

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. 0383

Ханина М.И.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить режимы адресации и формирование исполнительного адреса с помощью готовой программы. Найти ошибки и записать протокол работы в отладчике.

Задание.

Лабораторная работа 2 предназначена для изучения режимов адресации, использует готовую программу `lr2_compr.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.

Вариант 8.

`vec1 DB 28,27,26,25,21,22,23,24`

`vec2 DB 20,30,-20,-30,40,50,-40,-50`

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

Выполнение работы.

1. Данные варианта были записаны в исходный код программы.
2. При трансляции программы были обнаружены следующие ошибки:

1) `mov mem3,[bx]`

`lab2.asm(42): error A2052: Improper operand type`

Нельзя положить ячейку памяти в ячейку памяти.

2) `mov cx,vec2[di]`

lab2.asm(49): warning A4031: Operand types must match

Несовпадение размеров операндов:

нельзя положить элементы `vec2` размером 1 байт в регистр `CX` размером в 2 байта.

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

lab2.asm(53): warning A4031: Operand types must match

Несовпадение размеров операндов: нельзя положить в регистр `CX` размером 2 байта элементы матрицы `matr` длиной 1 байт.

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

lab2.asm(54): error A2055: Illegal register value

Нельзя одновременно менять значение регистра и класть в него информацию.

5) `mov ax,matr[bp+bx]`

lab2.asm(73): error A2046: Multiple base registers

Нельзя использовать несколько базовых регистров одновременно.

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

lab2.asm(74): error A2047: Multiple index registers

Нельзя использовать несколько индексных регистров.

7) `Main ENDP`

lab2.asm(81): error A2006: Phase error between passes

Текст исправленной программы lab2_fix.asm представлен в приложении Б, а файл листинга для этой программы — в приложении Г.

Начальные значения: (CS)=1A0A, (DS)=19F5, (ES)=19F5, (SS)=1A05

Таблица 1. Протокол выполнения программы lab2_fix.asm

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	(AX)=0000 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0018 (CS)=1A0A	(AX)=0000 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0016 (CS)=1A0A

			(DS)=19F5 (ES)=19F5 (SS)=1A05 (IP)=0000 Stack: +0 0000	(DS)=19F5 (ES)=19F5 (SS)=1A05 (IP)=0001 Stack: +0 19F5
0001	SUB AX,AX	2BC0	(AX)=0000 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0016 (CS)=1A0A (DS)=19F5 (ES)=19F5 (SS)=1A05 (IP)=0001 Stack: +0 19F5	(AX)=0000 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0016 (CS)=1A0A (DS)=19F5 (ES)=19F5 (SS)=1A05 (IP)=0003 Stack: +0 19F5
0003	PUSH AX	50	(AX)=0000 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0016 (CS)=1A0A	(AX)=0000 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A

			(DS)=19F5 (ES)=19F5 (SS)=1A05 (IP)=0003 Stack: +0 19F5 Stack: +2 0000	(DS)=19F5 (ES)=19F5 (SS)=1A05 (IP)=0004 Stack: +0 0000 Stack: +2 19F5
0004	MOV AX,1A07	B8071A	(AX)=0000 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=19F5 (ES)=19F5 (SS)=1A05 (IP)=0004 Stack: +0 0000 Stack: +2 19F5	(AX)=1A07 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=19F5 (ES)=19F5 (SS)=1A05 (IP)=0007 Stack: +0 0000 Stack: +2 19F5
0007	MOV DS,AX	8ED8	(AX)=1A07 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000	(AX)=1A07 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000

			(SP)=0014 (CS)=1A0A (DS)=19F5 (ES)=19F5 (SS)=1A05 (IP)=0007 Stack: +0 0000 Stack: +2 19F5	(SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0009 Stack: +0 0000 Stack: +2 19F5
0009	MOV AX,01F4	B8F401	(AX)=1A07 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0009 Stack: +0 0000 Stack: +2 19F5	(AX)=01F4 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=000C Stack: +0 0000 Stack: +2 19F5
000C	MOV CX,AX	8BC8	(AX)=01F4 (BX)=0000 (CX)=00B0 (DX)=0000 (SI)=0000	(AX)=01F4 (BX)=0000 (CX)=01F4 (DX)=0000 (SI)=0000

