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

Студент гр. 0382 _____ Тихонов С.В.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2021

Цель работы.

Изучить режимы адресации и формирование 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.

Ход работы

1. Описание обнаруженных при первоначальной трансляции ошибок и их объяснение :
    - ***mov mem3, [bx]*** - Нельзя читать из памяти и писать в память одной командой .
    - ***mov cx, vec2[di]*** - Несоответствие типов операндов.
    - ***mov cx, matr[bx][di]*** - Несоответствие типов операндов.
    - ***mov ax, matr[bx*4][di]*** - Нельзя умножать 16-битные регистры, нужно использовать регистры EXX.
    - ***mov ax, matr[bp+bx]*** - Нельзя использовать более одного базового регистра.
    - ***mov ax, matr[bp+di+si]*** - Нельзя использовать более одного индексного регистра.

2. Запуск программы под управлением отладчика с пошаговым выполнением и занесением данных в таблицу 1:

Начальное значение сегментных регистров:

CS = 1A0A ; DS = 19F5 ;

ES = 19F5 ; SS = 1A05 ;

Таблица 1 - Отладка LR2_COMP.EXE

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения	После выполнения
0000	PUSH DS	1E	DS = 19F5 IP = 0000 SP = 0018 Stack +0 0000 +2 0000 +4 0000 +6 0000	DS = 19F5 IP = 0001 SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0001	SUB AX, AX	2BC0	AX = 0000 IP = 0001	AX = 0000 IP = 0003
0003	PUSH AX	50	AX = 0000 IP = 0003 SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000	AX = 0000 IP = 0004 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0004	MOV AX, 1A07	B8071A	IP = 0004 AX = 0000	IP = 0007 AX = 1A07
0007	MOV DS, AX	8ED8	IP = 0007 DS = 19F5	IP = 0009 DS = 1A07
0009	MOV AX, 01F4	B8F401	AX = 1A07	AX = 01F4

			IP = 0009	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	C7060200CEFF	IP = 0012 DS:0002 = 00 DS:0003 = 00	IP = 0018 DS:0002 = CE DS:0003 = FF
0018	MOV BX, 0006	BB0600	BX = CE24 IP = 0018	BX = 0006 IP = 001B
001B	MOV [0000], AX	A30000	IP = 001B AX = 01F4 DS:0000 = 00 DS:0001 = 00	IP = 001E AX = 01F4 DS:0000 = F4 DS:0001 = 01
001E	MOV AL, [BX]	8A07	AX = 01F4 BX = 0006 IP = 001E	AX = 0101 BX = 0006 IP = 0020
0020	MOV AL, [BX+03]	8A4703	AX = 0101 BX = 0006 IP = 0020	AX = 0104 BX = 0006 IP = 0023
0023	MOV CX, [BX+03]	8B4F03	CX = 01F4 BX = 0006 IP = 0023	CX = 0804 BX = 0006 IP = 0026
0026	MOV DI, 0002	BF0200	DI = 0000 IP = 0026	DI = 0002 IP = 0029
0029	MOV AL,[000E+DI]	8A850E00	AX = 0104 DI = 0002	AX = 010A DI = 0002

			IP = 0029 DS:0010 = 0A	IP = 002D DS:0010 = 0A
002D	MOV BX, 0003	BB0300	BX = 0006 IP = 0002D	BX = 0003 IP = 00030
0030	MOV AL,[0016+BX+DI]	8A811600	AX = 010A DX = 0003 DI = 0002 IP = 0030 DS:001B = FD	AX = 01FD DX = 0003 DI = 0002 IP = 0034 DS:001B = FD
0034	MOV AX, 1A07	B8071A	IP = 0034 AX = 01FD	IP = 0037 AX = 1A07
0037	MOV ES,AX	8EC0	ES = 19F5 AX = 1A07 IP = 0037	ES = 1A07 AX = 1A07 IP = 0039
0039	MOV AX, ES:[BX]	268B07	AX = 1A07 ES = 1A07 BX = 0003 IP = 0039 DS:0003 = FF DS:0004 = 00	AX = 00FF ES = 1A07 BX = 0003 IP = 003C DS:0003 = FF DS:0004 = 00
003C	MOV AX, 0000	B80000	IP = 003C AX = 00FF	IP = 003F AX = 0000
003F	MOV ES, AX	8EC0	IP = 003F ES = 1A07 AX = 0000	IP = 0041 ES = 0000 AX = 0000
0041	PUSH DS	1E	DS = 1A07 IP = 0041 SP = 0014 Stack +0 0000 +2 19F5 +4 0000	DS = 1A07 IP = 0042 SP = 0012 Stack +0 1A07 +2 0000 +4 19F5

