

**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

Парфенов В.М.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Изучать как устроены ТРАНСЛЯЦИЯ, ОТЛАДКА И ВЫПОЛНЕНИЕ ПРОГРАММ НА ЯЗЫКЕ АССЕМБЛЕРА

### **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.

### **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.

### **Вариант 7**

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

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

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

### **Ход работы**

При трансляции программы с данными исходными данными были обнаружены следующие ошибки:

- lr2\_comp.asm(41): error A2052: Improper operand type. Попытка сохранить одну ячейку памяти в другую. Такое недопустимо. Необходимо сначала сохранить данные из одной ячейки памяти в регистр, а затем положить в другую.
- lr2\_comp.asm(49): warning A4031: Operand types must match. Попытка положить в ячейку памяти размером 2 байт информацию размером 1 байта, вследствие чего была вызвана ошибка, так как размеры должны совпадать. DB – один байт, CX – два байта.
- lr2\_comp.asm(53): warning A4031: Operand types must match. Попытка положить в ячейку памяти размером 2 байта информацию размером 1 байт, что недопустимо, так как размер должен совпадать. CX – 2 байта, matr[bx][di] – один байт.
- lr2\_comp.asm(54): error A2055: Illegal register value. Недопустимое значение регистра.
- lr2\_comp.asm(73): error A2046: Multiple base registers. Попытка использовать несколько базовых регистров для адресации, что вызывает ошибку.
- lr2\_comp.asm(74): error A2047: Multiple index registers. Попытка использовать несколько индексных регистров для адресации, что вызывает ошибку.

Начальное значение сегментных регистров:

CS = 1A0A ;      DS = 19F5;

ES = 19F5 ;      SS = 1A05 ;

Таблица 3 - Отладка hello2.exe

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 = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0000 SP = 0018 BP = 0010 Stack +0 0000 +2 0000 +4 0000 +6 0000	AX = 0000 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0001 SP = 0016 BP = 0010 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0001	SUB AX, AX	2BC0	AX = 0000 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A	AX = 0000 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A

			ES = 19F5 SS = 1A05 IP = 0001 SP = 0016 BP = 0010 Stack +0 19F5 +2 0000 +4 0000 +6 0000	ES = 19F5 SS = 1A05 IP = 0003 SP = 0016 BP = 0010 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0003	PUSH AX	50	AX = 0000 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0003 SP = 0016 BP = 0010 Stack +0 19F5 +2 0000 +4 0000 +6 0000	AX = 0000 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0004 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0004	MOV AX, 1A07	B8071A	AX = 0000 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A	AX = 1A07 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A

			ES = 19F5 SS = 1A05 IP = 0004 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 19F5 SS = 1A05 IP = 0007 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0007	MOV DS, AX	B8071A	AX = 1A07 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0007 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 1A07 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0009 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0009	MOV AX, 01F4	B8F401	AX = 1A07 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A	AX = 01F4 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A

			ES = 19F5 SS = 1A05 IP = 0009 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 19F5 SS = 1A05 IP = 000C SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
000C	MOV CX, AX	8BC8	AX = 01F4 BX = 0000 CX = 0000 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 000C SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 01F4 BX = 0000 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 000E SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
000E	MOV BL, 24	B324	AX = 01F4 BX = 0000 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A	AX = 01F4 BX = 2400 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A

			ES = 19F5 SS = 1A05 IP = 000E SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 19F5 SS = 1A05 IP = 0010 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0010	MOV BH, CE	B7CE	AX = 01F4 BX = 2400 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0010 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 01F4 BX = 24CE CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0012 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0012	MOV [0002], FFCE	C7060200CEFF	AX = 01F4 BX = 24CE CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A	AX = 01F4 BX = 24CE CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A



			ES = 19F5 SS = 1A05 IP = 0012 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 19F5 SS = 1A05 IP = 0018 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0018	MOV BX, 0006	BB0600	AX = 01F4 BX = 24CE CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0009 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 01F4 BX = 0006 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 001B SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
001B	MOV [0000], AX	A30000	AX = 01F4 BX = 0006 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A	AX = 01F4 BX = 0006 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A

			ES = 19F5 SS = 1A05 IP = 001B SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 19F5 SS = 1A05 IP = 001E SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
001E	MOV AL, [BX]	BB0600	AX = 01F4 BX = 0006 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 001E SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0115 BX = 0006 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0020 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0020	MOV AL, [BX+03]	8A4703	AX = 0115 BX = 0006 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A	AX = 0118 BX = 0006 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A

