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. 1381 _____ Тулегенова А.О.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить основные режимы адресации и формирования исполнительного адреса 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.

Ход выполнения работы

В ходе выполнения лабораторной работы был загружен файл lr2_comp.asm в каталог masm. Была произведена попытка протранслировать программу. Был получен файл диагностических сообщений со следующими ошибками и предупреждениями:

1. lr2_comp.asm(42): error A2052: Improper operand type
   mov mem3,[bx]
   Нельзя одновременно обращаться в память и записывать в нее.
2. lr2_comp.asm(49): warning A4031: Operand types must match
   mov cx,vec2[di]
   Несоответствие типов операндов, в cx должно храниться 2 байта, в vec2 1 байт.
3. lr2_comp.asm(53): warning A4031: Operand types must match
   mov cx,matr[bx][di]
   Несоответствие типов операндов, в cx должно храниться 2 байта, в matr 1 байт.
4. lr2_comp.asm(54): error A2055: Illegal register value
   mov ax,matr[bx*4][di]
   Нельзя масштабировать на наборе инструкций 086.
5. lr2_comp.asm(73): error A2046: Multiple base registers
   mov ax, matr[bp+bx]
   При обращении к операнду нельзя складывать два базовых регистра bp и bx.
6. lr2_comp.asm(74): error A2047: Multiple index registers
   mov ax,matr[bp+di+si]

При обращении к операнду нельзя складывать два индексных регистра di и si.

7. lr2_comp.asm(81): error A2006: Phase error between passes
   Main ENDP

Данная ошибка говорит о наличии ошибок в main.

Далее строки с ошибками были закомментированы, исправленная программа была протранслирована с созданием объектного файла и файла диагностических сообщений. Был скомпонован загрузочный файл с созданием исполняемого файла. Далее была выполнена программа в автоматическом режиме. Была произведена отладка программы в пошаговом режиме с помощью отладчика.

Таблица 1. Начальное значение регистров

CS	DS	ES	SS
1A0A	19F5	19F5	1A05

Таблица 2. Протокол работы программы lr2_comp

Адрес команды	Символьный код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	PUSH DS	1E	(DS)=19F5 (IP)=0000 (SP)=0018 Stack +0 0000 +2 0000 +4 0000 +6 0000	(DS)=19F5 (IP)=0001 (SP)=0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0001	SUB AX, AX	2BC0	(IP)=0001	(IP)=0003
0003	PUSH AX	50	(AX)=0000 (IP)=0003 (SP)=0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000	(AX)=0000 (IP)=0004 (SP)=0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000

0004	MOV AX, 1A07	B8071A	(AX)=0000 (IP)=0004	(AX)=1A07 (IP)=0007
0007	MOV DS, AX	8ED8	(AX)=1A07 (DS)=19F5 (IP)=0007	(AX)=1A07 (DS)=1A07 (IP)=0009
0009	MOV AX, 01F4	B8F401	(AX)=1A07 (IP)=0009	(AX)=01F4 (IP)=000C
000C	MOV CX, AX	8BC8	(AX)=4C07 (CX)=00B0 (IP)=000C	(AX)=1A07 (CX)=01F4 (IP)=000E
000E	MOV BL, 24	B324	(BL)=00 (IP)=000E	(BL)=24 (IP)=0010
0010	MOV BH, CE	B7CE	(BH)=00 (IP)=00010	(BH)=CE (IP)=0012
0012	MOV [0002],FFCE	C7060200CEFF	(IP)=012	(IP)=0018
0018	BB0600	MOV BX, 0006	(BX)=CE24 (IP)=0018	(BX)=0006 (IP)=001B
001B	A30000	MOV [0000], AX	(AX)=01F4 (IP)=001B	(AX)=01F4 (IP)=001E
001E	MOV AL, [BX]	8A07	(BX)=0006 (AL)=F4 (IP)=002E	((BX)=0006 (AL)=0B (IP)=0020
0020	MOV AL, [BX+03]	8A4703	(AL)=0B (BX)=0006 (IP)=0020	(AL)=0E (BX)=1A07 (IP)=0023
0023	MOV CX, [BX+03]	8B4F03	(CX)=01F4 (BX)=0006 (IP)=0023	(CX)=120E (BX)=0006 (IP)=0026
0026	MOV DI, 0002	BF0200	(DI)=0000 (IP)=0026	(DI)=0002 (IP)=0029
0029	MOV AL, [000E+DI]	8A850500	(AL)=0E (DI)=0002 (IP)=0029	(AL)=F6 (DI)=0002 (IP)=002D
002D	MOV BX, 0003	BB0300	(BX)=0006 (IP)=002D	(BX)=0003 (IP)=0030

