**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. 0383 |  | Сабанов П.А. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2021

**Цель работы.**

Изучить работу программы lr2\_comp.asm, написанной на языке ассемблера masm, проанализировать и исправить ошибки в ней, изучить режимы адресации и формирования исполнительного адреса.

**Текст задания.**

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.

**Порядок выполнения 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. 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.

**Ход работы.**

Вариант 5.

Значения массивов:

│ vec1 │ 11, 12, 13, 14, 18, 17, 16, 15 |

│ vec2 │ 10, 20, -10, -20, 30, 40, -30, -40 |

│ matr │ 1, 2, -4, -3, 3, 4, -2, -1, 5, 6, -8, -7, 7, 8, -6, -5 |

Сообщение компилятора при компиляции программы с ошибками:

Microsoft (R) Macro Assembler Version 5.10

Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

lr2\_comp.ASM(42): error A2052: Improper operand type

lr2\_comp.ASM(49): warning A4031: Operand types must match

lr2\_comp.ASM(53): warning A4031: Operand types must match

lr2\_comp.ASM(54): error A2055: Illegal register value

lr2\_comp.ASM(73): error A2046: Multiple base registers

lr2\_comp.ASM(74): error A2047: Multiple index registers

lr2\_comp.ASM(81): error A2006: Phase error between passes

47262 + 459998 Bytes symbol space free

2 Warning Errors

5 Severe Errors

Ошибка в строчке 42: попытка записать данные из одной ячейки оперативной памяти в другую ячеку оперативной памяти одной командой.

Ошибка в строчке 54: неправильная адресация.

Ошибка в строчке 73: использование при адресации вместе регистров bp и bx.

Ошибка в строчке 74: использование при адресации вместе регистров di и si.

Ошибка в строчке 81 пропадает при закомментировании строчек, содержащих предыдущие ошибки.

Также в коде есть ошибка, из-за которой программа не прекращает своё выполнение. Вместо команды ret 2 в конце программы нужно написать две команды: add sp, 4 и ret.

Сообщение компилятора после компиляции исправленной программы:

Microsoft (R) Macro Assembler Version 5.10

Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

lr2\_comp.ASM(49): warning A4031: Operand types must match

lr2\_comp.ASM(53): warning A4031: Operand types must match

47800 + 461507 Bytes symbol space free

2 Warning Errors

0 Severe Errors

**Выводы.**

Была изучена работа программы lr2\_comp.asm, написанной на языке ассемблера masm, были проанализированы и исправлены ошибки в ней, были изучены режимы адресации и форматирования исполнительного адреса.

