**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.**

**Вариант №4**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6382 |  | Сергеев А.Д. |
| Преподаватель |  | Кирьянчиков А.В, |

Санкт-Петербург

2019

**Цель работы.**

Изучить основные принципы трансляции, отладки и выполнения программ на языке Ассемблера.

**Задание.**

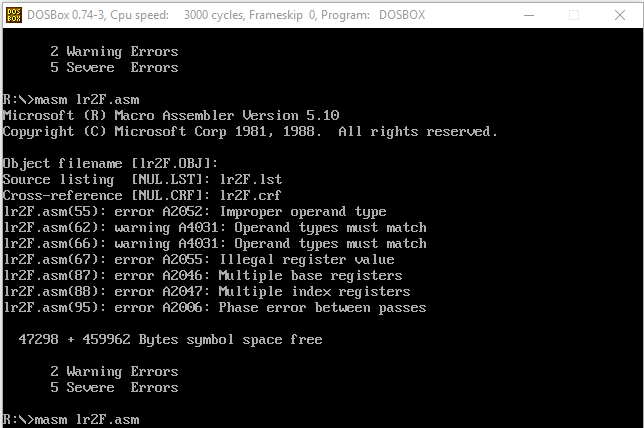
Лабораторная работа 2 предназначена для изучения режимов адресации, использует готовую программу на Ассемблере, которая в автоматическом режиме выполняться не должна, так как не имеет самостоятельного функционального назначения, а только тестирует режимы адресации. Поэтому ее выполнение должно производиться 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 предыдущей лабораторной работы. На защите студенты должны уметь объяснить результат выполнения каждой команды.

**Ход работы.**

1. Изменил набор значений исходных данных (массивов) vec1, vec2 и matr, согласно своему варианту.
2. Протранслировал программу с созданием файла диагностических сообщений (см. ниже). Объяснил обнаруженные ошибки и закомментировал соответствующие операторы в тексте программы.



1) Ошибка lab2F.asm (55): error A2052: Improper operand type (Неверный тип операнда)

**Строка 55**: mov mem3, [bx]

Тип операнда, нельзя читать из памяти и писать в память одной командой. Сначала нужно перенести информацию в регистр, а потом из регистра перенести в память.

Предложенное исправление:

mov ax,[bx]

mov mem3,ax

2) Предупреждение lab2F.asm (62): warning A4031: Operand types must match

**Строка 62**: mov сx, vec2[di]

Несоответствие типов операндов, cx – 2 байта, элементы vec – 1 байт.

Предложенное исправление:

mov сl, vec2[di]

3) Предупреждение lab2F.asm (66): warning A4031: Operand types must match

**Строка 66:** mov cx, matr[bx][di]

Несоответствие типов операндов, cx-2 байта, элементы matr – 1 байт.

Предложенное исправление:

mov cl, matr[bx][di]

4)Ошибка lab2F.asm (67): error A2055: Illegal register value

**Строка 67**: mov ax,matr[bx\*4][di]

Здесь используется базово-индексная адресация. При данном типе адресации надо сначала изменить значение регистра, затем уже переводить информацию.

5) Ошибка lab2F.asm (87): error A2046: Multiple base registers (несколько индексных регистров).

**Строка 87**: mov ax,matr[bp+bx]

Нельзя складывать регистры bp и bx. В базированной адресации необходимо указывать базовый регистр, затем производить смещение с помощью индексного. Так как здесь оба базовые, надо сначала сложить значения регистров, и затем уже передавать информацию указателю из одного регистра.

Предложенное исправление:

add bp,bx

mov al,matr[bp]

6) Ошибка lab2F.asm (88): error A2047: Multiple index registers (несколько индексных регистров)

**Строка 88**: mov ax,matr[bp+di+si]

Нельзя складывать регистры di и si, необходимо сначала в регистр di занести общую сумму, затем уже производить смещение.