			(DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=000C Stack: +0 0000 Stack: +2 19F5	(DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=000E Stack: +0 0000 Stack: +2 19F5
000E	MOV BL,24	B324	(AX)=01F4 (BX)=0000 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=000E Stack: +0 0000 Stack: +2 19F5	(AX)=01F4 (BX)=0024 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0010 Stack: +0 0000 Stack: +2 19F5
0010	MOV BH,CE	B7CE	(AX)=01F4 (BX)=0024 (CX)=01F4	(AX)=01F4 (BX)=CE24 (CX)=01F4

			(DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0010 Stack: +0 0000 Stack: +2 19F5	(DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0012 Stack: +0 0000 Stack: +2 19F5
0012	MOV [0002],FFCE	C7060200C- EFF	(AX)=01F4 (BX)=CE24 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0012 Stack: +0 0000 Stack: +2 19F5	(AX)=01F4 (BX)=CE24 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0018 Stack: +0 0000 Stack: +2 19F5
0018	MOV BX,0006	BB0600	(AX)=01F4	(AX)=01F4

			(BX)=CE24 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0018 Stack: +0 0000 Stack: +2 19F5	(BX)=0006 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=001B Stack: +0 0000 Stack: +2 19F5
001B	MOV [0000],AX	A30000	(AX)=01F4 (BX)=0006 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=001B Stack: +0 0000	(AX)=01F4 (BX)=0006 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=001E Stack: +0 0000

			Stack: +2 19F5	Stack: +2 19F5
001E	AL,[BX]	8A07	(AX)=01F4 (BX)=0006 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=001E Stack: +0 0000 Stack: +2 19F5	(AX)=011C (BX)=0006 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0020 Stack: +0 0000 Stack: +2 19F5
0020	MOV AL,[BX+03]	8A4703	(AX)=011C (BX)=0006 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05	(AX)=0119 (BX)=0006 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05

			(IP)=0020 Stack: +0 0000 Stack: +2 19F5	(IP)=0023 Stack: +0 0000 Stack: +2 19F5
0023	MOV CX,[BX+03]	8B4F03	(AX)=0119 (BX)=0006 (CX)=01F4 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0023 Stack: +0 0000 Stack: +2 19F5	(AX)=0119 (BX)=0006 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0026 Stack: +0 0000 Stack: +2 19F5
0026	DI,0002	BF0200	(AX)=0119 (BX)=0006 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0000 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07	(AX)=0119 (BX)=0006 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07

			(ES)=19F5 (SS)=1A05 (IP)=0026 Stack: +0 0000 Stack: +2 19F5	(ES)=19F5 (SS)=1A05 (IP)=0029 Stack: +0 0000 Stack: +2 19F5
0029	MOV AL,[000E+DI]	8A850E00	(AX)=0119 (BX)=0006 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0029 Stack: +0 0000 Stack: +2 19F5	(AX)=01EC (BX)=0006 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=002D Stack: +0 0000 Stack: +2 19F5
002D	MOV BX,0003	BB0300	(AX)=01EC (BX)=0006 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014	(AX)=01EC (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014

			(CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=002D Stack: +0 0000 Stack: +2 19F5	(CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0030 Stack: +0 0000 Stack: +2 19F5
0030	MOV AL, [0016+BX+DI]	8A811600	(AX)=01EC (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0030 Stack: +0 0000 Stack: +2 19F5	(AX)=01FB (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0034 Stack: +0 0000 Stack: +2 19F5
0034	MOV AX,1A07	B8071A	(AX)=01FB (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002	(AX)=1A07 (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002

			(BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0034 Stack: +0 0000 Stack: +2 19F5	(BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0037 Stack: +0 0000 Stack: +2 19F5
0037	MOV ES,AX	8EC0	(AX)=1A07 (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=19F5 (SS)=1A05 (IP)=0037 Stack: +0 0000 Stack: +2 19F5	(AX)=1A07 (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0039 Stack: +0 0000 Stack: +2 19F5
0039	MOV AS,EX:[BX]	268B07	(AX)=1A07 (BX)=0003 (CX)=1519 (DX)=0000	(AX)=00FF (BX)=0003 (CX)=1519 (DX)=0000

			(SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0039 Stack: +0 0000 Stack: +2 19F5	(SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=003C Stack: +0 0000 Stack: +2 19F5
003C	MOV AX,0000	B80000	(AX)=00FF (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=003C Stack: +0 0000 Stack: +2 19F5	(AX)=0000 (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=003F Stack: +0 0000 Stack: +2 19F5
003F	MOV ES,AX	8EC0	(AX)=0000 (BX)=0003	(AX)=0000 (BX)=0003