			ES = 19F5 SS = 1A05 IP = 0020 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 19F5 SS = 1A05 IP = 0023 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0023	MOV CX, [BX+03]	8B4F03	AX = 0118 BX = 0006 CX = 01F4 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0023 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0118 BX = 0006 CX = 1C18 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0026 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0026	MOV DI, 0002	BF0200	AX = 0118 BX = 0006 CX = 1C18 DX = 0000 SI = 0000 DI = 0000 DS = 1A07 CS = 1A0A	AX = 0118 BX = 0006 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A

			ES = 19F5 SS = 1A05 IP = 0026 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 19F5 SS = 1A05 IP = 0029 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0029	MOV AL, [000E+DI]	8A850E00	AX = 0118 BX = 0006 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0029 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 01D8 BX = 0006 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 002D SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
002D	MOV BX, 0003	BB0300	AX = 01D8 BX = 0006 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A	AX = 01D8 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A

			ES = 19F5 SS = 1A05 IP = 002D SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 19F5 SS = 1A05 IP = 0030 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0030	AL, [0016+BX+DI]	8A811600	AX = 01D8 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0030 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0108 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0034 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0034	MOV AX, 1A07	B8071A	AX = 0108 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A	AX = 1A07 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A

			ES = 19F5 SS = 1A05 IP = 0034 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 19F5 SS = 1A05 IP = 0037 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0037	MOV ES, AX	8EC0	AX = 1A07 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 19F5 SS = 1A05 IP = 0037 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 1A07 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 1A07 SS = 1A05 IP = 0039 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0039	MOV AX, ES:[BX]	268B07	AX = 1A07 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A	AX = 00FF BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A

			ES = 1A07 SS = 1A05 IP = 0039 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 1A07 SS = 1A05 IP = 003C SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
003C	MOV AX, 0000	B80000	AX = 00FF BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 1A07 SS = 1A05 IP = 003C SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0000 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 1A07 SS = 1A05 IP = 003F SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
003F	MOV ES, AX	8EC0	AX = 0000 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A	AX = 0000 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A

			ES = 1A07 SS = 1A05 IP = 003F SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 0000 SS = 1A05 IP = 0041 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0041	PUSH DS	1E	AX = 0000 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 0041 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0000 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 0042 SP = 0012 BP = 0010 Stack +0 1A07 +2 0000 +4 19F5 +6 0000
0042	POP	07	AX = 0000 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A	AX = 0000 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A



			ES = 0000 SS = 1A05 IP = 0042 SP = 0012 BP = 0010 Stack +0 1A07 +2 0000 +4 19F5 +6 0000	ES = 0000 SS = 1A05 IP = 0043 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0043	CX, ES:[BX-01]	268B4FFF	AX = 0000 BX = 0003 CX = 1C18 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 0043 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0000 BX = 0003 CX = FFCE DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 0047 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0047	XCHG	91	AX = 0000 BX = 0003 CX = FFCE DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A

			ES = 0000 SS = 1A05 IP = 0047 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 0000 SS = 1A05 IP = 0048 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0048	MOV DI, 0002	BF0200	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 0048 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 004B SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
004B	MOV ES:[BX+DI], AX	268901	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A

			ES = 0000 SS = 1A05 IP = 0042 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 1A07 SS = 1A05 IP = 0042 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
004E	MOV BP, SP	8BEC	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 004E SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 0050 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0050	PUSH [0000]	FF360000	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A

			ES = 0000 SS = 1A05 IP = 0050 SP = 0014 BP = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000	ES = 0000 SS = 1A05 IP = 0054 SP = 0014 BP = 0010 Stack +0 01A4 +2 0000 +4 19F5 +6 0000
0054	PUSH [0002]	FF360200	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 0054 SP = 0012 BP = 0010 Stack +0 01F4 +2 0000 +4 19F5 +6 0000	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 0058 SP = 0010 BP = 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5
0058	MOV BP, SP	8BEC	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A

			ES = 0000 SS = 1A05 IP = 0058 SP = 0010 BP = 0014 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5	ES = 0000 SS = 1A05 IP = 005A SP = 0010 BP = 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5
005A	MOV DX [BP+02]	8B5602	AX = FFCE BX = 0003 CX = 0000 DX = 0000 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 005A SP = 0010 BP = 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5	AX = FFCE BX = 0003 CX = 0000 DX = 01F4 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A ES = 0000 SS = 1A05 IP = 005D SP = 0010 BP = 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5
005D	RET FAR 0002	CA0200	AX = FFCE BX = 0003 CX = 0000 DX = 01F4 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A	AX = FFCE BX = 0003 CX = 0000 DX = 01F4 SI = 0000 DI = 0002 DS = 1A07 CS = 1A0A

			ES = 0000 SS = 1A05 IP = 005D SP = 0010 BP = 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5	ES = 0000 SS = 1A05 IP = FFCE SP = 0016 BP = 0010 Stack +0 19F5 +2 0000 +4 0000 +6 0000