Предложенное исправление:

add di,si

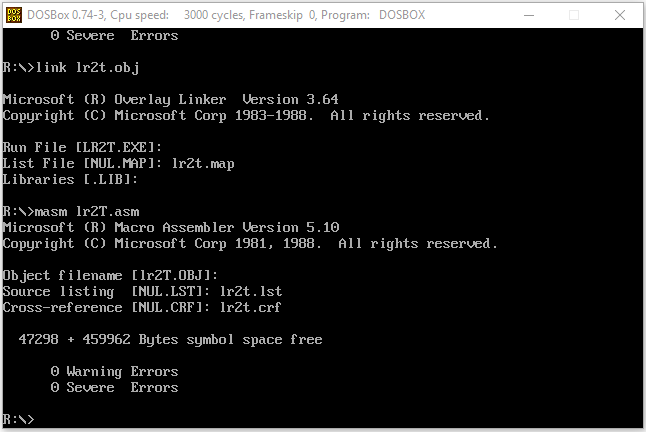
mov al,matr[bp+di]

7) Ошибка lab2F.asm (95): error A2006: Phase error between passes

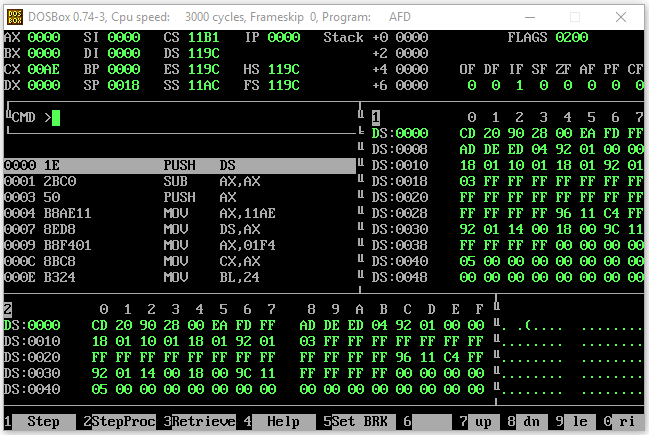
**Строка 95:** Main ENDP

Данная ошибка свидетельствует о том, что в функции main содержатся ошибки.

3. Переименовав файл lab2F.asm, в котором были закомментированы ошибки в lab2T.asm, снова протранслировал программу и скомпоновал загрузочный модуль.



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



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Символьный код команды | 16-ричный код команды | Содержимое регистров и ячеек памяти | |
| До выполнения | После выполнения |
| 0000 | PUSH DS | 1E | (SP)=0018  (IP)=0000 | (SP)=0016  (IP)=0001 |
| 0001 | SUB AX. AX | 2BC0 | (AX)=0000  (IP)=0001 | (IP)=0003  (AX)=0000 |
| 0003 | PUSH AX | 50 | (SP)= 0016  (IP)= 0003  (AX)=0000 | (SP)= 0014  (IP)= 0004  (AX)=0000 |
| 0004 | MOV AX, 11 AE | B8AE11 | (AX)=0000  (IP)=0004 | (AX)= 11AE  (IP)=0007 |
| 0007 | MOV DS,AX | 8ED8 | (DS)=119С  (IP)= 0007  (AX)= 11AE | (DS)=11AE  (IP)= 0009  (AX)= 11AE |
| 0009 | MOV AX, 01F4 | B8F401 | (AX)=11AE  (IP)=0009 | (AX)= 01F4  (IP)= 000C |
| 000C | MOV CX,AX | 8BC8 | (IP)=000C  (CX)=00BO | (IP)=000E  (CX)=01F4 |
| 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  (IP)=001E | (AX)=0108  (IP)=0020 |
| 0020 | MOV AL,[BX+03] | 8A4703 | (IP) = 0020  (AX)= 0108 | (IP)= 0023  (AX) = 0105 |
| 0023 | MOV CX, [BX+03] | 8B4F03 | (CX) = 01F4  (IP) = 0023 | (CX) = 120E  (IP) = 0026 |
| 0026 | MOV DI, 0002 | BF0200 | (DI) = 0000  (IP) = 0026 | (DI) = 0002  (IP) = 0029 |
| 0029 | MOV AL, [DI+ 000E] | 8A850E00 | (AX) = 0105  (IP) = 0029 | (AX)= 01F6  (IP)= 002D |
| 002D | MOV CX, [DI+ 000E] | 8B8D0E00 | (CX) = 120E  (IP) = 002D | (CX) = ECF6  (IP) = 0030 |
| 0031 | MOV BX, 0003 | BB03000 | (IP) = 0031  (BX) = 0006 | (IP) = 0034  (BX) = 0003 |
| 0034 | MOV AL, [BX+DI+0016] | 8A811600 | (IP) = 0034  (AX) = 01F6 | (IP) = 0038  (AX) = 0104 |
| 0038 | MOV CX, [BX+DI+0016] | 8B891600 | (IP) = 0038  (CX) = ECF6 | (IP) = 003C  (CX) = FE04 |
| 003C | MOV AX, 11 AE | B8AE11 | (AX) = 0104  (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 | (IP)= 0049  (SP)= 0014 | (IP)= 004A  (SP)= 0012 |
| 004A | POP ES | 07 | (SP)= 0012  (ES)=0000  (IP)= 004A | (SP) = 0014  (ES)=11AE  (IP)= 004B |
| 004B | MOV CX, ES:[BX—01] | 268B4FFF | (CX) = FE04  (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 | (IP) = 0056 |
| 0056 | MOV BP, SP | 8BEC | (IP) = 0056  (BP) = 0000 | (IP) = 0058  (BP) = 0014 |
| 0058 | PUSH [0000] | FF360000 | (IP) = 0058  (SP)=0014 | (IP) = 005C  (SP)=0012 |
| 005C | PUSH [0002] | FF360200 | (IP) = 005C  (SP) = 0012 | (IP) = 0060  (SP) = 0010 |
| 0060 | MOV BP, SP | 8BEC | (IP) = 0060  (BP) = 0014 | (IP) = 0062  (BP) = 0010 |
| 0062 | MOX DX, [BP+02] | 8B5602 | (IP) = 0062  (DX) = 0000 | (IP) = 0065  (DX) = 01F4 |
| 0065 | RET FAR 0002 | CA0200 | (IP) = 0065  (SP) = 0010  (CS)=11B1 | (IP) = FFCE  (SP)= 0016  (CS)=01F4 |