			(CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=003F Stack: +0 0000 Stack: +2 19F5	(CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=0000 (SS)=1A05 (IP)=0041 Stack: +0 0000 Stack: +2 19F5
0041	PUSH DS	1E	(AX)=0000 (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=0000 (SS)=1A05 (IP)=0041 Stack: +0 0000 Stack: +2 19F5	(AX)=0000 (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0012 (CS)=1A0A (DS)=1A07 (ES)=0000 (SS)=1A05 (IP)=0042 Stack: +0 1A07 Stack: +2 0000

			Stack: +4 0000	Stack: +4 19F5
0042	POP ES	07	(AX)=0000 (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0012 (CS)=1A0A (DS)=1A07 (ES)=0000 (SS)=1A05 (IP)=0042 Stack: +0 1A07 Stack: +2 0000 Stack: +4 19F5	(AX)=0000 (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0043 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000
0043	MOV CX,ES:[BX-01]	268B4FFF	(AX)=0000 (BX)=0003 (CX)=1519 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07	(AX)=0000 (BX)=0003 (CX)=FFCE (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07

			(SS)=1A05 (IP)=0043 Stack: +0 0000 Stack: +2 19F5	(SS)=1A05 (IP)=0047 Stack: +0 0000 Stack: +2 19F5
0047	XCHG AX,CX	91	(AX)=0000 (BX)=0003 (CX)=FFCE (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0047 Stack: +0 0000 Stack: +2 19F5	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0048 Stack: +0 0000 Stack: +2 19F5
0048	MOV DI,0002	BF0200	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A

			(DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0048 Stack: +0 0000 Stack: +2 19F5	(DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=004B Stack: +0 0000 Stack: +2 19F5
004B	MOV [BX+DI],AX	268901	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=004B Stack: +0 0000 Stack: +2 19F5	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=004E Stack: +0 0000 Stack: +2 19F5
004E	MOV BP,SP	8BEC	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0000	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0014

			(SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=004E Stack: +0 0000 Stack: +2 19F5	(SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0050 Stack: +0 0000 Stack: +2 19F5
0050	PUSH [0000]	FF360000	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0014 (SP)=0014 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0050 Stack: +0 0000 Stack: +2 19F5 Stack: +4 0000	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0014 (SP)=0012 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0054 Stack: +0 01F4 Stack: +2 0000 Stack: +4 19F5
0054	PUSH [0002]	FF360200	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000

			(SI)=0000 (DI)=0002 (BP)=0014 (SP)=0012 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0054 Stack: +0 01F4 Stack: +2 0000 Stack: +4 19F5 Stack: +6 0000	(SI)=0000 (DI)=0002 (BP)=0014 (SP)=0010 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0058 Stack: +0 FFCE Stack: +2 01F4 Stack: +4 0000 Stack: +6 19F5
0058	MOV BP,SP	8BEC	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0014 (SP)=0010 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=0058 Stack: +0 FFCE Stack: +2 01F4	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0010 (SP)=0010 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=005A Stack: +0 FFCE Stack: +2 01F4

			Stack: +4 0000 Stack: +6 19F5	Stack: +4 0000 Stack: +6 19F5
005A	MOV DX,[BP+02]	8B5602	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=0000 (SI)=0000 (DI)=0002 (BP)=0010 (SP)=0010 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=005A Stack:+0 FFCE Stack: +2 01F4 Stack: +4 0000 Stack: +6 19F5	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=01F4 (SI)=0000 (DI)=0002 (BP)=0010 (SP)=0010 (CS)=1A0A (DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=005D Stack: +0 FFCE Stack: +2 01F4 Stack: +4 0000 Stack: +6 19F5
005D	RET FAR 0002	CA0200	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=01F4 (SI)=0000 (DI)=0002 (BP)=0010 (SP)=0010 (CS)=1A0A	(AX)=FFCE (BX)=0003 (CX)=0000 (DX)=01F4 (SI)=0000 (DI)=0002 (BP)=0010 (SP)=0016 (CS)=01F4

			(DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=005D Stack: +0 FFCE Stack: +2 01F4 Stack: +4 0000 Stack: +6 19F5	(DS)=1A07 (ES)=1A07 (SS)=1A05 (IP)=FFCE Stack: +0 19F5 Stack: +2 0000 Stack: +4 0000 Stack: +6 0000
--	--	--	--	--

Тексты исходных файлов программ представлены в приложении А и Б.
Тексты файлов диагностических сообщений представлены в приложении В и Г.

Выводы.

В ходе выполнения лабораторной работы мною была изучена работа с режимами адресации на языке Ассемблер.