0030	MOV AL, [0016+BX+DI]	8A811600	(AL)=F6 (BX)=0003 (DI)=0002 (IP)=0030	(AL)=04 (BX)=0003 (DI)=0002 (IP)=0034
0034	MOV AX, 1A07	B8071A	(AX)=0104 (IP)=0034	(AX)=1A07 (IP)=0037
0037	MOV ES, AX	8EC0	(ES)=19F5 (AX)=1A07 (IP)=0037	(ES)=1A07 (AX)=1A07 (IP)=0039
0039	MOV AX, ES:[BX]	268B07	(AX)=1A07 (ES)=1A07 (BX)=0003 (IP)=0039	(AX)=00FF (ES)=1A07 (BX)=0003 (IP)=003C
003C	MOV AX, 0000	B80000	(AX)=00FF (IP)=003C	(AX)=0000 (IP)=003F
003F	MOV ES, AX	8EC0	(AX)=0000 (ES)=1A07 (IP)=003F	(AX)=0000 (ES)=0000 (IP)=0041
0041	PUSH DS	1E	(DS)=1A07 (SP)=0014 (IP)=0041 Stack +0 0000 +2 19F5 +4 0000 +6 0000	(DS)=1A07 (SP)=0012 (IP)=0042 Stack +0 1A07 +2 0000 +4 19F5 +6 0000
0042	POP ES	07	(ES)=0000 (SP)=0012 (IP)=0042 Stack +0 1A07 +2 0000 +4 19F5 +6 0000	(ES)=1A07 (SP)=0014 (IP)=0043 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0043	MOV CX, ES:[BX-01]	268B4FFF	(CX)=120E (ES)=1A07 (BX)=0003 (IP)=0043	(CX)=FFCE (ES)=1A07 (BX)=0003 (IP)=0047
0047	XCHG AX, CX	91	(AX)=0000 (CX)=FFCE (IP)=0047	(AX)=FFCE (CX)=0000 (IP)=0048
0048	MOV DI, 0002	BF0200	(DI)=0002 (IP)=0048	(DI)=0002 (IP)=004B
004B	MOV ES:[BX+DI]	268901	(ES)=1A07 (BX)=0003 (DI)=0002 (IP)=004B	(ES)=1A07 (BX)=0003 (DI)=0002 (IP)=004E

004E	MOV BP. SP	8BEC	(BP)=0000 (SP)=0014 (IP)=004E	(BP)=0014 (SP)=0014 (IP)=0050
0050	PUSH [0000]	FF360000	(SP)=0014 (IP)=0050 Stack +0 0000 +2 19F5 +4 0000 +6 0000	(SP)=0012 (IP)=0054 Stack +0 01F4 +2 0000 +4 19F5 +6 0000
0054	PUSH [0002]	FF360200	(SP)=0012 (IP)=0054 Stack +0 01F4 +2 0000 +4 19F5 +6 0000	(SP)=0010 (IP)=0058 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5
0058	MOV BP, SP	8BEC	(BP)=0014 (SP)=0010 (IP)=0058	(BP)=0010 (SP)=0010 (IP)=005A
005A	MOV DX, [BP+01]	8B5602	(DX)=0000 (BP)=0010 (IP)=005A	(DX)=01F4 (BP)=0010 (IP)=005D
005D	RET Far 0002	CA0200	(IP)=005D (CS)=1A01 (SP)=0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5	(IP)=FFCE (CS)=01F4 (SP)=0016 Stack +0 19F5 +2 00000 +4 0000 +6 0000

