP) = 0048	(DI) = 0002 (IP) = 004B
004B	mov es:[bx+di], ax	268901	(AX) = FFCE (BX) = 0003 (ES) = 1A07 (DI) = 0002 (IP) = 004B	(AX) = FFCE (BX) = 0003 (ES) = 1A07 (DI) = 0002 (IP) = 004E
004E	mov bp, sp	8BEC	(BP) = 0000 (SP) = 0014 (IP) = 004E	(BP) = 0014 (SP) = 0014 (IP) = 0050
0050	push [0000]	FF360000	(SP) = 0014 (IP) = 0050 Stack+0 0000	(SP) = 0012 (IP) = 0054 Stack+0 01F4

			Stack+2 19F5 Stack+4 0000	Stack+2 0000 Stack+4 19F5
0054	push [0002]	FF360200	(AX) = FFCE (SP) = 0012 (IP) = 0054 Stack+0 01F4 Stack+2 0000 Stack+4 19F5 Stack+6 0000	(AX) = FFCE (SP) = 0010 (IP) = 0058 Stack+0 FFCE Stack+2 01F4 Stack+4 0000 Stack+6 19F5
0058	mov bp, sp	8BEC	(BP) = 0014 (SP) = 0010 (IP) = 0058	(BP) = 0010 (SP) = 0010 (IP) = 005A
005A	mov dx, [bp+02]	8B5602	(DX) = 0000 (BP) = 0010 (IP) = 005A	(DX) = 01F4 (BP) = 0010 (IP) = 005D
005D	ret far 0002	CA0200	(IP) = 005D Stack+0 FFCE Stack+2 01F4 Stack+4 0000 Stack+6 19F5	(IP) = FFCE Stack+0 19F5 Stack+2 0000 Stack+4 0000 Stack+6 000

Текст исходной программы, исправленной и файл листинга представлены в приложении А.

Выводы.

В результате изучения исходного кода программы были выявлены и исправлены основные ошибки. На примере исправленной программы была рассмотрена работа с режимами адресации на языке Ассемблер.

ПРИЛОЖЕНИЕ А

Код программы lab2.asm, вариант 3

; Программа изучения режимов адресации процессора IntelX86

EOL EQU '\$'

ind EQU 2

n1 EQU 500

n2 EQU -50

; 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 8,7,6,5,1,2,3,4

vec2 DB -30,-40,30,40,-10,-20,10,20

matr DB -1,-2,-3,-4,8,7,6,5,-5,-6,-7,-8,4,3,2,1

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

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

```

Файл листинга программы lab2.asm

Microsoft (R) Macro Assembler Version 5.10

9/28/21 11:44:19

Page 1-1

; Программа изучения режимов адресации
процессо

pa IntelX86

= 0024 EOL EQU '\$'

= 0002 ind EQU 2

= 01F4 n1 EQU 500

=-0032 n2 EQU -50

; Стек программы

0000 AStack SEGMENT STACK

0000 000C[DW 12 DUP(?)

????

]

0018 AStack ENDS

; Данные программы

0000 DATA SEGMENT

; Директивы описания данных

0000 0000 mem1 DW 0

0002 0000 mem2 DW 0

0004 0000 mem3 DW 0

0006 08 07 06 05 01 02 vec1 DB 8,7,6,5,1,2,3,4

03 04

000E E2 D8 1E 28 F6 EC vec2 DB -30,-40,30,40,-10,-20,10,20

0A 14

0016 FF FE FD FC 08 07 matr DB -1,-2,-3,-4,8,7,6,5,-5,-6,-7,-8,4,3,2,1

06 05 FB FA F9 F8

04 03 02 01

```

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 АДРЕСАЦИИ НА
УРОВНЕ СМЕЩЕНИЙ

          ; 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  mov mem2,n2

0018 BB 0006 R    mov bx,OFFSET vec1

001B A3 0000 R    mov mem1,ax

          ; Косвенная адресация

001E 8A 07          mov al,[bx]

          mov mem3,[bx]

lab2.asm(42): error A2052: Improper operand type

          ; Базированная адресация

7

lab2.asm(44): warning A4001: Extra characters on line

0020 8A 47 03          mov al,[bx]+3

```

0023 8B 4F 03 mov cx,3[bx]

Microsoft (R) Macro Assembler Version 5.10

9/28/21 11:44:19

Page 1-2

; 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]

lab2.asm(50): warning A4031: Operand types must match

; Адресация с базированием и индексированием

0031 BB 0003 mov bx,3

0034 8A 81 0016 R mov al,matr[bx][di]

0038 8B 89 0016 R mov cx,matr[bx][di]

lab2.asm(54): warning A4031: Operand types must match

003C 8B 85 0022 R mov ax,matr[bx*4][di]

lab2.asm(55): error A2055: Illegal register value

; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С

УЧЕТОМ СЕГМЕНТОВ

; Переопределение сегмента

; ----- вариант 1

0040 B8 ---- R mov ax, SEG vec2

0043 8E C0 mov es, ax

0045 26: 8B 07 mov ax, es:[bx]

0048 B8 0000 mov ax, 0

; ----- вариант 2

004B 8E C0 mov es, ax

004D 1E push ds

004E 07 pop es

004F 26: 8B 4F FF mov cx, es:[bx-1]

0053 91 xchg cx,ax

; ----- вариант 3

