

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 системы»
Тема: Изучение режимов адресации и форматирования
исполнительного адреса.
Вариант 5

Студента гр. 1383

Панов М.Ю.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучение режимов адресации и форматирования исполнительного адреса на языке ассемблер.

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.

При трансляции программы возникли следующие ошибки:

- 1) `mov mem3, [dx]` - (error A2052: improper operand type) пересылка память-память запрещена.
- 2) `mov cx,vec2[di]` - (warning A4031: Operand types must match) операнды имеют разную длину (`cx` - 2 байта, `vec2[di]` - 1 байт).
- 3) `mov cx,matr[bx][di]` - (warning A4031: Operand types must match) операнды имеют разную длину (`cx` - 2 байта, `matr[bx][di]` - 1 байт).
- 4) `mov ax,matr[bx*4][di]` - (error A2055: Illegal register value) нельзя масштабировать 16 битные регистры.

5) `mov ax,matr[bp+bx]` - (error A2046: Multiple base registers) Нельзя

использовать более одного базового регистра для адресации.

6) `mov ax,matr[bp+di+si]` - (error A2047: Multiple index registers) Нельзя

использовать более одного базового регистра для адресации.

Начальное содержимое сегментных регистров:

(CS) = 1A0A

(DS) = 19F5

(SS) = 1A05

(ES) = 19F5

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 +2 0000 +4 0000 +6 0000	(SP)=0016 (IP)=0001 Stack: +0 19F5 +2 0000 +4 0000 +6 0000
0001	SUB AX, AX	2B C0	(IP)=0001 AX=0000	(IP)=0003 AX=0000
0003	PUSH AX	50	(SP)=0016 (IP)=0003 Stack: +0 19F5 +2 0000 +4 0000 +6 0000	(SP)=0014 (IP)=0004 Stack: +0 0000 +2 19F5 +4 0000 +6 0000
0004	<code>mov AX, 1A07</code>	B8 07 1A	(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	B8 F4 01	(IP)=0009 (AX) = 1A07	(IP)=000C (AX) = 01F4
000C	mov CX, AX	8B C8	(CX)=00B0 (IP)=000C	(CX)=01F4 (IP)=000E
000E	mov BL,24	B3 24	(IP) = 000E (BX) = 0000	(IP) = 0010 (BX) = 0024
0010	mov BH, CE	B7CE	(IP) = 0010 (BX) = 0024	(IP) = 0012 (BX) = CE24
0012	mov [0002], FFCE	C7060200EFF	(IP) = 0012 DS: 0000 00 00 00 00 00 00	(IP) = 0018 (SP) = 0014 0000 00 00 CE FF 00 00
0018	mov BX,0006	BB 06 00	(IP) = 0018 (BX) = CE24	(IP) =001B (DX) = 0006
001B	mov [0000], AX	A30000	(IP) = 001B DS: 0000 00 00 CE FF 00 00	(IP) = 001E DS: 0000 F4 01 CE FF 00 00
001E	mov AL,[BX]	8A 07	(IP) = 001E (AX)= 01F4	(IP)= 0020 (AX)=010B
0020	mov AL,[BX+03]	8A 47 03	(IP) = 0020 (AX) = 010B	(IP) = 0023 (AX)=010E
0023	mov CX,[BX +03]	8B4F03	(CX) = 01F4 (IP) = 0023	(CX) = 120E (IP) = 0026
0026	mov DI, 0002	BF0200	(IP) = 0026 (DI) = 0000	(IP) = 0029 (DI) = 0002

0029	mov AL, [000E+DI]	8A850E00	(IP) = 0029 (AX)= 010E	(IP) = 002D (AX) = 01F6
002D	mov BX, 0003	BB 03 00	(IP)= 002D (BX) = 0006	(IP) = 0030 BX= 0003
0030	mov AL, [0016 + BX + DI]	8A811600	(IP) = 0030 (AX) = 01F6	(IP) = 0034 (AX) = 0104
0034	mov AX, 1A07	BB 07 1A	(IP) = 0034 (AX) = 0104	(IP) = 0037 (AX) = 1A07
0037	mov ES,AX	8E C0	(IP) = 0037 (ES) = 19F5	(IP) = 0039 (ES) = 1A07
0039	mov AX, ES:[BX]	26 8B 07	(IP) = 0039 (AX)= 1A07	(IP)= 003C (AX)= 00FF
003C	mov AX, 0000	B8 0000	(IP) = 003C (AX) = 00FF	(IP) = 003F (AX) = 0000
003F	mov ES, AX	8EC0	(IP) = 003F (ES) = 1A07	(IP) = 0041 (ES) = 0000
0041	PUSH DS	1E	(IP) = 0041 (SP) = 0014 Stack: +0 0000 +2 19F5 +4 0000 +6 0000	(IP) = 0042 (SP) = 0012 Stack: +0 1A07 +2 0000 +4 19F5 +6 0000
0042	POP ES	07	(IP) = 0042 (SP) = 0012 (ES) = 0000 Stack: +0 1A07 +2 0000 +4 19F5 +6 0000	(IP) = 0043 (SP) = 0014 (ES) = 1A07 Stack: +0 0000 +2 19F5 +4 0000 +6 0000
0043	mov CX, ES:[BX - 01]	268B4FFF	(IP) = 0043 (CX) = 120E	(IP) = 0047 (CX) = FFCE
0047	XCHG AX,CX	91	(IP) = 0047 (AX) = 0000 (CX) = FFCE	(IP)= 0048 (AX) = FFCE (CX) = 0000

