

**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 основной памяти**

Студент гр. 7383

\_\_\_\_\_

Кирсанов А.Я.

Преподаватель

\_\_\_\_\_

Кирияничиков В.А.

Санкт-Петербург

2018

## **Цель работы**

Изучить режимы адресации основной памяти, используя программу на языке Ассемблер, выполнение которой производится под управлением отладчика в пошаговом режиме.

## **Задание**

Объяснить и закомментировать ряд ошибок в коде программы, составить протокол выполнения программы в пошаговом режиме отладчика.

## **Порядок выполнения задания**

1. Получить вариант выбора значений исходных данных (массивов) `vec1`, `vec2` и `matr` из файла `lr2.dat` и занести свои данные вместо значений, указанных в приведенной для образца программе.

2. Протранслировать программу с созданием файла диагностических сообщений и объяснить обнаруженные ошибки (`error`) и предупреждения (`warning`). Закомментировать операторы с ошибками в тексте программы, а операторы с предупреждениями оставить без изменения. Объяснения ошибок и предупреждений должны быть приведены в отчете по лабораторной работе.

3. Снова протранслировать программу и скомпоновать загрузочный модуль. Учесть, что программа - учебная и может выполняться только под отладчиком. В автоматическом режиме она выполняться не должна.

4. Выполнить программу в пошаговом режиме под управлением отладчика с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения каждой команды. Разобраться в используемых режимах адресации и получаемых результатах.

5. Результаты прогона программы под управлением отладчика должны быть подписаны преподавателем и представлены в отчете по лабораторной работе в виде, аналогичном указанному в лабораторной работе №1.

## Ход работы.

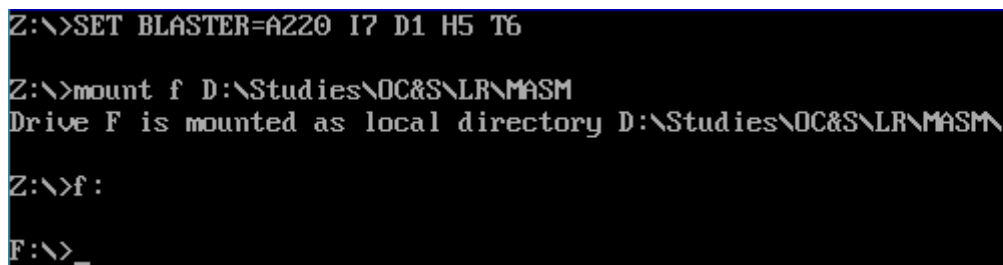
1. Открыт файл программы LR2\_comp.ASM и изменены значения, указанные для образца на свои данные в соответствии со своим вариантом.

mem1	DW	0
mem2	DW	0
mem3	DW	0
vec1	DB	38,37,36,35,31,32,33,34
vec2	DB	70,80,-70,-80,50,60,-50,-60
matr	DB	-2,-1,5,6,-8,-7,3,4,-4,-3,7,8,-6,-5,1,2

2. Запущена программа DOSBox0.74-2 и подключена папка MASM с помощью строки:

```
> mount f D:\Studies\OC&S\LR\MASM
```

```
>f:
```



```
Z:\>SET BLASTER=A220 I7 D1 H5 T6  
Z:\>mount f D:\Studies\OC&S\LR\MASM  
Drive F is mounted as local directory D:\Studies\OC&S\LR\MASM\  
Z:\>f:  
F:\>_
```

Протранслирована программа с помощью строки:

```
> MASM LR2_comp.ASM
```

с созданием файла диагностических сообщений (файла листинга) и объяснением обнаруженных ошибок (error) и предупреждений (warning). Перечень всех ошибок и предупреждений в начале трансляции программы и их объяснение представлено ниже:

- Строка 55. Ошибка в операторе `mov mem3,[bx]`. Эта команда переводит информацию из сегмента памяти в другой, что невозможно в языке Assembler. В данном случае необходимо перевести информацию из памяти в регистр, а затем уже в необходимый сегмент информацию перевести из регистра.
- Строка 62. `mov ex, vec2[di]`. LR2.ASM (62): warning A4031: Operand types must match (Несоответствие типов операндов). Размер элементов массива 'vec2' 1 байт, а 'cx' - 2 байта