```

0054 BF 0002          mov di,ind
0057 26: 89 01      mov es:[bx+di],ax
                  ; ----- вариант 4
005A 8B EC          mov bp,sp
005C 3E: 8B 86 0016 R      mov ax,matr[bp+bx]
lab2.asm(74): error A2046: Multiple base registers
0061 3E: 8B 83 0016 R      mov ax,matr[bp+di+si]
lab2.asm(75): 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
lab2.asm(82): error A2006: Phase error between passes
0076              CODE ENDS
                  END Main

```

Microsoft (R) Macro Assembler Version 5.10

9/28/21 11:44:19

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	0076	PARA	NONE
DATA	0026	PARA	NONE

Symbols:

N a m e	Type	Value	Attr	
EOL	NUMBER	0024		
IND	NUMBER	0002		
MAIN	F PROC	0000	CODE	Length = 0076
MATR	L BYTE	0016	DATA	
MEM1	L WORD	0000	DATA	
MEM2	L WORD	0002	DATA	
MEM3	L WORD	0004	DATA	
N1	NUMBER	01F4		
N2	NUMBER	-0032		
VEC1	L BYTE	0006	DATA	
VEC2	L BYTE	000E	DATA	
@CPU	TEXT	0101h		
@FILENAME	TEXT	lab2		
@VERSION	TEXT	510		
84 Source Lines				
84 Total Lines				
19 Symbols				
47810 + 459450 Bytes symbol space free				

3 Warning Errors

5 Severe Errors

lab2_fix.asm:

; Программа изучения режимов адресации процессора IntelX86

EOL EQU '\$'

ind EQU 2

n1 EQU 500

n2 EQU -50

; 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 1,2,3,4,8,7,6,5

vec2 DB -10,-20,10,20,-30,-40,30,40

matr DB 1,2,3,4,-4,-3,-2,-1,5,6,7,8,-8,-7,-6,-5

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
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

```

Файл листинга программы lab2_fix.asm

Microsoft (R) Macro Assembler Version 5.10

9/28/21 19:34:46

Page 1-1

процессо ; Программа изучения режимов адресации

pa IntelX86

= 0024 EOL EQU '\$'

= 0002 ind EQU 2

= 01F4 n1 EQU 500

=-0032 n2 EQU -50

; Стек программы

0000 AStack SEGMENT STACK

0000 000C[DW 12 DUP(?)

????

]

0018 AStack ENDS

; Данные программы

0000 DATA SEGMENT

; Директивы описания данных

0000 0000 mem1 DW 0

0002 0000 mem2 DW 0

0004 0000 mem3 DW 0

0006 01 02 03 04 08 07 vec1 DB 1,2,3,4,8,7,6,5

06 05

000E F6 EC 0A 14 E2 D8 vec2 DB -10,-20,10,20,-30,-40,30,40

1E 28

```

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

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 АДРЕСАЦИИ НА
УРОВНЕ СМЕЩЕНИЙ

                                ; 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       mov mem2,n2

0018 BB 0006 R               mov bx,OFFSET vec1

001B A3 0000 R               mov mem1,ax

                                ; Косвенная адресация

001E 8A 07                    mov al,[bx]

                                ; mov mem3,[bx]

                                ; Базированная адресация

```

0020 8A 47 03 mov al,[bx]+3

0023 8B 4F 03 mov cx,3[bx]

Microsoft (R) Macro Assembler Version 5.10

9/28/21 19:34:46

Page 1-2

; Индексная адресация

0026 BF 0002 mov di,ind

0029 8A 85 000E R mov al,vec2[di]

; mov cx,vec2[di]

; Адресация с базированием и индексированием

002D BB 0003 mov bx,3

0030 8A 81 0016 R 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

0034 B8----- R mov ax, SEG vec2

0037 8E C0 mov es, ax

0039 26: 8B 07 mov ax, es:[bx]

003C B8 0000 mov ax, 0

; ----- вариант 2

003F 8E C0 mov es, ax

0041 1E push ds

0042 07 pop es

0043 26: 8B 4F FF mov cx, es:[bx-1]

0047 91 xchg cx,ax

; ----- вариант 3

```

0048 BF 0002          mov di,ind
004B 26: 89 01      mov es:[bx+di],ax
                  ; ----- вариант 4
004E 8B EC          mov bp,sp
                  ; mov ax,matr[bp+bx]
                  ; mov ax,matr[bp+di+si]
                  ; Использование сегмента стека
0050 FF 36 0000 R    push mem1
0054 FF 36 0002 R    push mem2
0058 8B EC          mov bp,sp
005A 8B 56 02      mov dx,[bp]+2
005D CA 0002      ret 2
0060                Main ENDP
0060                CODE ENDS
                END Main

```

Microsoft (R) Macro Assembler Version 5.10

9/28/21 19:34:46

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	0060	PARA	NONE
DATA	0026	PARA	NONE

Symbols:

N a m e	Type	Value	Attr
EOL	NUMBER	0024	

IND	NUMBER	0002	
MAIN	F PROC	0000	CODE Length = 0060
MATR	L BYTE	0016	DATA
MEM1	L WORD	0000	DATA
MEM2	L WORD	0002	DATA
MEM3	L WORD	0004	DATA
N1	NUMBER	01F4	
N2	NUMBER	-0032	
VEC1	L BYTE	0006	DATA
VEC2	L BYTE	000E	DATA
@CPU	TEXT	0101h	
@FILENAME	TEXT	lab2fix	
@VERSION	TEXT	510	
84 Source Lines			
84 Total Lines			
19 Symbols			
47796 + 459464 Bytes symbol space free			
0 Warning Errors			
0 Severe Errors			