

**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. 0382

Самулевич В.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Изучить различные режимы адресации и способы формирования исполнительного адреса в языке программирования ассемблер.

### **Задание.**

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.

### **Выполнение работы.**

Вариант 6

Набор значений исходных данных представлен на рисунке 1.

<code>vec1</code>	<code>18, 17, 16, 15, 11, 12, 13, 14</code>
<code>vec2</code>	<code>30, 40, -30, -40, 10, 20, -10, -20</code>
<code>matr</code>	<code>-4, -3, 1, 2, -2, -1, 3, 4, 5, 6, 7, 8, -8, -7, -6, -5</code>

Рисунок 1-Данные для варианта 6

Объяснение полученных ошибок и предупреждений:

1) Строка 46: `mov mem3, [bx]`

Ошибка: errorA2052: Improper operand type

Причина: команда `mov` не может переслать содержимое одной ячейки памяти в другую ячейку памяти.

2) Строка 53: `mov cx,vec2[di]`

Предупреждение: warning A4301: Operand types must match

Причина: Размеры приемника и источника в команде не совпадают: `cx` имеет размер 2 байта, а `vec2[di]`-один байт

3) Строка 57: `mov cx,matr[bx][di]`

Предупреждение: warning A4301: Operand types must match

Причина: Аналогично предыдущему случаю размеры приемника и источника не совпадают. `cx`- 2 байта, а `matr[bx][di]` – 1 байт.

4) Строка 58: `mov ax,matr[bx*4][di]`

Ошибка: errorA2055: Illegal register value

Причина: Нельзя умножать 16 битовые регистры.

5) Строка 78: `mov ax,matr[bp+bx]`

Ошибка: errorA2046: Multiple base registers

Причина: Множественно использованы базовые регистры, при том что разрешен только один.

6) Строка 79: `mov ax,matr[bp+di+si]`

Ошибка: errorA2047: Multiple index registers

Причина: Множественно использованы индексные регистры, при том что разрешен только 1.

Протокол:

Адрес команд	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	(SP)=0018 (IP)=0000 Stack:+0 0000	(SP)=0016 (IP)=0001 Stack:+0 19F5 +2 0000
0001	SUB AX,AX	2BC0	(AX)=0000 (IP)=0001	(AX)=0000 (IP)=0003
0003	PUSH AX	50	(SP)=0016 (IP)=0003 Stack: :+0 19F5 +2 0000	(SP)=0014 (IP)=0004 Stack: :+0 0000 +2 19F5 +4 0000
0004	MOV AX, 1A07	B8071A	(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	B8F401	(AX)=1A07 (IP)=0009	(AX)=01F4 (IP)=000C
000C	MOV CX,AX	8BC8	(CX)=00B0 (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	C7060200CEF F	DS:0002=00 DS:0003=00 (IP)=0012	DS:0002=CE DS:0003=FF (IP)=0018
0018	MOV BX,0006	BB0600	(BX)=CE24 (IP)=0018	(BX)=0006 (IP)=001B
001B	MOV [0000],AX	A30000	DS:0000=00 DS:0001=00 (IP)=001B	DS:0000=F4 DS:0001=01 (IP)=001E
001E	MOV AL,[BX]	8A07	(AX)=01F4 (IP)=001E	(AX)=0112 (IP)=0020
0020	MOV AL,[BX+03]	8A4703	(AX)=0112 (IP)=0020	(AX)=010F (IP)=0023
0023	MOV CX, [BX+03]	8B4F03	(CX)=01F4 (IP)=0023	(CX)=0B0F (IP)=0026
0026	MOV DI,0002	BF0200	(DI)=0000 (IP)=0026	(DI)=0002 (IP)=0029
0029	MOV AL,[000E+DI]	8A850E00	(AX)=010F (IP)=0029	(AX)=01E2 (IP)=002D
002D	MOV BX,0003	BB0300	(BX)=0006 (IP)=002D	(BX)=0003 (IP)=0030
0030	MOV AL, [0016+BX+DI]	8A811600	(AX)=01E2 (IP)=0030	(AX)=01FF (IP)=0034
0034	MOV AX,1A07	B8071A	(AX)=01FF (IP)=0034	(AX)=1A07 (IP)=0037

0037	MOV ES,AX	8EC0	(ES)=19F5 (IP)=0037	(ES)=1A07 (IP)=0039
0039	MOV AX,ES:[BX]	268B07	(AX)=1A07 (IP)=0039	(AX)=00FF (IP)=003C
003C	MOV AX,0000	B80000	(AX)=00FF (IP)=003C	(AX)=0000 (IP)=003F
003F	MOV ES,AX	8EC0	(ES)=1A07 (IP)=003F	(ES)=0000 (IP)=0041
0041	PUSH DS	1E	(SP)=0014 (IP)=0041 Stack: :+0 0000 +2 19F5 +4 0000	(SP)=0012 (IP)=0042 Stack: +0 1A07 +2 0000 +4 19F5 +6 0000
0042	POP ES	07	SP)=0012 (IP)=0042 (ES)=0000 Stack: +0 1A07 +2 0000 +4 19F5 +6 0000	(SP)=0014 (IP)=0043 (ES)=1A07 Stack: +0 0000 +2 19F5 +4 0000
0043	MOV CX, ES:[BX-01]	268B4FFF	(CX)=0B0F (IP)=0043	(CX)=FFCE (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	DS:0005=00	DS:0005=CE