Протокол

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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  (SS) = 1A05  [SS+SP] = 0000 | (IP) = 0001  (SP) = 0016  (SS) = 1A05  [SS+SP] = 19F5 |
| 0001 | SUB AX, AX | 2BC0 | (AX) = 0000  (IP) = 0001 | (AX) = 0000  (IP) = 0003 |
| 0003 | PUSH AX | 50 | (AX) = 0000  (IP) = 0003  (SP) = 0016  (SS) = 1A05  [SS+SP] = 19F5  [SS+SP+2] = 0000 | (AX) = 0000  (IP) = 0004  (SP) = 0014  (SS) = 1A05  [SS+SP] = 0000  [SS+SP+2] = 19F5 |
| 0004 | MOV AX, 1A07 | B8071A | (AX) = 0000  (IP) = 0004 | (AX) = 1A07  (IP) = 0007 |
| 0007 | MOV DS, AX | 8ED8 | (AX) = 1A07  (DS) = 19F5  (IP) = 0007 | (AX) = 1A07  (DS) = 1A07  (IP) = 0009 |
| 0009 | MOV AX, 01F4 | B8F401 | (AX) = 1A07  (IP) = 0009 | (AX) = 01F4  (IP) = 000C |
| 000C | MOV CX, AX | 8BC8 | (AX) = 01F4  (CX) = 00B9  (IP) = 000C | (AX) = 01F4  (CX) = 01F4  (IP) = 000E |
| 000E | MOV BL, 24 | B324 | (BL) = 00 (IP) = 000E | (BL) = 24  (IP) = 0010 |
| 0010 | MOV BH, CE | B7CE | (BH) = 00  (IP) = 0010 | (BH) = CE  (IP) = 0012 |
| 0012 | MOV [0002], FFCE | C7060200CEFF | (IP) = 0012  [DS+0002] = 0000 | (IP) = 0018  [DS+0002] = CEFF |
| 0018 | MOV BX, 0006 | BB0600 | (BX) = CE24  (IP) = 0018 | (BX) = 0006  (IP) = 001B |
| 001B | MOV [0000], AX | A30000 | (AX) = 01F4  (IP) = 001B  [DS] = 0000 | (AX) = 01F4  (IP) = 001E  [DS] = F401 |
| 001E | MOV AL, [BX] | 8A07 | (AL) = F4  (BX) = 0006  (IP) = 001E  [DS+BX] = 0B | (AL) = 0B  (BX) = 0006  (IP) = 0020  [DS+BX] = 0B |
| 0020 | MOV AL, [BX+03] | 8A4703 | (AL) = 0B  (BX) = 0006  (IP) = 0020  [DS+BX+3] = 0E | (AL) = 0E  (BX) = 0006  (IP) = 0023  [DS+BX+3] = 0E |
| 0023 | MOV CX, [BX+03] | 8B4F03 | (BX) = 0006  (CX) = 01F4  (IP) = 0023  [DS+BX+3] = 0E12 | (BX) = 0006  (CX) = 120E  (IP) = 0026  [DS+BX+3] = 0E12 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0026 | MOV DI, 0002 | BF0200 | (DI) = 0000  (IP) = 0026 | (DI) = 0002  (IP) = 0029 |
| 0029 | MOV AL, [000E+DI] | 8A850E00 | (AL) = 0E  (DI) = 0002  (IP) = 0029  [DS+DI+000E] = F6 | (AL) = F6  (DI) = 0002  (IP) = 002D  [DS+DI+000E] = F6 |
| 002D | MOV CX,  [000E+DI] | 8B8D0E00 | (CX) = 120E  (DI) = 0002  (IP) = 002D  [DS+DI+000E] = F6EC | (CX) = ECF6  (DI) = 0002  (IP) = 0031  [DS+DI+000E] = F6EC |
| 0031 | MOV BX, 0003 | BB0300 | (BX) = 0006  (IP) = 0031 | (BX) = 0003  (IP) = 0034 |
| 0034 | MOV AL, [0016+BX+DI] | 8A811600 | (AL) = F6  (BX) = 0003  (DI) = 0002  (IP) = 0034  [DS+DI+BX+0016] = 04 | (AL) = 04  (BX) = 0003  (DI) = 0002  (IP) = 0038  [DS+DI+BX+0016] = 04 |
| 0038 | MOV CX, [0016+BX+DI] | 8B891600 | (CX) = ECF6  (BX) = 0003  (DI) = 0002  (IP) = 0038  [DS+DI+BX+0016] = 04FE | (CX) = FE04  (BX) = 0003  (DI) = 0002  (IP) = 003C  [DS+DI+BX+0016] = 04FE |
| 003C | MOV AX, 1A07 | B8071A | (AX) = 0104  (IP) = 003C | (AX) = 1A07  (IP) = 003F |
