

**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 и формирования исполнительного адреса**

Студентка гр. 9383

\_\_\_\_\_

Пономаренко С. А.

Преподаватель

\_\_\_\_\_

Ефремов М. А.

Санкт-Петербург

2020

### **Задание.**

Лабораторная работа 2 предназначена для изучения режимов адресации, использует

готовую программу `lr2_compr.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.

### **Тексты исходных файлов программ `hello1` и `hello2`.**

`lb2.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 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

```

*lb2.lst*

#Microsoft (R) Macro Assembler Version 5.10

11/19/20 00:04:1

Page 1-1

```

= 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
                                ?x

0000 0000                            mem1 DW 0
0002 0000                            mem2 DW 0
0004 0000                            mem3 DW 0
0006 0B 0C 0D 0E 12 11                vec1 DB 11,12,13,14,18,17,16,15
                                10 0F
000E 0A 14 F6 EC 1E 28                vec2 DB 10,20,-10,-20,30,40,-30,-40
                                E2 D8
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
                                FE FF 05 06 F8 F9
                                07 08 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 АДРЕСА		
ИИ 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]
lb2.asm(41): error A2052: Improper operand type		
; Базированная адресация		
0020	8A 47 03	mov al,[bx]+3
0023	8B 4F 03	mov cx,3[bx]
; 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]
```

lb2.asm(48): warning A4031: Operand types must match

; Адресация с базирование❖

❖ 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]
```

lb2.asm(52): warning A4031: Operand types must match

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

lb2.asm(53): error A2055: Illegal register value

; ПРОВЕРКА РЕЖИМОВ АДРЕСА❖

❖ИИ С УЧЕТОМ СЕГМЕНТОВ

; Переопределение сегмент

а

; ----- 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]
lb2.asm(72): error A2046: Multiple base registers
0061 3E: 8B 83 0016 R   mov ax,matr[bp+di+si]
lb2.asm(73): error A2047: Multiple index registers
                        ; Использование сегмента ?
                        ?тека
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
lb2.asm(80): error A2006: Phase error between passes
0076                   CODE ENDS
                        END Main

```

## 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 1b2

@VERSION ..... TEXT 510

82 Source Lines

82 Total Lines

19 Symbols

47842 + 459418 Bytes symbol space free

2 Warning Errors

5 Severe Errors

### **Исходные данные.**

Вариант 5.

vec1 11,12,13,14,18,17,16,15

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

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

### **Закомментированные ошибки.**

1. строка 41: mov mem3,[bx] — нельзя перемещать из области памяти в область памяти напрямую, надо через регистры общего назначения  
error A2052: Improper operand type (Неподходящий тип операнда)

2. строка 53: mov ax,matr[bx\*4][di] — 16-битовые регистры нельзя умножать  
error A2055: Illegal register value (Недопустимое значение регистра)

3. строка 72: mov ax,matr[bp+bx] — складывается два базовых регистра (могут складываться базовый с индексным)  
error A2046: Multiple base registers (Несколько базовых регистров)

4. строка 73: `mov ax,matr[bp+di+si]` — складывается два индексных регистра (могут складываться базовый с индексным)  
error A2047: Multiple index registers (Несколько индексных регистров)

**Предупреждения.**

1. строка 48: `mov cx,vec2[di]` — перемещение слова (1 байт) в 16-битовый регистр (2 байта)  
warning A4031: Operand types must match (Типы операндов должны совпадать)

2. строка 52: `mov cx,matr[bx][di]` — перемещение слова (1 байт) в 16-битовый регистр (2 байта)  
warning A4031: Operand types must match (Типы операндов должны совпадать)

**Листинг успешной трансляции.**

*lb2.asm*

```
= 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 данн?
                ?x

