

**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. 1383

Сапожников А.Э.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Получение знаний о режимах адресации на языке Ассемблере.

### **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. Изменен файл `lr2_comp.asm`.

В файле `lr2_comp.asm` изменены `vec1`, `vec2`, `lr2` в соответствии с условием задания

2. Протранслирована программа `lr2_comp.asm`, с помощью программы:

>MASM.EXE

В результате получены ошибки:

`mov mem3,[bx]` (пересылка m,m)

`mov cx,vec2[di]` (разная длина операндов)

`mov cx,matr[bx][di]` (разная длина операндов)

mov ax,matr[bx\*4][di] (нельзя умножать базовый регистр)

mov ax,matr[bp+bx] (>1 базовый регистр)

mov ax,matr[bp+di+si] (>1 индексный регистр)

Данные строчки были закомментированны и программа была заново протранслирована

3. Скомпонован объектный файл LR2\_COMP.OBJ, с помощью программы:  
>LINK.EXE

В результате создан исполняемый файл LR2\_COMP.EXE

4. Запущена программа отладки:

>AFDPRO.EXE HELLO1.EXE

5. Начальное содержимое сегментных регистров

CS = 1A0A, DS = 19F5, SS = 1A0A, ES = 19F5

Таблица 1- исполнение файла LR2\_COMP.EXE

Адрес команды	Символический код команды	16-ричный код команды	Содержимое ячеек до выполнения	Содержимое ячеек после выполнения
0000	PUSH DS	1E	SP = 18 IP = 0000 Stack: +0 0000 +2 0000 +4 0000 +6 0000	SP = 16 IP = 0001 Stack: +0 19F5 +2 0000 +4 0000 +6 0000
0001	SUB AX, AX	2BC0	IP = 0001 AX = 0000	IP = 0003 AX = 0000
0003	PUSH AX	50	SP = 16 IP = 0003 Stack: +0 19F5	SP = 14 IP = 0004 Stack: +0 0000

			+2 0000 +4 0000 +6 0000	+2 19F5 +4 0000 +6 0000
0004	MOV AX, 1A07	B8071A	AX = 0000 IP = 0004	AX = 1A07 IP = 0007
0007	MOV DS, AX	8ED8	DS = 19F5 IP = 0007	DS = 0A07 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	C7060200CE FF	IP = 0012 DS:0000 00 00 00 00 00 00 15 16	IP = 0018 DS:0000 00 00 CE FF 00 00 15 16
0018	MOV BX, 0006	BB0600	BX = CE24 IP = 0018	BX = 0006 IP = 001B
001B	MOV [0000], AX	A30000	IP = 001B DS:0000 00 00 CE FF 00 00 15 16	IP = 001E DS:0000 F4 01 CE FF 00 00 15 16
001E	MOV AL, [BX]	8F07	AX = 01F4 IP = 001E	AX = 0115 IP = 0020
0020	MOV AL, [BX+03]	8A4703	AX = 0115 IP = 0020	AX = 0018 IP = 0023
0023	MOV CX, [BX+03]	8B4F03	CX = 01F4 IP = 0023	CX = 1C18 IP = 0026

0026	MOV DI, 0002	BF0200	DI = 0000 IP = 0026	DI = 0002 IP = 0029
0029	MOV AL, [000E+DI]	8A850E00	AX = 0118 IP = 0029	AX = 01D8 IP = 002D
002D	MOV BX, 0003	BB0300	BX = 0006 IP = 002D	BX = 0003 IP = 0030
0030	MOV AL, [0016+BX+DI]	8A811600	AX = 01D8 IP = 0030	AX = 0108 IP = 0034
0034	MOV AX, 1A07	B8071A	AX = 0108 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 +6 0000	SP = 0012 IP = 0042 Stack: +0 1A07 +2 0000 +4 19F5 +6 0000
0042	POP ES	07	ES = 0000 SP = 0012 IP = 0042  Stack: +0 1A07 +2 0000	ES = 1A07 SP = 0014 IP = 0043  Stack: +0 0000 +2 19F5

			+4 19F5 +6 0000	+4 0000 +6 0000
0043	MOV CX, ES:[BX-01]	268B4FFF	CX = 1C18 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 DI = 004B
004B	MOV ES:[BX+DI], AX	268901	IP = 004B DS:0000 F4 01 CE FF 00 00 15 16	IP = 004E DS:0000 F4 01 CE FF 00 CE FF 16
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 +6 0000	SP = 0012 IP = 0054 Stack: +0 01F4 +2 0000 +4 19F5 +6 0000
0054	PUSH [0002]	FF360200	SP = 0012 IP = 0050 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 FAR 0002	CA0200	SP = 0010 CS = 1A0A	SP = 0016 CS = 01F4

			IP = 005D Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5	IP = FFCE Stack: +0 19F5 +2 0000 +4 0000 +6 0000
--	--	--	---	---

Программный код см. в приложении А.

