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 режима адресации и формирования исполнительного адреса.

Студент гр. 0384 ___________________ Кусмарцев А.И.

Преподаватель ___________________ Ефремов М. А.

Санкт-Петербург

2021

Цель работы.

Изучить работу режимов адресации. Найти и исправить ошибки при компиляции программы. Зафиксировать содержимое используемых регистров и ячеек памяти.

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.

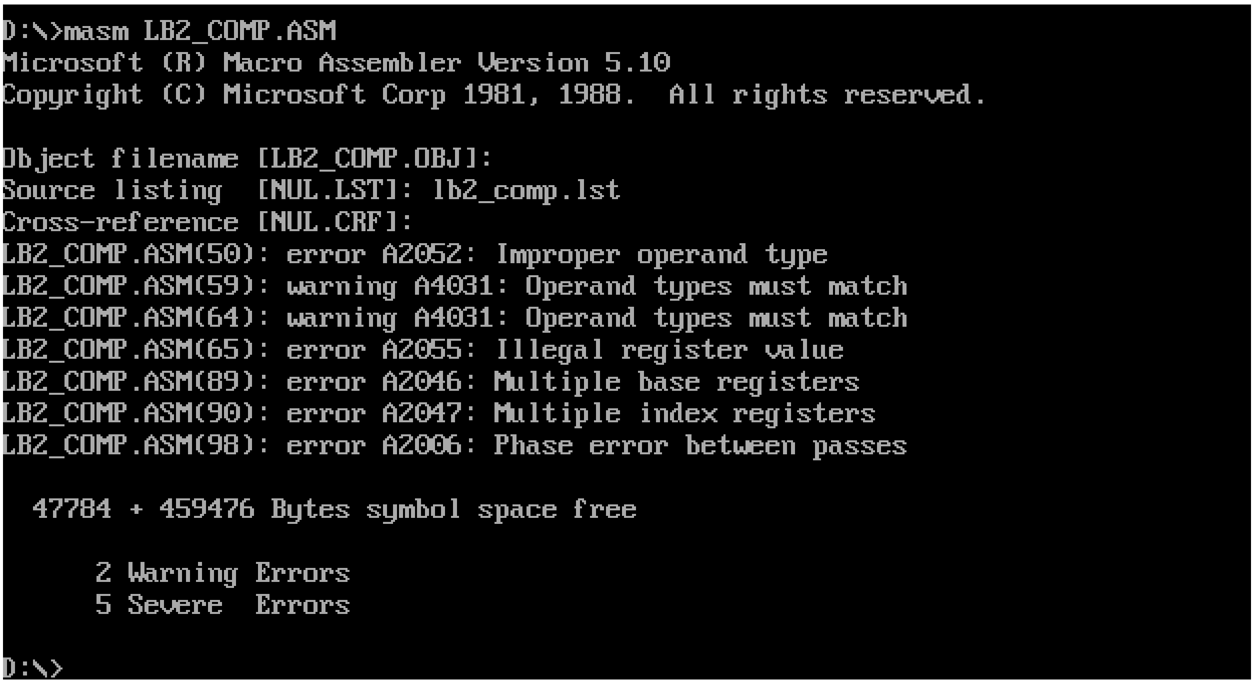
Вариант 4

vec1 12,11,10,9,5,6,7,8

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

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

Создаем файл lb2_comp.asm в соответствии с кодом из методического пособия. Меняем оригинальные значения массивов vec1, vec2 и matr в соответствии с вариантом №4 из файла lr2.dat.



```
D:\>masm LB2_COMP.ASM
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.

Object filename [LB2_COMP.OBJ]:
Source listing  [NUL.LST]: lb2_comp.lst
Cross-reference [NUL.CRF]:
LB2_COMP.ASM(50): error A2052: Improper operand type
LB2_COMP.ASM(59): warning A4031: Operand types must match
LB2_COMP.ASM(64): warning A4031: Operand types must match
LB2_COMP.ASM(65): error A2055: Illegal register value
LB2_COMP.ASM(89): error A2046: Multiple base registers
LB2_COMP.ASM(90): error A2047: Multiple index registers
LB2_COMP.ASM(98): error A2006: Phase error between passes

  47784 + 459476 Bytes symbol space free

      2 Warning Errors
      5 Severe  Errors

D:\>
```

