

**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 формирование**  
**исполнительного адреса»**

Студентка гр. 1381

Васильева О. М.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

### **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.**

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	<code>vec1</code>	<code>5,6,7,8,12,11,10,9</code>
	<code>vec2</code>	<code>-20,-30,20,30,-40,-50,40,50</code>
	<code>matr</code>	<code>-5,-6,-7,-8,4,3,2,1,-1,-2,-3,-4,8,7,6,5</code>

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX -
47784 + 459476 Bytes symbol space free
2 Warning Errors
5 Severe Errors
C:\>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
Cross-reference [NUL.CRF]:
lr2_comp.asm(47): error A2052: Improper operand type
lr2_comp.asm(54): warning A4031: Operand types must match
lr2_comp.asm(58): warning A4031: Operand types must match
lr2_comp.asm(59): error A2055: Illegal register value
lr2_comp.asm(75): error A2006: Phase error between passes
47784 + 459476 Bytes symbol space free
2 Warning Errors
3 Severe Errors
C:\>_
```

2. Протранслировать программу с созданием файла диагностических сообщений; объяснить обнаруженные ошибки и закомментировать соответствующие операторы в тексте программы.

1) Неподходящий тип операндов. Нельзя читать из памяти и писать в память одной командой.

```
65                                mov mem3,[bx]
66 lr2_comp.asm(47): error A2052: Improper operand type
```

2) Типы операндов должны соответствовать. Размер элементов массива 'vec2' 1 байт, а 'cx' - 2 байта

```
73 002D 8B 8D 000E R                mov cx,vec2[di]
74 lr2_comp.asm(54): warning A4031: Operand types must match
```

3) Типы операндов должны соответствовать. Размер элементов матрицы 'matr' 1 байт, а 'cx' - 2 байта

```
79 0038 8B 89 0016 R                mov cx,matr[bx][di]
80 lr2_comp.asm(58): warning A4031: Operand types must match
```

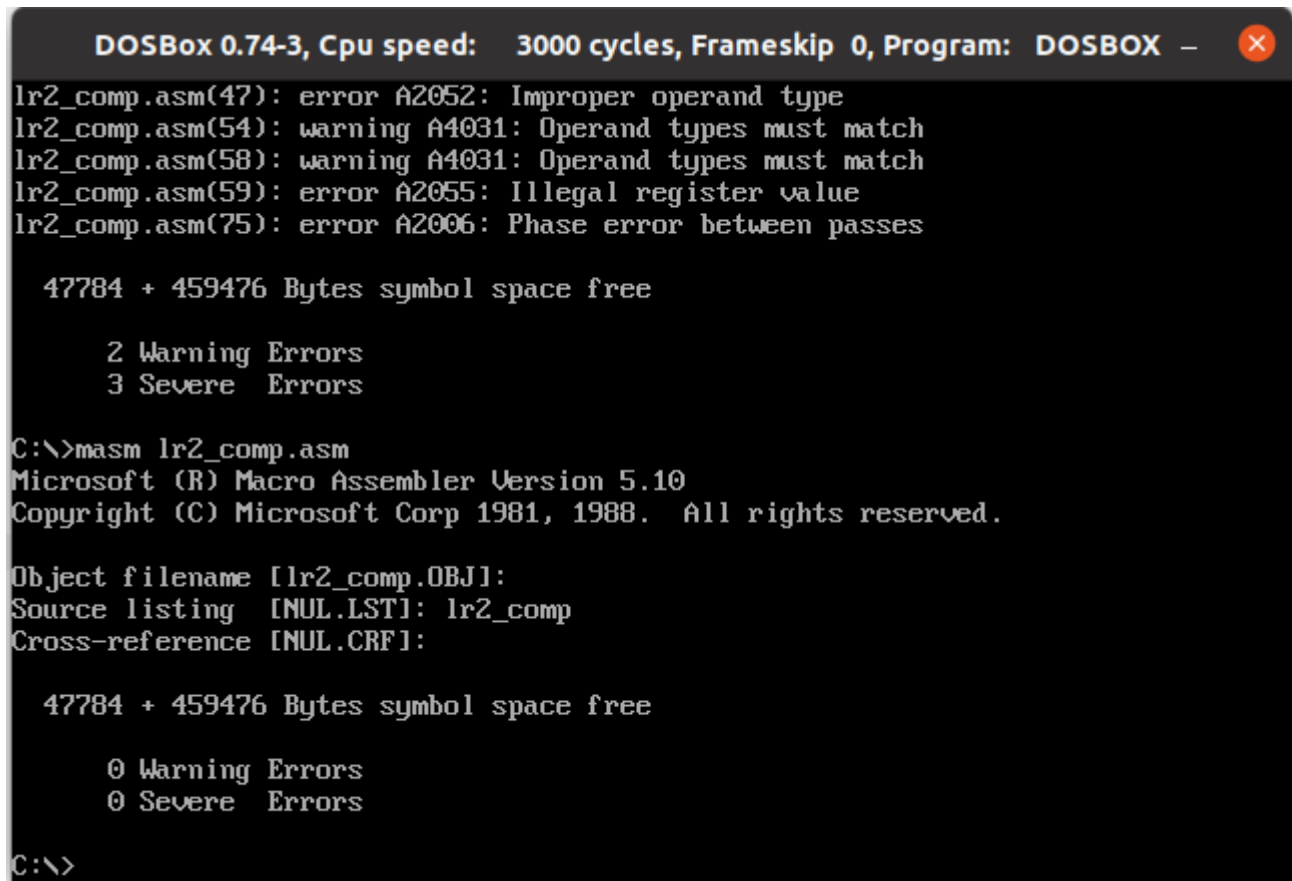
4) Недопустимое значение регистра. Нельзя умножать 16-битные регистры.

