

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»

Студентка гр. 1303

Куклина Ю.Н.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Разобраться в используемых режимах адресации и получаемых результатах на языке программирования Ассемблер.

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.

Исходные тексты программ и файлов листинга представлены в приложении «А»

Ход работы.

1. Изменены значения исходных данных(массивов) `vec1`, `vec2` и `matr`, согласно своему варианту.
2. Протранслирована программа с созданием файла диагностических сообщений; обнаруженные ошибки объяснены и закомментированы соответствующие операторы в тексте программы.

```
C:\>masm lb2.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lb2.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
lb2.asm(41): error A2052: Improper operand type
lb2.asm(48): warning A4031: Operand types must match
lb2.asm(52): warning A4031: Operand types must match
lb2.asm(53): error A2055: Illegal register value
lb2.asm(72): error A2046: Multiple base registers
lb2.asm(73): error A2047: Multiple index registers
lb2.asm(80): error A2006: Phase error between passes

49904 + 459406 Bytes symbol space free

2 Warning Errors
5 Severe Errors
```

- lr2_comp.asm(41): error A2052: Improper operand type

mov mem3,[bx]

Машинные команды не могут манипулировать одновременно двумя операндами, находящимися в оперативной памяти, то есть в команде только 1 операнд может указывать на ячейку памяти, другой операнд должен быть либо регистром, либо непосредственным значением.

- lr2_comp.asm(48): warning A4031: Operand types must match

mov cx,vec2[di]

Несоответствие типов операндов, cx – слово, а vec2[di] – размерность 1 байт

- lr2_comp.asm(52): warning A4031: Operand types must match

mov cx,matr[bx][di]

Несоответствие типов операндов, cx – слово, а matr[bx][di] – размерность 1 байт

- lr2_comp.asm(53): error A2055: Illegal register value

mov ax,matr[bx*4][di]

Ошибка возникает из-за масштабирования базового регистра bx, который нельзя использовать при индексной адресации с масштабированием.

- lr2_comp.asm(72): error A2046: Multiple base register

mov ax,matr[bp+bx]

В косвенной адресации с индексированием исполнительный адрес берется в виде суммы адресов, находящихся в базовом и индексном регистрах, а в данной строке оба регистра базовые.

- lr2_comp.asm(73): error A2047: Multiple index register

mov ax,matr[bp+di+si]

В непосредственной адресации с базированием и индексированием берется сумма базового и индексного регистра, к которым добавляется непосредственно фигурирующее в команде смещение, а в данной строке фигурируют 2 индексных регистра и 1 базовый.

- lr2_comp.asm(80): error A2006: Phase error between passes

Main ENDP

Ошибка говорит о том, что в функции main допущены ошибки.

3. Программа снова протранслирована и скомпонован загрузочный модуль.

```
C:\>masm lb2.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lb2.OBJ]:
Source listing [NUL.LST]: lb2
Cross-reference [NUL.CRF]:

47842 + 459418 Bytes symbol space free

0 Warning Errors
0 Severe Errors
```

4. Выполнена программа в пошаговом режиме под управлением отладчика с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды.

Протоколы пошагового исполнения каждой из программ под управлением отладчика.

Таблица 1- Протокол пошагового исполнения программы lb2.asm

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	(IP) = 0001 (SP) = 0016 STACK(+0) = 19F5
0001	SUB AX,AX	2BC0	(AX) = 0000	(AX) = 0000

			(IP) = 0001	(IP) = 0003
0003	PUSH AX	50	(IP) = 0003 (SP) = 0016 STACK(+0) = 19F5 STACK(+2) = 0000	(IP) = 0004 (SP) = 0014 STACK(+0) = 0000 STACK(+2) = 19F5
0004	MOV AX, 1A07	B8071A	(AX) = 0000 (IP) = 0004	(AX) = 1A07 (IP) = 0007
0007	MOV DS, AX	8ED8	(DS) = 19F5 (IP) = 0007	(DS) = 1A07 (IP) = 0009
0009	MOV AX, 01F4	B8F401	(AX) = 1A07 (IP) = 0009	(AX) = 01F4 (IP) = 000C
000C	MOV CX, AX	8BC8	(IP) = 000C (CX) = 00B0	(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],FFFCE	C7060200CEFF	(IP) = 0012	(IP) = 0018
0018	MOV BX,0006	BB0600	(IP) = 0012 (BX) = CE24	(IP) = 001B (BX) = 0006
001B	MOV [0000],AX	A30000	(IP) = 001B	(IP) = 001E
001E	MOV AL,[BX]	8A07	(AX) = 01F4 (IP) = 001E	(AX) = 0112 (IP) = 0020
0020	MOV AL,[BX+03]	8A4703	(AX) = 0112 (IP) = 0020	(AX) = 010F (IP) = 0023
0023	MOV CX,[BX+03]	8B4F03	(IP) = 0023 (CX) = 01F4	(IP) = 0026 (CX) = 0B0F
0026	MOV DI,0002	BF0200	(IP) = 0026 (DI) = 0000	(IP) = 0029 (DI) = 0002