ПРИЛОЖЕНИЕ А.
ТЕКСТ ИСХОДНОГО ФАЙЛА ПРОГРАММЫ.

Название файла: lab2.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 28,27,26,25,21,22,23,24
vec2 DB 20,30,-20,-30,40,50,-40,-50
matr DB -8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-2,-1,5,6
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.
ТЕКСТ ИСХОДНОГО ФАЙЛА ПРОГРАММЫ.

Название файла: lab2_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 28,27,26,25,21,22,23,24
vec2 DB 20,30,-20,-30,40,50,-40,-50
matr DB -8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-2,-1,5,6
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 В.
ТЕКСТ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ
СООБЩЕНИЙ ПРОГРАММЫ.

Название файла: lab2.lst

Microsoft (R) Macro Assembler Version 5.10

9/29/21 24:17:50

Page 1-1

```

                                ; Программа изучения режимов адресации
процессо
                                pa 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 1C 1B 1A 19 15 16        vec1 DB 28,27,26,25,21,22,23,24
                                17 18
000E 14 1E EC E2 28 32        vec2 DB 20,30,-20,-30,40,50,-40,-50
                                D8 CE

```

```

0016 F8 F9 03 04 FA FB      matr DB -8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-
      2,-1,5,6
      01 02 FC FD 07 08
      FE FF 05 06
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]
lab2.asm(42): error A2052: Improper operand type
                                ; Базированная адресация
0020 8A 47 03                                mov al,[bx]+3
0023 8B 4F 03                                mov cx,3[bx]
                                ; Индексная 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]
lab2.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]
lab2.asm(53): warning A4031: Operand types must match
003C 8B 85 0022 R     mov ax,matr[bx*4][di]
lab2.asm(54): error A2055: Illegal register value
                ; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ
                С УЧЕТОМ СЕГМЕНТОВ
                ; Переопределение сегмента
                ;-----вариант 1
0040 B8 -----R     mov ax, SEG vec2
0043 8E C0           mov es, ax
0045 26: 8B 07       mov ax, es:[bx]
0048 B8 0000         mov ax, 0
                ;-----вариант 2
004B 8E C0           mov es, ax
004D 1E             push ds
004E 07             pop es
004F 26: 8B 4F FF     mov cx, es:[bx-1]
0053 91             xchg cx,ax
                ;-----вариант 3
0054 BF 0002          mov di,ind
0057 26: 89 01       mov es:[bx+di],ax
                ;-----вариант 4
005A 8B EC           mov bp,sp
005C 3E: 8B 86 0016 R  mov ax,matr[bp+bx]
```

lab2.asm(73): error A2046: Multiple base registers

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

lab2.asm(74): error A2047: Multiple index registers

 ; Использование сегмента стека

0066 FF 36 0000 R push mem1

006A FF 36 0002 R push mem2

006E 8B EC mov bp,sp

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

0073 CA 0002 ret 2

0076 Main ENDP

lab2.asm(81): error A2006: Phase error between passes

0076 CODE ENDS

 END Main

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	lab2		
@VERSION	TEXT	510		

83 Source Lines

83 Total Lines

19 Symbols

47826 + 461481 Bytes symbol space free

2 Warning Errors

5 Severe Errors

ПРИЛОЖЕНИЕ Г.
ТЕКСТ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ
СООБЩЕНИЙ ПРОГРАММЫ.

Название файла: lab2_fix.lst

Microsoft (R) Macro Assembler Version 5.10

9/29/21 24:27:09

Page 1-1

```

; Программа изучения режимов адресации процессо
pa 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 1C 1B 1A 19 15 16  vec1 DB 28,27,26,25,21,22,23,24
              17 18
000E 14 1E EC E2 28 32  vec2 DB 20,30,-20,-30,40,50,-40,-50
              D8 CE
0016 F8 F9 03 04 FA FB  matr DB -8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-2,-1,5,6
              01 02 FC FD 07 08
              FE FF 05 06
```

```

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

```

```
0026 BF 0002          mov di,ind
0029 8A 85 000E R      mov al,vec2[di]
                      ;mov cx,vec2[di]
                      ; Адресация с базированием и индексированием
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 АДРЕСАЦИИ С УЧЕТОМ
                        СЕГМЕНТОВ
                      ; Переопределение сегмента
                      ; ----- 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

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	lab2_fix	
@VERSION	TEXT	510	

83 Source Lines

83 Total Lines

19 Symbols

47798 + 461509 Bytes symbol space free

0 Warning Errors

0 Severe Errors