**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №2**

**по дисциплине «Организация ЭВМ и систем»**

# **Тема: Изучение режимов адресации и формирования исполнительного адреса.**

|  |  |
| --- | --- |
| Студент гр. 3342 | Мохамед М.Х. |
| Преподаватель | Кирьянчиков В.А. |

Санкт-Петербург

2024

**Цель работы.**

Изучить режимы адресации и формирования исполнительного адреса на языке Ассемблер.

**Постановка задачи.**

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.

**Выполнение работы.**

1. В каталог с компилятором MASM был загружен файл LR2\_comp.ASM.
2. Исходный код программы был просмотрен в режиме редактирования. Была изучена структура и реализация каждого сегмента программы.
3. Согласно варианту лабораторной работы, был изменён набор значений в файле LR2\_comp.ASM.
4. Подготовка к работе и трансляция исходного кода

1) Монтирование каталога в программе-эмуляторе DosBox:

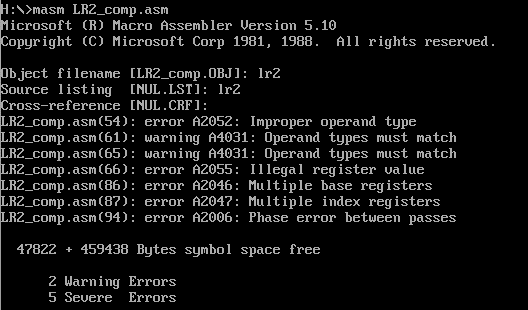


2) Переход к смонтированному каталогу:

3) Установка русификатора:

****

4) Трансляция программы с помощью команды **masm LR2\_comp.ASM**



В процессе трансляции был создан файл листинга lr2.LST. Файл листинга содержит диагностическую информацию в виде сообщений о двух предупреждениях (Warning errors) и пяти ошибках (Severe errors).

Демонстрация кода файла листинга приведена в приложении.

1. Объяснение ошибок.

**Ошибка №1:**

mov mem3,[bx]

LR2\_comp.ASM(54): error A2052: Improper operand type

Инструкция mov неспособна перенести значение из одной ячейки памяти в другую.

**Ошибка №2:**

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

LR2\_comp.ASM(66): error A2055: Illegal register value

Недопустимое значение регистра, так как масштабирование применимо только к 32-битным регистрам, а здесь используется базово-индексная адресация.

**Ошибка №3:**

mov ax, matr[bp+bx]

LR2\_comp.ASM(86): error A2046: Multiple base registers

Невозможно использовать несколько базовых регистров для адресации в операндах.

**Ошибка №4:**

mov ax, matr[bp+di+si]

LR2\_comp.ASM(87): error A2047: Multiple index registers

Невозможно использовать несколько индексных регистров для адресации в операндах.

**Ошибка №5:**

LR2\_comp.asm(94): error A2052: Phase error between passes

Внутренняя ошибка компилятора, вызванная другими ошибками в программе.

**Предупреждение №1:**

mov cx, vec2[di]

LR2\_comp.asm(61): warning A4031: Oerand types must match

Команда пытается переместить данные из вектора размером 1 байт в регистр с размером 2 байта.

**Предупреждение №2:**

mov cx, matr[bx][di]

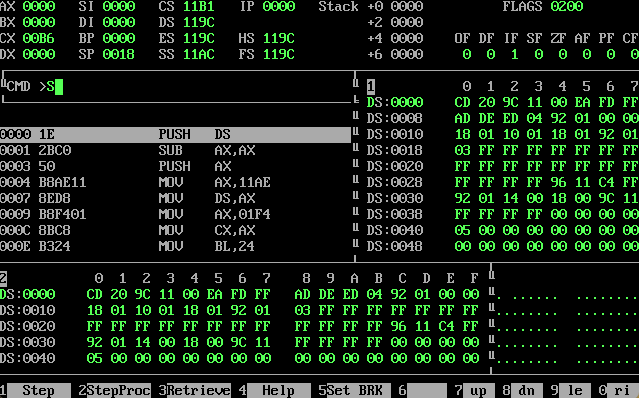
LR2\_comp.asm(65): warning A4031: Operand types must match

Команда пытается переместить данные из массива размером 1 байт в регистр с размером 2 байта.

