

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, формирование

дополнительного адреса»

Студент гр. 1383

Петров А.С.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы.

Изучить режимы адресации на языке Ассемблера, исправить ошибки в готовой программе `lr2_comp.asm`.

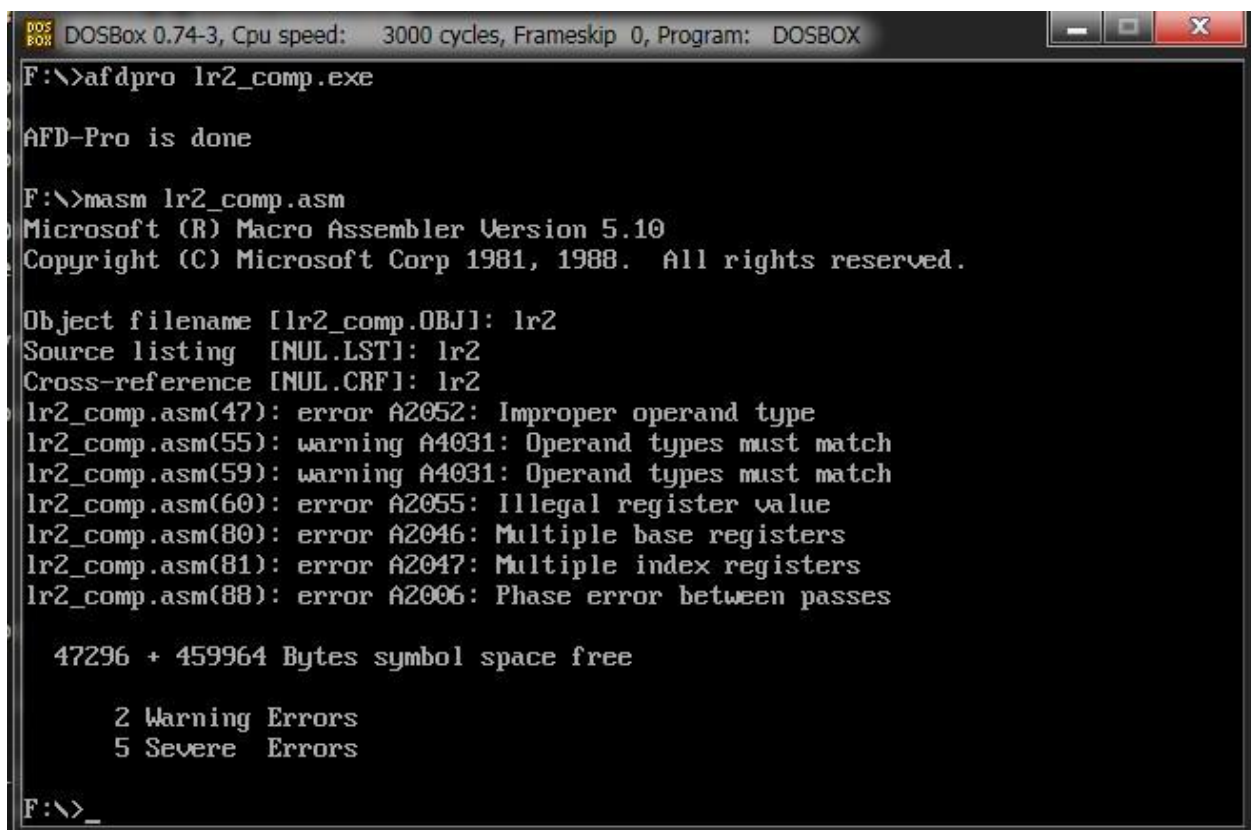
Задание на лабораторную работу.

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.

Выполнение работы.

1. Изменены исходные значения `vec1`, `vec2`, `matr` согласно варианту.
2. Протранслирована программа `lr2_comp.asm` с получением ошибок трансляции. Ошибки представлены на рисунке 1.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
F:\>afdprow lr2_comp.exe

AFD-Pro is done

F:\>masm lr2_comp.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lr2_comp.OBJ]: lr2
Source listing [NUL.LST]: lr2
Cross-reference [NUL.CRF]: lr2
lr2_comp.asm(47): error A2052: Improper operand type
lr2_comp.asm(55): warning A4031: Operand types must match
lr2_comp.asm(59): warning A4031: Operand types must match
lr2_comp.asm(60): error A2055: Illegal register value
lr2_comp.asm(80): error A2046: Multiple base registers
lr2_comp.asm(81): error A2047: Multiple index registers
lr2_comp.asm(88): error A2006: Phase error between passes