Изменение стека

|  |  |
| --- | --- |
| **Адрес команды** | **Стек** |
| Начальное положение | +0 0000  +2 0000  +4 0000  +6 0000 |
| 0000 | +0 119С  +2 0000  +4 0000  +6 0000 |
| 0003 | +0 0000  +2 119С  +4 0000  +6 0000 |
| 0049 | +0 11AE  +2 0000  +4 119C  +6 0000 |
| 004A | +0 0000  +2 119C  +4 0000  +6 0000 |
| 0058 | +0 01F4  +2 0000  +4 119C  +6 0000 |
| 005C | +0 FFCE  +2 01F4  +4 0000  +6 119C |
| 0065 | +0 119C  +2 0000  +4 0000  +6 0000 |

Вывод: В ходе выполнения лабораторной работы были получены основные навыки программирования на ассемблере, изучены основные режимы адресации памяти. Результатом выполнения работы стала программа lab2t.exe.

**Приложение.**

***1)Текст файла lab2T.asm (исходный код без ошибок)***

; 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 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 ax,[bx] ;suggested ;mov mem3,ax ;suggested  
; Базированная адресация  
mov al,[bx]+3  
mov cx,3[bx]  
; Индексированная адресация  
mov di,ind  
mov al,vec2[di]  
;mov cl,vec2[di] ;suggested  
; Адресация с базированием и индексированием  
mov bx,3  
mov al,matr[bx][di]  
;mov cl,matr[bx][di] ;suggested  
;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  
;add bp,bx ;suggested ;mov al,matr[bp] ;suggested  
;add di,si ;suggested ;mov al,matr[bp+di] ;suggested  
; Использование сегмента стека  
push mem1  
push mem2  
mov bp,sp  
mov dx,[bp]+2  
ret  
Main ENDP  
CODE ENDS  
END Main

***2)Текст файла lab2T.lst (листинг без ошибок)***

Microsoft (R) Macro Assembler Version 5.10 10/23/19 03:42:3

Page 1-1

1 ; Учебная программа лабораторной работ

ы №2 по дисциплине "Архитектура компьют

ера"

2 ;

3 ;

4 = 0024 EOL EQU '$'

5 = 0002 ind EQU 2

6 = 01F4 n1 EQU 500

7 =-0032 n2 EQU -50

8

9 ; Стек программы

10

11 0000 AStack SEGMENT STACK

12 0000 000C[ DW 12 DUP(?)

13 ????

14 ]

