

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.

Студентка гр. 0382

Здобнова К.Д.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить работу режимов адресации процессора IntelX86.

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.

Выполнение 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 преподавателем и представлены в отчете.

Пример используемой программы приведен в приложении.

Объяснения обнаруженных ошибок представлены в табл. 1.

Таблица 1 – Объяснения обнаруженных ошибок

Error A2052: Improper operand type <i>mov mem3,[bx]</i>	Нельзя перемещать и записывать данные из памяти в память (источник и приемник не могут быть одновременно памятью).
Warning A4031: Operand types must match <i>mov cx,vec2[di]</i>	Несоответствие типов операндов, размер регистра cx – 2 байта, размер элемента массива vec2 – 1.
Warning A4031: Operand types must match <i>mov cx,matr[bx][di]</i>	Несоответствие типов операндов, размер регистра cx – 2 байта, размер элемента массива matr – 1.
Error A2055: Illegal register value <i>mov ax,matr[bx*4][di]</i>	Нельзя масштабировать по 2-х байтовым регистрам.
Error A2046: Multiple base registers <i>mov ax,matr[bp+bx]</i>	Нельзя использовать более одного регистра общего назначения для адресации.
Error A2047: Multiple index registers <i>mov ax,matr[bp+di+si]</i>	Нельзя использовать более одного индексного регистра для адресации.

Начальные значения сегментных регистров:

(CS) = 1A0A; (DS) = 19F5; (ES) = 19F5; (SS) = 1A05.

Результаты пошагового выполнения представлены в табл. 2.

Таблица 2 – Результаты выполнения

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	(IP) = 0000 (DS) = 19F5 (SP) = 0018 Stack +0 0000	(IP) = 0001 (DS) = 19F5 (SP) = 0016 Stack +0 19F5
0001	SUB AX,AX	2BC0	(IP) = 0001 (AX) = 0000	(IP) = 0003 (AX) = 0000
0003	PUSH AX	50	(IP) = 0003 (AX) = 0000 (SP) = 0016 Stack +0 19F5 +2 0000	(IP) = 0004 (AX) = 0000 (SP) = 0014 Stack +0 0000 +2 19F5
0004	MOV AX,1A07	B8071A	(IP) = 0004 (AX) = 0000	(IP) = 0007 (AX) = 1A07
0007	MOV DS, AX	8ED8	(IP) = 0007 (DS) = 19F5	(IP) = 0009 (DS) = 1A07
0009	MOV AX,01F4	B8F401	(IP) = 0009 (AX) = 1A07	(IP) = 000C (AX) = 01F4
000C	MOV CX,AX	8BC8	(IP) = 000C (CX) = 00B0	(IP) = 000E (CX) = 01F4
000E	MOV BL,24	B324	(IP) = 000E (BX) = 0000	(IP) = 0010 (BX) = 0024
0010	MOV BH,CE	B7CE	(IP) = 0010 (BX) = 0024	(IP) = 0012 (BX) = CE24
0012	MOV [0002],FFCE	C7060200CEFF	(IP) = 0012	(IP) = 0018

			DS:0002 = 00 DS:0003 = 00	DS:0002 = CE DS:0003 = FF
0018	MOV BX,0006	BB0600	(IP) = 0018 (BX) = CE24	(IP) = 001B (BX) = 0006
001B	MOV [0000],AX	A30000	(IP) = 001B DS:0000 = 00 DS:0001 = 00	(IP) = 001E DS:0000 = F4 DS:0001 = 01
001E	MOV AL,[BX]	8A07	(IP) = 001E (AX) = 01F4	(IP) = 0020 (AX) = 010B
0020	MOV AL,[BX+03]	8A4703	(IP) = 0020 (AX) = 010B	(IP) = 0023 (AX) = 010E
0023	MOV CX,[BX+03]	8B4F03	(IP) = 0023 (CX) = 01F4	(IP) = 0026 (CX) = 120E
0026	MOV DI,0002	BF0200	(IP) = 0026 (DI) = 0002	(IP) = 0029 (DI) = 0002
0029	MOV AL,[000E+DI]	8A850E00	(IP) = 0029 (AX) = 010E	(IP) = 002D (AX) = 01F6
002D	MOV BX,0003	BB0300	(IP) = 002D (BX) = 0006	(IP) = 0030 (BX) = 0003
0030	MOV AL,[0016+BX+DI]	8A811600	(IP) = 0030 (AX) = 01F6	(IP) = 0034 (AX) = 0104
0034	MOV AX,1A07	B8071A	(IP) = 0034 (AX) = 0104	(IP) = 0037 (AX) = 1A07
0037	MOV ES,AX	8EC0	(IP) = 0037 (ES) = 19F5	(IP) = 0039 (ES) = 1A07
0039	MOV AX,ES:[BX]	268B07	(IP) = 0039 (AX) = 1A07	(IP) = 003C (AX) = 00FF
003C	MOV AX,0000	B80000	(IP) = 003C (AX) = 00FF	(IP) = 003F (AX) = 0000
003F	MOV ES,AX	8EC0	(IP) = 003F (ES) = 1A07	(IP) = 0041 (ES) = 0000