1. В режиме редактирования были закомментированы строки с ошибками, *строки с предупреждениями остались без изменений.* Повторно проведена трансляция программы с созданием файла листинга lr2\_1.LST. Осталось только два предупреждения, ошибок обнаружено не было. Код программы содержится в приложении.



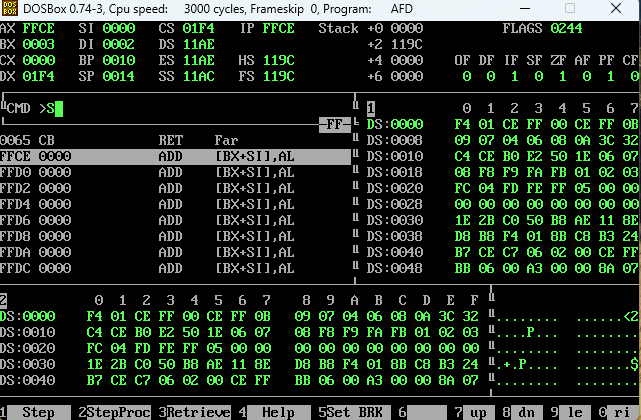
1. Командой **link lr.obj** был скомпонован **lr2.exe** и создан **lr2.map.** Программа была выполнена в режиме отладки командой **afd lr2.exe.**



Содержимое регистров до выполнения прогона программы: AX: 0000, BX: 0000, CX: 00B6, DX: 0000, CS: 11B1, DS: 119C, ES: 119C, SS: 11AC, IP: 0000

