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. 0383 _____ Арсентьева. Д.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2021

Цель работы.

Изучение режимов адресации и формирования исполнительного адреса

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.

Порядок выполнения работы: 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.

Вариант 2:

vec1 DB 5,6,7,8,12,11,10,9

vec2 DB -20,-30,20,30,-40,-50,40,50

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

Выполнение работы.

1. Получен вариант вариант набора значений исходных данных (массивов) vec1, vec2 и matr из файла lr2.dat (Вариант 2) и занести свои данные вместо значений примера программы в методических материалах (файл lr2_comp.asm).

2. Программу протранслирована с созданием файла диагностических сообщений. Закомментированы соответствующие операторы в тексте программы (файл lr2_comp_fix.asm). Ниже объяснены обнаруженные ошибки:

- Mov mem3,[BX] lr2_comp.asm(46): error A2052: Improper operand type

- Неподходящий тип операндов. Нельзя напрямую перекладывать данные из одной ячейки памяти, в другую.

- Mov CX,vec2[DI] lr2_comp.asm(53): warning A4031: Operand typES must match

- Типы операндов должны совпадать. Размер регистра ‘cx’ равен 2 байтам, а размер элементов массива ‘vec2’ - 1 байт.

- Mov CX,matr[BX][DI] lr2_comp.asm(57): warning A4031: Operand typES must match

- Типы операндов должны совпадать. Размер регистра ‘cx’ равен 2 байтам, а размер элементов матрицы ‘matr’ - 1 байт.

- Mov AX,matr[BX*4][DI] lr2_comp.asm(58): error A2055: IllegAL register vALue

- Недопустимое значение регистра.

- Mov AX,matr[BP+BX] lr2_comp.asm(78): error A2046: Multiple base registers

- Несколько базовых регистров. Нельзя использовать несколько базовых регистров для адресации.

❖ Mov AX,matr[BP+DI+si] lr2_comp.asm(79): error A2047: Multiple index registers

- Несколько индексных регистров. Нельзя использовать несколько индексных регистров для адресации.

3. Программа снова протранслирована, и скомпонован загрузочный модуль

4. Программу выполнена в пошаговом режиме под управлением отладчика с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды. Ниже приведена таблица 1.

Начальное содержимое сегментных регистров: (CS) = 1A0A, (DS) = 19F5, (ES) = 19F5 и (SS) = 1A05 .

Таблица 1 – Результат выполнения пункта 4.

Адрес Команды	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 Stack +2 0000	(SP) = 0016 (IP) = 0001 Stack +0 19F5 Stack +2 0000
0001	Sub AX, AX	2BC0	(IP) = 0001	(IP) = 0003
0003	Push AX	50	(SP) = 0016 (IP) = 0003 Stack +0 19F5 Stack +2 0000 Stack +4 0000	(SP) = 0014 (IP) = 0004 Stack +0 0000 Stack +2 19F5 Stack +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	(IP) = 0010	(IP) = 0012
0012	Mov [0002], FFCE	C7060200CEFF	(IP) = 0012	(IP) = 0018
0018	Mov BX, 0006	BB0600	(BX) = CE24 (IP) = 0018	(BX) = 0006 (IP) = 001B
001B	Mov [0000], AX	A30000	(IP) = 001B	(IP) = 000E
001E	Mov AL, [BX]	8A07	(AX) = 01F4 (IP) = 000E	(AX) = 0105 (IP) = 0020
0020	Mov AL, [BX+03]	8A4703	(AX) = 0105 (IP) = 0020	(AX) = 0108 (IP) = 0023
0023	Mov CX, [BX+03]	8B4F03	(CX) = 01F4 (IP) = 0023	(CX) = 0C08 (IP) = 0026
0026	Mov DI, 0002	DF0200	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	Mov AL, [000E+DI]	8A850E00	(AX) = 0122 (IP) = 0029 Stack +0 001A Stack +2 0000 Stack +4 19F5 Stack +6 0000	(AX) = 0114 (IP) = 002D Stack +0 0000 Stack +2 19F5 Stack +4 0000 Stack +6 0000
002D	Mov BX, 0003	BB0300	(BX) = 0006	(BX) = 0003