47296 + 459964 Bytes symbol space free

2 Warning Errors
5 Severe Errors

F:\>_
```

Рисунок 1 – ошибки трансляции.

3. Закомментированы следующие строки:
 - `mov mem3,[bx]` – в качестве одного из операндов `mov` должен использоваться регистр, пересылки типа память – память запрещены. (error A2052: Improper operand type).
 - `mov cx,vec2[di]` – оба операнда должны иметь одинаковую длину, `cx` – 2 байта, `DB` – 1 байт. (warning A4031: Operand types must match).
 - `mov ax,matr[bx*4][di]` – при базово-индексной адресации нельзя умножать базовый регистр `bx`, умножать можно индексный регистр `di`. (error A2055: Illegal register value).

- `mov ax,matr[bp+bx]` – должен быть только 1 базовый регистр.
(error A2046: Multiple base registers).
- `mov ax,matr[bp+di+si]` – должен быть только 1 индексный регистр.
(error A2047: Multiple index registers).

- Успешно проведена трансляция программы `lr2_comp.asm`.
- Запущено выполнение программы `LR2_COMP.EXE` под управлением отладчика.
- Начальное содержимое сегментных регистров:
`CS = 1A0A; DS = 19F5; ES = 19F5; SS = 1A05`

Таблица 1 – Результат запуска программы `LR2_COMP.EXE` в отладчике.

| Адрес команды | Символический код команды | 16-ричный код команды | Содержимое регистров и ячеек памяти | |
|---------------|---------------------------|-----------------------|--|---|
| | | | До выполнения | После выполнения |
| 0000 | PUSH DS | 1E | (SP) = 0018 (IP) = 0000 Stack: + 0 0000 | (SP) = 0016 (IP) = 0001 Stack: +0 19F5 |
| 0001 | SUB AX, AX | 2B C0 | (IP) = 0001 | (IP) = 0003 |
| 0003 | PUSH AX | 50 | (SP) = 0016 (IP) = 0003 | (SP) = 0014 (IP) = 0004 |
| | | | Stack: +0 19F5 +2 0000 | Stack: +0 0000 +2 19F5 |

| | | | | |
|------|--------------------|----------------------|----------------------------|----------------------------|
| 0004 | Mov AX,1A07 | B8 07 1A | (AX) = 0000 (IP) = 0004 | (AX) = 1A07 (IP) = 0007 |
| 0007 | MOV DS, AX | 8E D8 | (DS) = 19F5 (IP) = 0007 | (DS) = 1A07 (IP) = 0009 |
| 0009 | Mov AX,01F4 | B8 F4 01 | (AX) = 1A07 (IP) = 0009 | (AX) = 01F4 (IP) = 000C |
| 000C | Mov CX,AX | 8B C8 | (CX) = 00B0 (IP) = 000C | (CX) = 01F4 (IP) = 000E |
| 000E | MOV BL,24 | B3 24 | (BX) = 0000 (IP) = 000E | (BX) = 0024 (IP) = 0010 |
| 0010 | MOV BH,CE | B7 CE | (BX) = 0024 (IP) = 0010 | (BX) = CE24 (IP) = 0012 |
| 0012 | MOV [0002],FFCE | C7 06 02 00 CE FF | (IP) = 0012 | (IP) = 0018 |
| 0018 | MOV BX, 0006 | BB 06 00 | (BX) = CE24 (IP) = 0018 | (BX) = 0006 (IP) = 001B |
| 001B | MOV [0000],AX | A3 00 00 | (IP) = 001B | (IP) = 001E |
| 001E | MOV AL,[BX] | 8A 07 | (AX) = 01F4 (IP) = 001E | (AX) = 0112 (IP) = 0020 |
| 0020 | MOV AL, [BX+03] | 8A 47 03 | (AX) = 0112 (IP) = 0020 | (AX) = 010F (IP) = 0023 |
| 0023 | MOV CX,[BX+03] | 8B 4F 03 | (CX) = 01F4 (IP) = 0023 | (CX) = 0B0F (IP) = 0026 |

| | | | | |
|------|-------------------------|-------------|----------------------------|----------------------------|
| 0026 | MOV DI,0002 | BF 02 00 | (DI) = 0000 (IP) = 0026 | (DI) = 0002 (IP) = 0029 |
| 0029 | MOV AL,[000E+DI] | 8A 85 0E 00 | (AX) = 010F (IP) = 0029 | (AX) = 01E2 (IP) = 002D |
| 002D | MOV BX,0003 | BB 03 00 | (BX) = 0006 (IP) = 002D | (BX) = 0003 (IP) = 0030 |
| 0030 | MOV AL, [0016+BX+DI] | 8A 81 16 00 | (AX) = 01E2 (IP) = 0030 | (AX) = 01FF (IP) = 0034 |
| 0034 | MOV CX, [0016+BX+DI] | 8B 89 16 00 | (CX) = 0B0F (IP) = 0034 | (CX) = 03FF (IP) = 0038 |
| 0038 | MOV AX,1A07 | B8 07 1A | (AX) = 01FF (IP) = 0038 | (AX) = 1A07 (IP) = 003B |
| 003B | MOV ES,AX | 8E C0 | (ES) = 19F5 (IP) = 003B | (ES) = 1A07 (IP) = 003D |
| 003D | MOV AX,ES:[BX] | 26 8B 07 | (AX) = 1A07 (IP) = 003D | (AX) = 00FF (IP) = 0040 |
| 0040 | MOV AX,0000 | B8 00 00 | (AX) = 00FF (IP) = 0040 | (AX) = 0000 (IP) = 0043 |
| 0043 | MOV ES,AX | 8E C0 | (ES) = 1A07 (IP) = 0043 | (ES) = 0000 (IP) = 0045 |

| | | | | |
|------|----------------------|-------------|--|--|
| 0045 | PUSH DS | 1E | (SP) = 0014 (IP) = 0045 Stack: +0 0000 +2 19F5 +4 0000 | (SP) = 0012 (IP) = 0046 Stack: +0 1A07 +2 0000 +4 19F5 |
| 0046 | POP ES | 07 | (SP) = 0012 (ES) = 0000 (IP) = 0046 Stack: +0 1A07 +2 0000 +4 19F5 | (SP) = 0014 (ES) = 1A07 (IP) = 0047 Stack: +0 0000 +2 19F5 +4 0000 |
| 0047 | MOV CX,ES:[BX-01] | 26 8B 4F FF | (CX) = 0B0F (IP) = 0047 | (CX) = FFCE (IP) = 004B |
| 004B | XCHG AX,CX | 91 | (AX) = 0000 (CX) = FFCE (IP) = 004B | (AX) = FFCE (CX) = 0000 (IP) = 004C |
| 004C | MOV DI,0002 | BF 02 00 | (DI) = 0002 (IP) = 004C | (DI) = 0002 (IP) = 004F |
| 004F | MOV ES:[BX+DI],AX | 26 89 01 | (AX) = FFCE (IP) = 004F | (AX) = FFCE (IP) = 0052 |

| | | | | |
|------|-------------------|-------------|--|--|
| 0052 | MOV BP,SP | 8B EC | (BP) = 0000 (IP) = 0052 | (BP) = 0014 (IP) = 0054 |
| 0054 | PUSH [0000] | FF 36 00 00 | (SP) = 0014 (IP) = 0054 Stack: +0 0000 +2 19F5 +4 0000 | (SP) = 0012 (IP) = 0058 Stack: +0 01F4 +2 0000 +4 19F5 |
| 0058 | PUSH [0002] | FF 36 02 00 | (SP) = 0012 (IP) = 0058 Stack: +0 01F4 +2 0000 +4 19F5 +6 0000 | (SP) = 0010 (IP) = 005C Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5 |
| 005C | MOV BP,SP | 8B EC | (BP) = 0014 (IP) = 005C | (BP) = 0010 (IP) = 005E |
| 005E | MOV DX,[BP+02] | 8B 56 02 | (DX) = 0000 (IP) = 005E | (DX) = 01F4 (IP) = 0061 |
| 0061 | RET FAR 0002 | CA 02 00 | (SP) = 0010 (CS) = 1A0A (IP) = 0061 Stack: | (SP) = 0016 (CS) = 01F4 (IP) = FFCE Stack: |

| | | | | |
|--|--|--|---------|---------|
| | | | +0 FFCE | +0 19F5 |
| | | | +2 01F4 | +2 0000 |
| | | | +4 0000 | +4 0000 |
| | | | +6 19F5 | +6 0000 |

Выводы.

В ходе выполнения работы были изучены режимы адресации процессора в ассемблере.

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

ЛИСТИНГ ПРОГРАММЫ

Название файла: LR2_COMP.LST

Microsoft (R) Macro Assembler Version 5.10

10/30/22 13:12:1

Page 1-1