Вывод

При выполнении лабораторной работы были изучены основные режимы адресации и формирования исполнительного адреса на языке Ассемблера

ПРИЛОЖЕНИЕ А
ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ

Неисправленный файл lr2_comp.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 11,12,13,14,18,17,16,15
     vec2 DB 10,20,-10,-20,30,40,-30,-40
     matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7,8,-6,-5
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
```

Исправленный файл lr2_comp.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 11,12,13,14,18,17,16,15
     vec2 DB 10,20,-10,-20,30,40,-30,-40
     matr DB 1,2,-4,-3,3,4,-2,-1,5,6,-8,-7,7,8,-6,-5
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
ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Неисправленный файл lr2_comp.lst:

```
          MICROSOFT    (R)    MACRO    ASSEMBLER    VERSION    5.10
10/9/22 11:17:41

PAGE     1-1


                        ;   РЦСЂРSРІСЂР°РЈРЈР°   РЁР·СЃС‡РµРSРЁСЦ
СЂРµР¶РЁР
                        ЈРSРІ                    Р°РЃСЂРµСЃР°С†РЁРЁ
РїСЂРSС†РµСЃСЃРSСЂР° І
                        NTELX86
      = 0024                 EOL EQU '$'
      = 0002                 IND EQU 2
      = 01F4                 N1 EQU 500
      =-0032                 N2 EQU -50
                        ; РЎС,РµРЄ РїСЂРSРІСЂР°РЈРЈС‹
      0000                   ASTACK SEGMENT STACK
      0000  000C[              DW 12 DUP(?)
             ????
                   ]

      0018                   ASTACK ENDS
                        ; Р”Р°РSРSС‹ Рµ РїСЂРSРІСЂР°РЈРЈС‹
      0000                   DATA SEGMENT
                        ;   Р”РЁСЂРµРЄС,РЁРІС‹   РSРїРЁСЃР°РSРЁСЦ
РЃР°РSРSС
                        ‹С…
      0000  0000             MEM1 DW 0
      0002  0000             MEM2 DW 0
      0004  0000             MEM3 DW 0
      0006  0B 0C 0D 0E 12 11     VEC1 DB 11,12,13,14,18,17,16,15
            10 0F
      000E  0A 14 F6 EC 1E 28     VEC2  DB  10,20,-10,-20,30,40,-
30,-40
            E2 D8
      0016  01 02 FC FD 03 04     MATR    DB    1,2,-4,-3,3,4,-2,-
1,5,6,-8,-7,7,8,-6,-5
            FE FF 05 06 F8 F9
            07 08 FA FB
      0026                   DATA ENDS
                        ; РЂРSРЃ РїСЂРSРІСЂР°РЈРЈС‹
      0000                   CODE SEGMENT
                             ASSUME CS:CODE, DS:DATA, SS:ASTACK
                        ; Р"PSP»PSPIPSP°СЦ РїСЂРSС†РµРЃСЃСЂР°
      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
                           ;      РЦР РЋР'Р•Р РЉРЂ     Р Р•Р-Р□РЊРЋР'
РЂР"Р Р•РЎРЂР
                           ¦Р□Р□ РЌРЂ РЈР РЋР'РЌР• РЎРЊР•Р©Р•РЌР□Р™
                           ;                  Р РµРІРЁСЃС,СЂРЅРІР°СЦ
Р°РґСЂРµСЃР°С†РЁСЦ
      0009  B8 01F4                  MOV AX,N1
      000C  8B C8                    MOV CX,AX
      000E  B3 24                    MOV BL,EOL
      0010  B7 CE                    MOV BH,N2
                           ; РЦСЂСЦРЈР°СЦ Р°РґСЂРµСЃР°С†РЁСЦ
      0012  C7 06 0002 R FFCE              MOV MEM2,N2
      0018  BB 0006 R                MOV BX,OFFSET VEC1
      001B  A3 0000 R                MOV MEM1,AX
                           ; РЉРЅСЃРІРµРЅРЅР°СЦ Р°РґСЂРµСЃР°С†РЁСЦ
      001E  8A 07                    MOV AL,[BX]
                              MOV MEM3,[BX]
     LR2_COMP.ASM(42): ERROR A2052: IMPROPER OPERAND TYPE
                           ;                   Р‘Р°Р·РЁСЂРЅРІР°РЅРЅР°СЦ
Р°РґСЂРµСЃР°С†РЁСЦ
      MICROSOFT    (R)    MACRO    ASSEMBLER    VERSION    5.10
10/9/22 11:17:41

