

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. 9382

Павлов Р.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

Цель работы.

Изучить режимы адресации в ассемблере и формирование исполнительного адреса.

Задание.

Лабораторная работа 2 предназначена для изучения режимов адресации, использует готовую программу `lr2_compr.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.

Ход работы.

Часть 1.

1. Данные, соответствующие варианту (4), занесены в программу.
2. В ходе трансляции файла были обнаружены следующие ошибки:
 - LAB2.ASM(50): error A2048: Must be index or base register
в выражении `mov mem3,[bh]` слову `mem3` присваивается значение байта `bh`, поэтому возникает несоответствие из-за косвенной адресации
 - LAB2.ASM(65): error A2055: Illegal register value
в выражении `mov ax,matr[bx*4][di]` первый индекс является

недопустимым, т. к. это не базовый и не индексный регистр

- LAB2.ASM(90): error A2046: Multiple base registers
в выражении **mov ax,matr[bp+bx]** одновременно используются два базовых регистра, что недопустимо
- LAB2.ASM(91): error A2047: Multiple index registers
в выражении **mov ax,matr[bp+di+si]** одновременно используются два индексных регистра, что недопустимо.

3. Ошибки закомментированы, программа протранслирована ещё раз, скомпонован загрузочный модуль.

4. В отладчике afdpro пошагово выполнена программа, результаты прогона представлены ниже.

Начальные значения регистров:

AX	BX	CX	DX	DI	BP	SP	CS	DS	ES	IP
0000	0000	0000	0000	0000	0000	0018	1A0A	19F5	19F5	0000

Начальные значения регистров, которые не меняются в ходе программы:

SI	SS	HS	FS
0000	1A05	19F5	19F5

Изменение значений регистров в ходе работы программы:

Таблица 1.

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения

0000	PUSH DS	1E	(DS) = 19F5 (SP) = 0018 (IP) = 0000 Stack +0 7244 +2 0000 +4 0000 +6 0000	(DS) = 19F5 (SP)=0016 (IP) = 0001 Stack +0 19F5 +2 7244 +4 0000 +6 0000
0001	SUB AX, AX	2BC0	(AX) = 0000 (IP) = 0001	(AX) = 0000 (IP) = 0003
0003	PUSH AX	50	(AX) = 0000 (SP) = 0016 (IP) = 0003 Stack +0 19F5 +2 7244 +4 0000 +6 0000	(AX) = 0000 (SP) = 0014 (IP) = 0004 Stack +0 0000 +2 19F5 +4 7244 +6 0000
0004	MOV AX, 1A07	B8071A	(AX) = 0000 (IP) = 0004	(AX) = 1A07 (IP) = 0007
0007	MOV DS, AX	8ED8	(DS) = 19F5 (AX) = 1A07 (IP) = 0007	(DS) = 1A07 (AX) = 1A07 (IP) = 0009
0009	MOV AX, 01F4	B8F401	(AX) = 1A07 (IP) = 0009	(AX) = 01F4 (IP) = 000C
000C	MOV CX, AX	8BC8	(CX) = 0000 (AX) = 01F4 (IP) = 000C	(CX) = 01F4 (AX) = 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	(AX) = 01F4 (IP) = 001B	(AX) = 01F4 (IP) = 001E
001E	MOV AL, [BX]	8A07	(AX) = 01F4 (BX) = 0006 (IP) = 001E	(AX) = 01FF (BX) = 0006 (IP) = 0020
0020	MOV AL, [BX + 03]	8A4703	(AX) = 01FF (BX) = 0006 (IP) = 0020	(AX) = 0109 (BX) = 0006 (IP) = 0023