```
81 003C 8B 85 0022 R                mov ax,matr[bx*4][di]
82 lr2_comp.asm(59): error A2055: Illegal register value
```

5) Ошибка фазы между проходами. В основной функции содержатся ошибки.

```
101 0059                               Main    ENDP
102 lr2 comp.asm(75): error A2006: Phase error between passes
```

3. Снова протранслировать программу и скомпоновать загрузочный модуль.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX -
lr2_comp.asm(47): error A2052: Improper operand type
lr2_comp.asm(54): warning A4031: Operand types must match
lr2_comp.asm(58): warning A4031: Operand types must match
lr2_comp.asm(59): error A2055: Illegal register value
lr2_comp.asm(75): error A2006: Phase error between passes

47784 + 459476 Bytes symbol space free

2 Warning Errors
3 Severe Errors

C:\>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
Cross-reference [NUL.CRF]:

47784 + 459476 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>
```

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.

### Протокол пошагового исполнения.

Табл. 1- lr2\_comp.exe

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

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	1E	PUSH DS	(AX) = 0000 (IP) = 0000 (DS) = 19F5 STACK (+0) = 0000 (SP) = 0018	(AX) = 1A07 (IP) = 0001 (DS) = 19F5 STACK (+0) = 19F5 (SP) = 0016
0001	2BC0	SUB AX, AX	(AX) = 0000 (IP) = 0001 (DS) = 19F5	(AX) = 0000 (IP) = 0003 (DS) = 19F5
0003	50	PUSH AX	(AX) = 0000 (IP) = 0003 (DS) = 19F5 STACK (+0) = 19F5 Stack (+2) = 0000 (SP) = 0016	(AX) = 0000 (IP) = 0004 (DS) = 19F5 STACK (+0) = 0000 STACK (+2) = 19F5 (SP) = 0014
0004	B8071A	MOV AH, 1A07	(AX) = 0000 (IP) = 0004 (DS) = 19F5	(AX) = 1A07 (IP) = 0007 (DS) = 19F5
0007	8ED8	MOV DS, AX	(AX) = 1A07 (IP) = 0007 (DS) = 19F5	(AX) = 1A07 (IP) = 0009 (DS) = 1A07
0009	B8F401	MOV AX, 01F4	(AX) = 1A07 (IP) = 0009	(AX) = 01F4 (IP) = 000C
000C	8BC8	MOV CX, AX	(CX) = 009D	(DS) = 01F4