0000 0000      mem1 DW 0
0002 0000      mem2 DW 0
0004 0000      mem3 DW 0
0006 0B 0C 0D 0E 12 11      vec1 DB 11,12,13,14,18,17,16,15
                10 0F
000E 0A 14 F6 EC 1E 28      vec2 DB 10,20,-10,-20,30,40,-30,-40
                E2 D8
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
                FE FF 05 06 F8 F9
                07 08 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 АДРЕСА ❖
❖ ИИ 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]

; 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]
```

lb2.asm(48): warning A4031: Operand types must match

; Адресация с базирование

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]
```

lb2.asm(52): warning A4031: Operand types must match

; mov ax,matr[bx\*4][di]

; ПРОВЕРКА РЕЖИМОВ АДРЕСА

ИИ С УЧЕТОМ СЕГМЕНТОВ

; Переопределение сегмент

а

; ----- вариант 1

```
003C B8 ---- R        mov ax, SEG vec2
003F 8E C0              mov es, ax
0041 26: 8B 07          mov ax, es:[bx]
0044 B8 0000            mov ax, 0
```

; ----- вариант 2

```
0047 8E C0              mov es, ax
0049 1E                 push ds
004A 07                 pop es
004B 26: 8B 4F FF        mov cx, es:[bx-1]
004F 91                 xchg cx,ax
```

```

; ----- вариант 3
0050 BF 0002          mov di,ind
0053 26: 89 01       mov es:[bx+di],ax
; ----- вариант 4
0056 8B EC          mov bp,sp
; mov ax,matr[bp+bx]
; mov ax,matr[bp+di+si]
; Использование сегмента ?
? тека
0058 FF 36 0000 R    push mem1
005C FF 36 0002 R    push mem2
0060 8B EC          mov bp,sp
0062 8B 56 02       mov dx,[bp]+2
0065 CA 0002       ret 2
0068               Main ENDP
0068               CODE ENDS
END Main

```



## Symbols-1

## Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK .....	0018	PARA		STACK
CODE .....	0068	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 = 0068
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 1b2

@VERSION ..... TEXT 510

82 Source Lines

82 Total Lines

19 Symbols

47842 + 459418 Bytes symbol space free

2 Warning Errors

0 Severe Errors

### **Протокол пошагового исполнения программы.**

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти до	после выполнения
0	PUSH DS	1E	SP = 0018 IP = 0000 Stack +0 0000	SP = 0016 IP = 0001 Stack +0 19F5
1	SUB AX, AX	2BC0	IP = 0001 AX = 0000	IP = 0003 AX = 0000
3	PUSH AX	50	SP = 0016 IP = 0003 Stack +0 19F5 Stack +2 0000	SP = 0014 IP = 0004 Stack +0 0000 Stack +2 19F5
4	MOV AX, 1A07	B8071A	AX = 0000 IP = 0004	AX = 1A07 IP = 0007
7	MOV DS, AX	8ED8	IP = 0007 DS = 19F5	IP = 0009 DS = 1A07
9	MOV AX, 01F4	B8F401	AX = 1A07 IP = 0009	AX = 01F4 IP = 000C
000C	MOV CX, AX	8BC8	CX = 00B8 IP = 000C	CX = 01F4 IP = 000E

