

**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**

Студент гр. 9382

\_\_\_\_\_

Докукин В.М.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Изучить режимы адресации, указать на ошибки в программе и объяснить их.

### **Основные теоретические положения.**

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.

Исходный код программы:

```
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 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
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
; 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
        bmov 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

### Ход работы.

Листинг успешной трансляции программы:

```
= 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  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
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
                                ; 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]
lr2_comp.asm(35): error A2052: Improper operand type
                                ; Базированная адресация
                                6mov al,[bx]+3
lr2_comp.asm(37): warning A4001: Extra characters on line
0020    8B 4F 03                                mov cx,3[bx]
                                ; Индексная адресация
0023    BF 0002                                mov di,ind
0026    8A 85 000E R                                mov al,vec2[di]
002A    8B 8D 000E R                                mov cx,vec2[di]
lr2_comp.asm(42): warning A4031: Operand types must match

```



Page 1-2

```

0039 8B 85 0022 R      mov ax,matr[bx*4][di]
lr2_comp.asm(47): error A2055: Illegal register value
                ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ
СЕКМЕНТОВ

                ; Переопределение сегмента
                ; ----- вариант 1
003D B8 ---- R      mov ax, SEG vec2
0040 8E C0          mov es, ax
0042 26: 8B 07      mov ax, es:[bx]
0045 B8 0000        mov ax, 0
                ; ----- вариант 2
0048 8E C0          mov es, ax
004A 1E            push ds
004B 07            pop es
004C 26: 8B 4F FF   mov cx, es:[bx-1]
0050 91            xchg cx,ax
                ; ----- вариант 3
0051 BF 0002        mov di,ind
0054 26: 89 01      mov es:[bx+di],ax
                ; ----- вариант 4
0057 8B EC          mov bp,sp
0059 3E: 8B 86 0016 R  mov ax,matr[bp+bx]
lr2_comp.asm(66): error A2046: Multiple base registers
005E 3E: 8B 83 0016 R  mov ax,matr[bp+di+si]
lr2_comp.asm(67): error A2047: Multiple index registers
                ; Использование сегмента стека
0063 FF 36 0000 R    push mem1

```



```

0067  FF 36 0002 R          push mem2
006B  8B EC                mov bp,sp
006D  8B 56 02              mov dx,[bp]+2
0070  CA 0002              ret 2
0073                      Main ENDP
lr2_comp.asm(74): error A2006: Phase error between passes
0073                      CODE ENDS
                        END Main

```

Были обнаружены и закомментированы 4 ошибки:

```

mov mem3,[bx]
mov ax,matr[bx*4][di]
mov ax,matr[bp+bx]
mov ax,matr[bp+di+si]

```

Подробное описание ошибок:

1) `mov mem3,[bx]`

Ошибка: “Improper operand type”

Нельзя прямо передавать объекты с памяти в память. Если нужно передать данные из ячейки `[bx]` в ячейку, на которую ссылается переменная `mem3` то это следует делать через регистр `AX`.

2) `mov ax,matr[bx*4][di]`

Ошибка: “Illegal register value”

Операцию умножение на число можно применять только к регистрам с префиксом `E`.

3) `mov ax,matr[bp+bx]`

Ошибка: “Multiple base registers”

Нельзя использовать более одного базового регистра.

4) `mov ax, matr[bp+di+si]`

Ошибка: “Multiple index registers”

Нельзя использовать более одного индексного регистра, а также нельзя использовать более двух регистров.

### **Выводы.**

В ходе выполнения лабораторной работы получены навыки в области отладки программы на языке ассемблера и нахождения ошибок в готовой программе; кроме того, усвоены знания в области регистровой адресации.

## ПРОТОКОЛ

Начальные значения регистров:

CS = 1A0A, DS=19F5, ES=19F5, SS=1A05

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 STACK=+0 0000 IP=0000	DS= 19F5 SP=0016 STACK=+0  19F5 IP=0001
0001	SUB AX,AX	2BC0	AX=0000 IP=0001	AX=0000 IP=0003
0003	PUSH AX	50	AX=0000 SP=0016 STACK=+0 19F5 IP=0003	AX=0000 SP=0014 STACK=+0 0000 +2 19F5 IP=0004
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	C7060200CE FF	IP=0012 DS[0002]=00 DS[0003]=00	IP=0018 DS[0002]=CE DS[0003]=FF
0018	MOV BX, 0006	BB0600	BX=CE24	BX=0006

			IP=0018	IP=001B
001B	MOV [0000],AX	A30000	AX=01F4 IP=001B DS[0000]=00 DS[0001]=00	AX=01F4 IP=001E DS[0000]=F4 DS[0001]=01
001E	MOV AL,[BX]	8A07	AX=01F4 DS[BX]= DS[0006]=15 IP=001E	AX=0115 DS[BX]= DS[0006]=15 IP=0020
0020	MOV CX,[BX+03]	8B4F03	CX=01F4 DS[BX+03]= DS[0009]=18 DS[000A]=1C IP=0020	CX=1C18 DS[BX+03]= DS[0009]=18 DS[000A]=1C IP=0023
0023	MOV DI, 0002	BF0200	DI=0000 IP=0023	DI=0002 IP=0026
0026	MOV AL,[000E+DI]	8A850E00	AX=0115 DS[000E+DI]= DS[0010]=D8 IP=0026	AX=01D8 DS[000E+DI]= DS[0010]=D8 IP=002A
002A	MOV CX,[000E+DI]	8B8D0E00	CX=1C18 DS[000E+DI] = DS[0010] = D8 DS[0011] = CE IP=002A	CX=CED8 IP=002E
002E	MOV BX, 0003	BB0300	BX = 0006 IP = 002E	BX = 0003 IP = 0031
0031	MOV AL,[0016 + BX + DI]	8A811600	AX=010A DS[0016+BX+DI]= DS[001B]=08 IP=0031	AX=0108 DS[0016+BX+DI]= DS[001B]=08 IP=0035
0035	MOV CX, [0016 + BX + DI]	8B891600	CX = CED8 DS[0016 + BX + DI] = DS[001B] = 08 DS[001C] = FA IP = 0035	CX = FA08 IP = 0039
0039	MOV AX,1A07	B8071A	AX=0108	AX=1A07

			IP=0039	IP=003C
003C	MOV ES,AX	8EC0	ES=19F5 AX=1A07 IP=003C	ES=1A07 AX=1A07 IP=003E
003E	MOV AX,ES:[BX]	268B07	AX=1A07 ES=1A07 ES[BX]=ES[0003]=FF ES[0004]=00 IP=003E	AX=00FF ES=1A07 ES[BX]=ES[0003]=FF ES[0004]=00 IP=0041
0041	MOV AX,0000	B80000	AX=00FF IP=0041	AX=0000 IP=0044
0044	MOV ES,AX	8EC0	ES=1A07 AX=0000 IP=0044	ES=0000 AX=0000 IP=0046
0046	PUSH DS	1E	DS=1A07 SP=0014 STACK=+0 0000 +2 19F5 IP=0046	DS=1A07 SP=0012 STACK=+0 1A07 +2 0000 +4 19F5 IP=0047
0047	POP ES	07	SP=0012 ES=0000 STACK=+0 1A07 +2 0000 +4 19F5 IP=0047	SP=0014 ES=1A07 STACK=+0 0000 +2 30B0 IP=0048
0048	MOV CX,ES:[BX-01]	268B4FFF	CX=FA08 ES=1A07 ES[BX-01]= ES[0002]=CE ES[0003]=FF IP=0048	CX=FFCE ES=1A07 ES[BX-01] =ES[0002]=CE ES[0003]=FF IP=004C
004C	XCHG AX,CX	91	AX = 0000 CX = FFCE IP=004C	AX=FFCE CX=0000 IP=004D
004D	MOV DI,0002	BF0200	DI=0002 IP=004D	DI=0002 IP=0050
0050	MOV ES:	268901	ES=1A07	ES=1A07

	[BX+DI],AX		ES[BX+DI] = [0005]= 00 ES[0006] = 15 AX=FFCE IP=0050	ES[0005] = CE ES[0006] = FF IP=0053
0053	MOV BP,SP	8BEC	BP=0000 SP=0014 IP=0053	BP=0014 SP=0014 IP=0055
0055	PUSH [0000]	FF360000	DS[0000] = F4 DS[0001] = 01 SP = 0014 STACK = +0 0000 +2 19F5 IP=0055	DS[0000] = F4 DS[0001] = 01 SP = 0012 STACK= +0 01F4 +2 0000 +4 19F5 IP=0059
0059	PUSH [0002]	FF360200	DS[0002] = CE DS[0003] = FF SP = 0012 STACK=+0 01F4 +2 0000 +4 19F5 IP=0059	DS[0002] = CE DS[0003] = FF SP = 0010 STACK=+0 FFCE +2 01F4 +4 0000 +6 19F5 IP=005D
005D	MOV BP,SP	8BEC	SP=0010 BP=0014 IP=005D	SP=0010 BP=0010 IP=005F
005F	MOV DX,[BP+02]	8B5602	DX=0000 SS[BP+02] = SS[0012]=F4 SS[0013]=01 IP=005F	DX=01F4 SS[BP+02] = SS[0012]=F4 SS[0013] = 01 IP=0062
0062	RET FAR 0002	CA0200	CS=1A0A SP=0010 IP=0062 STACK=+0 FFCE +2 01F4 +4 0000	CS=01F4 SP=0016 IP=FFCE STACK=+0 19F5

			+6 19F5	
--	--	--	---------	--