**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Архитектура компьютера»**

**Тема: Изучение режимов адресации основной памяти**

Вариант 4

|  |  |  |
| --- | --- | --- |
| Студент гр. 5383 |  | Допира В. Е. |
| Преподаватель |  | Кирьянчиков В. А. |

Санкт-Петербург

2016

**Цель работы**

Ознакомиться с основными режимами адресации основной памяти.

**Теоретические сведения**

Режимы адресации — это различные способы указания местоположения операндов

1. Неявная адресация

Местоположение операнда фиксировано и определяется кодом операции

Пример: cbw

1. Прямая адресация

При непосредственной адресации значение операнда является частью машинной команды

Пример: mov al, 5

1. Абсолютная прямая адресация

В машинной команде содержится адрес операнда, находящегося в памяти.

mov dx, [5]

1. Относительная прямая адресация

Этот режим используется в командах передачи управления. В машинной команде содержится смещение, которое прибавляется к значению указателя команд IP.

Label:

…

loop Label

1. Регистровая адресация

Операнд находится в регистре.

add ax, bx

1. Косвенная регистровая (базовая) адресация

Адрес операнда находится в одном из регистров BX, SI или DI.

add ax, [si]

1. Косвенная регистровая (базовая) адресация со смещением

Адрес операнда вычисляется как сумма содержимого регистра BX, BP, SI или DI и 8- или 16-разрядного смещения.

mov dx, [bx+2]

1. Косвенная базовая индексная адресация

Адрес операнда вычисляется как сумма содержимого одного из базовых регистров BX или BP и одного из индексных регистров SI или DI.

add ax, [bp+di]

1. Косвенная базовая индексная адресация со смещением

Адрес операнда вычисляется как сумма содержимого одного из базовых регистров BX или BP, одного из индексных регистров SI или DI и 8- или 16-разрядного смещения.

mov bl, [bp+si+5]

**Ход работы :**

1. Получить у преподавателя вариант выбора значений исходных данных (массивов) vec1, vec2 и matr из файла lr2.dat и занести свои данные вместо значений, указанных в приведенной для образца программе.

2. Протранслировать программу с созданием файла диагностических сообщений и объяснить обнаруженные ошибки (error) и предупреждения (warning). Закомментировать операторы с ошибками в тексте программы, а операторы с предупреждениями оставить без изменения. Объяснения ошибок и предупреждений должны быть приведены в отчете по лабораторной работе.

3. Снова протранслировать программу и скомпоновать загрузочный модуль. Учесть, что программа - учебная и может выполняться только под отладчиком. В автоматическом режиме она выполняться не должна.

4. Выполнить программу в пошаговом режиме под управлением отладчика с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения каждой команды. Разобраться в используемых режимах адресации и получаемых результатах.

5. Результаты прогона программы под управлением отладчика должны быть подписаны преподавателем и представлены в отчете по лабораторной работе в виде, аналогичном указанному в лаб.работе №1.