| 003F | MOV ES, AX | 8EC0 | (AX) = 1A07  (ES) = 19F5  (IP) = 003F | (AX) = 19F5  (ES) = 1A07  (IP) = 0041 |
| 0041 | MOV AX, ES:[BX] | 268B07 | (AX) = 1A07  (IP) = 0041  [ES+BX] = FF00 | (AX) = 00FF  (IP) = 0044  [ES+BX] = FF00 |
| 0044 | MOV AX, 0000 | B80000 | (AX) = 00FF  (IP) = 0044 | (AX) = 0000  (IP) = 0047 |
| 0047 | MOV ES, AX | 8EC0 | (AX) = 0000  (ES) = 1A07  (IP) = 0047 | (AX) = 0000  (ES) = 0000  (IP) = 0049 |
| 0049 | PUSH DS | 1E | (DS) = 1A07  (IP) = 0049  (SP) = 0014  (SS) = 1A05  [SS+SP] = 0000  [SS+SP+2] = 19F5  [SS+SP+4] = 0000 | (DS) = 1A07  (IP) = 004A  (SP) = 0012  (SS) = 1A05  [SS+SP] = 1A07  [SS+SP+2] = 0000  [SS+SP+4] = 19F5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 004A | POP ES | 07 | (ES) = 0000  (IP) = 004A  (SP) = 0012  (SS) = 1A05  [SS+SP] = 1A07  [SS+SP+2] = 0000  [SS+SP+4] = 19F5 | (ES) = 1A07  (IP) = 004B  (SP) = 0014  (SS) = 1A05  [SS+SP] = 0000  [SS+SP+2] = 19F5  [SS+SP+4] = 0000 |
| 004B | MOV CX, ES:[BX-01] | 268B4FFF | (BX) = 0003  (CX) = FE04  (IP) = 004B  [ES+BX-01] = CEFF | (BX) = 0003  (CX) = FFCE  (IP) = 004F  [ES+BX-01] = CEFF |
| 004F | XCHG AX, CX | 91 | (AX) = 0000  (CX) = FFCE  (IP) = 004F | (AX) = FFCE  (CX) = 0000  (IP) = 0050 |
| 0050 | MOV DI, 0002 | BF0200 | (DI) = 0002  (IP) = 0050 | (DI) = 0002  (IP) = 0053 |
| 0053 | MOV ES:[BX+DI], AX | 268901 | (AX) = FFCE  (BX) = 0003  (DI) = 0002  (IP) = 0053  [ES+BX+DI] = 000B | (AX) = FFCE  (BX) = 0003  (DI) = 0002  (IP) = 0056  [ES+BX+DI] = CEFF |
| 0056 | MOV BP, SP | 8BEC | (BP) = 0000  (SP) = 0014  (IP) = 0056 | (BP) = 0014  (SP) = 0014  (IP) = 0058 |
| 0058 | PUSH [0000] | FF360000 | (IP) = 0058  (SP) = 0014  (SS) = 1A05  [SS+SP] = 0000  [SS+SP+2] = 19F5  [SS+SP+4] = 0000  [DS] = F401 | (IP) = 005C  (SP) = 0012  (SS) = 1A05  [SS+SP] = 01F4  [SS+SP+2] = 0000  [SS+SP+4] = 19F5  [DS] = F401 |
| 005C | PUSH [0002] | FF360200 | (IP) = 005C  (SP) = 0012  (SS) = 1A05  [SS+SP] = 01F4  [SS+SP+2] = 0000  [SS+SP+4] = 19F5  [SS+SP+6] = 0000  [DS+0002] = CEFF | (IP) = 0060  (SP) = 0010  (SS) = 1A05  [SS+SP] = FFCE  [SS+SP+2] = 01F4  [SS+SP+4] = 0000  [SS+SP+6] = 19F5  [DS+0002] = CEFF |
| 0060 | MOV BP, SP | 8BEC | (BP) = 0014  (SP) = 0010  (IP) = 0060 | (BP) = 0010  (SP) = 0010  (IP) = 0062 |
| 0062 | MOV DX, [BP+02] | 8B5602 | (DX) = 0000  (BP) = 0010  (IP) = 0062  [SS+BP+02] = 01F4 | (DX) = 01F4  (BP) = 0010  (IP) = 0065  [SS+BP+02] = 01F4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0065 | ADD SP, 0004 | 83C404 | (IP) = 0065  (SP) = 0010  (SS) = 1A05  [SS+SP] = FFCE  [SS+SP+2] = 01F4  [SS+SP+4] = 0000  [SS+SP+6] = 19F5 | (IP) = 0068  (SP) = 0014  (SS) = 1A05  [SS+SP] = 0000  [SS+SP+2] = 19F5  [SS+SP+4] = 0000  [SS+SP+6] = 0000 |
| 0068 | RET Far | CB | (CS) = 1A0A  (IP) = 0068  (SP) = 0014  [SS+SP] = 0000  [SS+SP+2] = 19F5 | (CS) = 19F5  (IP) = 0000  (SP) = 0018  [SS+SP] = 0000  [SS+SP+2] = 0000 |
| 0000 | INT 20 | CD20 |  |  |