0023	MOV CX, [BX + 03]	8B4F03	(CX) = 01F4 (BX) = 0006 (IP) = 0023	(CX) = 0509 (BX) = 0006 (IP) = 0026
0026	MOV DI, 0002	BF0200	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	MOV AL, [000E + DI]	8A850E00	(AX) = 0109 (DI) = 0002 (IP) = 0029	(AX) = 0128 (DI) = 0002 (IP) = 002D
002D	MOV CX, [000E + DI]	8B8D0E00	(CX) = 0509 (DI) = 0002 (IP) = 002D	(CX) = 3228 (DI) = 0002 (IP) = 0031
0031	MOV BX, 0003	BB0300	(BX) = 0006 (IP) = 0031	(BX) = 0003 (IP) = 0034
0034	MOV AL, [0016 + BX + DI]	8A811600	(AX) = 0128 (BX) = 0003 (DI) = 0002 (IP) = 0034	(AX) = 01F9 (BX) = 0003 (DI) = 0002 (IP) = 0038
0038	MOV CX, [0016 + BX + DI]	8B891600	(CX) = 3228 (BX) = 0003 (DI) = 0002 (IP) = 0038	(CX) = FAF9 (BX) = 0003 (DI) = 0002 (IP) = 003C
003C	MOV AX, 1A07	B8071A	(AX) = 01F9 (IP) = 003C	(AX) = 1A07 (IP) = 003F
003F	MOV ES, AX	8EC0	(AX) = 1A07 (ES) = 19F5 (IP) = 003F	(AX) = 1A07 (ES) = 1A07 (IP) = 0041
0041	MOV AX, ES:[BX]	268B07	(AX) = 1A07 (BX) = 0003 (ES) = 1A07 (IP) = 0041	(AX) = 00FF (BX) = 0003 (ES) = 1A07 (IP) = 0044
0044	MOV AX, 0000	B80000	(AX) = 00FF (IP) = 0044	(AX) = 0000 (IP) = 0047
0047	MOV ES, AX	8EC0	(AX) = 0000 (ES) = 1A07 (IP) = 0047	(AX) = 0000 (ES) = 0000 (IP) = 0049
0049	PUSH DS	1E	(DS) = 1A07 (SP) = 14 (IP) = 0049 Stack +0 0000 +2 19F5 +4 7244 +6 0000	(DS) = 1A07 (SP) = 12 (IP) = 004A Stack +0 1A07 +2 0000 +4 19F5 +6 7244

004A	POP ES	07	(ES) = 0000 (SP) = 12 (IP) = 004A Stack +0 1A07 +2 0000 +4 19F5 +6 7244	(ES) = 1A07 (SP) = 14 (IP) = 004B Stack +0 0000 +2 19F5 +4 7244 +6 0000
004B	MOV CX, ES:[BX - 01]	268B4FFF	(BX) = 0003 (CX) = FAF9 (ES) = 1A07 (IP) = 004B	(BX) = 0003 (CX) = FFCE (EX) = 1A07 (IP) = 004F
004F	XCHG AX, CX	91	(AX) = 0000 (CX) = FFCE (IP) = 004F	(AX) = FFCE (CX) = 0000 (IP) = 0050
0050	MOV DI, 0002	BF0200	(DI) = 0002 (IP) = 0050	(DI) = 0002 (IP) = 0053
0053	MOV ES:[BX + DI], AX	268901	(AX) = FFCE (BX) = 0003 (DI) = 0002 (ES) = 1A07 (IP) = 0053	(AX) = FFCE (BX) = 0003 (DI) = 0002 (ES) = 1A07 (IP) = 0056
0056	MOV BP, SP	8BEC	(BP) = 0000 (SP) = 0014 (IP) = 0056	(BP) = 0014 (SP) = 0014 (IP) = 0058
0058	PUSH [0000]	FF360000	(SP) = 0014 (IP) = 0058 Stack +0 0000 +2 19F5 +4 7244 +6 0000	(SP) = 0012 (IP) = 005A Stack +0 01F4 +2 0000 +4 19F5 +6 7244
005C	PUSH [0002]	FF360200	(SP) = 0012 (IP) = 005A Stack +0 01F4 +2 0000 +4 19F5 +6 7244	(SP) = 0010 (IP) = 0060 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5
0060	MOV BP, SP	8BEC	(BP) = 0014 (SP) = 0010 (IP) = 0060	(BP) = 0010 (SP) = 0010 (IP) = 0062
0062	MOV DX, [BP + 02]	8B5602	(DX) = 01F4 (BP) = 0010 (IP) = 0062	(DX) = 01F4 (BP) = 0010 (IP) = 0065

0065	RET Far 0002	CA0200	(CS) = 1A0A (SP) = 0010 (IP) = 0065 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5	(CS) = 01F4 (SP) = 0016 (IP) = FFCE Stack +0 19F5 +2 0000 +4 0000 +6 0000
------	--------------	--------	---	---

Выводы.