15

16 0018 AStack ENDS

17

18 ; Данные программы

19

20 0000 DATA SEGMENT

21

22 ; Директивы описания данных

23

24 0000 0000 mem1 DW 0

25 0002 0000 mem2 DW 0

26 0004 0000 mem3 DW 0

27 0006 0C 0B 0A 09 05 06 vec1 DB 12,11,10,9,5,6,7,8

28 07 08

29 000E D8 CE 28 32 EC E2 vec2 DB -40,-50,40,50,-20,-30,2

0,30

30 14 1E

31 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

32 FA FB 01 02 03 04

33 FC FD FE FF

34

35 0026 DATA ENDS

36

37 ; Код программы

38

39 0000 CODE SEGMENT

40 ASSUME CS:CODE, DS:DATA, SS:A

Stack

41

42 ; Головная процедура

43 0000 Main PROC FAR

44 0000 1E push DS

45 0001 2B C0 sub AX,AX

46 0003 50 push AX

47 0004 B8 —— R mov AX,DATA

48 0007 8E D8 mov DS,AX

49

Microsoft (R) Macro Assembler Version 5.10 10/23/19 03:42:3

Page 1-2

50 ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ

СМЕЩЕНИЙ

51 ; Регистровая адресация

52 0009 B8 01F4 mov ax,n1

53 000C 8B C8 mov cx,ax

54 000E B3 24 mov bl,EOL

55 0010 B7 CE mov bh,n2

56 ; Прямая адресация

57 0012 C7 06 0002 R FFCE mov mem2,n2

58 0018 BB 0006 R mov bx,OFFSET vec1

59 001B A3 0000 R mov mem1,ax

60 ; Косвенная адресация

61 001E 8A 07 mov al,[bx]

62 ;mov ax,[bx] ;suggested ;mov

mem3,ax ;suggested

63 ; Базированная адресация

64 0020 8A 47 03 mov al,[bx]+3

65 0023 8B 4F 03 mov cx,3[bx]

66 ; Индексированная адресация

67 0026 BF 0002 mov di,ind

68 0029 8A 85 000E R mov al,vec2[di]

69 ;mov cl,vec2[di] ;suggested

70 ; Адресация с базированием и индексиро

ванием

71 002D BB 0003 mov bx,3

72 0030 8A 81 0016 R mov al,matr[bx][di]

73 ;mov cl,matr[bx][di] ;sugges

ted

74 ;mov ax,matr[bx\*4][di]

75

76 ; ПРОВЕРКА АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТО

В

77 ; Переопределение сегмента

78 ; —--— вариант 1

79 0034 B8 —— R mov ax, SEG vec2

80 0037 8E C0 mov es, ax

81 0039 26: 8B 07 mov ax, es:[bx]

82 003C B8 0000 mov ax, 0

83 ; —--— вариант 2

84 003F 8E C0 mov es, ax

85 0041 1E push ds

86 0042 07 pop es

87 0043 26: 8B 4F FF mov cx, es:[bx-1]

88 0047 91 xchg cx,ax

89 ; —--— вариант 3

90 0048 BF 0002 mov di,ind

91 004B 26: 89 01 mov es:[bx+di],ax

92 ; —--— вариант 4

93 004E 8B EC mov bp,sp

94 ;add bp,bx ;suggested ;mov

al,matr[bp] ;suggested

95 ;add di,si ;suggested ;mov

al,matr[bp+di] ;suggested

96 ; Использование сегмента стека

Microsoft (R) Macro Assembler Version 5.10 10/23/19 03:42:3

Page 1-3

97 0050 FF 36 0000 R push mem1

98 0054 FF 36 0002 R push mem2

99 0058 8B EC mov bp,sp

100 005A 8B 56 02 mov dx,[bp]+2

101 005D CB ret

102 005E Main ENDP

103 005E CODE ENDS

104 END Main

Microsoft (R) Macro Assembler Version 5.10 10/23/19 03:42:3

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

ASTACK . . . . . . . . . . . . . 0018 PARA STACK

CODE . . . . . . . . . . . . . . 005E 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 = 005E

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 lr2T

@VERSION . . . . . . . . . . . . TEXT 510

97 Source Lines

97 Total Lines

19 Symbols

47298 + 459962 Bytes symbol space free

0 Warning Errors

0 Severe Errors

***2)Текст файла lab2F.lst (листинг с ошибками)***

Microsoft (R) Macro Assembler Version 5.10 10/23/19 03:38:1

Page 1-1

1 ; Учебная программа лабораторной работ

ы №2 по дисциплине "Архитектура компьют

ера"

2 ;

3 ;

4 = 0024 EOL EQU '$'

5 = 0002 ind EQU 2

6 = 01F4 n1 EQU 500

7 =-0032 n2 EQU -50

8

9 ; Стек программы

10

11 0000 AStack SEGMENT STACK

12 0000 000C[ DW 12 DUP(?)

13 ????