**ПРИЛОЖЕНИЕ А**

**Исходный код программы с ошибками**

; Программа изучения режимов адресации процессора IntelX86

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 11,12,13,14,18,17,16,15

vec2 DB 10,20,-10,-20,30,40,-30,-40

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

Main ENDP

CODE ENDS

END Main

**ПРИЛОЖЕНИЕ Б**

**Листинг компиляции программы с ошибками**

#Microsoft (R) Macro Assembler Version 5.10 10/15/21 20:39:3

Page 1-1

1 ; Программа изучения режимов адресации

процессора IntelX86

2 = 0024 EOL EQU '$'

3 = 0002 ind EQU 2

4 = 01F4 n1 EQU 500

5 =-0032 n2 EQU -50

6 ; Стек программы

7 0000 AStack SEGMENT STACK

8 0000 000C[ DW 12 DUP(?)

9 ????

10 ]

11

12 0018 AStack ENDS

13 ; Данные программы

14 0000 DATA SEGMENT

15 ; Директивы описания данных

16 0000 0000 mem1 DW 0

17 0002 0000 mem2 DW 0

18 0004 0000 mem3 DW 0

19 0006 0B 0C 0D 0E 12 11 vec1 DB 11,12,13,14,18,17,16,15

20 10 0F

21 000E 0A 14 F6 EC 1E 28 vec2 DB 10,20,-10,-20,30,40,-30,-40

22 E2 D8

23 0016 01 02 FC FD 03 04 matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7

,8,-6,-5

24 FE FF 05 06 F8 F9

25 07 08 FA FB

26 0026 DATA ENDS

27 ; Код программы

28 0000 CODE SEGMENT

29 ASSUME CS:CODE, DS:DATA, SS:AStack

30 ; Головная процедура

31 0000 Main PROC FAR

32 0000 1E push DS

33 0001 2B C0 sub AX,AX

34 0003 50 push AX

35 0004 B8 ---- R mov AX,DATA

36 0007 8E D8 mov DS,AX

37 ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ

СМЕЩЕНИЙ

38 ; Регистровая адресация

39 0009 B8 01F4 mov ax,n1

40 000C 8B C8 mov cx,ax

41 000E B3 24 mov bl,EOL

42 0010 B7 CE mov bh,n2

43 ; Прямая адресация

44 0012 C7 06 0002 R FFCE mov mem2,n2

45 0018 BB 0006 R mov bx,OFFSET vec1

46 001B A3 0000 R mov mem1,ax

47 ; Косвенная адресация

48 001E 8A 07 mov al,[bx]

49 mov mem3,[bx]

lr2\_comp.ASM(42): error A2052: Improper operand type

50 ; Базированная адресация

51 0020 8A 47 03 mov al,[bx]+3

#Microsoft (R) Macro Assembler Version 5.10 10/15/21 20:39:3

Page 1-2

52 0023 8B 4F 03 mov cx,3[bx]

53 ; Индексная адресация

54 0026 BF 0002 mov di,ind

55 0029 8A 85 000E R mov al,vec2[di]

56 002D 8B 8D 000E R mov cx,vec2[di]

lr2\_comp.ASM(49): warning A4031: Operand types must match

57 ; Адресация с базированием и индексиров

анием

58 0031 BB 0003 mov bx,3

59 0034 8A 81 0016 R mov al,matr[bx][di]