000E	MOV BL, 24	B324	BX = 0000 IP = 000E	BX = 0024 IP = 0010
10	MOV BH, CE	B7CE	BX = 0024 IP = 0010	BX = CE24 IP = 0012
12	MOV [0002], FFCE	C7060200CEFF	IP = 0012 DS: 0002 = 00 DS: 0003 = 00	IP = 0018 DS: 0002 = CE DS: 0003 = FF
18	MOV BX, 0006	BB0600	BX = CE24 IP = 0018	BX = 0006 IP = 001B
001B	MOV [0000], AX	A30000	IP = 001B DS: 0000 = 00 DS: 0001 = 00	IP = 001E DS: 0002 = F4 DS: 0003 = 01
001E	MOV AL, [BX]	8A07	AX = 01F4 IP = 001E	AX = 010B IP = 0020
20	MOV AL, [BX+03]	8A4703	AX = 010B IP = 0020	AX = 010E IP = 0023
23	MOV CX, [BX+03]	8B4F03	CX = 01F4 IP = 0023	CX = 120E IP = 0026
26	MOV DI, 0002	BF0200	DI = 0000 IP = 0026	DI = 0002 IP = 0029
29	MOV AL, [000E+DI]	8A850E00	AX = 010E IP = 0029	AX = 01F6 IP = 002D
002D	MOV CX, [000E+DI]	8B8D0E00	CX = 120E IP = 002D	CX = ECF6 IP = 0031
31	MOV BX, 0003	BB0300	BX = 0006 IP = 0031	BX = 0003 IP = 0034
34	MOV AL, [0012+BX+DI]	8A811600	AX = 01F6 IP = 0034	AX = 0104 IP = 0038
38	MOV CX, [0016+BX+DI]	8B891600	CX = ECF6 IP = 0038	CX = FE04 IP = 003C
003C	MOV AX, 1A07	B8071A	AX = 0104 IP = 003C	AX = 1A07 IP = 003F
003F	MOV ES, AX	8EC0	ES = 19F5 IP = 003F	ES = 1A07 IP = 0041
41	MOV AX, ES: [BX]	268B07	AX = 1A07 IP = 0041	AX = 00FF IP = 0044
44	MOV AX, 0000	B80000	AX = 00FF IP = 0044	AX = 0000 IP = 0047
47	MOV ES, AX	8EC0	ES = 1A07 IP = 0047	ES = 0000 IP = 0049
49	PUSH DS	1E	Stack +0 0000 Stack +2 19F5 Stack +4 0000 IP = 0049	Stack +0 1A07 Stack +2 0000 Stack +4 19F5 IP = 004A

004A	POP ES	7	ES = 0000 Stack +0 1A07 Stack +2 0000 Stack +4 19F5 IP = 004A	ES = 1A07 Stack +0 0000 Stack +2 19F5 Stack +4 0000 IP = 004B
004B	MOV CX, ES: [BX-01]	268B4FFF	CX = FE04 IP = 004B	CX = FFCE IP = 004F
004F	XCHG AX, CX	91	AX = 0000 CX = FFCE IP = 004F	AX = FFCE CX = 0000 IP = 0050
50	MOV DI, 0002	BF0200	DI = 0002 IP = 0050	DI = 0002 IP = 0053
53	MOV ES: [BX+DI], AX	268901	IP = 0053 DS: 0005 = 00 DS: 0006 = 0B	IP = 0056 DS: 0005 = CE DS: 0006 = FF
56	MOV BP, SP	8BEC	BP = 0000 IP = 0056	BP = 0014 IP = 0058
58	PUSH [0000]	FF360000	IP = 0058 Stack +0 0000 Stack +2 19F5 Stack +4 0000	IP = 005C Stack +0 01F4 Stack +2 0000 Stack +4 19F5
005C	PUSH [0002]	FF360200	IP = 005C Stack +0 01F4 Stack +2 0000 Stack +4 19F5 Stack +6 0000	IP = 0060 Stack +0 FFCE Stack +2 01F4 Stack +4 0000 Stack +6 19F5
60	MOV BP, SP	8BEC	BP = 0014 IP = 0060	BP = 0010 IP = 0062
62	MOV DX, [BP+02]	8B5602	DX = 0000 IP = 0062	DX = 01F4 IP = 0065
65	RET Far, 0002	CA0200	SP = 0010 CS = 1A0A Stack +0 FFCE Stack +2 01F4 Stack +4 0000 Stack +6 19F5	SP = 0016 CS = 01F4 Stack +0 19F5 Stack +2 0000 Stack +4 0000 Stack +6 0000

### **Вывод.**

Изучены режимы адресации и формирования исполнительного адреса, рассмотрены примеры; устранены ошибки; скомпилирована программа.