			DS:0006=12 (IP)=004B	DS:0006=FF (IP)=004E
004E	MOV BP,SP	8BEC	(BP)=0000 (IP)=004E	(BP)=0014 (IP)=0050
0050	PUSH [0000]	FF360000	(SP)=0014 (IP)=0050 Stack: +0 0000 +2 19F5 +4 0000	(SP)=0012 (IP)=0054 Stack: +0 01F4 +2 0000 +4 19F5 +6 0000
0054	PUSH [0002]	FF360200	(SP)=0012 (IP)=0054 Stack: +0 01F4 +2 0000 +4 19F5 +6 0000	(SP)=0010 (IP)=0058 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5
0058	MOV BP,SP	8BEC	(BP)=0014 (IP)=0058	(BP)=0010 (IP)=005A
005A	MOV DX,[BP+02]	8B5602	(DX)=0000 (IP)=005A	(DX)=01F4 (IP)=005D
005D	RET	CA0200	(SP)=0010 (IP)=005D Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5	(SP)=0016 (IP)=FFCE Stack: +0 19F5 +2 0000

Исходный код программы представлен в приложении А, содержимое файла диагностических сообщений- в приложении Б.

### **Выводы.**

Были изучены режимы адресации в ассемблере, а также исправлены ошибки в предоставленной программе



# ПРИЛОЖЕНИЕ А

## ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr2\_com.asm

```
EOL EQU '$'
ind EQU 2
n1 EQU 500
n2 EQU -50

AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS

DATA SEGMENT

mem1 DW 0
mem2 DW 0
mem3 DW 0
vec1 DB 18,17,16,15,11,12,13,14
vec2 DB 30,40,-30,-40,10,20,-10,-20
matr DB -4,-3,1,2,-2,-1,3,4,5,6,7,8,-8,-7,-6,-5
DATA ENDS

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

Main PROC FAR
    push DS
    sub AX,AX
    push AX
    mov AX,DATA
    mov DS,AX
    mov ax,n1
    mov cx,ax
    mov bl,EOL
    mov bh,n2
    mov mem2,n2
    mov bx,OFFSET vec1
    mov mem1,ax
    mov al,[bx]
    ;mov mem3,[bx]
    mov al,[bx]+3
    mov cx,3[bx]
    mov di,ind
    mov al,vec2[di]
    ;mov cx,vec2[di]
    mov bx,3
    mov al,matr[bx][di]
    ;mov cx,matr[bx][di]
    ;mov ax,matr[bx*4][di]

    mov ax, SEG vec2
    mov es, ax
```

```

    mov ax, es:[bx]
    mov ax, 0
    mov es, ax
    push ds
    pop es
    mov cx, es:[bx-1]
    xchg cx,ax
    mov di,ind
    mov es:[bx+di],ax
    mov bp,sp
    ;mov ax,matr[bp+bx]
    ;mov ax,matr[bp+di+si]
    push mem1
    push mem2
    mov bp,sp
    mov dx,[bp]+2
    ret 2
Main ENDP
CODE ENDS
    END Main

```

## ПРИЛОЖЕНИЕ Б

### ФАЙЛ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lr2\_com.lst

Microsoft (R) Macro Assembler Version 5.10  
20:02:53

10/6/21

Page 1-1

```
= 0024          EOL EQU '$'
= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50

0000          AStack SEGMENT STACK
0000 000C[      DW 12 DUP(?)
      ????
      ]

0018          AStack ENDS

0000          DATA SEGMENT

0000 0000      mem1 DW 0
0002 0000      mem2 DW 0
0004 0000      mem3 DW 0
0006 12 11 10 0F 0B 0C      vec1 DB 18,17,16,15,11,12,13,14

      0D 0E
000E 1E 28 E2 D8 0A 14      vec2 DB 30,40,-30,-40,10,20,-10,-20
      F6 EC
0016 FC FD 01 02 FE FF      matr DB -4,-3,1,2,-2,-1,3,4,5,6,7,8,-8,-7,-6,-5

      03 04 05 06 07 08
      F8 F9 FA FB
0026          DATA ENDS

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

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
0009 B8 01F4          mov ax,n1
000C 8B C8          mov cx,ax
000E B3 24          mov bl,EOL
0010 B7 CE          mov bh,n2
0012 C7 06 0002 R FFCE      mov mem2,n2
0018 BB 0006 R      mov bx,OFFSET vec1
001B A3 0000 R      mov mem1,ax
```



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	lr2_com	
@VERSION . . . . .	TEXT	510	

70 Source Lines  
70 Total Lines  
19 Symbols

47814 + 461493 Bytes symbol space free

0 Warning Errors  
0 Severe Errors