**Выполнение работы:**

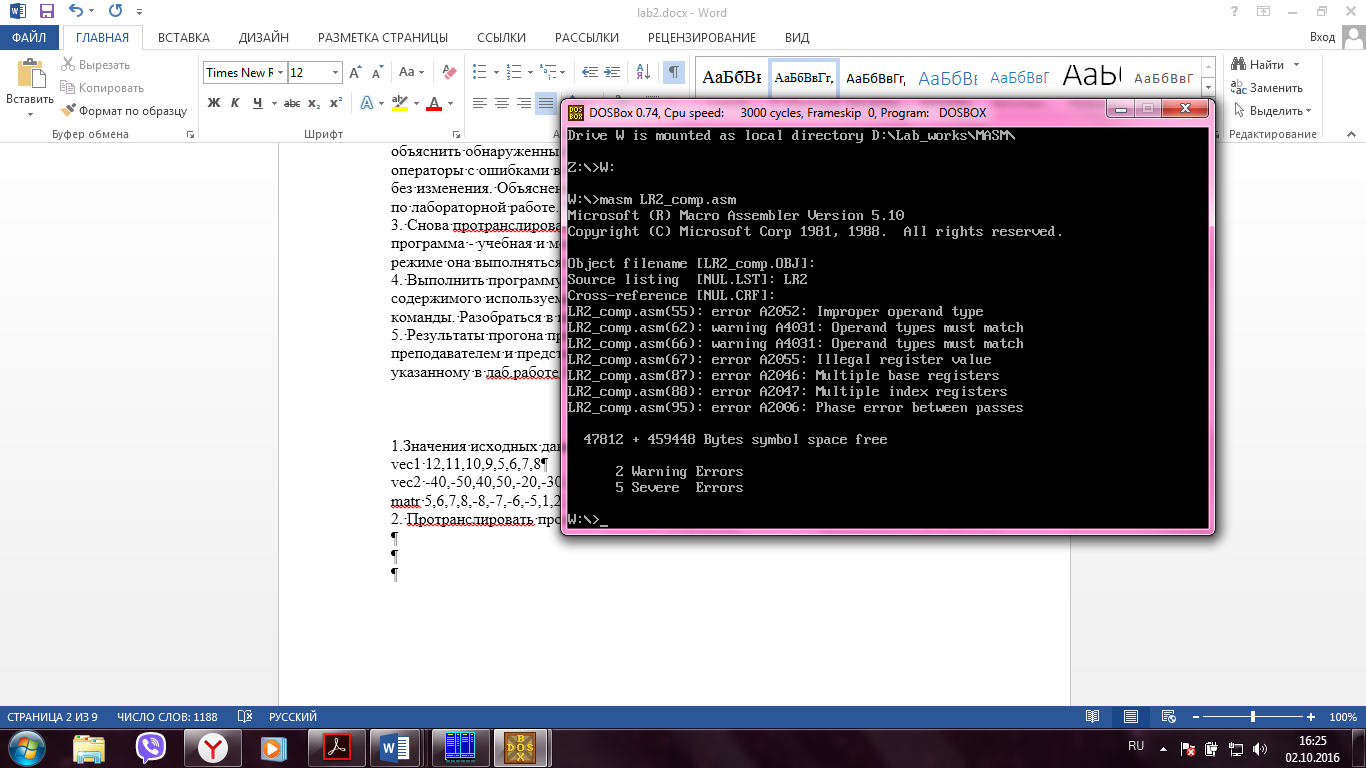
1.Заменены значения исходных данных массивов на следующие в файле LR2\_comp.ASM при помощи FAR:

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

2. Запущен DOSBOX. Протранслирована программа и создан файл диагностических сообщений lr2.lst.



**Файл листинга:**

; Учебная программа лабораторной работы №2 по

дисциплине "Архитектура компьютера"

;

= 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 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

; 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]

LR2\_comp.asm(55): error A2052: Improper operand type

; Базированная адресация

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

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

; Индексированная 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(62): 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(66): warning A4031: Operand types must match

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

LR2\_comp.asm(67): error A2055: Illegal register value

; ПРОВЕРКА 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]

LR2\_comp.asm(87): error A2046: Multiple base registers

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

LR2\_comp.asm(88): 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

LR2\_comp.asm(95): error A2006: Phase error between passes

0076 CODE ENDS

END Main

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

97 Source Lines

97 Total Lines

19 Symbols

47812 + 459448 Bytes symbol space free

2 Warning Errors

5 Severe Errors

Карта памяти:

Start Stop Length Name Class

00000H 00017H 00018H ASTACK

00020H 00045H 00026H DATA

00050H 000AFH 00060H CODE

Program entry point at 0005:0000

**Пояснения имеющихся некорректных конструкций**

**Синтаксические ошибки:**

mov cx,vec2[di]

LR2\_comp.asm(62): warning A4031: Operand types must match

*Несоответствие типов операндов*

Размер элементов массива ‘vec2’ 1 байт, а ‘сх’ - 2 байта

mov cx,matr[bx][di]

LR2\_comp.asm(66): warning A4031: Operand types must match

*Несоответствие типов операндов*

Размер элементов матрицы ‘matr' 1 байт, а ‘сх' - 2 байта

mov mem3, [bx]