			(IP) = 002D	(IP) = 0030
0030	Mov AL, [0016+BX+DI]	8A811600	(AX) = 0114 (IP) = 0030	(AX) = 0103 (IP) = 0034
0034	Mov AX, 1A07	B8071A	(AX) = 0103 (IP) = 0034	(AX) = 1A07 (IP) = 0037
0037	Mov ES, AX	8ECO	(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	8ECO	(ES) = 1A07 (IP) = 003F	(ES) = 0000 (IP) = 0041
0041	Push DS	1E	(SP) = 0014 (IP) = 0041 Stack +0 0000 Stack +2 19F5 Stack +4 0000 Stack +6 0000	(SP) = 0012 (IP) = 0042 Stack +1A07 Stack +2 0000 Stack +4 19F5 Stack +6 0000
0042	Pop ES	07	(SP) = 0012 (ES) = 0000 (IP) = 0042 Stack +1A07 Stack +2 0000 Stack +4 19F5 Stack +6 0000	(SP) = 0014 (ES) = 1A07 (IP) = 0043 Stack +0 0000 Stack +2 19F5 Stack +4 0000 Stack +6 0000
0043	Mov CX, ES:[BX-01]	268B4FFF	(CX) = 0C08 (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	(ES) = 1A07 (IP) = 004B	(ES) = 1A07 (IP) = 004E
004E	Mov BP, SP	8BEC	(SP) = 0012 (BP) = 0000 (IP) = 004E	(SP) = 0014 (BP) = 0014 (IP) = 0050
0050	Push [0000]	FF360000	(SP) = 0014 (IP) = 0050 Stack +0 0000 Stack +2 19F5 Stack +4 0000 Stack +6 0000	(SP) = 0012 (IP) = 0054 Stack +0 01F4 Stack +2 0000 Stack +4 19F5 Stack +6 0000
0054	Push [0002]	FF360200	(SP) = 0012 (IP) = 0054 Stack +0 01F4 Stack +2 0000 Stack +4 19F5 Stack +6 0000	(SP) = 0010 (IP) = 0058 Stack +0 FFCE Stack +2 01F4 Stack +4 0000 Stack +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	(CS) = 1A0A (SP) = 0010 (IP) = 005D	(CS) = 01F4 (SP) = 0016 (IP) = FFCE

			Stack+0 FFCE Stack +2 01F4 Stack +4 0000 Stack +6 19F5	Stack +0 19F5 Stack +2 0000 Stack +4 0000 Stack +6 0000

Тексты исходных файлов программ см. в приложении А.

Тексты файлов диагностических сообщений см. в приложении В.

Выводы.

Были изучены режимы адресации и формирования исполнительного адреса.

ПРИЛОЖЕНИЕ А
ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: lr2_comp.asm

```
; Программа изучения режимов адресации процессора IntelX86
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 5,6,7,8,12,11,10,9
vec2 DB -20,-30,20,30,-40,-50,40,50
matr DB -5,-6,-7,-8,4,3,2,1,-1,-2,-3,-4,8,7,6,5
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 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
```

Название файла: lr2_comp_fix.asm

```
; Программа изучения режимов адресации процессора IntelX86
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 5,6,7,8,12,11,10,9
vec2 DB -20,-30,20,30,-40,-50,40,50
matr DB -5,-6,-7,-8,4,3,2,1,-1,-2,-3,-4,8,7,6,5
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 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
ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lr2_comp.lst

```
Microsoft (R) Macro Assembler Version 5.10              6/10/21 20:43:17
                                                        Page     1-1


                        ; Программа изучения режи♦
                        ♦ов адресации процессора I
                        ntelX86
= 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 данны
                        x
0000  0000              mem1    DW   0
0002  0000              mem2    DW   0
0004  0000              mem3    DW   0
0006  05 06 07 08 0C 0B        vec1    DB   5,6,7,8,12,11,10,9
    0A 09
000E  EC E2 14 1E D8 CE        vec2    DB   -20,-30,20,30,-40,-50,40,50
    28 32
0016  FB FA F9 F8 04 03        matr    DB   -5,-6,-7,-8,4,3,2,1,-1,-2,-3,-4
                        ,8,7,6,5
    02 01 FF FE FD FC
    08 07 06 05
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
                        ЦИИ 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

Microsoft (R) Macro Assembler Version 5.10          6/10/21 20:43:17
                                    Page     1-2