			(AX) = 01F4 (IP) = 000C (DS) = 1A07	(AX) = 01F4 (IP) = 000E (DS) = 1A07
000E	B324	MOV BL, 24	(BX) = 0000 (AX) = 01F4 (IP) = 000E	(BX) = 0024 (AX) = 01F4 (IP) = 0010
0010	B7CE	MOV BH, CE	(BX) = 0024 (IP) = 0010	(BX) = CE24 (IP) = 0012
0012	C7060200CEFF	MOV [0002], FFCE	(IP) = 0012	(IP) = 0018
0018	BB0600	MOV BX, 0006	(BX) = CE24 (IP) = 0018	(BX) = 0006 (IP) = 001B
001B	A30000	MOV [0000], AX	(AX) = 01F4 (IP) = 001B	(AX) = 01F4 (IP) = 001E
001E	8A07	MOV AL, [BX]	(AX) = 01F4 (IP) = 001E	(AX) = 0105 (IP) = 0020
0020	8A4703	MOV AL, [BX+03]	(AX) = 0105 (BX) = 0006 (IP) = 0020	(AX) = 0108 (BX) = 0006 (IP) = 0023
0023	8B4F03	MOV CX, [BX+03]	(CX) = 01F4 (IP) = 0023	(CX) = 0C08 (IP) = 0026
0026	BF0200	MOV DI, 0002	(DI) = 0000 (IP) = 0026	(DI) = 0002 (IP) = 0029
0029	8A850E00	MOV AL, [000E+DI]	(AX) = 0108 (IP) = 0029 (DI) = 0002	(AX) = 0114 (IP) = 002D (DI) = 0002
002D	BB0300	MOV BX, 0003	(IP) = 002D (BX) = 0006	(IP) = 0030 (BX) = 0003
0030	8A811600	MOV AL, [0016+BX+DI]	(AX) = 0114 (IP) = 0030 (BX) = 0003 (DI) = 0002	(AX) = 0103 (IP) = 0034 (BX) = 0003 (DI) = 0002
0034	8ECO	MOV ES, AX	(ES) = 19F5 (IP) = 0034 (AX) = 0103	(ES) = 0103 (AX) = 0103 (IP) = 0036

0036	1E	PUSH DS	(AX) = (IP) = 0036 STACK (+0) = 0000 STACK (+2) = 19F5 STACK (+4) = 0000 (SP) = 0014	(AX) = 0103 (IP) = 0037 STACK (+0) = 1A07 STACK (+2) = 0000 STACK (+4) = 19F5 (SP) = 0012
00037	07	POP ES	(ES) = 0103 (IP) = 0037 (SP) = 0012 STACK (+0) = 1A07 STACK (+2) = 0000 STACK (+4) = 19F5	(ES) = 1A07 (IP) = 0038 (SP) = 0014 STACK (+0) = 0000 STACK (+2) = 19F5 STACK (+4) = 0000
0038	268B4FFF	MOV CX, ES: [BX-01]	(CX) = 0C08 (IP) = 0038	(CX) = FFCE (IP) = 003C
003C	91	XCHG AX, AX	(AX) = 0103 (IP) = 003C	(AX) = FFCE (IP) = 003D
003D	FF360000	PUSH [0000]	(AX) = FFCE (IP) = 003D STACK (+0) = 0000 STACK (+2) = 19F5 STACK (+4) = 0000 (SP) = 0014	(AX) = FFCE (IP) = 0041 STACK (+0) = 01F4 STCK (+2) = 0000 STACK (+4) = 19F5 (SP) = 0012
0041	FF360200	PUSH [0002]	(IP) = 0041 (SP) = 0012 STACK (+0) = 01F4 STCK (+2) = 0000 STACK (+4) = 19F5 STACK (+6) = 0000	(IP) = 0045 (SP) = 0010 STACK (+0) = FFCE STCK (+2) = 01F4 STACK (+4) = 0000 STCK (+6) = 19F5
0045	8BEC	MOV BP, SP	(IP) = 0045 (BP) = 0000	(IP) = 0047 (BP) = 0010
0047	8B5602	MOV DX, [BP+02]	(IP) = 0047 (DX) = 0000	(IP) = 004A (DX) = 01F4

004A	CA0200	RET Far 0002	(IP) = 004A (CS) = 1A0A (SP) = 0010 STACK (+0) = FFCE STACK (+2) = 01F4 STACK (+4) = 0000 STACK(+6) = 19F5	(IP) = FFCE (CS) = 01F4 (SP) = 0016 STACK (+0) = 19F5 STCK (+2) = 0000 STACK (+4) = 0000 STACK (+6) = 0000
------	--------	--------------	--	--

### **Вывод.**

В ходе выполнения лабораторной работы были получены основные навыки программирования на ассемблере, изучены основные режимы адресации памяти. Результат выполнения работы стали программы lr2\_comp.exe.



## ПРИЛОЖЕНИЕ А. Код программ.

Имя файла: lr2\_comp.asm

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 5,6,7,8,12,11,10,9

vec2 DB -20,-30,20,30,-40,-50,40,50

matr DB -5,-6,-7,-8,4,3,2,1,-1,-2,-3,-4,8,7,6,5

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]

;

; 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 2

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

; Использование сегмента стека

Main  
CODE

xchg cx,ax

push mem1

push mem2

mov bp,sp

mov dx,[bp]+2

ret 2

ENDP

ENDS

END Main