- Строка 67. Ошибка в операторе `mov ax,matr[bx*4][di]`. Здесь используется базово-индексная адресация. При данном типе адресации надо сначала изменить значение регистра, затем уже переводить информацию.
- Строка 87. Ошибка в операторе `mov ax,matr[bp+bx]`. LR2.ASM(86): errorA2046: Multiple base registers. Слишком много базовых регистров. Нельзя использовать более одного базового регистра. Несоответствие типов операндов. Размер элементов матрицы 'matr' 1 байт, а 'ax' - 2 байта
- Строка 88. Ошибка в операторе `mov ax,matr[bp+di+si]`. LR2.ASM(87): error A2047: Multiple index registers. Слишком много индексных регистров. Нельзя использовать более одного индексного регистра. Слишком много регистров. Нельзя использовать более двух регистров. Несоответствие типов операндов. Размер элементов матрицы 'matr' 1 байт, а 'ax' - 2 байта
- Строка 95. Фазовая ошибка CODE ENDS. Исчезает после исправления всех вышеперечисленных ошибок.
- Семантическая ошибка. `push mem1, push mem2`. Для завершения программы и возвращения в DOS необходимо, чтобы вершина стека содержала смещение и сегмент начала PSP, а в случае выполнения этих двух команд вершина стека содержит `mem2` и `mem1`, поэтому при выполнении команды `ret 2` управление перейдет на адрес `mem1: mem2` и программа не сможет корректно завершиться.

```

Z:\>f:

F:\>MASM LR2_comp.ASM
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [LR2_comp.OBJ]:
Source listing [NUL.LST]: LR2_comp.LST
Cross-reference [NUL.CRF]:
LR2_comp.ASM(1): warning A4001: Extra characters on line
LR2_comp.ASM(55): error A2052: Improper operand type
LR2_comp.ASM(62): warning A4031: Operand types must match
LR2_comp.ASM(66): warning A4031: Operand types must match
LR2_comp.ASM(67): error A2055: Illegal register value
LR2_comp.ASM(87): error A2046: Multiple base registers
LR2_comp.ASM(88): error A2047: Multiple index registers
LR2_comp.ASM(95): error A2006: Phase error between passes

47800 + 459460 Bytes symbol space free

3 Warning Errors
5 Severe Errors

F:\>_

```

Закомментированы операторы с ошибками в тексте программы, а операторы с предупреждениями оставлены без изменения.

3. Программа протранслирована еще раз, скомпонован загрузочный модуль с помощью строки:

> LINK LR2\_comp.OBJ

с созданием карты памяти LR2\_comp.MAP и исполняемого файла LR2\_comp.EXE.

```

F:\>MASM LR2_comp.ASM
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [LR2_comp.OBJ]:
Source listing [NUL.LST]: LR2_comp.LST
Cross-reference [NUL.CRF]:
LR2_comp.ASM(1): warning A4001: Extra characters on line
LR2_comp.ASM(62): warning A4031: Operand types must match
LR2_comp.ASM(66): warning A4031: Operand types must match

47800 + 459460 Bytes symbol space free

3 Warning Errors
0 Severe Errors

F:\>_

```

```

F:\>LINK LR2_comp.OBJ

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LR2_COMP.EXE]:
List File [NUL.MAP]: LR2_comp.MAP
Libraries [.LIB]:

F:\>

```

4. Запущено выполнение программы под управлением отладчика с помощью команды:

> AFD LR2\_comp.EXE

Программа выполнена в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды. Результаты прогона программы под управлением отладчика представлены в таблице 1.

