

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.
Вариант №8.

Студентка гр. 0382

Чегодаева Е.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить режимы адресации и формирования исполнительного адреса на языке.

Задание.

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.

Данные для варианта №8:

`vec1` 28,27,26,25,21,22,23,24

`vec2` 20,30,-20,-30,40,50,-40,-50

`matr` -8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-2,-1,5,6

Исходная программа была протранслирована с созданием файла диагностических сообщений – *lb-1.lst*, при этом было выявлено 6 ошибок, представленных в таблице 1. Затем соответствующие операторы были закомментированы.

Таблица 1. Ошибки, обнаруженные при транслировании программы.

| Операторы | Ошибка | Объяснение |
|----------------------------|------------------------------------|---|
| <code>mov mem3,[bx]</code> | error A2052: Improper operand type | Перемещение из памяти в память запрещено на уровне архитектуры. |

| | | |
|-----------------------|---|--|
| mov cx,vec2[di] | warning A4031: Operand types must match | Несоответствие размера регистра (2 байта) и элемента массива (1 байт). |
| mov cx,matr[bx][di] | warning A4031: Operand types must match | Несоответствие размера регистра (2 байта) и элемента массива (1 байт). |
| mov ax,matr[bx*4][di] | error A2055: Illegal register value | Умножение двухбайтовых регистров. |
| mov ax,matr[bp+bx] | error A2046: Multiple base registers | Использование нескольких базовых регистров для адресации. |
| mov ax,matr[bp+di+si] | error A2047: Multiple index registers | Использование нескольких базовых регистров для адресации. |

Затем программа была снова протранслирована, с созданием нового файла листинга – *lb-2.lst*. Скомпонован загрузочный модуль исправленной программы.

Далее программа выполнена в пошаговом режиме под управлением отладчика с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды. Результаты выполнения *lb2.exe* в отладчике представлены в таблице 2.

Таблица 2. Результаты выполнения *lb2.exe* в отладчике.

Начальные значения сегментных регистров:

(CS)=1A0A, (DS)=19F5, (ES)=19F5, (SS)=1A05.

| Адрес команды | Символический код команды | 16-ричный код команды | Содержимое регистров и ячеек памяти | |
|---------------|---------------------------|-----------------------|-------------------------------------|--------------------------------------|
| | | | До выполнения | После выполнения |
| 0000 | PUSH DS | 1E | SP = 0018 Stack: +0 0000 | SP = 0016 Stack: +0 19F5 |
| 0001 | SUB AX, AX | 2BC0 | AX 0000 | AX 0000 |
| 0003 | PUSH AX | 50 | SP 0016 Stack: +0 19F5 | SP 0014 Stack: +0 0000 +2 19F5 |

| | | | | |
|------|-------------------------|--------------|---------------------------------------|---|
| 0004 | MOV AX, 1A07 | B8071A | AX 0000 | AX 1A07 |
| 0007 | MOV DS, AX | 8ED8 | DS 019F5 | DS 1A07 |
| 0009 | MOV AX, 01F4 | B8F401 | AX 1A07 | AX 01F4 |
| 000C | MOV CX, AX | 8BC8 | CX 00B0 | CX 01F4 |
| 000E | MOV BL, 24 | B324 | BX 0000 | BX 0024 |
| 0010 | MOV BH, CE | B7CE | BX 0014 | BX CE24 |
| 0012 | MOV [0002], FFCE | C7060200CEFF | DS:0000 00 00 00 00 00 00 1C 1B | DS:0000 00 00 CE FF 00 00 1C 1B |
| 0018 | MOV BX, 0006 | BB0600 | BX CE24 | BX 0006 |
| 001B | MOV [0000], AX | A30000 | DS:0000 00 00 CE FF 00 00 1C 1B | DS:0000 F4 01 CE FF 00 00 1C 1B |
| 001E | MOV AL, [BX] | 8A07 | AX 01F4 | AX 011C |
| 0020 | MOV AL, [BX+03] | 8A4703 | AX 011C | AX 0119 |
| 0023 | MOV CX, [BX+03] | 8B4F03 | CX 01F4 | CX 1519 |
| 0026 | M | BF0200 | DI 0000 | DI 0002 |
| 0029 | MOV AL, [000E+DI] | 8A850E00 | AX 0119 | AX 01EC |
| 002D | MOV BX, 0003 | BB0300 | BX 0006 | BX 0003 |
| 0030 | MOV AL, [0016+BX+DI] | 8A811600 | AX 01EC | AX 01FB |
| 0034 | MOV AX, 1A07 | B8071A | AX 01FB | AX 1A07 |
| 0037 | MOV ES, AX | 8EC0 | ES 19F5 | ES 1A07 |
| 0039 | MOV AX, ES:[BX] | 268B07 | AX 1A07 | AX 00FF |
| 003C | MOV AX, 0000 | B80000 | AX 00FF | AX 0000 |
| 003F | MOV ES, AX | 8ECO | ES 1A07 | ES 0000 |
| 0041 | PUSH DS | 1E | SP 0014 Stack: +0 0000 +2 19F5 | SP 0012 Stack: +0 1A07 +2 0000 +4 19F5 |

