

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Изучение режимов адресации и формирования исполнительного**  
**адреса.**

Студент гр. 9382

\_\_\_\_\_

Иерусалимов Н.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Изучение режимов адресации и формирования исполнительного адреса на языке Ассемблер.

### **Постановка задачи.**

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.**

1. В каталок был загружен файл `main.asm` компилятор `masm` и линковщик
2. Код программы был рассмотрен в Visual Studio. Изучена структура и реализация каждого сегмента программы.
3. Были занесен набор значений, заменяющих приведённые в образце программы.

- а) Трансляция исходного кода.
- б) Была открыта программа DOSBOX.

- в) Транслируем программу в DOSBOX следующим образом: монтируем директорию «**mount C C:\masm**» и совершаем переход в нее «**C:**». Транслируем программу с помощью команды «> **masm main.asm**»

В процессе трансляции был создан файл листина main.LST. Файл листинга содержит диагностическую информацию в виде сообщений о двух предупреждениях (Warning errors) и пяти ошибках (Severe errors).

- г) Демонстрация кода файла листинга приведена в приложении А. Ошибки и предупреждения выделены жирным.

#### **Объяснение ошибок.**

1. LR2\_COMP.ASM(55): error A2052: Improper operand type
2. LR2\_COMP.ASM(62): warning A4031: Operand types must match
3. LR2\_COMP.ASM(66): warning A4031: Operand types must match
4. LR2\_COMP.ASM(67): error A2055: Illegal register value
5. LR2\_COMP.ASM(87): error A2046: Multiple base registers
6. LR2\_COMP.ASM(88): error A2047: Multiple index registers

- 1) `mov mem3,[bx]`

Эта команда переводит информацию из сегмента памяти в другой, что невозможно в языке Assembler. В данном случае необходимо перевести информацию из памяти в регистр, а затем уже в необходимый сегмент информацию перевести из регистра.

- 2) `mov ax,vec2[di]`

Объем регистра ax составляет 2 байта, в то время, как вес элемента в массиве составляет 1 байт, в данном случае надо вместо регистра ax использовать, например al

- 3) см.2

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

Здесь используется базово-индексная адресация. При данном типе адресации надо сначала изменить значение регистра, затем уже переводить информацию.

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

В базированной адресации необходимо указывать базовый регистр, затем производить смещение с помощью индексного. Так как здесь оба базовые, надо сначала сложить значения регистров, затем уже передавать информацию указателю из одного регистра.

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

Здесь ошибка, похожая на ошибку в п.4, за одним исключением- повторяются индексные регистры. В таком случае необходимо сначала а регистр `di` занести общую сумму, затем уже производить смещение.

Ошибки закомментированы путем редактирования варнинги не трогали.

Снова запустили `masm main.asm`. Исправленный код в приложении Б

Командой **link main.ASM** был скомпонован **main.exe**. Программа была выполнена в режиме отладки командой **afd main.exe**.

Результат работы программы.

| Адрес команды | Символический код команды | 16-ричный код команды | Содержимое регистров до выполнения команды | Содержимое регистров после выполнения команды | Содержимое стека до выполнения команды   | Содержимое стека после выполнения команды |
|---------------|---------------------------|-----------------------|--|---|--|---|
| 0000          | PUSH DS                   | 1E                    | SP=0018<br>IP=0000                         | SP=0016<br>IP=0001                            | +0 0000<br>+2 0000<br>+4 0000<br>+6 0000 | +0 19F5<br>+2 0000<br>+4 0000<br>+6 0000  |
| 0001          | SUB AX, AX                | 2BC0                  | AX=0000<br>IP=0001                         | AX=0000<br>IP=0003                            |  |   |
| 0003          | PUSH AX                   | 50                    | SP=0016<br>IP=0003                         | SP=0014<br>IP=0004                            | +0 19F5<br>+2 0000<br>+4 0000<br>+6 0000 | +0 0000<br>+2 19F5<br>+4 0000<br>+6 0000  |
| 0004          | MOV AX, 1A07              | B8AE11                | AX=0000<br>IP=0004                         | AX=1A07<br>IP=0007                            |  |   |