			+6 0000	+6 0000
0042	POP ES	07	ES = 0000 IP = 0042 SP = 0012 Stack +0 1A07 +2 0000 +4 19F5	ES = 1A07 IP = 0043 SP = 0014 Stack +0 0000 +2 19F5 +4 0000
0043	MOV CX, ES:[BX-01]	268B4FFF	CX = 0804 IP = 0043 BX = 0003 ES =1A07 DS:0002 = CE DS:0003 = FF	CX = FFCE IP = 0047 BX = 0003 ES =1A07 DS:0002 = CE DS:0003 = FF
0047	XCHG AX,CX	91	AX = 0000 CX = FFCE IP = 0047	AX = FFCE CX = 0000 IP = 0048
0048	MOV DI, 0002	BF0200	DI = 0002 PI = 0048	DI = 0002 PI = 004B
004B	MOV ES:[BX+DI], AX	268901	AX= FFCE IP = 004B BX = 0003 ES =1A07 DI = 0002 DS:0005 = 00 DS:0006 = 01	AX= FFCE IP = 004E BX = 0003 ES =1A07 DI = 0002 DS:0005 = CE DS:0006 = FF
004E	MOV BP, SP	8BEC	SP = 0014 BP = 0000 PI = 004E	SP = 0014 BP = 0014 PI = 0050
0050	PUSH [0000]	FF360000	IP = 0050 SP = 0014 DS:0000 = F4	IP = 0054 SP = 0012 DS:0000 = F4

			DS:0001 = 01 Stack +0 0000 +2 19F5 +4 0000 +6 0000	DS:0001 = 01 Stack +0 01F4 +2 0000 +4 19F5 +6 0000
0054	PUSH [0002]	FF360000	IP = 0054 SP = 0012 DS:0002 = CE DS:0003 = FF Stack +0 01F4 +2 0000 +4 19F5 +6 0000	IP = 0058 SP = 0010 DS:0002 = CE DS:0003 = FF Stack +0 FFCE +2 01F4 +4 0000 +6 19F5
0058	MOV BP, SP	8BEC	BP = 0014 SP = 0010 IP = 0058	BP = 0010 SP = 0010 IP = 005A
005A	MOV DX, [BP+02]	8B5602	BP = 0010 DX = 0000 IP = 005A SS:0012 = 01F4	BP = 0010 DX = 01F4 IP = 005D SS:0012 = 01F4
005D	RET Far 0002	CA0200	IP = 005D SP = 0010 CS = 1A0A Stack +0 FFCE +2 01F4 +4 0000 +6 19F5	IP = FFCE SP = 0016 CS = 01F4 Stack +0 19F5 +2 0000 +4 0000 +6 0000

Вывод.

Были изучены режимы адресации и то как формируется исполнительный адрес. В ходе работы был исправлен и пошагово отлажен исходный файл.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb2.asm

```
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 28,27,26,25,21,22,23,24
vec2 DB 20,30,-20,-30,40,50,-40,-50
matr DB -8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-2,-1,5,6
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
      mov ax,n1
      mov cx,ax
      mov bl,EOL
      mov bh,n2
      mov mem2,n2
      mov bx,OFFSET vec1
      mov mem1,ax
      mov al,[bx]
      ;     mov mem3,[bx]
      mov al,[bx]+3
      mov cx,3[bx]
      mov di,ind
      mov al,vec2[di]
      ;     mov cx,vec2[di]
      mov bx,3
      mov al,matr[bx][di]
      ;     mov cx,matr[bx][di]
      ;     mov ax,matr[bx*4][di]
      mov ax, SEG vec2
      mov es, ax
      mov ax, es:[bx]
      mov ax, 0
      mov es, ax
      push ds
      pop es
      mov cx, es:[bx-1]
```

```
      xchg cx,ax
      mov di,ind
      mov es:[bx+di],ax
      mov bp,sp
      ;     mov ax,matr[bp+bx]
      ;     mov ax,matr[bp+di+si]
      push mem1
      push mem2
      mov bp,sp
      mov dx,[bp]+2
      ret 2
Main ENDP
CODE ENDS
      END Main
```