1. Результат работы программы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Символический код команды | 16-ричный код команды | Содержимое регистров и ячеек памяти | |
| До выполнения | После выполнения |
| 0000 | PUSH DS | 1E | (IP) = 0000  (SP)=0018  Stack  +0 0000  +2 0000  +4 0000  +6 0000 | (IP) = 0001  (SP)=0016  Stack  +0 119C  +2 0000  +4 0000  +6 0000 |
| 0001 | SUB AX, AX | 2BC0 | (IP) = 0001  (AX) = 0000 | (IP) = 0003  (AX) = 0000 |
| 0003 | PUSH AX | 50 | (IP) = 0003  (SP)=0016  Stack  +0 119C  +2 0000  +4 0000  +6 0000 | (IP) = 0004  (SP)=0014  Stack  +0 0000  +2 119C  +4 0000  +6 0000 |
| 0004 | MOV AX,  11AE | B8AE11 | (IP) = 0004  (AX) = 0000 | (IP) = 0007  (AX) = 11AE |
| 0007 | MOV DS, AX | 8ED8 | (IP) = 0007  (DS) = 119C | (IP) = 0009  (DS) = 11AE |
| 0009 | MOV AX,  01F4 | B8F401 | (IP) = 0009  (AX) = 11AE | (IP) = 000C  (AX) = 01F4 |
| 000C | MOV CX, AX | 8BC8 | (IP) = 000C  (CX)=00B6 | (IP) = 000E  (CX)=01F4 |
| 000E | MOV BL, 24 | B324 | (IP) = 000E  (BX) = 0000 | (IP) = 0010  (BX) = 0024 |
| 0010 | MOV BH, CE | B7CE | (IP) = 0010  (BX) = 0024 | (IP) = 0012  (BX) = CE24 |
| 0012 | MOV [0002], FFCE | C7060200C EFF | (IP) = 0012  DS:0002 00  DS:0003 00 | (IP) = 0018  DS:0002 CE  DS:0003 FF |
| 0018 | MOV BX, 0006 | BB0600 | (IP) = 0018  (BX) = CE24 | (IP) = 001B  (BX) = 0006 |
| 001B | MOV [0000], AX | A30000 | (IP) = 001B  DS:0000 00  DS:0001 00 | (IP) = 001E  DS:0000 F4  DS:0001 01 |
| 001E | MOV AL,  [BX] | 8A07 | (IP) = 001E  (AX) = 01F4 | (IP) = 0020  (AX) = 010D |
| 0020 | MOV AL,  [BX+03] | 8A4703 | (IP) = 0020  (AX) = 010D | (IP) = 0023  (AX) = 0107 |
| 0023 | MOV CX,  [BX+03] | 8B4F03 | (IP) = 0023  (CX) = 01F4 | (IP) = 0026  (CX) = 0407 |
| 0026 | MOV DI,  0002 | BF0200 | (IP) = 0026  (DI) = 0000 | (IP) = 0029  (DI) = 0002 |
| 0029 | MOV AL,  [DI+000E] | 8A850E00 | (IP) = 0029  (AX) = 0107 | (IP) = 002D  (AX) = 01C4 |
| 002D | MOX CX, [DI+000E] | 8B8D0E00 | (IP) = 002D  (CX) =0407 | (IP) = 0031  (CX) = CEC4 |
| 0031 | MOV BX,  0003 | BB0300 | (IP) = 0031  (BX) = 0006 | (IP) = 0034  (BX) = 0003 |