Рисунок 1 — результат трансляции файла lb2_comp.asm

1) mov mem3, [bx]

LB2_COMP.ASM(50): error A2052: Improper operand type

Вызвана ошибка, так как на месте источника и приемника стоят ячейки памяти.

2) mov cx,vec2[di]

LB2_COMP.ASM(59): warning A4031: Operand types must match

Попытка записать в регистр размера двух байтов данных размером в 1 байт(так как vec2 определялась директивой DB)

3) mov cx,matr[bx][di]

LB2_COMP.ASM(64): warning A4031: Operand types must match

Попытка записать в регистр размера двух байтов данных размером в 1 байт(так как matr определялась директивой DB)

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

LB2_COMP.ASM(65): error A2055: Illegal register value

Недопустимое значение регистра

5) mov ax,matr[bp+bx]

LB2_COMP.ASM(89): error A2046: Multiple base registers

Попытка использовать несколько базовых регистров для адресации

6) mov ax,matr[bp+di+si]

LB2_COMP.ASM(90): error A2047: Multiple index registers

Попытка превышения количества индексных регистров для адресации

7) Main ENDP

LB2_COMP.ASM(98): error A2006: Phase error between passes

Комментируем строки содержащие ошибки и удачно транслируем программу без ошибок. Был создан диагностический файл lb2.lst и объектный файл lb2_fix.obj. Соберем lb2_fix.exe. Запустим его в отладчике.

(CS) = 1A0A

(DS) = 19F5

(ES) = 19F5

(SS) = 1A05

(CX) = 00B0

(BP) = 0000

(DX) = 0000

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения .	После выполнения
0000	Push DS	1E	(AX) = 0000 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 19F5 (CS) = 1A0A (ES) = 19F5 (SP) = 0018 (IP) = 0000	(AX) = 0000 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 19F5 (CS) = 1A0A (ES) = 19F5 (SP) = 0016 (IP) = 0001

			Stack +0 0000	Stack +0 19F5
0001	Sub AX, AX	2BC0	(AX) = 0000 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 19F5 (CS) = 1A0A (ES) = 19F5 (SP) = 0016 (IP) = 0001 Stack +0 19F5	(AX) = 0000 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 19F5 (CS) = 1A0A (ES) = 19F5 (SP) = 0016 (IP) = 0003 Stack +0 19F5
0003	Push AX	50	(AX) = 0000 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 19F5 (CS) = 1A0A (ES) = 19F5 (SP) = 0016 (IP) = 0003 Stack +0 19F5	(AX) = 0000 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 19F5 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0004 Stack +0 0000 Stack +2 19F5
0004	Mov Ax,1A07	B8071A	(AX) = 0000 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 19F5 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0004 Stack +0 0000 Stack +2 19F5	(AX) = 1A07 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 19F5 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0007 Stack +0 0000 Stack +2 19F5
0007	Mov DS, AX	8ED8	(AX) = 1A07 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 19F5 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0007 Stack +0 0000	(AX) = 1A07 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0009 Stack +0 0000

			Stack +2 19F5	Stack +2 19F5
0009	Mov AX, 01F4	B8F401	(AX) = 1A07 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0009 Stack +0 0000 Stack +2 19F5	(AX) = 01F4 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 000C Stack +0 0000 Stack +2 19F5
000C	Mov CX, AX	8BC8	(AX) = 01F4 (DX) = 0000 (CX) = 00B0 (BX) = 0000 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 000C Stack +0 0000 Stack +2 19F5	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = 0000 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 000E Stack +0 0000 Stack +2 19F5
000E	Mov BL, 24	B324	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = 0000 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 000E Stack +0 0000 Stack +2 19F5	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = 0024 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0010 Stack +0 0000 Stack +2 19F5
0010	Mov BH, CE	B7CE	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = 0024 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0010	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = CE24 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0012

