МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ

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

Студент гр. 1383 _____ Депрейс А.С.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Изучить режимы адресации на языке Ассемблера.

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. Изменены исходные значения vec1, vec2, matr согласно варианту.
2. Была произведена попытка трансляции файла с получением ошибок продемонстрированных на рисунке 1.

Рисунлк 1 – Ошибки при первой трансляции файла.

3. Закомментированы следующие строки:

1) mov mem3, [bx] – Запрещено в качестве операндов команды mov не использовать регистр или значение, пересылки память-память запрещена. (error A2052: Improper operand type)

2) mov cx,vec2[di] – Операнды имеют разную длину, cx – 2 байта, vec2[di] – 1 байт. (warning A4031: Operand types must match)

3) mov cx,matr[bx][di] - Операнды имеют разную длину, cx – 2 байта, matr[bx][di] – 1 байт. (warning A4031: Operand types must match)

4) mov ax,matr[bx*4][di] – При базово-индексной адресации с масштабированием нельзя умножать базовый регистр, следует умножать регистр di в зависимости от размера элемента массива. (error A2055: Illegal register value)

5) mov ax,matr[bp+bx] – Должен быть только 1 базовый регистр. (error A2046: Multiple base registers)

6) mov ax,matr[bp+di+si] – Должен быть только 1 индексный регистр. (error A2047: Multiple index registers)

7) Добавлены два pop ax для успешного завершения программы.

4. Проведена успешная трансляция файла.

5. Начальное содержимое регистров:

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

Таблица 1 - результат прогона программы LR2_COMP.EXE в отладчике.

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	IP = 0000 SP = 0018 Stack: +0 0000	IP = 0001 SP = 0016 Stack: +0 19F5
0001	SUB AX,AX	2B C0	IP = 0001 AX = 0000	IP = 0003 AX = 0000
0003	PUSH AX	50	IP = 0003 SP = 0016 Stack: +0 19F5 +2 0000	IP = 0004 SP = 0014 Stack: +0 0000 +2 19F5
0004	MOV AX,1A07	B8 07 1A	IP = 0004 AX = 0000	IP = 0007 AX = 1A07
0007	MOV DS,AX	8E D8	IP = 0007 DS = 19F5	IP = 0009 DS = 1A07
0009	MOV AX,01F4	B8 F4 01	IP = 0009 AX = 1A07	IP = 000C AX = 01F4
000C	MOV CX,AX	8B C8	IP = 000C CX = 00AE	IP = 000E CX = 01F4
000E	MOV BL,24	B3 24	IP = 000E	IP = 0010

			BX = 0000	BX = 0024
0010	MOV BH,CE	B7CE	IP = 0010 BX = 0024	IP = 0012 BX = CE24
0012	MOV [0002],FFCE	C7 06 02 00 CE FF	IP = 0012 DS:0002 = 00 DS:0003 = 00	IP = 0018 DS:0002 = CE DS:0003 = FF
0018	MOV BX,0006	BB 06 00	IP = 0018 BX = CE24	IP = 001B BX = 0006
001B	MOV [0000],AX	A3 00 00	IP = 001B DS:0000 = 00 DS:0001 = 00	IP = 001E DS:0000 = F4 DS:0001 = 01
001E	MOV AL,[BX]	8A 07	IP = 001E AX = 01F4	IP = 0020 AX = 0115
0020	MOV AL,[BX+03]	8A 47 03	IP = 0020 AX = 0115	IP = 0023 AX = 0118
0023	MOV CX,[BX+03]	8B 4F 03	IP = 0023 CX = 01F4	IP = 0026 CX = 1C18
0026	MOV DI,0002	BF 02 00	IP = 0026 DI = 0000	IP = 0029 DI = 0002
0029	MOV AL,[DL]	8A 05	IP = 0029 AX = 0118	IP = 002B AX = 01CE
002B	MOV BX,0003	BB 03 00	IP = 002B BX = 0006	IP = 002E BX = 0003
002E	MOV AL,[0016+BX+DI]	8A 81 16 00	IP = 002E AX = 01CE	IP = 0032 AX = 0108
0032	MOV AX,1A07	B8 07 1A	IP = 0032 AX = 0108	IP = 0035 AX = 1A07
0035	MOV ES,AX	8E C0	IP = 0035 ES = 19F5	IP = 0037 ES = 1A07

