

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

**Кафедра Математического обеспечения электронно-вычислительных
машин**

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Организация ЭВМ и систем»
ТЕМА: ИЗУЧЕНИЕ РЕЖИМОВ АДРЕСАЦИИ И ФОРМИРОВАНИЯ
ИСПОЛНИТЕЛЬНОГО АДРЕСА.
Вариант 3

Студентка гр. 0382

Рубежова Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить режимы адресации и формирование исполнительного адреса в программах на языке Ассемблер.

Ход выполнения.

1. Протранслируем программу с созданием объектного файла *hello.obj* и файла диагностических сообщений *list.lst* с помощью строки :

```
> masm hello.asm
```

Файл диагностических сообщений см. приложение Б

2. В результате первоначальной трансляции программы были обнаружены ошибки и предупреждения:

1) Ошибка - попытка перемещения данных из памяти в память. Возможны лишь такие комбинации вида «пункт назначения – источник»: регистр – регистр, регистр – память, память – регистр, регистр – непосредственный операнд, память – непосредственный операнд.

```
mov mem3,[bx]
hello.asm(67): error A2052: Improper operand type
```

2) Предупреждение – размеры операндов должны совпадать. Элемент массива *vec2* размером в 1 байт, а регистр *CX* – двухбайтовый.

```
002D 8B 8D 000E R          mov cx,vec2[di]
hello.asm(80): warning A4031: Operand types must match
```

3) Предупреждение – размеры операндов должны совпадать. Элемент массива *matr* размером в 1 байт, а регистр *CX* – двухбайтовый.

```
0038 8B 89 0016 R          mov cx,matr[bx][di]
hello.asm(87): warning A4031: Operand types must match
```

4) Ошибка – недопустимое значение регистра. Масштабировать можно лишь расширенные регистры.

```
003C 8B 85 0022 R          mov ax,matr[bx*4][di]
hello.asm(88): error A2055: Illegal register value
```

- 5) Ошибка – допустимо использование лишь одного базового регистра для базово-индексной адресации.

```
005C 3E: 8B 86 0016 R          mov ax,matr[bp+bx]
hello.asm(119): error A2046: Multiple base registers
```

- 6) Ошибка – допустимо использование лишь одного индексного регистра для базово-индексной адресации.

```
0061 3E: 8B 83 0016 R          mov ax,matr[bp+di+si]
hello.asm(120): error A2047: Multiple index registers
```

3. Закомментируем соответствующие операторы в коде программы.
4. Снова протранслируем программу с созданием файла листинга *list1.lst*. И скомпилируем загрузочный модуль.
5. Выполним программу в пошаговом режиме под управлением отладчика с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды.

Начальное содержание сегментных регистров:

(CS)=1A0A, (DS)=19F5, (ES)=19F5, (SS)=1A05

Результаты прогона программы представлены в табл. 1.

Табл.1

Адрес команды	Символический код команды	16-ичный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	(SP)=0018 (IP)=0000 SS:SP +0 0000 +2 0000 +4 0000 +6 0000	(SP)=0016 (IP)=0001 SS:SP +0 19F5 +2 0000 +4 0000 +6 0000
0001	SUB AX, AX	2B C0	(IP) = 0001	(IP) = 0003
0003	PUSH AX	50	(SP)=0016 (IP)=0003 SS:SP +0 19F5 +2 0000 +4 0000	(SP)=0014 (IP)=0004 SS:SP +0 0000 +2 19F5 +4 0000

