

**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. 2384

\_\_\_\_\_

Кузьминых Е.М.

Преподаватель

\_\_\_\_\_

Морозов С.М.

Санкт-Петербург

2023

### **Цель работы.**

Проанализировать программу на языке Ассемблера и найти в ней ошибки, закомментировать код с ошибками, сделать трансляцию, компоновку и отладку программы, зафиксировать изменение регистров во время исполнения программы, изучить виды адресации.

### **Задание.**

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

3. Снова протранслировать программу и скомпоновать загрузочный модуль.

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.

### **Выполнение работы.**

1. Был получен вариант №4, изменен файл `LR2_comp.asm` (23-25 строки).
2. Произведена трансляция программы

```

F:\>masm.exe lr2.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lr2.OBJ]:
Source listing [NUL.LST]: lr2.lst
Cross-reference [NUL.CRF]:
lr2.asm(54): error A2052: Improper operand type
lr2.asm(61): warning A4031: Operand types must match
lr2.asm(65): warning A4031: Operand types must match
lr2.asm(66): error A2055: Illegal register value
lr2.asm(86): error A2046: Multiple base registers
lr2.asm(87): error A2047: Multiple index registers
lr2.asm(94): error A2006: Phase error between passes

47842 + 459418 Bytes symbol space free

2 Warning Errors
5 Severe Errors

```

Рисунок 1 – трансляция программы

В результате трансляции программы получаем файл lr2.lst, содержащий в себе информацию, представляющую из себя два предупреждения и пять ошибок.

Ошибки, полученные в результате трансляции программы:

1. error A2052: Improper operand type – ошибка возникает из-за строки `mov mem3, [bx]`, операция `mov` должна использовать промежуточное значение для переноса значения из одной ячейки в другую.
2. error A2055: Illegal register value – ошибка возникает из-за строки `mov ax,matr[bx*4][di]`, т.к. регистр содержит недопустимое значение. Используется базовая индексная адресация, а масштабирование применяется только к 32 битным регистрам. Необходимо изначально изменить значение регистра до перевода информации.
3. error A2046: Multiple base registers – ошибка возникает из-за строки `mov ax,matr[bp+bx]`, т.к. нельзя использовать несколько базовых регистров для адресации.
4. error 2047: Multiple index registers – ошибка возникает из-за

строки `mov ax,matr[bp+di+si]`, т.к. не допускается использование нескольких индексных регистров для адресации.

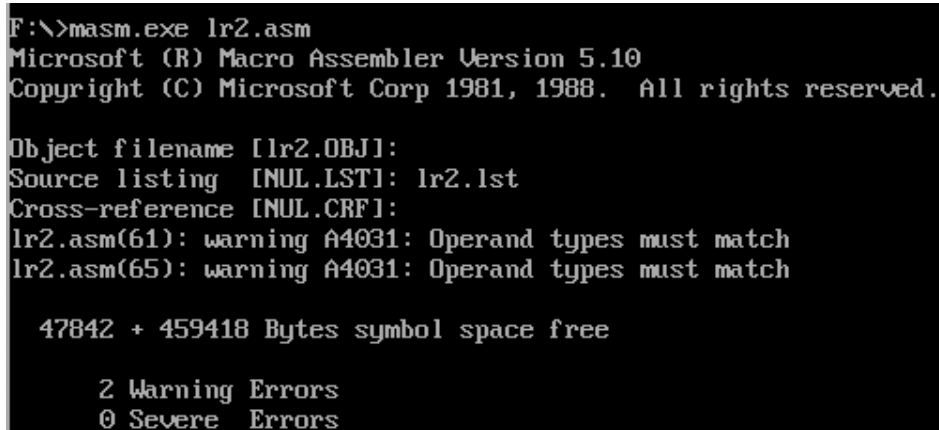
5. error A2006: Phase error between passes – ошибка компилятора из-а ошибок в предыдущих строках. Ошибка исчезает после исправления предыдущих ошибок.

Предупреждения, полученные в результате трансляции программы:

1. warning A4031: Operand types must match – возникает из-за строки `mov cx,vec2[di]`, т.к. происходит попытка перенести информацию из ячейки размером в 1 байт в регистр размером 2 байта.
2. warning A4031: Operand types must match – возникает из-за строки `mov cx,matr[bx][di]`, аналогично первому предупреждению.

В результате анализа полученных ошибок были закомментированы строчки кода, приводящие к поломке программы.

Произведена повторная трансляция программы.



```
F:\>masm.exe lr2.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lr2.OBJ]:
Source listing [NUL.LST]: lr2.lst
Cross-reference [NUL.CRF]:
lr2.asm(61): warning A4031: Operand types must match
lr2.asm(65): warning A4031: Operand types must match

47842 + 459418 Bytes symbol space free

2 Warning Errors
0 Severe Errors
```

Рисунок 2 – повторная трансляция программы

В результате был скомпилирован файл `lr2.exe` и создан файл `lr2.map`, был произведен запуск программы в режиме отладки.