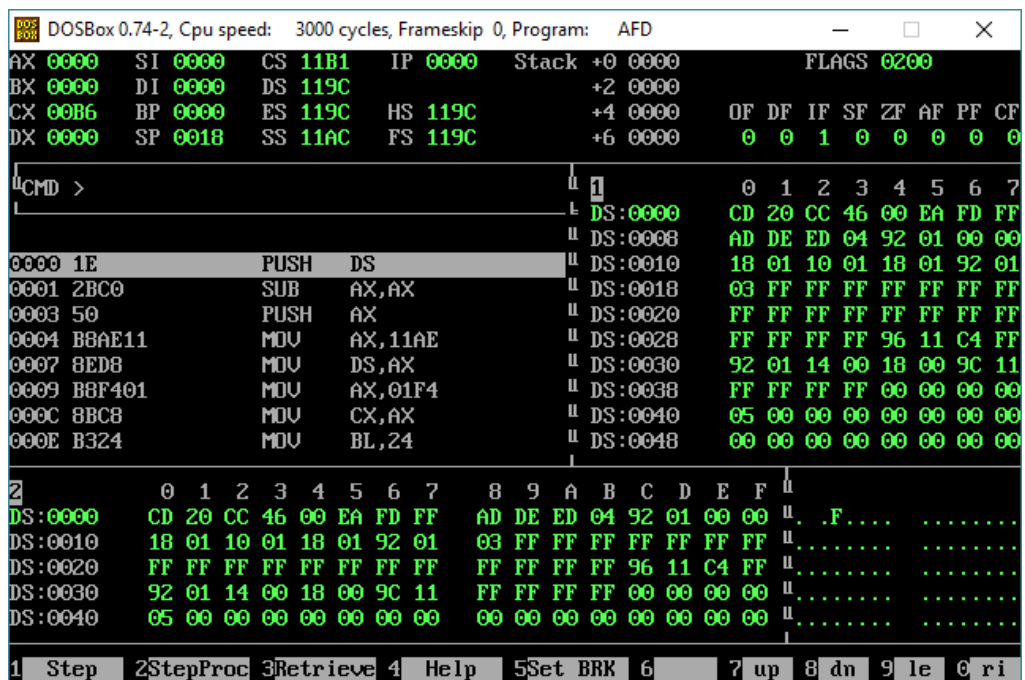


Таблица 1 - Результаты прогона программы под управлением отладчика

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	SP = 0016 IP = 0001 Stack +0 119C

			+2 0000 +4 0000 +6 0000	+2 0000 +4 0000 +6 0000
0001	Sub AX, AX	2BC0	IP = 0001	IP = 0003
0003	Push AX	50	SP = 0016 IP = 0003 Stack +0 119C +2 0000 +4 0000 +6 0000	SP = 0014 IP = 0004 Stack +0 0000 +2 119C +4 0000 +6 0000
0004	Mov AX, 11AE	B8AE11	AX = 0000 IP = 0004	AX = 11AE IP = 0007
0007	Mov DS, AX	8ED8	DS = 119C IP = 0007	DS = 11AE IP = 0009
0009	Mov AX, 01F4	B8F401	AX = 11AE IP = 0009	AX = 01F4 IP = 000C
000C	Mov CX, AX	8BC8	CX = 00B6 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 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 DS:0000 = 00 DS:0001 = 00	IP = 001E DS:0000 = F4 DS:0001 = 01
001E	Mov AL, [BX]	8A07	AX = 01F4 IP = 001E	AX = 0126 IP = 0020
0020	Mov AL, [BX+03]	8A4703	AX = 0126 IP = 0020	AX = 0123 IP = 0023
0023	Mov CX, [BX+03]	8B4F03	CX = 01F4 IP = 0023	CX = 1F23 IP = 0026
0026	Mov DI, 0002	BF0200	DI = 0000 IP = 0026	DI = 0002 IP = 0029
0029	Mov AL, [DI+000E]	8A850E00	AX = 0123 IP = 0029	AX = 01BA IP = 002D
002D	Mov CX, [DI+000E]	8B8D0E00	CX = 1F23 IP = 002D	CX = B0BA IP = 0031
0031	Mov BX, 0003	BB0300	BX = 0006 IP = 0031	BX = 0003 IP = 0034
0034	Mov AL, [BX+DI+0016]	8A811600	AX = 01BA IP = 0034	AX = 01F9 IP = 0038
0038	Mov CX, [BX+DI+0016]	8B891600	CX = B0BA IP = 0038	CX = 03F9 IP = 003C