0048	mov DI, 0002	DF0200	(IP) = 0048 (DI) = 0002	(IP) = 004B (DI) = 0002
004B	mov ES:[BX+DI], AX	268901	(IP) = 004B DS: 0000 F4 01 CE FF 00 00 0B	(IP) = 004E DS: 0000 F4 01 CE FF 00 CE FF
004E	mov BP,SP	8BEC	(IP) = 004E (BP) = 0000	(IP) = 0050 (BP) = 0014
0050	PUSH [0000]	FF360000	(SP) = 0014 (IP) = 0050 Stack: +0 0000 +2 19F5 +4 0000 +6 0000	(SP) = 0012 (IP) = 0054 Stack: +0 01F4 +2 0000 +4 19F5 +6 0000
0054	PUSH [0002]	FF360200	(SP) = 0012 (IP) = 0054 Stack: +0 01F4 +2 0000 +4 19F5 +6 0000	(SP) = 0010 (IP) = 0058 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5
0058	mov BP, SP	8BEC	(IP) = 0058 (BP) = 0014	(IP) = 005A (BP) = 0010
005A	mov DX, [BP+02]	8B5602	(IP) = 005A (DX) = 0000	(IP) = 005D (DX) = 01F4
005D	RET FAR 0002	CA0200	(IP) = 005D (CS) = 1A0A (SP) = 0010 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5	(IP) = FFCE (CS) = 01F4 (SP) = 0016 Stack: +0 19F5 +2 0000 +4 0000 +6 0000

Выводы.

В ходе работы были изучены режимы адресации на языке ассемблер.

Исходный код

```
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 11,12,13,14,18,17,16,15  
vec2 DB 10,20,-10,-20,30,40,-30,-40  
matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7,8,-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] ;error A2052: improper operand type

; Базированная адресация

mov al,[bx]+3

mov cx,3[bx]

; Индексная адресация

mov di,ind

mov al,vec2[di]

;mov cx,vec2[di]; warning A4031: Operand types must match

; Адресация с базированием и индексированием

mov bx,3

mov al,matr[bx][di]

;mov cx,matr[bx][di]; warning A4031: Operand types must match

; mov ax,matr[bx*4][di] ;error A2055: Illegal register value

```

; 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] ;error A2046: Multiple base registers
; mov ax,matr[bp+di+si] ;error A2047: Multiple index registers
; Использование сегмента стека
push mem1
push mem2
mov bp,sp
mov dx,[bp]+2
ret 2
Main ENDP
CODE ENDS
END Main

```

Листинг:

```
= 0024          EOL EQU '$'
= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50

; D;Ñ,DµD° D;Ñ€D³⁄₄D³Ñ€D°D¹⁄₄D¹⁄₄Ñ<

0000          AStack SEGMENT STACK
0000 000C[      DW 12 DUP(?)
               ???
               ]

0018          AStack ENDS

; D”D°D¹⁄₂D¹⁄₂Ñ<Dµ D;Ñ€D³⁄₄D³Ñ€D°D¹⁄₄D¹⁄₄Ñ<

0000          DATA SEGMENT
; D”D,Ñ€DµD°Ñ,D,D²Ñ< D³⁄₄D;D,ÑD°D¹⁄₂D,Ñ D’D°D¹⁄₂D¹⁄₂Ñ
<Ñ...

0000 0000          mem1 DW 0
0002 0000          mem2 DW 0
0004 0000          mem3 DW 0
0006 0B 0C 0D 0E 12 11   vec1 DB 11,12,13,14,18,17,16,15
               10 0F
000E 0A 14 F6 EC 1E 28   vec2 DB 10,20,-10,-20,30,40,-30,-40
               E2 D8
0016 01 02 FC FD 03 04   matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7,8,-6,-5
               FE FF 05 06 F8 F9
               07 08 FA FB