0041	PUSH DS	1E	(IP) = 0041 (DS) = 1A07 (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000	(IP) = 0042 (DS) = 1A07 (SP) = 0012 Stack +0 1A07 +2 0000 +4 19F5
0042	POP ES	07	(IP) = 0042 (ES) = 0000 (SP) = 0012 Stack +0 1A07 +2 0000 +4 19F5	(IP) = 0043 (ES) = 0000 (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000
0043	MOV CX,ES:[BX-01]	268B4FFF	(IP) = 0043 (CX) = 120E	(IP) = 0047 (CX) = FFCE
0047	XCHG AX,CX	91	(IP) = 0047 (AX) = 0000 (CX) = FFCE	(IP) = 0048 (AX) = FFCE (CX) = 0000
0048	MOV DI,0002	BF0200	(IP) = 0048 (DI) = 0002	(IP) = 004B (DI) = 0002
004B	MOV ES:[BX+ID],AX	268901	(IP) = 004B ES:[BX+ID] = DS:0005 = 0B00	(IP) = 004E ES:[BX+ID] = DS:0005 = FFCE
004E	MOV BP,SP	8BEC	(IP) = 004E (BP) = 0000	(IP) = 0050 (BP) = 0014
0050	PUSH [0000]	FF360000	(IP) = 0050 [0000] = 01F4 (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000	(IP) = 0054 [0000] = 01F4 (SP) = 0012 Stack +0 01F4 +2 0000 +4 19F5
0054	PUSH [0002]	FF360200	(IP) = 0054 [0002] = FFCE	(IP) = 0058 [0002] = FFCE

			(SP) = 0012 Stack +0 01F4 +2 0000 +4 19F5 +6 0000	(SP) = 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5
0058	MOV BP,SP	8BEC	(IP) = 0058 (BP) = 0014	(IP) = 005A (BP) = 0010
005A	MOV DX,[BP+02]	8B5602	(IP) = 005A (DX) = 0000	(IP) = 005D (DX) = 01F4
005D	RET Far 0002	CA0200	(IP) = 005D (SP) = 0010 (CS) = 1A0A Stack +0 FFCE +2 01F4 +4 0000 +6 19F5	(IP) = FFCE (SP) = 0016 (CS) = 01F4 Stack +0 19F5 +2 0000 +4 0000 +6 0000

Выводы.

Была изучена работа режимов адресации процессора IntelX86 с использованием программы на языке Ассемблера.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: *lab2.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 11,12,13,14,18,17,16,15
vec2 DB 10,20,-10,-20,30,40,-30,-40
matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7,8,-6,-5
DATA ENDS

; Код программы
CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack

; Главная 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

```

Название файла: *lab2.LST*

Microsoft (R) Macro Assembler Version 5.10
21:48:30

10/5/21

Page 1-1

```

1          ; Программа изучения э
           ; ежимов адресации про
           ; цессора IntelX86
2 = 0024          EOL EQU '$'
3 = 0002          ind EQU 2
4 = 01F4          n1 EQU 500
5 =-0032          n2 EQU -50
6
7          ; Стек программы
8 0000          AStack SEGMENT STACK
9 0000 000C[      DW 12 DUP(?)
10      ????
11          ]
12
13 0018          AStack ENDS
14
15          ; Данные программы
16 0000          DATA SEGMENT
17
18          ; Директивы описания И
           ; анных
19 0000 0000          mem1 DW 0

```

```

20 0002 0000 mem2 DW 0
21 0004 0000 mem3 DW 0
22 0006 0B 0C 0D 0E 12 11 vec1 DB 11,12,13,14,18,17,16,15
23 10 0F
24 000E 0A 14 F6 EC 1E 28 vec2 DB 10,20,-10,-20,30,40,-30,-40
25 E2 D8
26 0016 01 02 FC FD 03 04 matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7
,8,-6,-5
27 FE FF 05 06 F8 F9
28 07 08 FA FB

```

