

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «ОЭВМиС»
Тема: Изучение режимов адресации и формирования исполнительного
адреса

Студент гр. 9383

Крейсманн К.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить режимы адресации в языке Ассемблер.

Задание:

Часть 1.

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.

Текст исходного файла `lr2.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 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

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

; 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] нельзя умножать регистр и ax 2 байта

;-v1
    mov ax,SEG vec2
    mov es,ax
    mov ax,es:[bx]
    mov ax,0

;-v2
    mov es,ax
    push ds
    pop es
    mov cx,es:[bx-1]
    xchg cx,ax

;-v3
    mov di,ind
    mov es:[bx+di],ax

;-v4
    mov bp,sp

```

```

; mov ax,matr[bp+bx]
; mov ax,matr[bp+di+si] сложение индексных регистров

```

```

push mem1
push mem2
mov bp,sp
mov dx,[bp]+2
ret 2

```

MAIN ENDP

CODE ENDS

END MAIN

Файл диагностических сообщений lr2.lst:

Microsoft (R) Macro Assembler Version 5.10

10/6/20 38:58:37

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 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
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
    FA FB 01 02 FC FD
    03 04 FE FF

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
                                ; 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]*

Microsoft (R) Macro Assembler Version 5.10

10/6/20 38:58:37

Page 1-2

; *mov cx,vec2[di]*

; Адресация с базирование ?

? 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]* нельзя у

множить регистр и *ax* 2 байт ?

?

; -v1

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*

; -v2

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*

; -v3

0048 BF 0002 *mov di,ind*

004B 26: 89 01 *mov es:[bx+di],ax*

; -v4

004E 8B EC *mov bp,sp*

; *mov ax,matr[bp+bx]*

; mov ax,matr[bp+di+si] сложение ?

Индексных регистров

```
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 10/6/20 38:58:37
Symbols-1

Segments and Groups:

<i>N a m e</i>	<i>Length</i>	<i>Align</i>	<i>Combine</i>	<i>Class</i>
ASTACK.....	0018	PARA	STACK	
CODE.....	0060	PARA	NONE	
DATA.....	0026	PARA	NONE	

Symbols:

<i>N a m e</i>	<i>Type</i>	<i>Value</i>	<i>Attr</i>
EOL	NUMBER	0024	
IND	NUMBER	0002	
MAIN.....	F PROC	0000	CODE Length = 0060
MATR.....	L BYTE	0016	DATA
MEM1.....	L WORD	0000	DATA

<i>MEM2</i>	<i>L WORD</i>	<i>0002</i>	<i>DATA</i>
<i>MEM3</i>	<i>L WORD</i>	<i>0004</i>	<i>DATA</i>
<i>N1</i>	<i>NUMBER</i>	<i>01F4</i>	
<i>N2</i>	<i>NUMBER</i>	<i>-0032</i>	
<i>VEC1</i>	<i>L BYTE</i>	<i>0006</i>	<i>DATA</i>
<i>VEC2</i>	<i>L BYTE</i>	<i>000E</i>	<i>DATA</i>
<i>@CPU</i>	<i>TEXT</i>	<i>0101h</i>	
<i>@FILENAME</i>	<i>TEXT</i>	<i>lr2</i>	
<i>@VERSION</i>	<i>TEXT</i>	<i>510</i>	

85 Source Lines

85 Total Lines

19 Symbols

47842 + 461465 Bytes symbol space free

0 Warning Errors

0 Severe

Объяснение ошибок:

1) Строка 41: *mov mem3,[bx]*

Ошибка: error A2052: Improper operand type

Объяснение: Оба операнда не могут одновременно быть памятью

2) Строка 48 : *mov cx,vec2[di]*

Ошибка: warning A4031: Operand types must match

Объяснение: размер элементов *vec2* – 1 байт , а *cx* – 2 байта

3) Строка 52 : *mov cx,matr[bx][di]*

Ошибка: warning A4031: Operand types must match

Объяснение: размер элементов *matr* - 1 байт, а *cx* – два байта.