### **Вывод.**

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

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

```
Название файла lr2_comp.asm
EOL EQU '$'
ind EQU 2
n1 EQU 500
n2 EQU -50
; Стек программы
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
; Данные программы
DATA SEGMENT
; Директивы описания данных
mem1 DW 0
mem2 DW 0
mem3 DW 0
vec1 DB 21,22,23,24,28,27,26,25
vec2 DB 40,50,-40,-50,20,30,-20,-30
matr DB 5,6,-8,-7,7,8,-6,-5,1,2,-4,-3,3,4,-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 lr2_fix.asm
EOL EQU '$'
ind EQU 2
n1 EQU 500

```

```

n2 EQU -50
; Стек программы
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
; Данные программы
DATA SEGMENT
; Директивы описания данных
mem1 DW 0
mem2 DW 0
mem3 DW 0
vec1 DB 21,22,23,24,28,27,26,25
vec2 DB 40,50,-40,-50,20,30,-20,-30
matr DB 5,6,-8,-7,7,8,-6,-5,1,2,-4,-3,3,4,-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 lr2_comp.lst

```

```
= 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 данн?
?x

0000 0000          mem1 DW 0
0002 0000          mem2 DW 0
0004 0000          mem3 DW 0
0006 15 16 17 18 1C 1B      vec1 DB 21,22,23,24,28,27,26,25
    1A 19
000E 28 32 D8 CE 14 1E      vec2 DB 40,50,-40,-50,20,30,-20,-30
    EC E2
0016 05 06 F8 F9 07 08 matr DB 5,6,-8,-7,7,8,-6,-5,1,2,-4,-3,3,4,-2,-1
    FA FB 01 02 FC FD
    03 04 FE FF
0026          DATA ENDS
```

```

; Код программы
0000          CODE SEGMENT

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

; Головная процедура
0000          Main PROC FAR
0000 1E          push DS
0001 2B C0          sub AX,AX
0003 50          push AX
0004 B8 ---- R      mov AX,DATA
0007 8E D8          mov DS,AX

; ПРОВЕРКА РЕЖИМОВ АДРЕСА❖
❖ИИ 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]

lr2_comp.asm(41): 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]
```

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

; Адресация с базирование❖

❖ 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

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

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

; ПРОВЕРКА РЕЖИМОВ АДРЕСА❖

❖ИИ С УЧЕТОМ СЕГМЕНТОВ

; Переопределение сегмент

a

; ----- 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]
lr2_comp.asm(73): error A2046: Multiple base registers
0061 3E: 8B 83 0016 R     mov ax,matr[bp+di+si]
lr2_comp.asm(74): error A2047: Multiple index registers
                        ; Использование сегмента ?
                        ?тека
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
lr2_comp.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 lr2\_comp

@VERSION ..... TEXT 510

83 Source Lines

83 Total Lines

19 Symbols

47800 + 459460 Bytes symbol space free

2 Warning Errors

5 Severe Errors

Название файла lr2\_fix.lst

```
= 0024          EOL EQU '$'
= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50

; Стек программы
0000          AStack SEGMENT STACK
0000 000C[          DW 12 DUP(?)
    ???
    ]

0018          AStack ENDS

; Данные программы
0000          DATA SEGMENT

; Директивы описания данных
    ?x

0000 0000          mem1 DW 0
0002 0000          mem2 DW 0
0004 0000          mem3 DW 0
0006 15 16 17 18 1C 1B      vec1 DB 21,22,23,24,28,27,26,25
    1A 19
000E 28 32 D8 CE 14 1E      vec2 DB 40,50,-40,-50,20,30,-20,-30
    EC E2
0016 05 06 F8 F9 07 08 matr DB 5,6,-8,-7,7,8,-6,-5,1,2,-4,-3,3,4,-2,-1
    FA FB 01 02 FC FD
    03 04 FE FF
0026          DATA ENDS
```

```

; Код программы
0000          CODE SEGMENT

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

; Головная процедура
0000          Main PROC FAR
0000 1E          push DS
0001 2B C0          sub AX,AX
0003 50          push AX
0004 B8 ---- R      mov AX,DATA
0007 8E D8          mov DS,AX

; ПРОВЕРКА РЕЖИМОВ АДРЕСА
; ИИ 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 базирование❖
❖ 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 АДРЕСА❖
❖ИИ С УЧЕТОМ СЕГМЕНТОВ
; Переопределение сегмент
a
; ----- 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 ?
                  ?тека
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 lr2\_fix

@VERSION ..... TEXT 510

83 Source Lines

83 Total Lines

19 Symbols

47826 + 459434 Bytes symbol space free

0 Warning Errors

0 Severe Errors