003C	Mov AX, 11AE	B8AE11	AX = 01F9 IP = 003C	AX = 11AE IP = 003F
003F	Mov ES, AX	8EC0	ES = 119C IP = 003F	ES = 11AE IP = 0041
0041	Mov AX, ES:[BX]	268B07	AX = 11AE IP = 0041	AX = 00FF IP = 0044
0044	Mov AX, 0000	B80000	AX = 00FF IP = 0044	AX = 0000 IP = 0047
0047	Mov ES, AX	8EC0	ES = 11AE IP = 0047	ES = 0000 IP = 0049
0049	Push DS	1E	SP = 0014 IP = 0049 Stack +0 0000 +2 119C +4 0000 +6 0000	SP = 0012 IP = 004A Stack +0 11AE +2 0000 +4 119C +6 0000
004A	Pop ES	07	SP = 0012 ES = 0000 IP = 004A Stack +0 11AE +2 0000 +4 119C +6 0000	SP = 0014 ES = 11AE IP = 004B Stack +0 0000 +2 119C +4 0000 +6 0000
004B	Mov CX, ES:[BX-01]	268B4FFF	CX = 03F9 IP = 004B	CX = FFCE IP = 004F
004F	Xchg AX, CX	91	AX = 0000 CX = FFCE IP = 004F	AX = FFCE CX = 0000 IP = 0050
0050	Mov DI, 0002	BF0200	IP = 0050	IP = 0053
0053	Mov ES:[BX+DI], AX	268901	IP = 0053 DS:0005 = 00 DS:0006 = 26	IP = 0056 DS:0005 = CE DS:0006 = FF
0056	Mov BP, SP	8BEC	BP = 0000 IP = 0056	BP = 0014 IP = 0058
0058	Push [0000]	FF360000	SP = 0014 IP = 0058 Stack +0 0000 +2 119C +4 0000 +6 0000	SP = 0012 IP = 005C Stack +0 01F4 +2 0000 +4 119C +6 0000
005C	Push [0002]	FF360200	SP = 0012 IP = 005C Stack +0 01F4 +2 0000 +4 119C +6 0000	SP = 0010 IP = 0060 Stack +0 FFCE +2 01F4 +4 0000 +6 119C



0060	Mov BP, SP	8BEC	BP = 0014 IP = 0060	BP = 0010 IP = 0062
0062	Mov DX, [BP+02]	8B5602	DX = 0000 IP = 0062	DX = 01F4 IP = 0065
0065	Ret Far	CB	SP = 0010 CS = 11B1 IP = 0065 Stack +0 FFCE +2 01F4 +4 0000 +6 119C	SP = 0014 CS = 01F4 IP = FFCE Stack +0 0000 +2 119C +4 0000 +6 0000

### Выводы

В данной лабораторной работы изучены режимы адресации основной памяти, используя программу, выполнение которой производится под управлением отладчика в пошаговом режиме. Также исправлены ряд ошибок в программе и составлен протокол ее выполнения в пошаговом режиме отладчика.