4)Строка 53: *mov ax,matr[bx*4][di]*

Ошибка: error A2055: Illegal register value

Объяснение: нельзя умножать регистр

5) Строка 74: *mov ax,matr[bp+bx]*

Ошибка: error A2046: Multiple base registers

Объяснение: нельзя использовать два базовых регистра

6) Строка 75 : *mov ax,matr[bp+di+si]*

Ошибка: error A2047: Multiple index registers

Объяснение: нельзя использовать два индексных регистра

Протокол пошагового исполнения программы :

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти до	Содержимое регистров и ячеек памяти после
0000	Push DS	1E	(SP)=0018 Stack:+0 0000 (IP)=0000 +2 0000 +4 0000 +6 0000	(SP)=0016 Stack:+0 19F5 (IP)=0001 +2 0000 +4 0000 +6 0000
0001	Sub AX,AX	2BC0	(AX) = 0000 (IP) = 0001	(AX) = 0000 (IP) = 0003
0003	PUSH AX	50	(SP)=0016 Stack:+0 19F5 (IP)=0003 +2 0000 +4 0000 +6 0000	(SP)=0014 Stack:+0 0000 (IP)=0004 +2 19F5 +4 0000 +6 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	8BCB	(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	(BX) = 0024 (IP) = 0010	(BX) = CE24 (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) = 001E
001E	MOV AL,[BX]	8A07	(AX) = 01F4 (IP) = 001E	(AX) = 0115 (IP) = 0020
0020	MOV AL,[BX+03]	8A4703	(AX) = 0115	(AX) = 0118

			(IP) = 0020	(IP) = 0023
0023	MOV CX,[BX+03]	8B4F03	(CX) = 01F4 (IP) = 0023	(CX) = 1C18 (IP) = 0026
0026	MOV DI,0002	BF0200	(DI) = 0002 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	MOV AL,[000E+DI]	BA850E 00	(AX) = 0118 (IP) = 0029	(AX) = 01D8 (IP) = 002D
002D	MOV BX,0003	BB0300	(BX) = 0006 (IP) = 002D	(BX) = 0003 (IP) = 0030
0030	MOV AL,[0016+BX+DI]	8A81160 0	(AX) = 01D8 (IP) = 0030	(AX) = 0108 (IP) = 0034
0034	MOV AX,1A07	B8071A	(AX) = 0108 (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	(EF) = 0000 (IP) = 0041
0041	PUSH DS	1E	(SP)=0014 Stack:+0 0000 (IP)=0041 +2 19F5 +4 0000 +6 0000	(SP)=0012 Stack:+0 1A07 (IP)=0042 +2 0000 +4 19F5 +6 0000
0042	POP ES	07	(SP)=0012 Stack:+0 1A07 (IP)=0042 +2 0000 (ES) = 0000 +4 19F5 +6 0000	(SP)=0014 Stack:+0 0000 (IP)=0043 +2 19F5 (ES) = 1A07 +4 0000 +6 0000
0043	MOV CX,ES:[BX-01]	268B4FF F	(CX) = 1C1B (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	(IP) = 004B	(IP) = 004E
004E	MOV BP,SP	8BEC	(BP) = 0000 (IP) = 004E	(BP) = 0014 (IP) = 0050
0050	PUSH [0000]	FF36000 0	(SP)=0014 Stack:+0 0000 (IP)=0050 +2 19F5 +4 0000 +6 0000	(SP)=0012 Stack:+0 01F4 (IP)=0054 +2 0000 +4 19F5 +6 0000
0054	PUSH [0002]	FF36020 0	(SP)=0012 Stack:+0 01F4 (IP)=0054 +2 0000 +4 19F5 +6 0000	(SP)=0010 Stack:+0 FFCE (IP)=0058 +2 01F4 +4 0000 +6 19F5
0059	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	(SP)=0010 Stack: +0 FFCE (IP)=005D +2 01F4 +4 0000 +6 19F5	(SP)=0016 Stack: +0 19F5 (IP)=FFCE +2 0000 +4 0000 +6 0000

Вывод:

Изучены режимы адресации в языке ассемблер.