			Stack +0 0000 Stack +2 19F5	Stack +0 0000 Stack +2 19F5
0012	Mov [0002], FFCE	C7060200CEFF	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = CE24 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0012 Stack +0 0000 Stack +2 19F5	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = CE24 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0018 Stack +0 0000 Stack +2 19F5
0018	Mov BX, 0006	BB0600	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = CE24 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0018 Stack +0 0000 Stack +2 19F5	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = 0006 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 001B Stack +0 0000 Stack +2 19F5
001B	Mov [0000], AX	A30000	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = 0006 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 001B Stack +0 0000 Stack +2 19F5	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = 0006 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 001E Stack +0 0000 Stack +2 19F5
001E	Mov AL, [BX]	BA07	(AX) = 01F4 (DX) = 0000 (CX) = 01F4 (BX) = 0006 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014	(AX) = 011F (DX) = 0000 (CX) = 01F4 (BX) = 0006 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014

			(IP) = 001E Stack +0 0000 Stack +2 19F5	(IP) = 0020 Stack +0 0000 Stack +2 19F5
0020	Mov AL, [BX+03]	BA4703	(AX) = 011F (DX) = 0000 (CX) = 01F4 (BX) = 0006 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0020 Stack +0 0000 Stack +2 19F5	(AX) = 0122 (DX) = 0000 (CX) = 01F4 (BX) = 0006 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0023 Stack +0 0000 Stack +2 19F5
0023	Mov CX, [BX+03]	8B4F03	(AX) = 0122 (DX) = 0000 (CX) = 01F4 (BX) = 0006 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0023 Stack +0 0000 Stack +2 19F5	(AX) = 0122 (DX) = 0000 (CX) = 2622 (BX) = 0006 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0026 Stack +0 0000 Stack +2 19F5
0026	Mov DI, 0002	BF0200	(AX) = 0122 (DX) = 0000 (CX) = 2622 (BX) = 0006 (DI) = 0000 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0026 Stack +0 0000 Stack +2 19F5	(AX) = 0122 (DX) = 0000 (CX) = 2622 (BX) = 0006 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0029 Stack +0 0000 Stack +2 19F5
0029	Mov AL, [000E+DI]	BA850E00	(AX) = 0122 (DX) = 0000 (CX) = 2622 (BX) = 0006 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5	(AX) = 01CE (DX) = 0000 (CX) = 2622 (BX) = 0006 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5

			(SP) = 0014 (IP) = 0029 Stack +0 0000 Stack +2 19F5	(SP) = 0014 (IP) = 002D Stack +0 0000 Stack +2 19F5
002D	Mov BX, 0003	BB0300	(AX) = 01CE (DX) = 0000 (CX) = 2622 (BX) = 0006 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 002D Stack +0 0000 Stack +2 19F5	(AX) = 01CE (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0030 Stack +0 0000 Stack +2 19F5
0030	Mov AL, [0016+BX+DI]	8A811600	(AX) = 01CE (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0030 Stack +0 0000 Stack +2 19F5	(AX) = 01FF (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0034 Stack +0 0000 Stack +2 19F5
0034	Mov AX, 1A07	B8071A	(AX) = 01FF (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0034 Stack +0 0000 Stack +2 19F5	(AX) = 1A07 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 19F5 (SP) = 0014 (IP) = 0037 Stack +0 0000 Stack +2 19F5
0037	Mov ES, AX	8EC0	(AX) = 1A07 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A	(AX) = 1A07 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A