## ПРИЛОЖЕНИЕ

### Файл LR2\_comp.ASM:

```
; Учебная программа лабораторной работы №2 по дисциплине "Архитектура компьютера"
;
;
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 38,37,36,35,31,32,33,34
vec2 DB 70,80,-70,-80,50,60,-50,-60
matr DB -2,-1,5,6,-8,-7,3,4,-4,-3,7,8,-6,-5,1,2
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]
; Индексированная 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]
```



```
000E 46 50 BA B0 32 3C vec2 DB 70,80,-70,-80,50,60,-50,-60  
CE C4  
0016 FE FF 05 06 F8 F9 matr DB -2,-1,5,6,-8,-7,3,4,-4,-3,7,8,-  
6,-5,1,2  
03 04 FC FD 07 08  
FA FB 01 02  
0026 DATA ENDS  
; P«P»PsPr PïCßPsPiCßP°PjPjC<  
0000 CODE SEGMENT  
ASSUME CS:CODE, DS:DATA, SS:AStack  
; P“PsP»PsPIPSPP°C¼ PïCßPsC†PµPrCfCßP°  
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  
  
Microsoft (R) Macro Assembler Version 5.10 10/2/18 18:21:01  
Page 1-2  
; P¼P PhP’P•P P½Ph P P•P–P¶P½PhP’ PhP”P P•PŸPh  
P!P¶P¶ P¿Ph PJP PhP’P¿P• PŸP½P•P@P•P¿P¶P™  
; P PµPiPëCfC,CßPsPIPP°C¼ P°PrCßPµCfP°C†PëC¼  
0009 B8 01F4 mov ax,n1  
000C 8B C8 mov cx,ax  
000E B3 24 mov bl,EOL  
0010 B7 CE mov bh,n2  
; P¼CßC¼PjP°C¼ P°PrCßPµCfP°C†PëC¼  
0012 C7 06 0002 R FFCE mov mem2,n2  
0018 BB 0006 R mov bx,OFFSET vec1  
001B A3 0000 R mov mem1,ax  
; P½PsCfPIPPµPSPSP°C¼ P°PrCßPµCfP°C†PëC¼  
001E 8A 07 mov al,[bx]  
; mov mem3,[bx]  
; P‘P°P•PëCßPsPIPP°PSPSP°C¼ P°PrCßPµCfP°C†PëC¼  
0020 8A 47 03 mov al,[bx]+3  
0023 8B 4F 03 mov cx,3[bx]  
; P¶PSPGrµPeCfPëCßPsPIPP°PSPSP°C¼ P°PrCßPµCfP°C†PëC¼  
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(62): warning A4031: Operand types must match  
; PhPrCßPµCfP°C†PëC¼ Cf P±P°P•PëCßPsPIPP°PSPëPµ  
Pj Pë PëPSPrµPeCfPëCßPsPIPP°PSPëPµPj  
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(66): warning A4031: Operand types must match  
; mov ax,matr[bx*4][di]  
; P¼P PhP’P•P P½Ph PhP”P P•PŸPhP!P¶P¶ PŸ PJPŞP  
•PŸPhP½ PŸP•P“P½P•P¿PŸPhP’  
; P¼PµCßPµPsPïCßPµPrPµP»PµPSPëPµ CfPµPiPjPµPSC  
P°  
; ----- PIPP°CßPëP°PSC, 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  
; ----- PIPP°CßPëP°PSC, 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
; ----- PIP°CЪPëP°PSC, 3
0050 BF 0002                          mov  di,ind
0053 26: 89 01                      mov  es:[bx+di],ax
; ----- PIP°CЪPëP°PSC, 4
0056 8B EC                          mov  bp,sp
;                                mov  ax,matr[bp+bx]
;                                mov  ax,matr[bp+di+si]
; P°CfPïPsP»CЪP·PsPIP°PSPëPµ CfPµPiPjPµPSC,P°
CfC,PµPëP°
0058 FF 36 0000 R                    push mem1
Microsoft (R) Macro Assembler Version 5.10
10/2/18 18:21:01
Page 1-3

005C FF 36 0002 R                    push mem2
0060 8B EC                          mov  bp,sp
0062 8B 56 02                      mov  dx,[bp]+2
0065 CB                            ret
0066                                Main  ENDP
0066                                CODE  ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10 10/2/18 18:21:01  
Symbols-1

#### Segments and Groups:

	N a m e	Length	Align	Combine	Class
ASTACK	. . . . .	0018	PARA	STACK	
CODE	. . . . .	0066	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 = 0066
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		

```

97 Source Lines
97 Total Lines
19 Symbols
47800 + 459460 Bytes symbol space free
3 Warning Errors
0 Severe Errors

```

#### Файл LR2\_comp.MAP:

Start	Stop	Length	Name	Class
00000H	00017H	00018H	ASTACK	
00020H	00045H	00026H	DATA	
00050H	000B5H	00066H	CODE	

Program entry point at 0005:0000