```

29 0026 DATA ENDS

```

```

30

```

```

31 ; Код программы

```

```

32 0000 CODE SEGMENT

```

```

33 ASSUME CS:CODE, DS:DATA, SS:AStack

```

```

34

```

```

35 ; Головная процедура

```

```

36 0000 Main PROC FAR

```

```

37 0000 1E push DS

```

```

38 0001 2B C0 sub AX,AX

```

```

39 0003 50 push AX

```

```

40 0004 B8 ---- R mov AX,DATA

```

```

41 0007 8E D8 mov DS,AX

```

```

42

```

```

43 ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ СМЕЙ
©ЕНИЙ

```

```

44 ; Регистровая адресация

```

```

45 0009 B8 01F4 mov ax,n1

```

```

46 000C 8B C8 mov cx,ax

```

```

47 000E B3 24 mov bl,EOL

```

```

Microsoft (R) Macro Assembler Version 5.10
21:48:30

```

```

10/5/21

```

```

Page 1-2

```

```

48 0010 B7 CE mov bh,n2

```

```

49

```

```

50 ; Прямая адресация

```

```

51 0012 C7 06 0002 R FFCE mov mem2,n2

```

```

52 0018 BB 0006 R mov bx,OFFSET vec1

```

```

53 001B A3 0000 R mov mem1,ax

```

```

54

```

```

55                                     ; Косвенная адресация
56 001E 8A 07                         mov al,[bx]
57                                     ;mov mem3,[bx]
58
59                                     ; Базированная адреса
        ция
60 0020 8A 47 03                       mov al,[bx]+3
61 0023 8B 4F 03                       mov cx,3[bx]
62
63                                     ; Индексная адресация
64 0026 BF 0002                         mov di,ind
65 0029 8A 85 000E R                   mov al,vec2[di]
66                                     ;mov cx,vec2[di]
67
68                                     ; Адресация с базирован
        ием и индексировани
        ем
69 002D BB 0003                         mov bx,3
70 0030 8A 81 0016 R                   mov al,matr[bx][di]
71                                     ;mov cx,matr[bx][di]
72                                     ;mov ax,matr[bx*4][di]
73
74                                     ; ПРОВЕРКА РЕЖИМОВ АДРЕ
        САЦИИ С УЧЕТОМ СЕГМЕН
        ТОВ
75                                     ; Переопределение сег
        мента
76                                     ; ----- вариант 1
77 0034 B8 ---- R                       mov ax, SEG vec2
78 0037 8E C0                           mov es, ax
79 0039 26: 8B 07                       mov ax, es:[bx]
80 003C B8 0000                         mov ax, 0
81
82                                     ; ----- вариант 2
83 003F 8E C0                           mov es, ax
84 0041 1E                             push ds
85 0042 07                             pop es
86 0043 26: 8B 4F FF                   mov cx, es:[bx-1]
87 0047 91                             xchg cx,ax
88
89                                     ; ----- вариант 3
90 0048 BF 0002                         mov di,ind
91 004B 26: 89 01                       mov es:[bx+di],ax

```

```

92
93                ; ----- вариант 4
94 004E  8B EC                mov bp,sp
95                ;mov ax,matr[bp+bx]
Microsoft (R) Macro Assembler Version 5.10
21:48:30
10/5/21
Page 1-3

```

```

96                ;mov ax,matr[bp+di+si]
97
98                ; Использование сегме
                нта стека
99 0050  FF 36 0000 R        push mem1
100 0054  FF 36 0002 R        push mem2
101 0058  8B EC                mov bp,sp
102 005A  8B 56 02                mov dx,[bp]+2
103 005D  CA 0002                ret 2
104 0060                                Main ENDP
105 0060                                CODE ENDS
106                                END Main
Microsoft (R) Macro Assembler Version 5.10
21:48:30
10/5/21
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	lab2
@VERSION	TEXT	510

99 Source Lines

99 Total Lines

19 Symbols

47296 + 459964 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Название файла: *lab2wr.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 11,12,13,14,18,17,16,15

```
vec2 DB 10,20,-10,-20,30,40,-30,-40
matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7,8,-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

```

Название файла: *LAB2WR.LST*

Microsoft (R) Macro Assembler Version 5.10
00:28:46

10/6/21

Page 1-1

```

1          ; Программа изучения э
           ; ежимов адресации про
           ; цессора IntelX86

2  = 0024          EOL EQU '$'
3  = 0002          ind EQU 2
4  = 01F4          n1 EQU 500
5  =-0032          n2 EQU -50
6
7          ; Стек программы
8 0000          AStack SEGMENT STACK
9 0000 000C[      DW 12 DUP(?)
10      ???
11      ]
12
13 0018          AStack ENDS
14
15          ; Данные программы
16 0000          DATA SEGMENT
17
18          ; Директивы описания И
           ; анных
19 0000 0000          mem1 DW 0
20 0002 0000          mem2 DW 0
21 0004 0000          mem3 DW 0
22 0006 0B 0C 0D 0E 12 11 vec1 DB 11,12,13,14,18,17,16,15
23      10 0F
24 000E 0A 14 F6 EC 1E 28 vec2 DB 10,20,-10,-20,30,40,-30,-40
25      E2 D8
26 0016 01 02 FC FD 03 04 matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7
           ,8,-6,-5
27      FE FF 05 06 F8 F9
28      07 08 FA FB
29 0026          DATA ENDS
30
31          ; Код программы
32 0000          CODE SEGMENT
33          ASSUME CS:CODE, DS:DATA, SS:AStack
34
35          ; Головная процедура
36 0000          Main PROC FAR
37 0000 1E          push DS
38 0001 2B C0          sub AX,AX
39 0003 50          push AX