			+6 0000	+6 0000
0004	MOV AX, 1A07	B8 07 1A	(AX) = 0000 (IP) = 0004	(AX) = 1A07 (IP) = 0007
0007	MOV DS, AX	8E D8	(DS) = 19F5 (IP) = 0007	(DS) = 1A07 (IP) = 0009
0009	MOV AX, 01F4	B8 F4 01	(AX) = 1A07 (IP) = 0009	(AX) = 01F4 (IP) = 000C
000C	MOV CX, AX	8B C8	(CX) = 00B0 (IP) = 000C	(CX) = 01F4 (IP) = 000E
000E	MOV BL, 24	B3 24	(BX) = 0000 (IP) = 000E	(BX) = 0024 (IP) = 0010
0010	MOV BH, CE	B7 CE	(BX) = 0024 (IP) = 0010	(BX) = CE24 (IP) = 0012
0012	MOV [0002], FFCE	C7060200CEFF	DS:[0002] = 00 DS:[0003] = 00 (IP) = 0012	DS:[0002] = CE DS:[0003] = FF (IP) = 0018
0018	MOV BX, 0006	BB 06 00	(BX) = CE24 (IP) = 0018	(BX) = 0006 (IP) = 001B
001B	MOV [0000], AX	A3 00 00	DS:[0000] = 00 DS:[0001] = 00 (AX) = 01F4 (IP) = 001B	DS:[0000] = F4 DS:[0001] = 01 (IP) = 001E
001E	MOV AL, [BX]	8A 07	(AX) = 01F4 (BX) = 0006 DS:[0006] = 08 (IP) = 001E	(AX) = 0108 (IP) = 0020
0020	MOV AL, [BX+03]	8A 47 03	(AX) = 0108 (BX) = 0006 DS:[0009] = 05 (IP) = 0020	(AX) = 0105 (IP) = 0023
0023	MOV CX, [BX+03]	8A 47 03	(CX) = 01F4 (BX) = 0006 DS:[0009] = 05 (IP) = 0023	(CX) = 0105 (IP) = 0026
0026	MOV DI, 0002	8B 4F 03	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	MOV AL, [000E+DI]	8A 85 0E 00	(AX) = 0105 (DI) = 0002 DS:[0010] = 1E (IP) = 0029	(AX) = 011E (IP) = 002D
002D	MOV BX, 0003	BB 03 00	(BX) = 0006 (IP) = 002D	(BX) = 0003 (IP) = 0030
0030	MOV AL, [0016+BX+DI]	8A 81 16 00	(AX) = 011E (BX) = 0003 (DI) = 0002 DS:[001B] = 07 (IP) = 0030	(AX) = 0107 (IP) = 0034

0034	MOV AX, 1A07	B8 07 1A	(AX)=0107 (IP)=0034	(AX) = 1A07 (IP) = 0037
0037	MOV ES, AX	8E C0	(ES) = 19F5 (IP) = 0037	(ES) = 1A07 (IP) = 0039
0039	MOV AX, ES:[BX]	26 8B 07	(AX) = 1A07 (BX) = 0003 DS:[0003] = FF DS:[0004] = 00 (IP) = 0039	(AX) = 00FF (IP) = 003C
003C	MOV AX, 0000	B8 00 00	(AX) = 00FF (IP) = 003C	(AX) = 0000 (IP) = 003F
003F	MOV ES, AX	8E C0	(ES) = 1A07 (IP) = 003F	(ES) = 0000 (IP) = 0041
0041	PUSH DS	1E	(SP)=0014 (IP)=0041 SS:SP +0 0000 +2 19F5 +4 0000 +6 0000	(SP)=0012 (IP)=0042 SS:SP +0 1A07 +2 0000 +4 19F5 +6 0000
0042	POP ES	07	(ES) = 0000 (SP)=0012 (IP)=0042 SS:SP +0 1A07 +2 0000 +4 19F5 +6 0000	(ES) = 1A07 (SP)=0014 (IP)=0043 SS:SP +0 0000 +2 19F5 +4 0000 +6 0000
0043	MOV CX, ES:[BX-01]	26 8B 4F FF	(CX) = 0105 (BX) = 0003 DS:[0002] = CE DS:[0003] = FF (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	BF 02 00	(DI) = 0002 (IP) = 0048	(DI) = 0002 (IP) = 004B
004B	MOV ES:[BX+DI], AX	26 89 01	(AX) = FFCE (BX) = 0003 (DI) = 0002 DS:[0005] = 00 DS:[0006] = 08 (IP) = 004B	DS:[0005] = CE DS:[0006] = FF (IP) = 004E
004E	MOV BP, SP	8B EC	(BP) = 0000 (SP) = 0014 (IP) = 004E	(BP) = 0014 (IP) = 0050