Название файла: lst1.LST

```
#Microsoft (R) Macro Assembler Version 5.10                 11/11/21
01:03:2
                                                            Page
1-1


 = 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  1C 1B 1A 19 15 16    vec1 DB 28,27,26,25,21,22,23,24
       17 18
 000E  14 1E EC E2 28 32    vec2 DB 20,30,-20,-30,40,50,-40,-50
       D8 CE
 0016  F8 F9 03 04 FA FB    matr DB -8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-2,-
1,5,6
```

```
       01 02 FC FD 07 08
       FE FF 05 06
 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
 0009  B8 01F4                    mov ax,n1
 000C  8B C8                      mov cx,ax
 000E  B3 24                      mov bl,EOL
 0010  B7 CE                      mov bh,n2
 0012  C7 06 0002 R FFCE          mov mem2,n2
 0018  BB 0006 R            mov bx,OFFSET vec1
 001B  A3 0000 R            mov mem1,ax
 001E  8A 07                      mov al,[bx]
                            mov mem3,[bx]
HELLO1.ASM(32): error A2052: Improper operand type
 0020  8A 47 03                   mov al,[bx]+3
 0023  8B 4F 03                   mov cx,3[bx]
 0026  BF 0002                    mov di,ind
 0029  8A 85 000E R               mov al,vec2[di]
 002D  8B 8D 000E R               mov cx,vec2[di]
HELLO1.ASM(37): warning A4031: Operand types must match
 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]
HELLO1.ASM(40): warning A4031: Operand types must match
 003C  8B 85 0022 R               mov ax,matr[bx*4][di]
HELLO1.ASM(41): error A2055: Illegal register value
 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
```

```
 004B  8E C0                    mov es, ax
 004D  1E                   push ds
#Microsoft (R) Macro Assembler Version 5.10                  11/11/21
01:03:2
                                                             Page
1-2


 004E  07                   pop es
 004F  26: 8B 4F FF              mov cx, es:[bx-1]
 0053  91                   xchg cx,ax
 0054  BF 0002                   mov di,ind
 0057  26: 89 01            mov es:[bx+di],ax
 005A  8B EC                     mov bp,sp
 005C  3E: 8B 86 0016 R               mov ax,matr[bp+bx]
HELLO1.ASM(54): error A2046: Multiple base registers
 0061  3E: 8B 83 0016 R               mov ax,matr[bp+di+si]
HELLO1.ASM(55): error A2047: Multiple index registers
 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
HELLO1.ASM(61): error A2006: Phase error between passes
 0076                 CODE ENDS
                          END Main
#Microsoft (R) Macro Assembler Version 5.10                  11/11/21
01:03:2
                                                             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  HELLO1
@VERSION . . . . . .      TEXT  510


     63 Source  Lines
     63 Total   Lines
     19 Symbols

  47812 + 461495 Bytes symbol space free

      2 Warning Errors
      5 Severe  Errors
```

Название файла: lst2.LST

```
#Microsoft (R) Macro Assembler Version 5.10                 11/11/21
01:06:5
                                                            Page
1-1


 = 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  1C 1B 1A 19 15 16    vec1 DB 28,27,26,25,21,22,23,24
       17 18
 000E  14 1E EC E2 28 32    vec2 DB 20,30,-20,-30,40,50,-40,-50
       D8 CE
 0016  F8 F9 03 04 FA FB    matr DB -8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-2,-
1,5,6
       01 02 FC FD 07 08
       FE FF 05 06
 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
 0009  B8 01F4                  mov ax,n1
 000C  8B C8                    mov cx,ax
 000E  B3 24                    mov bl,EOL
 0010  B7 CE                    mov bh,n2
 0012  C7 06 0002 R FFCE        mov mem2,n2
 0018  BB 0006 R           mov bx,OFFSET vec1
 001B  A3 0000 R           mov mem1,ax
 001E  8A 07                    mov al,[bx]
                           ;    mov mem3,[bx]
 0020  8A 47 03                 mov al,[bx]+3
 0023  8B 4F 03                 mov cx,3[bx]
 0026  BF 0002                  mov di,ind
 0029  8A 85 000E R             mov al,vec2[di]
                           ;    mov cx,vec2[di]
 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]
 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
 003F  8E C0                    mov es, ax
 0041  1E                  push ds
#Microsoft (R) Macro Assembler Version 5.10                 11/11/21
01:06:5
                                                           Page
1-2


 0042  07                  pop es
 0043  26: 8B 4F FF             mov cx, es:[bx-1]
 0047  91                  xchg cx,ax
 0048  BF 0002                  mov di,ind
 004B  26: 89 01           mov es:[bx+di],ax
 004E  8B EC                    mov bp,sp
```

```
                        ;     mov ax,matr[bp+bx]
                        ;     mov ax,matr[bp+di+si]
 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                  11/11/21
01:06:5
                                                             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  HELLO1
@VERSION . . . . .  TEXT  510


     63 Source  Lines
     63 Total   Lines
     19 Symbols

  47826 + 461481 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```