LR2\_comp.ASM(55): error A2052: Improper operand type

*Неподходящий тип операндов*

Нельзя читать из памяти и писать в память одной командой

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

LR2\_comp.asm(67): error A2055: Illegal register value

*Незаконное использование регистра*

Нельзя умножать 16-битные регистры

*Несоответствие типов операндов*

Размер элементов матрицы ‘matr' 1 байт, а ‘ах' - 2 байта

mov ax,matr[bp+bx]

LR2\_comp.asm(87): error A2046: Multiple base registers

*Слишком много базовых регистров*

Нельзя использовать более одного базового регистра

*Несоответствие типов операндов*

Размер элементов матрицы ‘matr' 1 байт, а ‘ах' - 2 байта

mov ax,matr[bp+di+si]

LR2\_comp.asm(88): error A2047: Multiple index registers

*Слишком много индексных регистров*

Нельзя использовать более одного индексного регистра

*Слишком много регистров*

Нельзя использовать более двух регистров

*Несоответствие типов операндов*

Размер элементов матрицы ‘matr' 1 байт, а ‘ах' - 2 байта

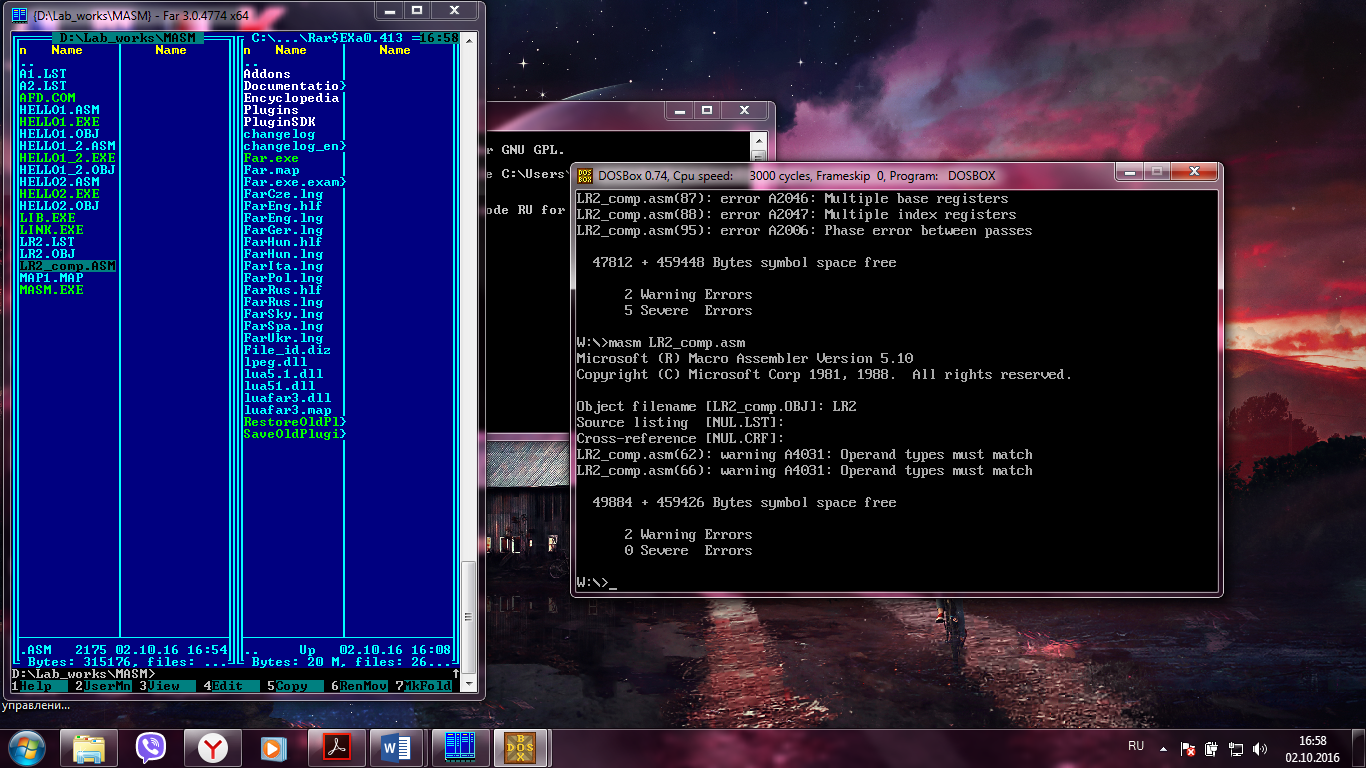
mov dx,[bp]+2

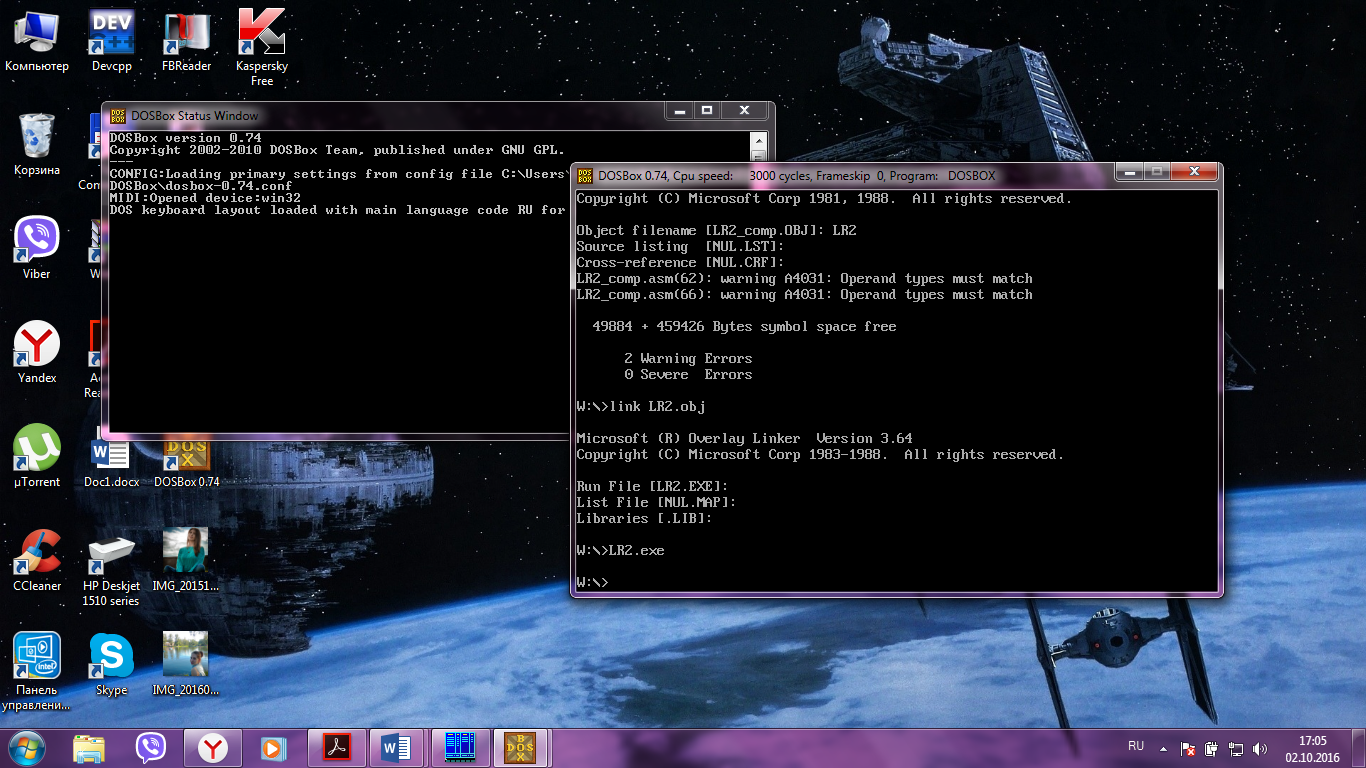
LR2\_comp.asm(95): error A2006: Phase error between passes

Неправильно используется индексная адресация со смещением.