#### 4. Запуск программы

```

AX 0000 SI 0000 CS 11B1 IP 0000 Stack +0 0000 FLAGS 0200
BX 0000 DI 0000 DS 119C
CX 00B6 BP 0000 ES 119C HS 119C +4 0000 OF DF IF SF ZF AF PF CF
DX 0000 SP 0018 SS 11AC FS 119C +6 0000 0 0 1 0 0 0 0 0

CMD >S
DS:0000 CD 20 CC 46 00 EA FD FF
DS:0008 AD DE ED 04 92 01 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 03 FF FF FF FF FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF 96 11 C4 FF
DS:0030 92 01 14 00 18 00 9C 11
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00

2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 CC 46 00 EA FD FF AD DE ED 04 92 01 00 00 .F....
DS:0010 18 01 10 01 18 01 92 01 03 FF FF FF FF FF FF FF ..
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF 96 11 C4 FF ..
DS:0030 92 01 14 00 18 00 9C 11 FF FF FF FF 00 00 00 00 ..
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..

1 Step 2StepProc 3Retrieve 4 Help 5Set BRK 6 7 up 8 dn 9 le 0 ri

```

Содержимое регистров до выполнения прогона программы: AX: 0000, BX: 0000, CX: 00B6, DX: 0000, CS: 11B1, DS: 119C, ES: 119C, SS: 11AC, IP: 0000

#### 5. Процесс отладки программы в *afd*

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	(IP) = 0000 (SP)=0018 Stack +0 0000 +2 0000 +4 0000 +6 0000	(IP) = 0001 (SP)=0016 Stack +0 119C +2 0000 +4 0000 +6 0000

0001	SUB AX, AX	2BC0	(IP) = 0001 (AX) = 0000	(IP) = 0003 (AX) = 0000
0003	PUSH AX	50	(IP) = 0003 (SP)=0016 Stack +0 119C +2 0000 +4 0000 +6 0000	(IP) = 0004 (SP)=0014 Stack +0 0000 +2 119C +4 0000 +6 0000
0004	MOV AX, 11AE	B8AE11	(IP) = 0004 (AX) = 0000	(IP) = 0007 (AX) = 11AE
0007	MOV DS, AX	8ED8	(IP) = 0007 (DS) = 119C	(IP) = 0009 (DS) = 11AE
0009	MOV AX, 01F4	B8F401	(IP) = 0009 (AX) = 11AE	(IP) = 000C (AX) = 01F4
000C	MOV CX, AX	8BC8	(IP) = 000C (CX)=00B6	(IP) = 000E (CX)=01F4
000E	MOV BL, 24	B324	(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	C7060200C EFF	(IP) = 0012 DS:0002 00 DS:0003 00	(IP) = 0018 DS:0002 CE DS:0003 FF

0018	MOV BX, 0006	BB0600	(IP) = 0018 (BX) = CE24	(IP) = 001B (BX) = 0006
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	(IP) = 001E (AX) = 01F4	(IP) = 0020 (AX) = 0112
0020	MOV AL, [BX+03]	8A4703	(IP) = 0020 (AX) = 010C	(IP) = 0023 (AX) = 0109
0023	MOV CX, [BX+03]	8B4F03	(IP) = 0023 (CX) = 01F4	(IP) = 0026 (CX) = 0509
0026	MOV DI, 0002	BF0200	(IP) = 0026 (DI) = 0000	(IP) = 0029 (DI) = 0002
0029	MOV AL, [DI+000E]	8A850E00	(IP) = 0029 (AX) = 0109	(IP) = 002D (AX) = 01D8
002D	MOV CX, [DI+000E]	8B8D0E00	(IP) = 002D (CX) = 0509	(IP) = 0031 (CX) = CED8
0031	MOV BX, 0003	BB0300	(IP) = 0031 (BX) = 0006	(IP) = 0034 (BX) = 0003
0034	MOV AL, [BX+DI+0016]	8A811600	(IP) = 0034 (AX) = 01D8	(IP) = 0038 (AX) = 01F9
0038	MOV CX, [BX+DI+0016]	8B891600	(IP) = 0038 (CX) = CED8	(IP) = 003C (CX) = FAF9

003C	MOV AX, 11AE	B8AE11	(IP) = 003C (AX) = 01F9	(IP) = 003F (AX) = 11AE
003F	MOV ES, AX	8ECO	(IP) = 003F (ES) = 119C	(IP) = 0041 (ES) = 11AE
0041	MOV AX, ES:[BX]	268B07	(IP) = 0041 (AX) = 11AE	(IP) = 0044 (AX) = 00FF
0044	MOV AX, 0000	B80000	(IP) = 0044 (AX) = 00FF	(IP) = 0047 (AX) = 0000
0047	MOV ES, AX	8ECO	(IP) = 0047 (ES) = 11AE	(IP) = 0049 (ES) = 0000
0049	PUSH DS	1E	(IP) = 0049 (SP) = 0014 Stack +0 0000 +2 119C +4 0000 +6 0000	(IP) = 004A (SP) = 0012 Stack +0 11AE +2 0000 +4 119C +6 0000
004A	POP ES	07	(IP) = 004A (SP) = 0012 Stack +0 11AE +2 0000 +4 119C +6 0000	(IP) = 004B (SP) = 0014 Stack +0 0000 +2 119C +4 0000 +6 0000
004B	MOV CX, ES:[BX-01]	268B4FFF	(IP) = 004B (CX) = FAF9	(IP) = 004F (CX) = FFCE