PAGE     1-2


      0020  8A 47 03                     MOV AL,[BX]+3
      0023  8B 4F 03                     MOV CX,3[BX]
                           ; Р□РЅРґРµРЄСЃРЅРЅР°СЦ Р°РґСЂРµСЃР°С†РЁСЦ
      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
                           ;           РЂРґСЂРµСЃР°С†РЁСЦ          СЃ
Р±Р°Р·РЁСЂРЅРІР°РЅРЁРµР
                           J РЁ РЁРЅРґРґРµРЄСЃРЁСЂРЅРІР°РЅРЁРµРJ
      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
                           ;      РЦР РЋР'Р•Р РЉРЂ     Р Р•Р-Р□РЊРЋР'
РЂР"Р Р•РЎРЂР
                           ¦Р□Р□ РЎ РЈР§Р•РЎРЂРЊ РЎР•Р"РЊР•РЌРЎРЂР'
                           ;           РЦРµСЂРµРЅРЇСЂРµРґРµР»РµРЅРЁРµ
СЃРµРІРJРµРС,
                           Р°
                           ; ------ РІР°СЂРЁР°РЅС, 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
                    ; ------ PIP°CЪPËP°PSC, 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
                    ; ------ PIP°CЪPËP°PSC, 3
 0054  BF 0002              MOV DI,IND
 0057  26: 89 01            MOV ES:[BX+DI],AX
                    ; ------ PIP°CЪPËP°PSC, 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
                    ;              P□CЃPЇPSP»CЊP·PSPIP°PSPËPµ
CЃPµPIPJPµPSC,P° C
                    ЃC,PµPЄP°
 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
  MICROSOFT    (R)    MACRO    ASSEMBLER    VERSION    5.10
10/9/22 11:17:41

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

  47812 + 459448 BYTES SYMBOL SPACE FREE

      2 WARNING ERRORS
  5 SEVERE  ERRORS
```

Исправленный файл lr2_comp.lst

```
      MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10
10/9/22 12:19:14

PAGE     1-1


                        ; PЦCЂPSPICЂP°PJPJP° PËP·CЃC‡PµPSPËCЦ
CЂPµP¶PËP
                        JPSPI P°PЃCЂPµCЃP°C†PËPË
PЇCЂPSC†PµCЃCЃPSCЂP° I
                        NTELX86
      = 0024                      EOL EQU '$'
      = 0002                      IND EQU 2
      = 01F4                      N1 EQU 500
      =-0032                      N2 EQU -50

                        ; PЎC,PµPЄ PЇCЂPSPICЂP°PJPJC‹
      0000                   ASTACK SEGMENT STACK
      0000  000C[                      DW 12 DUP(?)
             ????
                    ]
```

```
      0018                        ASTACK ENDS

                                ; P”P°PSPSC‹Pµ PÏCЪPSPICЪP°PJPJC‹
      0000                        DATA SEGMENT
                                ; P”PËCЪPµPЄC,PËPIC‹ PSPÏPËCЃP°PSPËCЦ
PЃP°PSPSC
                                ‹C…
      0000  0000                        MEM1 DW 0
      0002  0000                        MEM2 DW 0
      0004  0000                        MEM3 DW 0
      0006  0B 0C 0D 0E 12 11           VEC1 DB
11,12,13,14,18,17,16,15
            10 0F
      000E  0A 14 F6 EC 1E 28           VEC2 DB 10,20,-10,-
20,30,40,-30,-40
            E2 D8
      0016  01 02 FC FD 03 04           MATR DB 1,2,-4,-3,3,4,-2,-
1,5,6,-8,-7,7
                                ,8,-6,-5
            FE FF 05 06 F8 F9
            07 08 FA FB
      0026                        DATA ENDS