0050	PUSH [0000]	FF 36 00 00	(SP)=0014 (IP)=0050 DS:[0000] = F4 DS:[0001] = 01 SS:SP +0 0000 +2 19F5 +4 0000 +6 0000	(SP)=0012 (IP)=0054 SS:SP +0 01F4 +2 0000 +4 19F5 +6 0000
0054	PUSH [0002]	FF 36 02 00	(SP)=0012 (IP)=0054 DS:[0002] = CE DS:[0003] = FF SS:SP +0 01F4 +2 0000 +4 19F5 +6 0000	(SP)=0010 (IP)=0058 SS:SP +0 FFCE +2 01F4 +4 0000 +6 19F5
0058	MOV BP, SP	8B EC	(BP) = 0014 (SP) = 0010 (IP) = 0058	(BP) = 0010 (IP) = 005A
005A	MOV DX, [BP+02]	8B 56 02	(DX) = 0000 (BP) = 0010 DS:[0012] = F6 DS:[0013] = EC (IP) = 005A	(DX) = 01F4 (IP) = 005D
005D	RET FAR 0002	CA 02 00	(CS) = 1A0A (SP) = 0010 (IP) = 005D SS:SP +0 FFCE +2 01F4 +4 0000 +6 19F5	(CS) = 01F4 (SP) = 0016 (IP) = FFCE SS:SP +0 19F5 +2 0000 +4 0000 +6 0000

Выводы.

В результате работы были изучены режимы адресации и формирование исполнительного адреса в программах, написанных на языке Ассемблер.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: hello.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 8,7,6,5,1,2,3,4
    vec2 DB -30,-40,30,40,-10,-20,10,20
    matr DB -1,-2,-3,-4,8,7,6,5,-5,-6,-7,-8,4,3,2,1
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,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 Б

ФАЙЛ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: list1.lst

Microsoft (R) Macro Assembler Version 5.10

10/3/21 23:57:41

Page 1-1

; 32-битовый процессор с архитектурой IntelX86

= 0024 EOL EQU '\$'

= 0002 ind EQU 2

= 01F4 n1 EQU 500

= -0032 n2 EQU -50

; 32-битовый процессор с архитектурой IntelX86

0000 AStack SEGMENT STACK

0000 000C[DW 12 DUP(?)

????

]

0018 AStack ENDS

; 32-битовый процессор с архитектурой IntelX86

0000 DATA SEGMENT

; 32-битовый процессор с архитектурой IntelX86

0000 0000 mem1 DW 0

0002 0000 mem2 DW 0

0004 0000 mem3 DW 0

0006 08 07 06 05 01 02 vec1 DB 8,7,6,5,1,2,3,4

03 04

000E E2 D8 1E 28 F6 EC vec2 DB -30,-40,30,40,-10,-20,10,20

0A 14


```

001E 8A 07                                mov al,[bx]

                                mov mem3,[bx]

hello.asm(67): error A2052: Improper operand type
; Ð°Ð°Ð·ÐžÑÐŸÐ²Ð°ÐœÐœÐ°Ñ Ð°ÐžÑÐµÑÐ°ÑÐžÑ

0020 8A 47 03                            mov al,[bx]+3

0023 8B 4F 03                            mov cx,3[bx]

; Ð°ÐœÐžÐµÐ°ÑÐœÐ°Ñ Ð°ÐžÑÐµÑÐ°ÑÐžÑÐžÑ

0026 BF 0002                            mov di,ind

0029 8A 85 000E R                        mov al,vec2[di]

002D 8B 8D 000E R                        mov cx,vec2[di]

hello.asm(80): warning A4031: Operand types must match

; Ð°ÐžÑÐµÑÐ°ÑÐžÑÑ Ð±Ð°Ð·ÐžÑÐŸÐ²Ð°ÐœÐžÐµÐœ Ðž
ÐžÐœÐžÐµÐ°ÑÐ°ÑÐžÑÐŸÐ²Ð°ÐœÐžÐµÐœ
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]

hello.asm(87): warning A4031: Operand types must match

003C 8B 85 0022 R                        mov ax,matr[bx*4][di]

hello.asm(88): error A2055: Illegal register value

; ÐŸÐžÐžÐÐÐ ÐÐÐ Ð ÐÐÐÐ~ÐœÐžÐ ÐÐÐÐ ÐÐ;ÐÐŠÐÐ Ð; ÐŁŚÐÐÐÐÐÐ
Ð;ÐÐÐÐœÐÐÐÐÐÐÐÐžÐ
; ÐµÑÐµŸÐŸÑÐµŸÐžÑÐµŸžÐµ»ÐµœÐžÐµÑÐµ³

Microsoft (R) Macro Assembler Version 5.10                                10/3/21 23:57:41

ÐœÐµÑÑÐ°Ñ°                                Page 1-3

```

[illegible]

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
	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	
= 0076	MAIN	F PROC	0000	CODE Length
	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	hello
@VERSION	TEXT	510

132 Source Lines

132 Total Lines

19 Symbols

47812 + 459448 Bytes symbol space free

2 Warning Errors

5 Severe Errors