| 0034 | MOV AL,  [BX+DI+0016] | 8A811600 | (IP) = 0034  (AX) = 01C4 | (IP) = 0038  (AX) = 01FA |
| 0038 | MOV CX, [BX+DI+0016] | 8B891600 | (IP) = 0038  (CX) = CEC4 | (IP)=003C  (CX)=FBFA |
| 003C | MOV AX,  11AE | B8AE11 | (IP) = 003C  (AX) = 01FA | (IP) = 003F  (AX) = 11AE |
| 003F | MOV ES, AX | 8ECO | (IP) = 003F  (ES) = 119C | (IP) = 0041  (ES) = 11AE |
| 0041 | MOV AX,  ES:[BX] | 268B07 | (IP) = 0041  (AX) = 11AE | (IP) = 0044  (AX) = 00FF |
| 0044 | MOVE AX,  0000 | B80000 | (IP) = 0044  (AX) = 00FF | (IP) = 0047  (AX) = 0000 |
| 0047 | MOVE ES, AX | 8ECO | (IP) = 0047  (ES) = 11AE | (IP) = 0049  (ES) = 0000 |
| 0049 | PUSH DS | 1E | (IP) = 0049  (SP) = 0014  Stack  +0 0000  +2 119C  +4 0000  +6 0000 | (IP) = 004A  (SP) = 0012  Stack  +0 11AE  +2 0000  +4 119C  +6 0000 |
| 004A | POP ES | 07 | (IP) = 004A  (SP) = 0012  Stack  +0 11AE  +2 0000  +4 119C  +6 0000 | (IP) = 004B  (SP) = 0014  Stack  +0 0000  +2 119C  +4 0000  +6 0000 |
| 004B | MOV CX,  ES:[BX-01] | 268B4FFF | (IP) = 004B  (CX) = FBFA | (IP) = 004F  (CX) = FFCE |
| 004F | XCHG | 91 | (IP) = 004F  (AX) = 0000  (CX) = FFCE | (IP) = 0050  (AX) = FFCE  (CX) = 0000 |
| 0050 | MOV DI, 0002 | BF0200 | (IP) = 0050  (DI) = 0002 | (IP) = 0053  (DI) = 0002 |
| 0053 | MOV  ES:[BX+DI],  AX | 268901 | (IP) = 0053  DS:0005 00  DS:0006 0D | (IP) = 0056  DS:0005 CE  DS:0006 FF |
| 0056 | MOV BP, SP | 8BEC | (IP) = 0056  (BP) = 0000 | (IP) = 0058  (BP) = 0014 |
| 0058 | PUSH [0000] | FF360000 | (IP) = 0058  (SP) = 0014  Stack  +0 0000  +2 119C  +4 0000  +6 0000 | (IP) = 005C  (SP) = 0012  Stack  +0 01F4  +2 0000  +4 119C  +6 0000 |
| 005C | PUSH [0002] | FF360200 | (IP) = 005C  (SP) = 0012  Stack  +0 01F4  +2 0000  +4 119C  +6 0000 | (IP) = 0060  (SP) = 0010  Stack  +0 FFCE  +2 01F4  +4 0000  +6 119C |
| 0060 | MOV BP, SP | 8BEC | (IP) = 0060  (BP) = 0014 | (IP) = 0062  (BP) = 0010 |
| 0062 | MOV DX,  [BP+02] | 8B5602 | (IP) = 0062  (DX) = 0000 | (IP) = 0065  (DX) = 01F4 |
| 0065 | RET FAR | CB | (IP) = 0065  (SP) = 0010  (CS) = 11B1  Stack  +0 FFCE  +2 01F4  +4 0000  +6 119C | (IP) = FFCE  (SP) = 0014  (CS) = 01F4  Stack  +0 0000  +2 119C  +4 0000  +6 0000 |