0029	MOV AL,[000E+DI]	8A850E00	(AX) = 010F (IP) = 0029	(AX) = 01E2 (IP) = 002D
002D	MOV BX,0003	BB0300	(IP) = 002D (BX) = 0006	(IP) = 0030 (BX) = 0003
0030	MOV AL,[0016+BX+DI]	8A811600	(AX) = 01E2 (IP) = 0030	(AX) = 01FF (IP) = 0034
0034	MOV AX,1A07	B8071A	(AX) = 01FF (IP) = 0034	(AX) = 1A07 (IP) = 0037
0037	MOV ES,AX	8EC0	(ES) = 19F5 (IP) = 0037	(ES) = 1A07 (IP) = 0039
0039	MOV AX,ES:[BX]	268B07	(AX) = 1A07 (IP) = 0039	(AX) = 00FF (IP) = 003C
003C	MOV AX,0000	B80000	(AX) = 00FF (IP) = 003C	(AX) = 0000 (IP) = 003F
003F	MOV ES,AX	8EC0	(ES) = 1A07 (IP) = 003F	(ES) = 0000 (IP) = 0041
0041	PUSH DS	1E	(IP) = 0041 (SP) = 0014 STACK(+0) = 0000 STACK(+2) = 19F5 STACK(+4) = 0000	(IP) = 0042 (SP) = 0012 STACK(+0) = 1A07 STACK(+2) = 0000 STACK(+4) = 19F5
0042	POP ES	07	(IP) = 0042 (SP) = 0012 (ES) = 0000 STACK(+0) = 1A07 STACK(+2) = 0000 STACK(+4) = 19F5	(IP) = 0043 (SP) = 0014 (ES) = 1A07 STACK(+0) = 0000 STACK(+2) = 19F5 STACK(+4) = 0000

0043	MOV CX,ES:[BX—01]	268B4FFF	(IP) = 0043 (CX) = 0B0F	(IP) = 0047 (CX) = FFCE
0047	XCHG AX,CX	91	(IP) = 0047 (CX) = FFCE (AX) = 0000	(IP) = 0048 (CX) = 0000 (AX) = FFCE
0048	MOV DI 0002	BF0200	(IP) = 0048 (DI) = 0002	(IP) = 004B (DI) = 0002
004B	MOV ES:[BX+DI], AX	268901	(IP) = 004B	(IP) = 004E
004E	MOV BP,SP	8BEC	(IP) = 004E (BP) = 0000	(IP) = 0050 (BP) = 0014
0050	PUSH [0000]	FF360000	(IP) = 0050 (SP) = 0014 STACK(+0) = 0000 STACK(+2) = 19F5 STACK(+4) = 0000	(IP) = 0054 (SP) = 0012 STACK(+0) = 01F4 STACK(+2) = 0000 STACK(+4) = 19F5
0054	PUSH [0002]	FF360200	(IP) = 0054 (SP) = 0012 STACK(+0) = 01F4 STACK(+2) = 0000 STACK(+4) = 19F5 STACK(+6) = 0000	(IP) = 0058 (SP) = 0010 STACK(+0) = FFCE STACK(+2) = 01F4 STACK(+4) = 0000 STACK(+6) = 19F5
0058	MOV BP,SP	8BEC	(IP) = 0058 (BP) = 0014	(IP) = 005A (BP) = 0010
005A	MOV DX,[BP+02]	8B5602	(IP) = 005A (DX) = 0000	(IP) = 005D (DX) = 01F4

005D	RET FAR 0002	CA0200	(IP) = 0058 (SP) = 0010 (CS) = 1A0A STACK(+0) = FFCE STACK(+2) = 01F4 STACK(+4) = 0000 STACK(+6) = 19F5	(IP) = FFCE (SP) = 0016 (CS) = 01F4 STACK(+0) = 19F5 STACK(+2) = 0000 STACK(+4) = 0000 STACK(+6) = 0000
------	--------------	--------	---	---

Выводы.