| | | | | |
|------|-----------------------|----------|--|--|
| 0042 | POP ES | 07 | ES 0000 Stack: +0 1A07 +2 0000 +4 19F5 | ES 1A07 Stack: +0 0000 +2 19F5 |
| 0043 | MOV CX, ES:[BX-01] | 268B4FFF | CX 1519 | CX FFCE |
| 0047 | XCHG AX, CX | 91 | AX 0000 CX FFCE | AX FFCE CX 0000 |
| 0048 | MOV DI, 0002 | BF0200 | DI 0002 | DI 0002 |
| 004B | MOV ES:[BX+DI], AX | 268901 | DS:0000 F4 01 CE FF 00 00 1C 1B | DS:0000 F4 01 CE FF 00 CE FF 1B |
| 004E | MOV BP, SP | 8BEC | BP 0010 | BP 0014 |
| 0050 | PUSH [0000] | FF360000 | SP 0014 Stack: +0 0000 +2 19F5 | SP 0012 Stack: +0 01F4 +2 0000 +4 19F5 |
| 0054 | PUSH [0002] | FF360200 | SP 0012 Stack: +0 01F4 +2 0000 +4 19F5 | SP 0010 Stack: +0 FFCE +2 01F4 +4 0000 +6 19F5 |
| 0058 | MOV BP, SP | 8BEC | BP 0014 | BP 0010 |
| 005A | MOV DX, [BP+02] | 8B5602 | DX 0000 | DX 01F4 |
| 005D | RET Far 0002 | CA0200 | CS 1A0A SP 0010 Stack: +0 01F4 +2 0000 +4 19F5 | CS 01F4 SP 0016 Stack: +0 19F5 |

Исходный код программы с закомментированными ошибками представлен в приложении А. Файлы листинга до и после исправлений представлены в приложении В.

Выводы.

В ходе работы были изучены режимы адресации и формирования исполнительного адреса.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММ

Название файла: lb2.asm

```
EOL EQU '$'
ind EQU 2
n1 EQU 500
n2 EQU -50
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
DATA SEGMENT
mem1 DW 0
mem2 DW 0
mem3 DW 0
vec1 DB 28,27,26,25,21,22,23,24
vec2 DB 20,30,-20,-30,40,50,-40,-50
matr DB -8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-2,-1,5,6
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack
Main PROC FAR
    push DS
    sub AX,AX
    push AX
    mov AX,DATA
    mov DS,AX
    mov ax,n1
    mov cx,ax
    mov bl,EOL
    mov bh,n2
    mov mem2,n2
    mov bx,OFFSET vec1
    mov mem1,ax
    mov al,[bx]
    ;    mov mem3,[bx]
    mov al,[bx]+3
    mov cx,3[bx]
    mov di,ind
    mov al,vec2[di]
    ;    mov cx,vec2[di]
    mov bx,3
    mov al,matr[bx][di]
    ;    mov cx,matr[bx][di]
    ;    mov ax,matr[bx*4][di]
    mov ax, SEG vec2
    mov es, ax
    mov ax, es:[bx]
    mov ax, 0
    mov es, ax
    push ds
    pop es
    mov cx, es:[bx-1]
    xchg cx,ax
    mov di,ind
    mov es:[bx+di],ax
```

```
    mov bp,sp
    ;    mov ax,matr[bp+bx]
    ;    mov ax,matr[bp+di+si]
    push mem1
    push mem2
    mov bp,sp
    mov dx,[bp]+2
    ret 2
Main ENDP
CODE ENDS
    END Main
```


ПРИЛОЖЕНИЕ В

ФАЙЛЫ ЛИСТИНГА

Название файла: lb-1.lst (До исправления ошибок)