Закомментированы операторы с ошибками в тексте программы, а операторы с предупреждениями оставлены без изменения.

3. Программа снова протранслирована, и скомпонован загрузочный модуль. В результате трансляции были получены 2 предупреждения и 0 ошибок.





4. Программа выполнена в пошаговом режиме под управлением отладчика afd с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения каждой команды.

Исходное состояние регистров:

|  |  |  |  |
| --- | --- | --- | --- |
| (AX)= 0000 | (BX)= 0000 | (CX)= 0000 | (DX)= 0000 |
| (SI)= 0000 | (DI)= 0000 | (BP)= 0000 | (SP)= 0018 |
| (CS)= 11B1 | (DS)= 119C | (ES)= 119C | (SS)= 11AC |
| (IP)=0000 | | | |

**Протокол пошагового исполнения программы**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Адрес коман-ды** | **Символьный код команды** | **16-ричный код команды** | **Содержимое регистров и ячеек памяти** | | | |
| **До выполнения** | | **После выполнения** | |
|  | **Stack** |  | **Stack** |
| 0000 | PUSH DS | 1E | (CX)=00B0  (SP)=0018  (SS)=11AC  (IP)=0000 | +0 0000  +2 0000  +4 0000  +6 0000 | (CX)=0000  (SP)=0016  (SS)=11AC  (IP)=0001 | +0 0246  +2 0000  +4 0000  +6 0000 |
| 0001 | SUB АХ, AX | 2ВС0 | (AX)=0000  (IP)=0001 | +0 119C  +2 0000  +4 0000  +6 0000 | (AX)=0000  (IP)=0003 | +0 119C  +2 0246  +4 0000  +6 0000 |
| 0003 | PUSH AX | 50 | (SP)=0016  (SS)=11AC  (AX)=0000  (IP)=0003 |  | (SP)=0016  (SS)=11AC  (AX)=0000  (IP)=0004 | +0 0000  +2 119C  +4 0246  +6 0000 |
| 0004 | MOV АХ, 11AE | В83625 | (AX)=0000  (IP)=0004 | +0 0000  +2 119C  +4 0000  +6 0000 | (AX)=11AE  (IP)=0007 |  |
| 0007 | MOV DS, AX | 8ED8 | (DS)=119C  (IP)=0007 |  | (DS)=11AE  (IP)=0009 |  |
| 0009 | MOV AX, 01F4 | B8F401 | (AX)=11AE  (IP)=0009 |  | (AX)=01F4  (IP)=000C |  |
| 000C | MOV CX,AX | 8ВС8 | (CX)=00B0  (IP)=000C |  | (CX)=01F4  (IP)=000E |  |
| 000E | MOV BL,24 | В324 | (BX)=0000  (IP)=000E |  | (BX)=0024  (IP)=0010 |  |
| 0010 | MOV BH, CE | В7СЕ | (BX)=0000  (IP)=0010 |  | (BX)=CE24  (IP)=0012 |  |
| 0012 | MOV [0002], FFCE | C7060200CEFF | (DS)=11AE  (IP)=0012 |  | (IP)=0018 |  |
| 0018 | MOV BX, 0006 | BB0E00 | (BX)=СЕ24  (IP)=0018 |  | (BX)=0006  (IP)=001B |  |
| 001B | MOV [0000], AX | А30000 | (DS)=11AE  (IP)=001B |  | (DS)=11AE  (IP)=001E |  |
| 001E | MOV AL, [BX] | 8АО7 | (AX)=01F4  (IP)=001E |  | (AX)=01FF  (IP)=0020 |  |
| 0020 | MOV AL, [BX+03] | 8А4703 | (AX)=010C  (DX)=0000  (IP)=0020 |  | (AX)=0109  (DX)=119C  (IP)=0023 |  |
| 0023 | MOV CX, [BX+03] | 8B4F03 | (CX)=01F4  (IP)=0023 |  | (CX)=0509  (IP)=0026 |  |