			(ES) = 19F5 (SP) = 0014 (IP) = 0037 Stack +0 0000 Stack +2 19F5	(ES) = 1A07 (SP) = 0014 (IP) = 0039 Stack +0 0000 Stack +2 19F5
0039	Mov AX, ES:[BX]	268B07	(AX) = 1A07 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 0039 Stack +0 0000 Stack +2 19F5	(AX) = 00FF (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 003C Stack +0 0000 Stack +2 19F5
003C	Mov AX, 0000	B80000	(AX) = 00FF (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 003C Stack +0 0000 Stack +2 19F5	(AX) = 0000 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 003F Stack +0 0000 Stack +2 19F5
003F	Mov ES, AX	8ECO	(AX) = 0000 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 003F Stack +0 0000 Stack +2 19F5	(AX) = 0000 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 0000 (SP) = 0014 (IP) = 0041 Stack +0 0000 Stack +2 19F5
0041	Push DS	1E	(AX) = 0000 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07	(AX) = 0000 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07

			(CS) = 1A0A (ES) = 0000 (SP) = 0014 (IP) = 0041 Stack +0 0000 Stack +2 19F5	(CS) = 1A0A (ES) = 0000 (SP) = 0012 (IP) = 0042 Stack +0 1A07 Stack +2 0000 Stack +4 19F5
0042	Pop ES	07	(AX) = 0000 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 0000 (SP) = 0012 (IP) = 0042 Stack +0 1A07 Stack +2 0000 Stack +4 19F5	(AX) = 0000 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 0043 Stack +0 0000 Stack +2 19F5
0043	Mov CX, ES:[BX-01]	268B4FFF	(AX) = 0000 (DX) = 0000 (CX) = 2622 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 0043 Stack +0 0000 Stack +2 19F5	(AX) = 0000 (DX) = 0000 (CX) = FFCE (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 0047 Stack +0 0000 Stack +2 19F5
0047	Xchg AX, CX	91	(AX) = 0000 (DX) = 0000 (CX) = FFCE (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 0047 Stack +0 0000 Stack +2 19F5	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 0048 Stack +0 0000 Stack +2 19F5
0048	Mov DI, 0002	BF0200	(AX) = FFCE (DX) = 0000 (CX) = 0000	(AX) = FFCE (DX) = 0000 (CX) = 0000

			(BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 0048 Stack +0 0000 Stack +2 19F5	(BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 004B Stack +0 0000 Stack +2 19F5
004B	Mov ES:[BX+DI], AX	268901	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 004B Stack +0 0000 Stack +2 19F5	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (IP) = 004E Stack +0 0000 Stack +2 19F5
004E	Mov BP, SP	8BEC	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (BP) = 0000 (IP) = 004E Stack +0 0000 Stack +2 19F5	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (BP) = 0014 (IP) = 0050 Stack +0 0000 Stack +2 19F5
0050	Push [0000]	FF360000	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0014 (BP) = 0014 (IP) = 0050 Stack +0 0000 Stack +2 19F5	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0012 (BP) = 0014 (IP) = 0054 Stack +0 01F4 Stack +2 0000

				Stack +4 19F5
0054	Push [0002]	FF360200	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0012 (BP) = 0014 (IP) = 0054 Stack +0 01F4 Stack +2 0000 Stack +4 19F5	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0010 (BP) = 0014 (IP) = 0058 Stack +0 FFCE Stack +2 01F4 Stack +4 0000 Stack +6 19F5
0058	Mov BP, SP	8BEC	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0010 (BP) = 0014 (IP) = 0058 Stack +0 FFCE Stack +2 01F4 Stack +4 0000 Stack +6 19F5	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0010 (BP) = 0010 (IP) = 005A Stack +0 FFCE Stack +2 01F4 Stack +4 0000 Stack +6 19F5
005A	Mov DX, [BP+02]	8B5602	(AX) = FFCE (DX) = 0000 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0010 (BP) = 0010 (IP) = 005A Stack +0 FFCE Stack +2 01F4 Stack +4 0000 Stack +6 19F5	(AX) = FFCE (DX) = 01F4 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0010 (BP) = 0010 (IP) = 005D Stack +0 FFCE Stack +2 01F4 Stack +4 0000 Stack +6 19F5
005D	Ret Far 0002	CA0200	(AX) = FFCE	(AX) = FFCE