Microsoft (R) Macro Assembler Version 5.10

10/16/21

20:50:3

Page

1-1

```
= 0024          EOL EQU '$'
= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50
0000            AStack SEGMENT STACK
0000 000C[      DW 12 DUP(?)
           ???
           ]

0018            AStack ENDS
0000            DATA SEGMENT
0000 0000          mem1 DW 0
0002 0000          mem2 DW 0
0004 0000          mem3 DW 0
0006 1C 1B 1A 19 15 16  vec1 DB 28,27,26,25,21,22,23,24
           17 18
000E 14 1E EC E2 28 32  vec2 DB 20,30,-20,-30,40,50,-40,-50
           D8 CE
0016 F8 F9 03 04 FA FB  matr DB -8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-
2,-1,5,6
           01 02 FC FD 07 08
           FE FF 05 06
0026            DATA ENDS
0000            CODE SEGMENT
           ASSUME CS:CODE, DS:DATA, SS:AStack
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
0009  B8 01F4                mov ax, n1
000C  8B C8                  mov cx, ax
000E  B3 24                  mov bl, EOL
0010  B7 CE                  mov bh, n2
0012  C7 06 0002 R FFCE     mov mem2, n2
0018  BB 0006 R              mov bx, OFFSET vec1
001B  A3 0000 R              mov mem1, ax
001E  8A 07                  mov al, [bx]
                                mov mem3, [bx]
lb.asm(32): error A2052: Improper operand type
0020  8A 47 03              mov al, [bx]+3
0023  8B 4F 03              mov cx, 3[bx]
0026  BF 0002                mov di, ind
0029  8A 85 000E R           mov al, vec2[di]
002D  8B 8D 000E R           mov cx, vec2[di]
lb.asm(37): warning A4031: Operand types must match
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]
lb.asm(40): warning A4031: Operand types must match
003C  8B 85 0022 R           mov ax, matr[bx*4][di]
lb.asm(41): error A2055: Illegal register value
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
004B  8E C0                  mov es, ax
004D  1E                      push ds

```

Microsoft (R) Macro Assembler Version 5.10

10/16/21

20:50:3

Page

1-2

```

004E  07                      pop es

```

```

004F 26: 8B 4F FF          mov cx, es:[bx-1]
0053 91                   xchg cx,ax
0054 BF 0002              mov di,ind
0057 26: 89 01          mov es:[bx+di],ax
005A 8B EC              mov bp,sp
005C 3E: 8B 86 0016 R      mov ax,matr[bp+bx]
lb.asm(54): error A2046: Multiple base registers
0061 3E: 8B 83 0016 R      mov ax,matr[bp+di+si]
lb.asm(55): error A2047: Multiple index registers
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
lb.asm(61): error A2006: Phase error between passes
0076                   CODE ENDS
                           END Main

```

Microsoft (R) Macro Assembler Version 5.10

10/16/21

20:50:3

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 | 1b | |
| @VERSION | TEXT | 510 | |

63 Source Lines
63 Total Lines
19 Symbols

47840 + 461467 Bytes symbol space free

2 Warning Errors
5 Severe Errors

Название файла: lb-2.lst (После исправления ошибок)

Microsoft (R) Macro Assembler Version 5.10
10/16/21 20:58:3

Page 1-1

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

```

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

0018          AStack ENDS
0000          DATA SEGMENT
0000 0000          mem1 DW 0
0002 0000          mem2 DW 0
0004 0000          mem3 DW 0
0006 1C 1B 1A 19 15 16  vec1 DB 28,27,26,25,21,22,23,24
      17 18
000E 14 1E EC E2 28 32  vec2 DB 20,30,-20,-30,40,50,-40,-50
      D8 CE
0016 F8 F9 03 04 FA FB  matr DB -8,-7,3,4,-6,-5,1,2,-4,-3,7,8,-2,-
1,5,6
      01 02 FC FD 07 08
      FE FF 05 06
0026          DATA ENDS
0000          CODE SEGMENT
      ASSUME CS:CODE, DS:DATA, SS:AStack
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
0009 B8 01F4        mov ax,n1
000C 8B C8          mov cx,ax
000E B3 24          mov bl,EOL
0010 B7 CE          mov bh,n2
0012 C7 06 0002 R FFCE  mov mem2,n2
0018 BB 0006 R      mov bx,OFFSET vec1
001B A3 0000 R      mov mem1,ax
001E 8A 07          mov al,[bx]
      ;          mov mem3,[bx]
0020 8A 47 03        mov al,[bx]+3
0023 8B 4F 03        mov cx,3[bx]
0026 BF 0002        mov di,ind
0029 8A 85 000E R    mov al,vec2[di]
      ;          mov cx,vec2[di]
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]
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
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
0048 BF 0002           mov di,ind
004B 26: 89 01        mov es:[bx+di],ax
004E 8B EC            mov bp,sp
                   ; mov ax,matr[bp+bx]
                   ; mov ax,matr[bp+di+si]

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
 10/16/21 20:58:3

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 | 1b | |
| @VERSION | | TEXT | 510 | |

63 Source Lines
 63 Total Lines

19 Symbols

47806 + 461501 Bytes symbol space free

0 Warning Errors

0 Severe Errors