```
1 = 0024          EOL EQU '$'
2 = 0002          ind EQU 2
3 = 01F4          n1 EQU 500
4 =-0032          n2 EQU -50
5
6                ; Стек программы
7 0000            AStack SEGMENT STACK
8 0000 000C[      DW 12 DUP(?)
9      ????)
10                ]
11
12 0018            AStack ENDS
13
14                ; Данные программы
15 0000            DATA SEGMENT
16
17                ; Директивы описания данных
18 0000 0000      mem1 DW 0
19 0002 0000      mem2 DW 0
20 0004 0000      mem3 DW 0
21 0006 12 11 10 0F 0B 0Cvec1 DB
18,17,16,15,11,12,13,14
22      0D 0E
23 000E 1E 28 E2 D8 0A 14vec2 DB 30,40,-30,-
40,10,20,-10
      ,-20
24      F6 EC
25 0016 FC FD 01 02 FE FFmatr DB -4,-3,1,2,-2,-
1,3,4,5,6
      ,7,8,-8,-7,-6,-5
26      03 04 05 06 07 08
27      F8 F9 FA FB
28 0026            DATA ENDS
29
30                ; Код программы
31 0000            CODE SEGMENT
32                ASSUME CS:CODE, DS:DATA,
SS:A
      Stack
33
34                ; Головная процедура
35 0000            Main PROC FAR
```