14 ]

15

16 0018 AStack ENDS

17

18 ; Данные программы

19

20 0000 DATA SEGMENT

21

22 ; Директивы описания данных

23

24 0000 0000 mem1 DW 0

25 0002 0000 mem2 DW 0

26 0004 0000 mem3 DW 0

27 0006 0C 0B 0A 09 05 06 vec1 DB 12,11,10,9,5,6,7,8

28 07 08

29 000E D8 CE 28 32 EC E2 vec2 DB -40,-50,40,50,-20,-30,2

0,30

30 14 1E

31 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

32 FA FB 01 02 03 04

33 FC FD FE FF

34

35 0026 DATA ENDS

36

37 ; Код программы

38

39 0000 CODE SEGMENT

40 ASSUME CS:CODE, DS:DATA, SS:A

Stack

41

42 ; Головная процедура

43 0000 Main PROC FAR

44 0000 1E push DS

45 0001 2B C0 sub AX,AX

46 0003 50 push AX

47 0004 B8 —— R mov AX,DATA

48 0007 8E D8 mov DS,AX

49

Microsoft (R) Macro Assembler Version 5.10 10/23/19 03:38:1

Page 1-2

50 ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ

СМЕЩЕНИЙ

51 ; Регистровая адресация

52 0009 B8 01F4 mov ax,n1

53 000C 8B C8 mov cx,ax

54 000E B3 24 mov bl,EOL

55 0010 B7 CE mov bh,n2

56 ; Прямая адресация

57 0012 C7 06 0002 R FFCE mov mem2,n2

58 0018 BB 0006 R mov bx,OFFSET vec1

59 001B A3 0000 R mov mem1,ax

60 ; Косвенная адресация

61 001E 8A 07 mov al,[bx]

62 mov mem3,[bx]

lr2F.asm(55): error A2052: Improper operand type

63 ; Базированная адресация

64 0020 8A 47 03 mov al,[bx]+3

65 0023 8B 4F 03 mov cx,3[bx]

66 ; Индексированная адресация

67 0026 BF 0002 mov di,ind

68 0029 8A 85 000E R mov al,vec2[di]

69 002D 8B 8D 000E R mov cx,vec2[di]

lr2F.asm(62): warning A4031: Operand types must match

70 ; Адресация с базированием и индексиро

ванием

71 0031 BB 0003 mov bx,3

72 0034 8A 81 0016 R mov al,matr[bx][di]

73 0038 8B 89 0016 R mov cx,matr[bx][di]

lr2F.asm(66): warning A4031: Operand types must match

74 003C 8B 85 0022 R mov ax,matr[bx\*4][di]

lr2F.asm(67): error A2055: Illegal register value

75

76 ; ПРОВЕРКА АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТО

В

77 ; Переопределение сегмента

78 ; —--— вариант 1

79 0040 B8 —— R mov ax, SEG vec2

80 0043 8E C0 mov es, ax

81 0045 26: 8B 07 mov ax, es:[bx]

82 0048 B8 0000 mov ax, 0

83 ; —--— вариант 2

84 004B 8E C0 mov es, ax

85 004D 1E push ds

86 004E 07 pop es

87 004F 26: 8B 4F FF mov cx, es:[bx-1]

88 0053 91 xchg cx,ax

89 ; —--— вариант 3

90 0054 BF 0002 mov di,ind

91 0057 26: 89 01 mov es:[bx+di],ax

92 ; —--— вариант 4

93 005A 8B EC mov bp,sp

94 005C 3E: 8B 86 0016 R mov ax,matr[bp+bx]

lr2F.asm(87): error A2046: Multiple base registers

95 0061 3E: 8B 83 0016 R mov ax,matr[bp+di+si]

lr2F.asm(88): error A2047: Multiple index registers

96 ; Использование сегмента стека

97 0066 FF 36 0000 R push mem1

98 006A FF 36 0002 R push mem2

99 006E 8B EC mov bp,sp

100 0070 8B 56 02 mov dx,[bp]+2

Microsoft (R) Macro Assembler Version 5.10 10/23/19 03:38:1

Page 1-3

101 0073 CB ret

102 0074 Main ENDP

lr2F.asm(95): error A2006: Phase error between passes

103 0074 CODE ENDS

104 END Main

Microsoft (R) Macro Assembler Version 5.10 10/23/19 03:38:1

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

ASTACK . . . . . . . . . . . . . 0018 PARA STACK

CODE . . . . . . . . . . . . . . 0074 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 = 0074

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 lr2F

@VERSION . . . . . . . . . . . . TEXT 510

97 Source Lines

97 Total Lines

19 Symbols

47298 + 459962 Bytes symbol space free

2 Warning Errors

5 Severe Errors

***2)Текст файла lab2T.map***

Start Stop Length Name Class

00000H 00017H 00018H ASTACK

00020H 00045H 00026H DATA

00050H 000ADH 0005EH CODE

Program entry point at 0005:0000