|      |                         |                  |                               |                               |  |  |
|------|-------------------------|------------------|-------------------------------|-------------------------------|--|--|
| 0007 | MOV DS, AX              | 8ED8             | DS=19F5<br>IP=0007            | DS=1A07<br>IP=0009            |  |  |
| 0009 | MOV AX,<br>01F4         | B8F401           | AX=1A07<br>IP=0009            | AX=01F4<br>IP=000C            |  |  |
| 000C | MOV CX, AX              | 8BC8             | CX=00B8<br>IP=000C            | CX=01F4<br>IP=000E            |  |  |
| 000E | MOV BL, 24              | B324             | BX=0000<br>IP=000E            | BX=0024<br>IP=0010            |  |  |
| 0010 | MOV BH, CE              | B7CE             | BX=0024<br>IP=0010            | BX=CE24<br>IP=0012            |  |  |
| 0012 | MOV [0002],<br>FFCE     | C70602<br>00CEFF | IP=0012                       | IP=0018                       |  |  |
| 0018 | MOV BX,<br>0006         | BB0600           | BX=CE24<br>IP=0018            | BX=0006<br>IP=001B            |  |  |
| 001B | MOV [0000],<br>AX       | A30000           | IP=001B                       | IP=001E                       |  |  |
| 001E | MOV AL,<br>[BX]         | 8A07             | AX=01F4<br>IP=001E            | AX=0101<br>IP=0020            |  |  |
| 0020 | MOV AL,<br>[BX+03]      | 8A4703           | AX=0101<br>IP=0020            | AX=0104<br>IP=0023            |  |  |
| 0023 | MOV CX,<br>[BX+03]      | 8B4F03           | CX=01F4<br>IP=0023            | CX=0804<br>IP=0026            |  |  |
| 0026 | MOV DI,<br>0002         | BF0200           | DI=0000<br>IP=0026            | DI=0002<br>IP=0029            |  |  |
| 0029 | MOV AL,<br>[DI+000E]    | 8A85<br>0E00     | AX=0104<br>IP=0029            | AX=010A<br>IP=002D            |  |  |
| 002D | MOV CX,<br>[DI+000E]    | 8B8D0E00         | CX=0804<br>IP=002D            | CX=140A<br>IP=0031            |  |  |
| 0031 | MOV BX,<br>0003         | BB0300           | BX=0006<br>IP=0031            | BX=0003<br>IP=0034            |  |  |
| 0034 | MOV AL,<br>[BX+DI+0016] | 8A811600         | AX=010A<br>IP=0034            | AX=01FD<br>IP=0038            |  |  |
| 0038 | MOV CX,<br>[BX+DI+0016] | 8B891600         | CX=140A<br>IP=0038            | CX=FEFD<br>IP=003C            |  |  |
| 003C | MOV AX,<br>1A07         | B8AE11           | AX=01FD<br>IP=003C            | AX=1A07<br>IP=003F            |  |  |
| 003F | MOV ES, AX              | 8ECO             | ES=19F5<br>IP=003F            | ES=1A07<br>IP=0041            |  |  |
| 0041 | MOV AX,<br>ES:[BX]      | 268B07           | AX=1A07<br>IP=0041            | AX=00FF<br>IP=0044            |  |  |
| 0044 | MOV AX,<br>0000         | B80000           | AX=00FF<br>IP=0044            | AX=0000<br>IP=0047            |  |  |
| 0047 | MOV ES, AX              | 8ECO             | ES=1A07<br>IP=0047            | ES=0000<br>IP=0049            |  |  |
| 0049 | PUSH DS                 | 1E               | SP=0014<br>IP=0049            | SP=0012<br>IP=004A            | +0 0000<br>+2 19F5<br>+4 0000<br>+6 0000 | +0 1A07<br>+2 0000<br>+4 19F5<br>+6 0000 |
| 004A | POP ES                  | 07               | SP=0012<br>ES=0000<br>IP=004A | SP=0014<br>ES=1A07<br>IP=004B | +0 1A07<br>+2 0000<br>+4 19F5<br>+6 0000 | +0 0000<br>+2 19F5<br>+4 0000<br>+6 0000 |