```

```

40 0004 B8 ---- R      mov AX,DATA
41 0007 8E D8          mov DS,AX
42
43                    ; ПРОВЕРКА РЕЖИМОВ АДІ
                     ЕСАЦІИ НА УРОВНЕ СМЕЙ
                     ©ЕНИЙ
44                    ; Регистровая адресац
                     ия
45 0009 B8 01F4          mov ax,n1
46 000C 8B C8          mov cx,ax
47 000E B3 24          mov bl,EOL
Microsoft (R) Macro Assembler Version 5.10      10/6/21
00:28:46

```

Page 1-2

```

48 0010 B7 CE          mov bh,n2
49
50                    ; Прямая адресация
51 0012 C7 06 0002 R FFCE mov mem2,n2
52 0018 BB 0006 R      mov bx,OFFSET vec1
53 001B A3 0000 R      mov mem1,ax
54
55                    ; Косвенная адресация
56 001E 8A 07          mov al,[bx]
57                    mov mem3,[bx]
lab2wr.asm(50): error A2052: Improper operand type
58
59                    ; Базированная адреса
                     ция
60 0020 8A 47 03          mov al,[bx]+3
61 0023 8B 4F 03          mov cx,3[bx]
62
63                    ; Индексная адресация
64 0026 BF 0002          mov di,ind
65 0029 8A 85 000E R      mov al,vec2[di]
66 002D 8B 8D 000E R      mov cx,vec2[di]
lab2wr.asm(59): warning A4031: Operand types must match
67
68                    ; Адресация с базировІ
                     ънием и индексировани
                     ем
69 0031 BB 0003          mov bx,3
70 0034 8A 81 0016 R      mov al,matr[bx][di]

```

```

71 0038 8B 89 0016 R      mov cx,matr[bx][di]
lab2wr.asm(64): warning A4031: Operand types must match
72 003C 8B 85 0022 R      mov ax,matr[bx*4][di]
lab2wr.asm(65): error A2055: Illegal register value
73
74                        ; ПРОВЕРКА РЕЖИМОВ АДЙ
                        ЕСАЦИИ С УЧЕТОМ СЕГМЙ
                        □НТОВ
75                        ; Переопределение сег
                        мента
76                        ; ----- вариант 1
77 0040 B8 ---- R      mov ax, SEG vec2
78 0043 8E C0          mov es, ax
79 0045 26: 8B 07      mov ax, es:[bx]
80 0048 B8 0000          mov ax, 0
81
82                        ; ----- вариант 2
83 004B 8E C0          mov es, ax
84 004D 1E            push ds
85 004E 07            pop es
86 004F 26: 8B 4F FF    mov cx, es:[bx-1]
87 0053 91            xchg cx,ax
88
89                        ; ----- вариант 3
90 0054 BF 0002          mov di,ind
91 0057 26: 89 01      mov es:[bx+di],ax
92
93                        ; ----- вариант 4
94 005A 8B EC          mov bp,sp
95 005C 3E: 8B 86 0016 R    mov ax,matr[bp+bx]
lab2wr.asm(88): error A2046: Multiple base registers
Microsoft (R) Macro Assembler Version 5.10          10/6/21
00:28:46

```

Page 1-3

```

96 0061 3E: 8B 83 0016 R    mov ax,matr[bp+di+si]
lab2wr.asm(89): error A2047: Multiple index registers
97
98                        ; Использование сегме
                        нта стека
99 0066 FF 36 0000 R    push mem1
100 006A FF 36 0002 R    push mem2
101 006E 8B EC          mov bp,sp

```

```

102 0070 8B 56 02          mov dx,[bp]+2
103 0073 CA 0002          ret 2
104 0076                  Main ENDP
lab2wr.asm(97): error A2006: Phase error between passes
105 0076                  CODE ENDS
106                      END Main

```

Microsoft (R) Macro Assembler Version 5.10
00:28:46

10/6/21

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	lab2wr	
@VERSION	TEXT	510	

99 Source Lines

99 Total Lines

19 Symbols

47278 + 457935 Bytes symbol space free

2 Warning Errors

5 Severe Errors