В результате выполнения данной лабораторной работы были изучены различные режимы адресации в ассемблере и принципы формирования исполнительного адреса.

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД

- имя файла : lb2_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 12,11,10,9,5,6,7,8
    vec2 DB -40,-50,40,50,-20,-30,20,30
    matr DB 5,6,7,8,-8,-7,-6,-5,1,2,3,4,-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,[bh]

; 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. ТЕКСТ ЛИСТИНГОВ

- имя файла: LB2_COMP.lst

#Microsoft (R) Macro Assembler Version 5.10

10/21/20 21:10:0

Page 1-1

```

;          Программа          изучения          режим
;mov адресации процессора I
ntelX86
= 0024          EOL EQU '$'
= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50

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

0018          AStack ENDS

; Данные программы
0000  DATA SEGMENT
      ;          Директивы          описания          й
      'анных
0000  0000          mem1 DW 0
0002  0000          mem2 DW 0
0004  0000          mem3 DW 0
0006  0C 0B 0A 09 05 06          vec1 DB 12,11,10,9,5,6,7,8
      07 08
000E  D8 CE 28 32 EC E2          vec2 DB -40,-50,40,50,-20,-30,20,30
      14 1E
0016  05 06 07 08 F8 F9          matr DB 5,6,7,8,-8,-7,-6,-5,1,2,3,4,-4,
      -3,-2,-1
      FA FB 01 02 03 04
      FC FD FE FF
0026          DATA ENDS

; Код программы
0000  CODE SEGMENT
      ASSUME CS:CODE, DS:DATA, SS:AStack

; Головная процедура
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

; ПРОВЕРКА РЕЖИМО
В          АДРЕСАЦИИ          НА          УРОВНЕ          СМЕЙ
©ЕНИЙ

; Регистровая адр
есация
0009  B8 01F4          mov ax,n1

```

```

000C  8B C8                mov cx,ax
000E  B3 24                mov bl,EOL

```

#Microsoft (R) Macro Assembler Version 5.10

10/21/20 21:10:0
Page 1-2

```

0010  B7 CE                mov bh,n2

; Прямая адресация
я
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,[bh]

; 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]
002D  8B 8D 000E R        mov cx,vec2[di]
lr2_comp.asm(59): warning A4031: Operand types must match

```

```

; Адресация с базой
; ,рованием и индексированием
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]
lr2_comp.asm(64): warning A4031: Operand types must match
;mov ax,matr[bx*4][di]

```

```

; ПРОВЕРКА РЕЖИМОВ
В АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ
0040  00 00 00 00

```

```

;
и сегмента

```

```

; ----- вариант 1
003C  B8 ---- R          mov ax, SEG vec2
003F  8E C0                mov es, ax
0041  26: 8B 07            mov ax, es:[bx]
0044  B8 0000            mov ax, 0

```

```

; ----- вариант 2
0047  8E C0                mov es, ax
0049  1E                    push ds
004A  07                    pop es
004B  26: 8B 4F FF            mov cx, es:[bx-1]
004F  91                    xchg cx,ax

```

```

0050  BF 0002                mov di,ind
0053  26: 89 01              mov es:[bx+di],ax

; ----- вариант 4
0056  8B EC                mov bp,sp
;mov ax,matr[bp+bx]
;mov ax,matr[bp+di+si]

; Использование с
егмента стека
0058  FF 36 0000 R          push mem1
005C  FF 36 0002 R          push mem2
0060  8B EC                mov bp,sp
0062  8B 56 02              mov dx,[bp]+2
0065  CA 0002              ret 2
0068                                Main ENDP

0068                                CODE ENDS
                                END Main

```

Segments and Groups:

	N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	0068	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 = 0068
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		

```
102 Source  Lines
102 Total   Lines
19 Symbols

47800 + 459460 Bytes symbol space free

2 Warning Errors
0 Severe  Errors
```