|      |                          |          |                               |                               |  |  |
|------|--------------------------|----------|-------------------------------|-------------------------------|--|--|
| 004B | MOV CX,<br>ES:[BX-01]    | 268B4FFF | CX=FEFD<br>IP=004B            | CX=FFCE<br>IP=004F            |  |  |
| 004F | XCHG                     | 91       | AX=0000<br>CX=FFCE<br>IP=004F | AX=FFCE<br>CX=0000<br>IP=0050 |  |  |
| 0050 | MOV DI,<br>0002          | BF0200   | DI=0002<br>IP=0050            | DI=0002<br>IP=0053            |  |  |
| 0053 | MOV<br>ES:[BX+DI],<br>AX | 268901   | ES=1A07<br>IP=0053            | ES=1A07<br>IP=0056            |  |  |
| 0056 | MOV BP, SP               | 8BEC     | BP=0000<br>IP=0056            | BP=0014<br>IP=0058            |  |  |
| 0058 | PUSH [0000]              | FF360000 | SP=0014<br>IP=0058            | SP=0012<br>IP=005C            | +0 0000<br>+2 19F5<br>+4 0000<br>+6 0000 | +0 01F4<br>+2 0000<br>+4 19F5<br>+6 0000 |
| 005C | PUSH [0002]              | FF360200 | SP=0012<br>IP=005C            | SP=0010<br>IP=0060            | +0 01F4<br>+2 0000<br>+4 19F5<br>+6 0000 | +0 FFCE<br>+2 01F4<br>+4 0000<br>+6 19F5 |
| 0060 | MOV BP, SP               | 8BEC     | BP=0014<br>IP=0060            | BP=0010<br>IP=0062            |  |  |
| 0062 | MOV DX,<br>[BP+02]       | 8B5602   | DX=0000<br>IP=0062            | DX=01F4<br>IP=0065            |  |  |
| 0065 | RET FAR 0002             | CA0200   | SP=0010,<br>CS=11B1           | SP=0016,<br>CS=01F4           | +0 FFCE<br>+2 01F4<br>+4 0000<br>+6 19F5 | +0 19F5<br>+2 0000<br>+4 0000<br>+6 0000 |

### Вывод.

В результате выполнения лабораторной работы были изучены различные виды адресации (регистровая, прямая, косвенная, базированная, индексированная адресации и адресация с базированием и индексированием).

# ПРИЛОЖЕНИЕ А

## ФАЙЛ ЛИСТИНГА С ОБЩИБКАМИ

Microsoft (R) Macro Assembler Version 5.10

11/9/20 00:40:40

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
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
                ; 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 АДРЕСАЦИИ 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]
MAIN.ASM(42): 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
__Microsoft (R) Macro Assembler Version 5.10
11/9/20 00:40:40
Page      1-2

0029  8A 85 000E R          mov al,vec2[di]
002D  8A 05                mov al,vec2[di]*/
MAIN.ASM(49): error A2027: Operand expected
002F  8B 8D 000E R          mov cx,vec2[di]
MAIN.ASM(50): warning A4031: Operand types must match
                                ; Адресация с базированием и индексированием
0033  BB 0003                mov bx,3
0036  8A 81 0016 R          mov al,matr[bx][di]

003A  8B 89 0016 R          mov cx,matr[bx][di]
MAIN.ASM(55): warning A4031: Operand types must match
003E  8B 85 0022 R          mov ax,matr[bx*4][di]
MAIN.ASM(56): error A2055: Illegal register value
                                ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ
                                ; Переопределение сегмента
                                ; ----- вариант 1
0042  B8 ---- R            mov ax, SEG vec2
0045  8E C0                mov es, ax
0047  26: 8B 07            mov ax, es:[bx]
004A  B8 0000                mov ax, 0
                                ; ----- вариант 2
004D  8E C0                mov es, ax
004F  1E                  push ds
0050  07                  pop es
0051  26: 8B 4F FF          mov cx, es:[bx-1]
0055  91                  xchg cx,ax
                                ; ----- вариант 3
0056  BF 0002                mov di,ind
0059  26: 89 01            mov es:[bx+di],ax
                                ; ----- вариант 4
005C  8B EC                mov bp,sp
005E  3E: 8B 86 0016 R      mov ax,matr[bp+bx]
MAIN.ASM(75): error A2046: Multiple base registers
0063  3E: 8B 83 0016 R      mov ax,matr[bp+di+si]
MAIN.ASM(76): error A2047: Multiple index registers
                                ; Использование сегмента стека
0068  FF 36 0000 R          push mem1
006C  FF 36 0002 R          push mem2
0070  8B EC                mov bp,sp
0072  8B 56 02                mov dx,[bp]+2
0075  CA 0002                ret 2
0078                                Main ENDP
MAIN.ASM(83): error A2006: Phase error between passes
0078                                CODE ENDS
                                END Main
__Microsoft (R) Macro Assembler Version 5.10
11/9/20 00:40:40
Symbols-1

```

Segments and Groups:

| N a m e          | Length | Align | Combine | Class |
|------------------|--------|-------|---------|-------|
| ASTACK . . . . . | 0018   | PARA  |         | STACK |



|                |      |      |      |
|----------------|------|------|------|
| CODE . . . . . | 0078 | 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 = 0078 |
| 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   | MAIN  |                    |
| @VERSION . . . . .  | TEXT   | 510   |                    |

85 Source Lines  
85 Total Lines  
19 Symbols

47828 + 461479 Bytes symbol space free

2 Warning Errors  
6 Severe Errors

## ПРИЛОЖЕНИЕ Б

### ИСПРАВЛЕННЫЙ КОД ПРОГРАММЫ

```

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