0026          DATA ENDS

; DšD³⁄₄D’ D;Ñ€D³⁄₄D³Ñ€D°D¹⁄₄D¹⁄₄Ñ<
```

```

0000                                CODE SEGMENT

                                ASSUME CS:CODE, DS:DATA, SS:AStack
                                ; Ð“Ð³¼Ð»Ð³¼Ð²Ð¹Ð°Ñ Ð¿Ñ€Ð³¼Ñ†ÐµÐŒÑƒÐŒƒÑ€Ð°
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
                                ; ÐŸÐ ÐŹÐ’Ð•Ð ÐšÐ Ð Ð•Ð–ÐŒÐžÐ’ ÐÐ”Ð Ð•Ð¿ÐÐ
                                |ÐŒÐŒ ÐÐ Ð£Ð ÐŹÐ’ÐÐ• Ð¿Ð€Ð•Ð©Ð•ÐÐŒÐ™
                                ; Ð ÐµÐ³Ð,ÑÑ,Ñ€Ð³¼Ð²Ð°Ñ Ð°ÐŒÑ€ÐµÑÐ°Ñ†Ð,Ñ
0009 B8 01F4                          mov ax,n1
000C 8B C8                            mov cx,ax
000E B3 24                            mov bl,EOL
0010 B7 CE                            mov bh,n2
                                ; ÐŸÑ€ÑÐ¹¼Ð°Ñ Ð°ÐŒÑ€ÐµÑÐ°Ñ†Ð,Ñ
0012 C7 06 0002 R FFCE    mov mem2,n2
0018 BB 0006 R                        mov bx,OFFSET vec1
001B A3 0000 R                        mov mem1,ax
                                ; ÐšÐ³¼ÑÐ²ÐµÐ¹¼Ð¹Ð°Ñ Ð°ÐŒÑ€ÐµÑÐ°Ñ†Ð,Ñ
001E 8A 07                            mov al,[bx]
                                ;mov mem3,[bx] ;error A2052: improper operand t
                                ype
                                ; Ð°Ð°Ð•Ð,Ñ€Ð³¼Ð²Ð°Ð¹¼Ð¹Ð°Ñ Ð°ÐŒÑ€ÐµÑÐ°Ñ†Ð,Ñ
0020 8A 47 03                        mov al,[bx]+3

```

```

0023 8B 4F 03      mov cx,3[bx]
                  ; D~D'½D'DμD°ÑD'½D°Ñ D°D'Ñ€DμÑD°Ñ†D,Ñ
0026 BF 0002      mov di,ind
0029 8A 85 000E R  mov al,vec2[di]
                  ;mov cx,vec2[di]; warning A4031: Operand types
                  must match
                  ; DD'Ñ€DμÑD°Ñ†D,Ñ Ñ D±D°D·D,Ñ€D¾D²D°D'½D,DμD
                  ¼ D, D,D'½D'DμD°ÑD,Ñ€D¾D²D°D'½D,DμD¼
002D BB 0003      mov bx,3
0030 8A 81 0016 R  mov al,matr[bx][di]
                  ;mov cx,matr[bx][di]; warning A4031: Operand ty
                  pes must match
                  ; mov ax,matr[bx*4][di] ;error A2055: Illegal r
                  egister value
                  ; DŸD DžD'D·D DšD D D·D–D~DœDžD' DD"D D·D;DD
                  |D~D~ D; D£D§D·D¢DžDœ D;D·D“DœD·DD¢DžD'
                  ; DŸDμÑ€DμD¾D;Ñ€DμD'DμD»DμD'½D,Dμ
ÑDμD³D¼DμD'½Ñ,
                  D°
                  ; ----- D²D°Ñ€D,D°D'½Ñ, 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
                  ; ----- D²D°Ñ€D,D°D'½Ñ, 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
                ; ----- D²D°Ñ€D, D°D½Ñ, 3
0048 BF 0002      mov di,ind
004B 26: 89 01    mov es:[bx+di],ax
                ; ----- D²D°Ñ€D, D°D½Ñ, 4
004E 8B EC        mov bp,sp
                ;mov ax,matr[bp+bx] ;error A2046: Multiple base
                registers
                ; mov ax,matr[bp+di+si] ;error A2047: Multiple
                index registers
                ; D~ÑD¿D³¼D»Ñ€D·D³¼D²D°D½D,Dµ ÑDµD³D¼DµD½Ñ,D°
Ñ
                Ñ,DµD°D°
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

10/19/22 18:31:5

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	1b2	
@VERSION	TEXT	510	

83 Source Lines

83 Total Lines

19 Symbols

47836 + 459424 Bytes symbol space free

0 Warning Errors

0 Severe Errors