                                ; PЉPSPЃ PÏCЪPSPICЪP°PJPJC‹
      0000                        CODE SEGMENT
                                  ASSUME CS:CODE, DS:DATA, SS:ASTACK
                                ; P‟PSP»PSPIPSP°CЦ PÏCЪPSC†PµPЃCЃCЪP°
      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
                                ; PЦP PЋP’P•P PЉPЂ P P•P–P□PЋPЋP’
PЂP”P P•PЎPЂP
                                ¦P□P□ PЌPЂ PJP PЋP’PЌP• PЎPЋP•P©P•PЌP□P™
                                ; P PµPIPËCЃC,CЪPSPIP°CЦ
P°PЃCЪPµCЃP°C†PËCЦ
      0009  B8 01F4                        MOV AX,N1
      000C  8B C8                           MOV CX,AX
      000E  B3 24                           MOV BL,EOL
      0010  B7 CE                           MOV BH,N2
                                ; PЦCЪCЦPJP°CЦ P°PЃCЪPµCЃP°C†PËCЦ
      0012  C7 06 0002 R FFCE                MOV MEM2,N2
      0018  BB 0006 R                       MOV BX,OFFSET VEC1
      001B  A3 0000 R                       MOV MEM1,AX
```

```
      MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10
10/9/22 12:19:14

PAGE     1-2


                        ; PЉPSCЃPIPµPSPSP°CЏ P°PЃCЂPµCЃP°C†PËCЏ
      001E  8A 07                   MOV AL,[BX]
                                  ;MOV MEM3,[BX]
                        ; P‘P°P·PËCЂPSPIP°PSPSP°CЏ
P°PЃCЂPµCЃP°C†PËCЏ
      0020  8A 47 03                    MOV AL,[BX]+3
      0023  8B 4F 03                    MOV CX,3[BX]
                        ; P□PSPЃPµPЄCЃPSP°CЏ P°PЃCЂPµCЃP°C†PËCЏ
      0026  BF 0002                 MOV DI,IND
      0029  8A 85 000E R            MOV AL,VEC2[DI]
                                  ;MOV CX,VEC2[DI]
                        ; PЂPЃCЂPµCЃP°C†PËCЏ CЃ
P±P°P·PËCЂPSPIP°PSPËPµP
                        J PË PËPSPЃPµPЄCЃPËCЂPSPIP°PSPËPµPJ
      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]
                        ; PЏP PЂP’P•P PЉPЂ P P•P–P□PЊPЂP’
PЂP”P P•PЎPЂP
                        ¦P□P□ PЎ PJP§P•PЎPЂPЊ PЎP•P‟PЊP•PЌPЎPЂP’
                        ; PЏPµCЂPµPSPЇCЂPµPЃPµP»PµPSPËPµ
CЃPµPIPJPµPSC,
                        P°
                        ; ------ PIP°CЂPËP°PSC, 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
                        ; ------ PIP°CЂPËP°PSC, 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
                        ; ------ PIP°CЂPËP°PSC, 3
      0048  BF 0002                 MOV DI,IND
      004B  26: 89 01               MOV ES:[BX+DI],AX
                        ; ------ PIP°CЂPËP°PSC, 4
      004E  8B EC                   MOV BP,SP
                                  ;MOV AX,MATR[BP+BX]
                                  ;MOV AX,MATR[BP+DI+SI]
                        ; P□CЃPЇPSP»CЊP·PSPIP°PSPËPµ
CЃPµPIPJPµPSC,P° C
                        ЃC,PµPЄP°
      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
 MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10
10/9/22 12:19:14

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_COMP
     @VERSION . . . . . .    TEXT  510


          86 SOURCE  LINES
          86 TOTAL   LINES
          19 SYMBOLS

       47800 + 459460 BYTES SYMBOL SPACE FREE
```

```
     0 WARNING ERRORS
0 SEVERE  ERRORS
```