001B  A3 0000 R               mov  mem1,ax
                        ;  Косвенная адресация
001E  8A 07                     mov  al,[bx]
                              mov  mem3,[bx]
lb2.asm(46): 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(53): warning A4031: Operand types must match
                        ;  Адресация с базирование
                        м 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(57): warning A4031: Operand types must match
 003C  8B 85 0022 R                    mov  ax,matr[bx*4][di]
lb2.asm(58): error A2055: Illegal register value

                                       ;  ПРОВЕРКА РЕЖИМОВ АДРЕСА
                                       ЦИИ С УЧЕТОМ СЕГМЕНТОВ
                                       ;  Переопределение сегмен
                                       a
                                       ;  ------ 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(78): error A2046: Multiple base registers
 0061  3E: 8B 83 0016 R                mov  ax,matr[bp+di+si]
lb2.asm(79): error A2047: Multiple index registers
                                       ;  Использование сегмента
                                       стека
 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(86): error A2006: Phase error between passes
 0076                                  CODE      ENDS
                                       END Main

Microsoft (R) Macro Assembler Version 5.10          6/10/21 20:43:17
                                                    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  lb2
@VERSION . . . . . . . . . . .  TEXT  510


    88 Source  Lines
    88 Total   Lines
    19 Symbols

  47826 + 459431 Bytes symbol space free

    2 Warning Errors
    5 Severe  Errors
```

Название файла: lr2_comp_fix.lst

```
Microsoft (R) Macro Assembler Version 5.10                6/10/21 21:15:49
                                                          Page     1-1


                                ; Программа изучения режи�
                                
                                ��ов адресации процессора I
                                
                                ntelX86
= 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 данны
                        x
0000  0000              mem1    DW   0
0002  0000              mem2    DW   0
0004  0000              mem3    DW   0
0006  05 06 07 08 0C 0B       vec1    DB   5,6,7,8,12,11,10,9
    0A 09
000E  EC E2 14 1E D8 CE       vec2    DB   -20,-30,20,30,-40,-50,40,50
    28 32
0016  FB FA F9 F8 04 03       matr    DB   -5,-6,-7,-8,4,3,2,1,-1,-2,-3,-4
                        ,8,7,6,5
    02 01 FF FE FD FC
    08 07 06 05
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
                        ЦИИ 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

Microsoft (R) Macro Assembler Version 5.10          6/10/21 21:15:49
                                                    Page     1-2


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]
                            ;mov  cx,vec2[di]
                        ; Адресация с базирование
                        м и индексированием
002D  BB 0003                   mov  bx,3
0030  8A 81 0016 R              mov  al,matr[bx][di]
                            ;mov  cx,matr[bx][di]
                            ;mov  ax,matr[bx*4][di]
```

```
                        ; ПРОВЕРКА РЕЖИМОВ АДРЕСА
                        ЦИИ С УЧЕТОМ СЕГМЕНТОВ
                        ; Переопределение сегмен►

                        ►a

                        ; ------ вариант 1
0034  B8 ---- R                 mov  ax, SEG vec2
0037  8E C0                           mov  es, ax
0039  26: 8B 07                 mov  ax, es:[bx]
003C  B8 0000                         mov  ax, 0
                        ; ------ вариант 2
003F  8E C0                           mov  es, ax
0041  1E                        push ds
0042  07                        pop  es
0043  26: 8B 4F FF                    mov  cx, es:[bx-1]
0047  91                        xchg cx,ax
                        ; ------ вариант 3
0048  BF 0002                         mov  di,ind
004B  26: 89 01                 mov  es:[bx+di],ax
                        ; ------ вариант 4
004E  8B EC                           mov  bp,sp
                        ;mov  ax,matr[bp+bx]
                        ;mov  ax,matr[bp+di+si]
                        ; Использование сегмента
                        стека
0050  FF 36 0000 R                    push  mem1
0054  FF 36 0002 R                    push  mem2
0058  8B EC                           mov   bp,sp
005A  8B 56 02                        mov   dx,[bp]+2
```

```
005D  CA 0002                    ret   2
0060                      Main    ENDP
0060                      CODE    ENDS
                          END Main
```

```
Microsoft (R) Macro Assembler Version 5.10          6/10/21 21:15:49
                                                    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  lb2
@VERSION . . . . . .       TEXT  510


    88 Source  Lines
    88 Total   Lines
    19 Symbols

 47814 + 459443 Bytes symbol space free

     0 Warning Errors
     0 Severe  Errors
```