```

36 0000 1E                                push  DS
37 0001 2B C0                            sub   AX,AX
38 0003 50                                push  AX
39 0004 B8 ---- R                        mov   AX,DATA
40 0007 8E D8                            mov   DS,AX
41
42                                ;   ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА
УРОВНЕ
                                СМЕЩЕНИЙ
43                                ;   Регистровая адресация
44 0009 B8 01F4                            mov  ax,n1
45 000C 8B C8                            mov  cx,ax
46 000E B3 24                            mov  bl,EOL
47 0010 B7 CE                            mov  bh,n2
48                                ;   Прямая   адресация
49 0012 C7 06 0002 R FFCE                mov  mem2,n2

```

Microsoft (R) Macro Assembler Version 5.10
10/30/22 13:12:1

Page 1-2

```

50 0018 BB 0006 R                        mov  bx,OFFSET vec1
51 001B A3 0000 R                        mov  mem1,ax
52                                ;   Косвенная адресация
53 001E 8A 07                            mov  al,[bx]
54                                ;mov  mem3,[bx]
55                                ;   Базированная адресация
56 0020 8A 47 03                        mov  al,[bx]+3
57 0023 8B 4F 03                        mov  cx,3[bx]
58
59                                ;   Индексная адресация
60 0026 BF 0002                            mov  di,ind
61 0029 8A 85 000E R                    mov  al,vec2[di]
62                                ;mov  cx,vec2[di]
63                                ;   Адресация с базированием и
индексиро
                                ванием
64 002D BB 0003                            mov  bx,3
65 0030 8A 81 0016 R                    mov  al,matr[bx][di]
66 0034 8B 89 0016 R                    mov  cx,matr[bx][di]
lr2_comp.asm(59): warning A4031: Operand types must match
67                                ;mov  ax,matr[bx*4][di]
68
69                                ;   ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С
УЧЕТОМ
                                СЕГМЕНТОВ
70                                ;   Переопределение сегмента
71                                ;   ----- вариант 1
72 0038 B8 ---- R                        mov  ax, SEG vec2
73 003B 8E C0                            mov  es, ax
74 003D 26: 8B 07                        mov  ax, es:[bx]

```

```

75 0040 B8 0000          mov ax, 0
76          ; ----- вариант 2
77 0043 8E C0          mov es, ax
78 0045 1E             push ds
79 0046 07             pop es
80 0047 26: 8B 4F FF    mov cx, es:[bx-1]
81 004B 91             xchg cx, ax
82          ; ----- вариант 3
83 004C BF 0002          mov di, ind
84 004F 26: 89 01      mov es:[bx+di], ax
85          ; ----- вариант 4
86 0052 8B EC          mov bp, sp
87          ;mov ax, matr[bp+bx]
88          ;mov ax, matr[bp+di+si]
89          ; Использование сегмента стека
90 0054 FF 36 0000 R    push mem1
91 0058 FF 36 0002 R    push mem2
92 005C 8B EC          mov bp, sp
93 005E 8B 56 02      mov dx, [bp]+2
94 0061 CA 0002          ret 2
95 0064                Main      ENDP
96 0064                CODE      ENDS
97                END Main

```

Microsoft (R) Macro Assembler Version 5.10
 10/30/22 13:12:1

Symbols-1

Segments and Groups:

| Class | N a m e | Length | Align | Combine |
|--------|-----------|--------|-------|---------|
| ASTACK | | 0018 | PARA | STACK |
| CODE | | 0064 | 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 = 0064 | | | |
| 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 | |

90 Source Lines
 90 Total Lines
 19 Symbols

47296 + 459964 Bytes symbol space free

1 Warning Errors
 0 Severe Errors