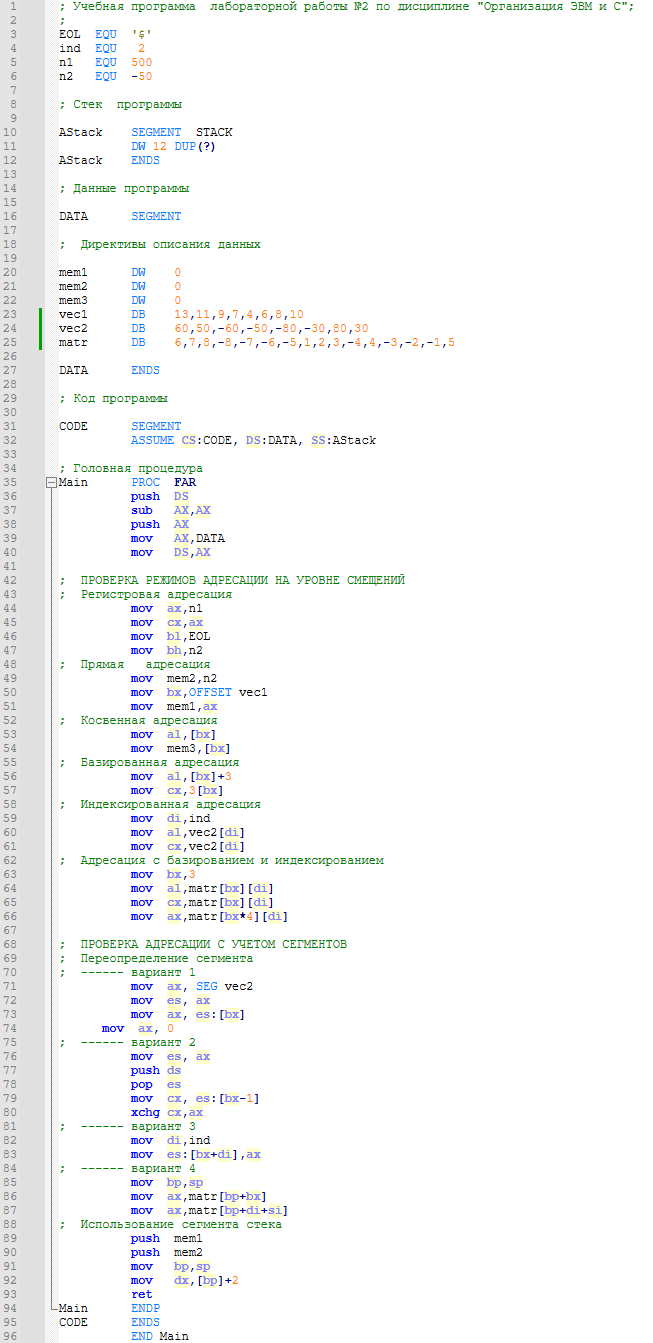


**Вывод.**

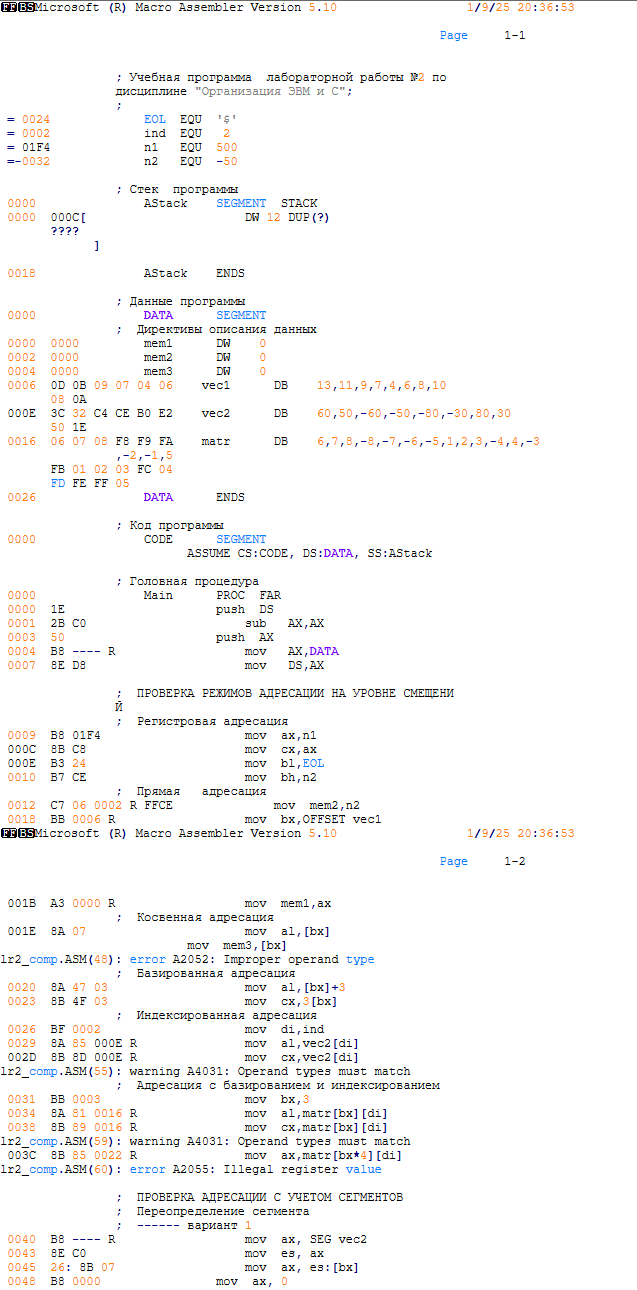
В результате выполнения лабораторной работы были изучены различные виды адресации (регистровая, прямая, косвенная, базированная, индексированная адресации и адресация с базированием и индексированием).