			(DX) = 01F4 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 1A0A (ES) = 1A07 (SP) = 0010 (BP) = 0010 (IP) = 005D Stack +0 FFCE Stack +2 01F4 Stack +4 0000 Stack +6 19F5	(DX) = 01F4 (CX) = 0000 (BX) = 0003 (DI) = 0002 (DS) = 1A07 (CS) = 01F4 (ES) = 1A07 (SP) = 0016 (BP) = 0010 (IP) = FFCE Stack +0 19F5

Разработанный программный код см. в приложении А.

Выводы.

В ходе выполнения работы были изучены режимы адресации и формирования исполнительного адреса.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb2_fix.asm

```
; 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 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]

; 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
```

Название файла: LB2.lst

```
     #Microsoft      (R)      Macro      Assembler      Version      5.10
10/28/21 15:52:2
                                                                       Page
1-1


                           ;     Ð Ñ ÐŸÐ³Ñ Ð°ÐŒÐŒÐ°      ÐžÐ·Ñ Ñ ÐµÐœÐžÑ
Ñ ÐµÐ¶ÐžÐ
                           ŒÐŸÐ² Ð°ÐŽÑ ÐµÑ·Ð°Ñ ÐžÐž Ð¿Ñ ÐŸÑ ÐµÑ·Ñ·ÐŸÑ Ð°
I
                           ntelX86
      = 0024                    EOL EQU '$'
      = 0002                    ind EQU 2
      = 01F4                    n1 EQU 500
      =-0032                    n2 EQU -50

                           ; Ð¡Ñ–ÐµÐº Ð¿Ñ ÐŸÐ³Ñ Ð°ÐŒÐŒÑ
      0000                 AStack SEGMENT STACK
      0000  000C[                   DW 12 DUP(?)
              ????
                      ]

      0018                 AStack ENDS

                           ; Ð Ð°ÐœÐœÑ Ðµ Ð¿Ñ ÐŸÐ³Ñ Ð°ÐŒÐŒÑ
      0000                 DATA SEGMENT

                           ;     Ð ÐžÑ ÐµÐºÑ–ÐžÐ²Ñ      ÐŸÐ¿ÐžÑ·Ð°ÐœÐžÑ
ÐŽÐ°ÐœÐœÑ
                            Ñ
      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  D8 CE 28 32 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

                           ; Ð ÐŸÐŽ Ð¿Ñ ÐŸÐ³Ñ Ð°ÐŒÐŒÑ
      0000                 CODE SEGMENT
                                ASSUME CS:CODE, DS:DATA, SS:AStack

                           ; Ð ÐŸÐ»ÐŸÐ²ÐœÐ°Ñ  Ð¿Ñ ÐŸÑ ÐµÐŽÑ Ñ Ð°
      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
```

```
                              ;       Ð�Ð  Ð�Ð�Ð�Ð  Ð�Ð�        Ð  Ð�Ð�Ð�Ð�Ð�Ð�
Ð�Ð�Ð  Ð�Ð¡Ð�Ð
                              ŠÐ�Ð�  Ð�Ð�  Ð£Ð  Ð�Ð�Ð�Ð�  Ð¡Ð�Ð�Ð©Ð�Ð�Ð�Ð�

                              ; Ð  ÐµÐ³ÐžÑ·Ñ–Ñ�ÐŸÐ²Ð°Ñ�  Ð°ÐŽÑ�ÐµÑ·Ð°Ñ�ÐžÑ�
       0009  B8 01F4                    mov ax,n1
       000C  8B C8                      mov cx,ax
       000E  B3 24                      mov bl,EOL
       0010  B7 CE                      mov bh,n2

                              ; Ð�Ñ�Ñ�ÐŒÐ°Ñ�  Ð°ÐŽÑ�ÐµÑ·Ð°Ñ�ÐžÑ�
     #Microsoft      (R)      Macro      Assembler      Version      5.10