Файлы диагностических сообщений см. в приложении Б.

### **Выводы.**

Получены знания о режимах адресации на языке Ассемблере.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr2\_comp.asm

```
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 21,22,23,24,28,27,26,25
vec2 DB 40,50,-40,-50,20,30,-20,-30
matr DB 5,6,-8,-7,7,8,-6,-5,1,2,-4,-3,3,4,-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
```

## ПРИЛОЖЕНИЕ В ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: LR2.lst

```
Microsoft (R) Macro Assembler Version 5.10
10/22/22 19:19:1
```

Page 1-1

```
1 = 0024          EOL EQU '$'
2 = 0002          ind EQU 2
3 = 01F4          n1 EQU 500
4 =-0032          n2 EQU -50
5                ;  Ð¡Ñ, ÐµÐ° Ð¡Ñ ÐŸ³Ñ Ð°ÐœÐœÑ<
6 0000           AStack SEGMENT STACK
7 0000 000C[      DW 12 DUP(?)
8  ????
```

9 ]

```
10
11 0018          AStack ENDS
12              ;          Ð”Ð°ÐœÐœÑ< Ðµ
Ð¡Ñ ÐŸ³Ñ Ð°ÐœÐœÑ<
13 0000          DATA SEGMENT
14              ;          Ð”ÐžÑ ÐµÐ°Ñ, ÐžÐ²Ñ<
ÐŸÐ¡ÐžÑ Ð°ÐœÐžÑ Ð
          ŽÐ°ÐœÐœÑ<Ñ...
15 0000 0000      mem1 DW 0
16 0002 0000      mem2 DW 0
17 0004 0000      mem3 DW 0
18 0006 15 16 17 18 1C 1B      vec1 DB
21,22,23,24,28,27,26,25
19 1A 19
20 000E 28 32 D8 CE 14 1E      vec2 DB
40,50,-40,-50,20,30,-20,-30
21 EC E2
22 0016 05 06 F8 F9 07 08      matr DB
5,6,-8,-7,7,8,-6,-5,1,2,-4,-3,3
          ,4,-2,-1
```

```

23          FA FB 01 02 FC FD
24          03 04 FE FF
25 0026                                DATA ENDS
26                                ;  ĐŠĐŸĐŽ ĐĴŃ ĐŸĐ³Ń Đ°ĐĖĐĚŃ<
27 0000                                CODE SEGMENT
28                                ASSUME  CS:CODE,  DS:DATA,
SS:ASt
                                ack
29                                ;          Đ“ĐŸĐ»ĐŸĐ²ĐœĐ°Ń
ĐĴŃ ĐŸŃ†ĐμĐŽŃfŃ Đ°
30 0000                                Main PROC FAR
31 0000 1E                                push DS
32 0001 2B C0                                sub AX,AX
33 0003 50                                push AX
34 0004 B8 ---- R                                mov AX,DATA
35 0007 8E D8                                mov DS,AX
36                                ;  ĐŸĐ  Đ Đ'Đ Đ  ĐŠĐ  Đ
Đ Đ-Đ~ĐœĐ Đ'  Đ Đ”Đ
                                Đ ĐĴĐ ĐŠĐ~Đ~  Đ Đ  ĐŁĐ  Đ Đ'Đ Đ
ĐĴĐœĐ Đ
                                ©Đ Đ Đ~Đ™
37                                ;  Đ  ĐμĐ³ĐŽŃ Ń,Ń ĐŸĐ²Đ°Ń
Đ°ĐŽŃ ĐμŃ Đ°Ń†
                                ĐŽŃ
38 0009 B8 01F4                                mov ax,n1
39 000C 8B C8                                mov cx,ax
40 000E B3 24                                mov bl,EOL
41 0010 B7 CE                                mov bh,n2
42                                ;          ĐŸŃ Ń ĐĖĐ°Ń
Đ°ĐŽŃ ĐμŃ Đ°Ń†ĐŽŃ
43 0012 C7 06 0002 R FFCE                                mov mem2,n2
44 0018 BB 0006 R                                mov bx,OFFSET vec1
45 001B A3 0000 R                                mov mem1,ax
46                                ;          ĐŠĐŸŃ Đ²ĐμĐœĐœĐ°Ń
Đ°ĐŽŃ ĐμŃ Đ°Ń†ĐŽŃ
47 001E 8A 07                                mov al,[bx]
48                                ;          mov mem3,[bx]
Microsoft (R) Macro Assembler Version 5.10
10/22/22 19:19:1