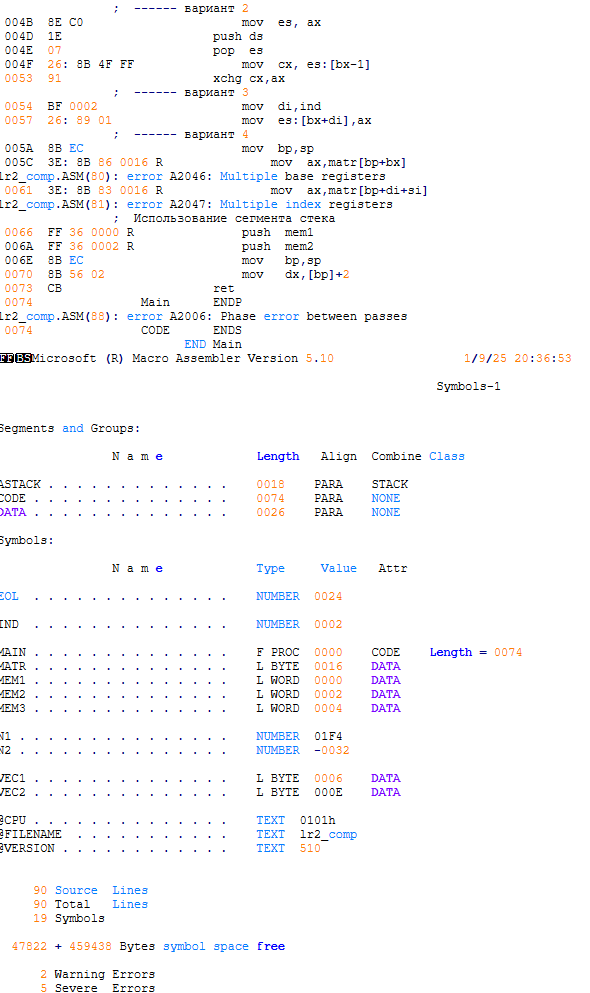
**Приложение**

*LR2\_comp.asm (до исправления)*

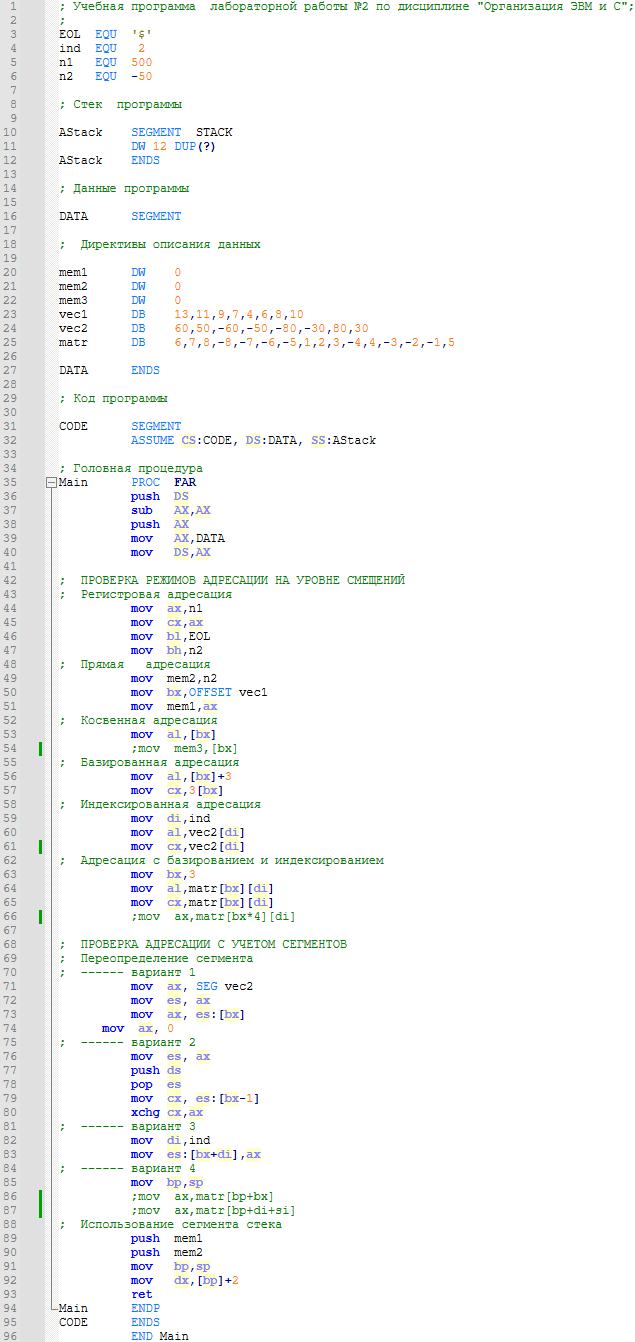
**

*LR2.lst (до исправления)*



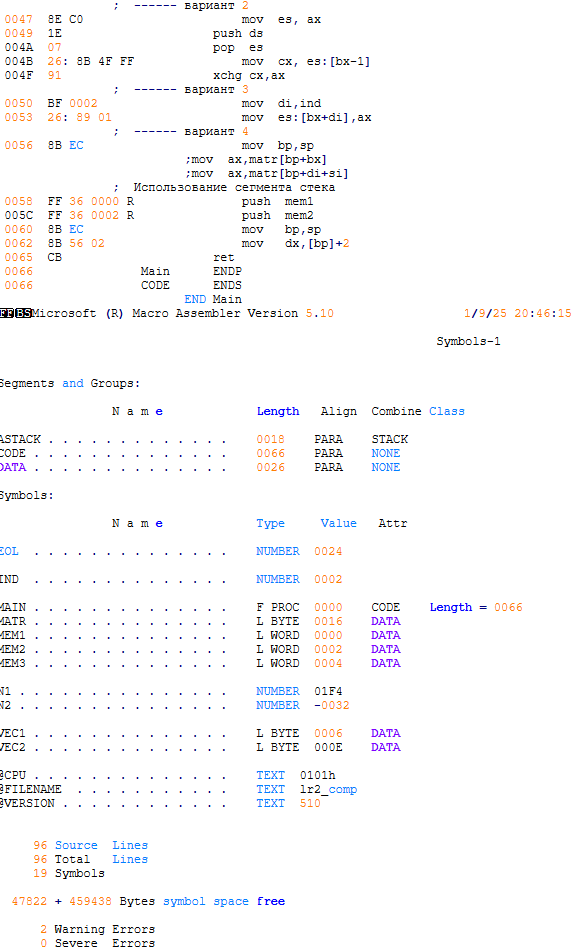


*LR2\_comp.asm (после исправления)*

**

*LR2.lst (после исправления)*





*lr2.MAP*

Start Stop Length Name Class

00000H 00017H 00018H ASTACK

00020H 00045H 00026H DATA

00050H 000B5H 00066H CODE

Program entry point at 0005:0000