0037	MOV AX,ES:[BX]	26 8B 07	IP = 0037 AX = 1A07	IP = 003A AX = 00FF
003A	MOV AX,0000	B8 00 00	IP = 003A AX = 00FF	IP = 003D AX = 0000
003D	MOV ES,AX	8E C0	IP = 003D ES = 1A07	IP = 003F ES = 0000
003F	PUSH DS	1E	IP = 003F SP = 0014 Stack: +0 0000 +2 19F5 +4 0000	IP = 0040 SP = 0012 Stack: +0 1A07 +2 0000 +4 19F5
0040	POP ES	07	IP = 0040 ES = 0000 SP = 0012 Stack: +0 1A07 +2 0000 +4 19F5	IP = 0041 ES = 1A07 SP = 0014 Stack: +0 0000 +2 19F5 +4 0000
0041	MOV CX,ES:[BX -01]	26 8B 4F FF	IP = 0041 CX = 1C18	IP = 0045 CX = FFCE
0045	XCHG AX,CX	91	IP = 0045 CX = FFCE AX = 0000	IP = 0046 CX = 0000 AX = FFCE
0046	MOV DI,0002	BF 02 00	IP = 0046 DI = 0002	IP = 0049 DI = 0002
0049	MOV ES:[BX+DI],AX	26 89 01	IP = 0049 DS:0005 = 00 DS:0006 = 15	IP = 004C DS:0005 = CE DS:0006 = FF

004C	MOV BP,SP	8B EC	IP = 004C BP = 0000	IP = 004E BP = 0014
004E	PUSH [0000]	FF 36 00 00	IP = 004E SP = 0014 Stack: +0 0000 +2 19F5 +4 0000	IP = 0052 SP = 0012 Stack: +0 01F4 +2 0000 +4 19F5
0052	PUSH [0002]	FF 36 02 00	IP = 0052 SP = 0012 Stack: +0 01F4 +2 0000 +4 19F5 +6 0000	IP = 0056 SP = 0010 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5
0056	MOV BP,SP	8B EC	IP = 0056 BP = 0014	IP = 0058 BP = 0010
0058	MOV DX,[BP+02]	8B 56 02	IP = 0058 DX = 0000	IP = 005B DX = 01F4
005B	POP AX	58	IP = 005B AX = FFCE SP = 0010 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5	IP = 005C AX = FFCE SP = 0012 Stack: +0 01F4 +2 0000 +4 19F5 +6 0000

005C	POP AX	58	IP = 005C AX = FFCE SP = 0012 Stack: +0 01F4 +2 0000 +4 19F5	IP = 005D AX = 01F4 SP = 0014 Stack: +0 0000 +2 19F5 +4 0000
005D	RET Far	CB	IP = 005D CS = 1A0A SP = 0014 Stack: +0 0000 +2 19F5	IP = 0000 CS = 19F5 SP = 0018 Stack: +0 0000 +2 0000
0000	INT 20	CD20		

Выводы.

В ходе лабораторной работы были изучены режимы адресации на языке Ассемблера.