В ходе выполнения лабораторной работы были получены основные навыки программирования на ассемблере, изучены основные режимы адресации памяти.

Приложение А. Коды программы.

Исходный текст программы lb2.asm.

lb2.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 18,17,16,15,11,12,13,14
vec2 DB 30,40,-30,-40,10,20,-10,-20
matr DB -4,-3,1,2,-2,-1,3,4,5,6,7,8,-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]//ошибка
; Базированная адресация
    mov al,[bx]+3
    mov cx,3[bx]
; Индексная адресация (косвенная со смещением)
    mov di,ind
    mov al,vec2[di]
    mov cx,vec2[di]//ошибка
; Адресация с базированием и индексированием
    mov bx,3
```

```

        mov al,matr[bx][di]
        mov cx,matr[bx][di]//ошибка
        mov ax,matr[bx*4][di]//ошибка
; 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]//ошибка
        mov ax,matr[bp+di+si]//ошибка
; Использование сегмента стека
        push mem1
        push mem2
        mov bp,sp
        mov dx,[bp]+2
        ret 2
Main ENDP//ошибка вроде
CODE ENDS
END Main

```

Текст файлов диагностических сообщений lb2.lst.

Microsoft (R) Macro Assembler Version 5.10
14:41:4

10/11/22

Page 1-

1

```

= 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(?)
        ?????
    ]

```

```

0018          AStack ENDS
          ; Данные программы
0000          DATA SEGMENT
          ; Директивы описания данных
          x

0000 0000          mem1 DW 0
0002 0000          mem2 DW 0
0004 0000          mem3 DW 0
0006 12 11 10 0F 0B 0C  vec1 DB 18,17,16,15,11,12,13,14
          0D 0E
000E 1E 28 E2 D8 0A 14  vec2 DB 30,40,-30,-40,10,20,-10,-20
          F6 EC
0016 FC FD 01 02 FE FF  matr DB -4,-3,1,2,-2,-1,3,4,5,6,7,8,-8,-7,-
6,-5
          03 04 05 06 07 08
          F8 F9 FA FB

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 РЕЖИМОВ АДРЕСА
          ИЛИ НА УРОВНЕ СМЕЩЕНИЙ
          ; Регистровая адресация
0009 B8 01F4          mov ax,n1
000C 8B C8          mov cx,ax
000E B3 24          mov bl,EOL
0010 B7 CE          mov bh,n2
          ; 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]
          ; Индексная адресация
Microsoft (R) Macro Assembler Version 5.10
14:41:4
2
          0026 BF 0002          mov di,ind
          0029 8A 85 000E R      mov al,vec2[di]
          ;mov cx,vec2[di]
          ; Адресация с базированием
          и индексированием

```

10/11/22

Page 1-

```

002D BB 0003          mov bx,3
0030 8A 81 0016 R     mov al,matr[bx][di]
                      ; mov cx,matr[bx][di]
                      ;mov ax,matr[bx*4][di]
                      ; ПРОВЕРКА РЕЖИМОВ АДРЕСА
                      ИИ С УЧЕТОМ СЕГМЕНТОВ
                      ; Переопределение сегмент
                      а
                      ; ----- вариант 1
0034 B8 ---- R       mov ax, SEG vec2
0037 8E C0           mov es, ax
0039 26: 8B 07       mov ax, es:[bx]
003C B8 0000         mov ax, 0
                      ; ----- вариант 2
003F 8E C0           mov es, ax
0041 1E             push ds
0042 07             pop es
0043 26: 8B 4F FF     mov cx, es:[bx-1]
0047 91             xchg cx,ax
                      ; ----- вариант 3
0048 BF 0002         mov di,ind
004B 26: 89 01       mov es:[bx+di],ax
                      ; ----- вариант 4
004E 8B EC           mov bp,sp
                      ; mov ax,matr[bp+bx]
                      ;mov ax,matr[bp+di+si]
                      ; Использование сегмента
                      стека
0050 FF 36 0000 R     push mem1
0054 FF 36 0002 R     push mem2
0058 8B EC           mov bp,sp
005A 8B 56 02       mov dx,[bp]+2
005D CA 0002         ret 2
0060                Main ENDP
0060                CODE ENDS
                      END Main
Microsoft (R) Macro Assembler Version 5.10      10/11/22
14:41:4

```

Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	0060	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	=
0060					
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	1b2			
@VERSION	TEXT	510			

82 Source Lines
82 Total Lines
19 Symbols

47842 + 459418 Bytes symbol space free

0 Warning Errors
0 Severe Errors