10/28/21 15:52:2
                                                                        Page
1-2


       0012  C7 06 0002 R FFCE          mov mem2,n2
       0018  BB 0006 R             mov bx,OFFSET vec1
       001B  A3 0000 R             mov mem1,ax

                              ; Ð�ÐŸÑ·Ð²ÐµÐœÐœÐ°Ñ�  Ð°ÐŽÑ�ÐµÑ·Ð°Ñ�ÐžÑ�
       001E  8A 07                      mov al,[bx]
                              ;    mov mem3,[bx]

                              ;                    Ð�Ð°Ð·ÐžÑ�ÐŸÐ²Ð°ÐœÐœÐ°Ñ�
Ð°ÐŽÑ�ÐµÑ·Ð°Ñ�ÐžÑ�
       0020  8A 47 03                   mov al,[bx]+3
       0023  8B 4F 03                   mov cx,3[bx]

                              ; Ð�ÐœÐŽÐµÐºÑ·ÐœÐ°Ñ�  Ð°ÐŽÑ�ÐµÑ·Ð°Ñ�ÐžÑ�
       0026  BF 0002                    mov di,ind
       0029  8A 85 000E R               mov al,vec2[di]
                              ;    mov cx,vec2[di]

                              ;            Ð�ÐŽÑ�ÐµÑ·Ð°Ñ�ÐžÑ�            Ñ·
Ð±Ð°Ð·ÐžÑ�ÐŸÐ²Ð°ÐœÐŽÐµÐ
                              Œ ÐŽ ÐŽÐœÐŽÐµÐºÑ·ÐžÑ�ÐŸÐ²Ð°ÐœÐŽÐµÐŒ
       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]

                              ;       Ð�Ð  Ð�Ð�Ð�Ð  Ð�Ð�        Ð  Ð�Ð�Ð�Ð�Ð�Ð�
Ð�Ð�Ð  Ð�Ð¡Ð�Ð
                              ŠÐ�Ð�  Ð¡  Ð£Ð§Ð�Ð¢Ð�Ð�  Ð¡Ð�Ð�Ð�Ð�Ð�Ð¢Ð�Ð�
                              ;             Ð�ÐµÑ�ÐµÐŸÐ¿Ñ�ÐµÐŽÐµÐ»ÐµÐœÐŽÐµ
Ñ·ÐµÐ³ÐŒÐµÐœÑ–

                              Ð°

                              ; ------ Ð²Ð°Ñ�ÐžÐ°ÐœÑ– 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
```

```
                          ; ------ Ð²Ð°ÑÐžÐ°ÐœÑ– 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

                          ; ------ Ð²Ð°ÑÐžÐ°ÐœÑ– 3
      0048  BF 0002                   mov di,ind
      004B  26: 89 01            mov es:[bx+di],ax

                          ; ------ Ð²Ð°ÑÐžÐ°ÐœÑ– 4
      004E  8B EC                     mov bp,sp
                          ;      mov ax,matr[bp+bx]
                          ;      mov ax,matr[bp+di+si]

                          ;                   ÐÑ·Ð¿ÐŸÐ»ÑÐ·ÐŸÐ²Ð°ÐœÐžÐµ
Ñ·ÐµÐ³ÐŒÐµÐœÑ–Ð° Ñ
                          ·Ñ–ÐµÐºÐ°
      0050  FF 36 0000 R              push mem1
     #Microsoft     (R)     Macro     Assembler     Version     5.10
10/28/21 15:52:2
                                                                Page
1-3


      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
10/28/21 15:52:2
                                                                Symb
ols-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  LB2_FIX
     @VERSION . . . . . . . . . . . .      TEXT  510


         100 Source  Lines
         100 Total   Lines
          19 Symbols

       47806 + 459454 Bytes symbol space free

           0 Warning Errors
           0 Severe  Errors
```