```

```

49                                ;      Ð‘Ð°Ð· ÐŸÐ²Ð°ÐœÐ°Ñ
Ð°ÐŸÑ ÐµÑ Ð°
                                Ñ†ÐŸÑ
50 0020 8A 47 03                mov al,[bx]+3
51 0023 8B 4F 03                mov cx,3[bx]
52                                ;      ÐœÐŸÐ°Ñ ÐœÐ°Ñ
Ð°ÐŸÑ ÐµÑ Ð°Ñ†ÐŸÑ
53 0026 BF 0002                mov di,ind
54 0029 8A 85 000E R            mov al,vec2[di]
55                                ;      mov cx,vec2[di]
56                                ;      ÐŸÑ ÐµÑ Ð°Ñ†ÐŸÑ      Ñ
Ð±Ð°Ð· ÐŸÐ²Ð
                                °ÐœÐŸÐœ
ÐœÐœÐŸÐ°Ñ ÐŸÑ ÐŸÑ ÐŸÐ²Ð°ÐœÐŸ
                                ÐµÐœ
57 002D BB 0003                mov bx,3
58 0030 8A 81 0016 R            mov
al,matr[bx][di]
59                                ;      mov cx,matr[bx][di]
60                                ;      mov ax,matr[bx*4][di]
61                                ;      ÐŸ Ð’ Ð Ð Š Ð
Ð Ð-Ðœ Ð’ Ð Ð”Ð
                                Ð Ð; Ð ŠÐ~ Ð Ð; ÐŁÐŚ ÐŸ Ðœ
Ð; Ð Ðœ
                                Ð ÐŸ Ð’
62                                ;
ÐŸÐµÑ ÐµÐŸ;Ñ ÐµÐŸÐœ»ÐµÐœÐŸ Ñ ÐµÐ³
                                ÐœÐœÑ,Ð°
63                                ; ----- Ð²Ð°Ñ ÐŸÐ°Ñ, 1
64 0034 B8 ---- R              mov ax, SEG vec2
65 0037 8E C0                  mov es, ax
66 0039 26: 8B 07              mov ax, es:[bx]
67 003C B8 0000                mov ax, 0
68                                ; ----- Ð²Ð°Ñ ÐŸÐ°Ñ, 2
69 003F 8E C0                  mov es, ax
70 0041 1E                      push ds
71 0042 07                      pop es
72 0043 26: 8B 4F FF            mov cx, es:[bx-1]
73 0047 91                      xchg cx,ax
74                                ; ----- Ð²Ð°Ñ ÐŸÐ°Ñ, 3
75 0048 BF 0002                mov di,ind
76 004B 26: 89 01              mov es:[bx+di],ax
77                                ; ----- Ð²Ð°Ñ ÐŸÐ°Ñ, 4
78 004E 8B EC                  mov bp,sp

```

```

79          ;      mov ax,matr[bp+bx]
80          ;      mov ax,matr[bp+di+si]
81          ;      Ð Ñ Ð;ÐŸÐ»Ñ(Ð ·ÐŸÐ²Ð°ÐœÐžÐµ
Ñ ÐµÐ³ÐœÐµ
          ÐœÑ,Ð° Ñ Ñ,ÐµÐ°Ð°
82 0050 FF 36 0000 R      push mem1
83 0054 FF 36 0002 R      push mem2
84 0058 8B EC            mov bp,sp
85 005A 8B 56 02         mov dx,[bp]+2
86 005D CA 0002          ret 2
87 0060                  Main ENDP
88 0060                  CODE ENDS
89                      END Main
Microsoft (R) Macro Assembler Version 5.10
10/22/22 19:19:1

```

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	LR2_COMP
@VERSION . . . . .	TEXT	510

82 Source Lines  
 82 Total Lines  
 19 Symbols

47286 + 457927 Bytes symbol space free

0 Warning Errors  
 0 Severe Errors