60 0038 8B 89 0016 R mov cx,matr[bx][di]

lr2\_comp.ASM(53): warning A4031: Operand types must match

61 003C 8B 85 0022 R mov ax,matr[bx\*4][di]

lr2\_comp.ASM(54): error A2055: Illegal register value

62 ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ С

ЕГМЕНТОВ

63 ; Переопределение сегмента

64 ; ------ вариант 1

65 0040 B8 ---- R mov ax, SEG vec2

66 0043 8E C0 mov es, ax

67 0045 26: 8B 07 mov ax, es:[bx]

68 0048 B8 0000 mov ax, 0

69 ; ------ вариант 2

70 004B 8E C0 mov es, ax

71 004D 1E push ds

72 004E 07 pop es

73 004F 26: 8B 4F FF mov cx, es:[bx-1]

74 0053 91 xchg cx,ax

75 ; ------ вариант 3

76 0054 BF 0002 mov di,ind

77 0057 26: 89 01 mov es:[bx+di],ax

78 ; ------ вариант 4

79 005A 8B EC mov bp,sp

80 005C 3E: 8B 86 0016 R mov ax,matr[bp+bx]

lr2\_comp.ASM(73): error A2046: Multiple base registers

81 0061 3E: 8B 83 0016 R mov ax,matr[bp+di+si]

lr2\_comp.ASM(74): error A2047: Multiple index registers

82 ; Использование сегмента стека

83 0066 FF 36 0000 R push mem1

84 006A FF 36 0002 R push mem2

85 006E 8B EC mov bp,sp

86 0070 8B 56 02 mov dx,[bp]+2

87 0073 CA 0002 ret 2

88 0076 Main ENDP

lr2\_comp.ASM(81): error A2006: Phase error between passes

89 0076 CODE ENDS

90 END Main

#Microsoft (R) Macro Assembler Version 5.10 10/15/21 20:39:3

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

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

CODE . . . . . . . . . . . . . . 0076 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 = 0076

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

83 Source Lines

83 Total Lines

19 Symbols

47262 + 459998 Bytes symbol space free

2 Warning Errors

5 Severe Errors

**ПРИЛОЖЕНИЕ В**

**Исходный код исправленной программы**

; Программа изучения режимов адресации процессора IntelX86

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 11,12,13,14,18,17,16,15

vec2 DB 10,20,-10,-20,30,40,-30,-40

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

add sp, 4

ret

Main ENDP

CODE ENDS

END Main

**ПРИЛОЖЕНИЕ Г**

**Листинг компиляции исправленной программы**

#Microsoft (R) Macro Assembler Version 5.10 10/15/21 20:49:0

Page 1-1

; Программа изучения режимов адресации процессо

ра IntelX86

= 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

0000 0000 mem1 DW 0

0002 0000 mem2 DW 0

0004 0000 mem3 DW 0

0006 0B 0C 0D 0E 12 11 vec1 DB 11,12,13,14,18,17,16,15

10 0F

000E 0A 14 F6 EC 1E 28 vec2 DB 10,20,-10,-20,30,40,-30,-40

E2 D8

0016 01 02 FC FD 03 04 matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7,8,-6,-5

FE FF 05 06 F8 F9

07 08 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 АДРЕСАЦИИ 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 10/15/21 20:49:0

Page 1-2

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(49): 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(53): warning A4031: Operand types must match

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

; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ

; 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

0062 8B 56 02 mov dx,[bp]+2

0065 83 C4 04 add sp, 4

0068 CB ret

0069 Main ENDP

0069 CODE ENDS

END Main

#Microsoft (R) Macro Assembler Version 5.10 10/15/21 20:49:0

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

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

CODE . . . . . . . . . . . . . . 0069 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 = 0069

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

84 Source Lines

84 Total Lines

19 Symbols

47800 + 461507 Bytes symbol space free

2 Warning Errors

0 Severe Errors

**ПРИЛОЖЕНИЕ Д**

**Карта памяти исправленной программы**

Start Stop Length Name Class

00000H 00017H 00018H ASTACK

00020H 00045H 00026H DATA

00050H 000B8H 00069H CODE

Program entry point at 0005:0000