ПРИЛОЖЕНИЕ А
ЛИСТИНГИ ПРОГРАММ

Название файла LR2_COMP.LST

```
Microsoft (R) Macro Assembler Version 5.10
10/8/22 23:29:47
                                                                Page
1-1


                        ; Программа изучения ♦
                        ♦ежимов адресации про
                        цессора IntelX86
 = 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 описани
                        я данных
 0000  0000                       mem1 DW 0
 0002  0000                       mem2 DW 0
 0004  0000                       mem3 DW 0
 0006  15 16 17 18 1C 1B          vec1 DB 21,22,23,24,28,27,26,25
       1A 19
 000E  28 32 D8 CE 14 1E          vec2 DB 40,50,-40,-50,20,30,-
20,-30
       EC E2
```

```
           23 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
           24       FA FB 01 02 FC FD
           25       03 04 FE FF
           26 0026                      DATA ENDS
           27                         ; Код программы
           28 0000                      CODE SEGMENT
           29                               ASSUME CS:CODE, DS:DATA, SS:AS
                                        tack
           30                           ; Головная процедур
                                        a
           31 0000                      Main     PROC   FAR
           32 0000  1E                           push   DS
           33 0001  2B C0                        sub    AX,AX
           34 0003  50                           push   AX
           35 0004  B8 ---- R                    mov    AX,DATA
           36 0007  8E D8                        mov    DS,AX
           37                           ; ПРОВЕРКА РЕЖИМОВ ?
                                        ?ДРЕСАЦИИ НА УРОВНЕ С?
                                        ?ЕЩЕНИЙ
           38                           ; Регистровая адрес
                                        ация
           39 0009  B8 01F4                          mov    ax,n1
           40 000C  8B C8                        mov    cx,ax
           41 000E  B3 24                        mov    bl,EOL
           42 0010  B7 CE                        mov    bh,n2
           43                           ; Прямая адресация
           44 0012  C7 06 0002 R FFCE            mov    mem2,n2
           45 0018  BB 0006 R                    mov    bx,OFFSET vec1
     Microsoft (R) Macro Assembler Version 5.10
10/8/22 23:29:47
                                                                    Page
1-2


           46 001B  A3 0000 R                    mov    mem1,ax
           47                           ; Косвенная адресац
```

```
                        ия
 48 001E  8A 07                      mov    al,[bx]
 49                                ;mov mem3,[bx]
 50                          ; Базированная адре
                        сация
 51 0020  8A 47 03                        mov    al,[bx]+3
 52 0023  8B 4F 03                        mov    cx,3[bx]
 53                          ; Индексная адресац
                        ия
 54 0026  BF 0002                         mov    di,ind
 55 0029  8A 05                      mov    al,[di]
 56                                ;mov    cx,vec2[di]
 57                          ; Адресация с базир?
                        ?ванием и индексирова
                        нием
 58 002B  BB 0003                         mov    bx,3
 59 002E  8A 81 0016 R                    mov
al,matr[bx][di]
 60                                ;mov    cx,matr[bx][di]
 61                                ;mov     ax,matr[bx*4][di]
 62                          ; ПРОВЕРКА РЕЖИМОВ ?
                        ?ДРЕСАЦИИ С УЧЕТОМ СЕ?
                        ?МЕНТОВ
 63                          ; Переопределение с
                        егмента
 64                          ; ------ вариант 1
 65 0032  B8 ---- R                 mov    ax, SEG vec2
 66 0035  8E C0                     mov    es, ax
 67 0037  26: 8B 07                 mov    ax, es:[bx]
 68 003A  B8 0000                        mov    ax, 0
 69                          ; ------ вариант 2
 70 003D  8E C0                     mov    es, ax
 71 003F  1E                        push   ds
 72 0040  07                        pop    es
 73 0041  26: 8B 4F FF                   mov    cx, es:[bx-1]
 74 0045  91                        xchg   cx,ax
 75                          ; ------ вариант 3
 76 0046  BF 0002                         mov    di,ind
```

```
          77 0049  26: 89 01                mov    es:[bx+di],ax
          78                      ; ------ вариант 4
          79 004C  8B EC                    mov    bp,sp
          80                           ;mov    ax,matr[bp+bx]
          81                           ;mov    ax,matr[bp+di+si]
          82                      ; Использование сег
                                  мента стека
          83 004E  FF 36 0000 R                  push   mem1
          84 0052  FF 36 0002 R                  push   mem2
          85 0056  8B EC                    mov    bp,sp
          86 0058  8B 56 02                      mov    dx,[bp]+2
          87
          88 005B  58                       pop ax
          89 005C  58                       pop ax
          90 005D  CB                       ret
     Microsoft (R) Macro Assembler Version 5.10
10/8/22 23:29:47
                                                                Page
1-3


          91 005E                   Main     ENDP
          92 005E                   CODE     ENDS
          93                           END Main
     Microsoft (R) Macro Assembler Version 5.10
10/8/22 23:29:47

Symbols-1


     Segments and Groups:

                N a m e         Length     Align     Combine Class

     ASTACK . . . . . .      0018  PARA  STACK
     CODE . . . . . .      005E  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 =
005E
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


     86 Source  Lines
     86 Total   Lines
     19 Symbols

  47262 + 457951 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```

ПРИЛОЖЕНИЕ В
КОД 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 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,[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

        pop ax
        pop ax
        ret
Main    ENDP
CODE    ENDS
        END Main
```