004F	XCHG	91	(IP) = 004F (AX) = 0000 (CX) = FFCE	(IP) = 0050 (AX) = FFCE (CX) = 0000
0050	MOV DI, 0002	BF0200	(IP) = 0050 (DI) = 0002	(IP) = 0053 (DI) = 0002
0053	MOV ES:[BX+DI], AX	268901	(IP) = 0053 DS:0005 00 DS:0006 0C	(IP) = 0056 DS:0005 CE DS:0006 FF
0056	MOV BP, SP	8BEC	(IP) = 0056 (BP) = 0000	(IP) = 0058 (BP) = 0014
0058	PUSH [0000]	FF360000	(IP) = 0058 (SP) = 0014 Stack +0 0000 +2 119C +4 0000 +6 0000	(IP) = 005C (SP) = 0012 Stack +0 01F4 +2 0000 +4 119C +6 0000
005C	PUSH [0002]	FF360200	(IP) = 005C (SP) = 0012 Stack +0 01F4 +2 0000 +4 119C +6 0000	(IP) = 0060 (SP) = 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 119C
0060	MOV BP, SP	8BEC	(IP) = 0060 (BP) = 0014	(IP) = 0062 (BP) = 0010

0062	MOV DX, [BP+02]	8B5602	(IP) = 0062 (DX) = 0000	(IP) = 0065 (DX) = 01F4
0065	RET FAR	CB	(IP) = 0065 (SP) = 0010 (CS) = 11B1 Stack +0 FFCE +2 01F4 +4 0000 +6 119C	(IP) = FFCE (SP) = 0014 (CS) = 01F4 Stack +0 0000 +2 119C +4 0000 +6 0000

Таблица 1

### **Выводы.**

В результате выполнения лабораторной работы были изучены виды адресации, были исправлены ошибки в коде программы lr2\_comp.asm, проведена трансляция программы и отладка с помощью afd.

## ПРИЛОЖЕНИЕ А

### Файл lr2\_comp.asm

```
; Учебная программа лабораторной работы №2 по дисциплине "Организация ЭВМ и
C";
;
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,[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]

;   ПРОВЕРКА 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
Main      ENDP
CODE      ENDS

```

END Main

SS:ASt  
ack

## Файл lr2.lst

Microsoft (R) Macro Assembler Version 5.10

10/8/23 12:18:13

Page 1-1

; Учебная программа лабораторной работы №2 по  
дисциплине "Организация ЭВМ и С";

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

; Стек программы

0000          AStack  SEGMENT  STACK
0000  000C[          DW 12 DUP(?)
      0000      ]
0018          AStack  ENDS

; Данные программы

0000          DATA    SEGMENT

; Директивы описания данных

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  28 32 D8 CE 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,
```

;
Головна
я
процеду
ра
0000

M
ain
PROC
FAR
0000
1E

push
DS
0001
2B C0

sub
AX,AX
0003
50

push
AX
0004
B8 ---
- R

mov
AX,DAT
A
0007
8E D8

mov
DS,AX

УРОВНЕ СМЕЩЕНИ	;	ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА
	Й	
	;	Регистровая адресация
0009 B8 01F4		mov ax,n1

Page 1-2

```

                                mov     cx,ax
                                dx,[bp
000C 8B C8                                ]+2
000E B3 24                                0065
0010 B7 CE                                CB
                                ; Прямая адресация
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,[bx]
                                ; Базированная адресация
0020 8A 47 03                        mov     al,[bx]+3
0023 8B 4F 03                        mov     cx,3[bx]
                                ; Индексированная 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(61): warning A4031: Operand types must match
                                ; Адресация с базированием 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(65): warning A4031: Operand types must match
                                ;mov     ax,matr[bx*4][di]

                                ; ПРОВЕРКА АДРЕСАЦИИ С УЧЕТОМ
СЕКМЕНТОВ
                                ; Переопределение 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
                                ; ----- вариант 3
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 сегмента стека
0058 FF 36 0000 R                    push    mem1
005C FF 36 0002 R                    push    mem2
0060 8B EC                                mov     bp,sp

```

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	

96 Source Lines  
96 Total Lines  
19 Symbols

47812 + 459448 Bytes symbol space free

2 Warning Errors  
0 Severe Errors



## Файл lr2.map

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

Program entry point at 0005:0000