| 0026 | MOV DI, 0002 | BF0200 | (DI)=0000  (IP)=0026 |  | (DI)=0002  (IP)=0029 |  |
| 0029 | MOV AL, [DI+000E] | 8А850Е00 | (AX)=0109  (IP)=0029 |  | (AX)=0128  (IP)=002D |  |
| 002D | MOV CX, [DI+000E] | 8B8D0E00 | (CX)=0509  (DX)=0000  (IP)=002D |  | (CX)=3228  (DX)=119C  (IP)=0031 |  |
| 0031 | MOV BX, 0003 | ВВ0300 | (BX)=0006  (IP)=0031 |  | (BX)=0003  (IP)=0034 |  |
| 0034 | MOV AL, [BX+DI+0016] | 8А811600 | (AX)=0128  (IP)=0034 |  | (AX)=01F9  (IP)=0038 |  |
| 0038 | MOV CX, [BX+DI+0016] | 8B891600 | (CX)=3228  (DX)=119C  (IP)=0038 |  | (CX)=FAF9  (DX)=0000  (IP)=003C |  |
| 003C | MOV AX, 11AE | B8AE11 | (AX)=01F9  (IP)=003C |  | (AX)=11AE  (IP)=003F |  |
| 003F | MOV ES, AX | 8ЕС0 | (ES)=119C  (IP)=003F |  | (ES)=11AE  (IP)=0041 |  |
| 0041 | MOV AX, ES:[BX] | 268В07 | (AX)=11AE  (DX)=0000  (IP)=0041 |  | (AX)=00FF  (DX)=119C  (IP)=0044 |  |
| 0044 | MOV AX,0000 | В80000 | (AX)=00FF  (IP)=0044 |  | (AX)=0000  (IP)=0047 |  |
| 0047 | MOV ES, AX | 8ЕС0 | (ES)=11AE  (IP)=0047 |  | (ES)=0000  (IP)=0049 |  |
| 0049 | PUSH DS | 1E | (SP)=0014  (ES)=0000  (IP)=0049 |  | (SP)=0012  (ES)=0000  (IP)=004A | +0 11AE  +2 0000  +4 119C  +6 0246 |
| 004A | POP ES | 07 | (SP)=0012  (ES)=0000  (IP)=004A | +0 11AE  +2 0000  +4 119C  +6 0000 | (SP)=0014  (ES)=11AE  (IP)=004B | +0 0000  +2 119C  +4 0246  +6 0000 |
| 004B | MOV CX, ES:[BX-01] | 268B4FFF | (CX)=FAF9  (IP)=004B | +0 0000  +2 119C  +4 0000  +6 0000 | (CX)=FACE  (IP)=004F |  |
| 004F | XCHG AX, CX | 91 | (АХ)=0000  (СХ)=FFCE  (IP)=004F |  | (АХ)=FFCE  (СХ)=0000  (IP)=0050 |  |
| 0050 | MOV DI, 0002 | BF0200 | (DI)=0002  (IP)=0050 |  | (IP)=0053 |  |
| 0053 | MOV ES:[BX+DI], AX | 268901 | (ES)=11AE  (IP)=0053 |  | (ES)=11AE  (IP)=0056 |  |
| 0056 | MOV BP, SP | 8ВЕС | (BP)=0000  (IP)=0056 |  | (BP)=0014  (IP)=0058 |  |
| 0058 | MOV BP, SP | 8B5602 | (DX)=0000  (IP)=0058 |  | (DX)=119C  (IP)=005A |  |
| 005A | MOV DX, [BP+02] | CA0200 | (SP)=0014  (IP)=005A |  | (IP)=00  (IP)=005D |  |

Для корректного завершения программы необходимо, чтобы вершина стека содержала смещение и сегмент начала PSP, но вершина стека содержит значения mem1 и mem2, поэтому при выполнении команды ret 2 управление перейдет по адресу mem1:mem2, и программа не завершится корректно

**Вывод**: В ходе выполнения данной лабораторной работы изучены режимы адресации в пошаговом режиме под отладчиком, найдены и разобраны ошибки, применены практические навыки работы с методами создания и отладки программ, написанных на языке ассемблера. Результат выполнения работы содержится в программах LR2.exe.