

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

| | | |
|------------------|-------|---------------|
| Студент гр. 1382 | _____ | Исайкин Г. И. |
| Преподаватель | _____ | Ефремов М. А |

Санкт-Петербург
2022

Цель работы.

Понять и изучить 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.

Выполнение работы.

Лабораторная работа была мной выполнена в следующих шагах:

1. Изменил в файле значения `vec1`, `vec2`, `matr`.
2. Отметил как комментарии строки, мешавшие компиляции, и занёс их в табл. 1.
3. Протранслировал программу через `masm` и получил файл `lr2.obj`.
4. Скомпоновал загрузочный файл и создал исполняемый файл `lr2.exe`.
5. Запустил отладчик для программы `lr2.exe` и зафиксировал все изменения в регистрах и ячейках памяти в табл. 2.

Таблица 1 — Ошибки, обнаруженные при первоначальной трансляции

| № строки | Код | Ошибка |
|----------|---------------------------|--|
| 41 | mov mem3,[bx] | Нельзя в mov использовать прямую адресацию с косвенной |
| 53 | mov ax,matr[bx*4] [di] | Нельзя так масштабировать |
| 72 | mov ax,matr[bp+bx] | Нельзя использовать bx с bp |
| 73 | mov ax,matr[bp+di+si] | Нельзя использовать di с si |
| 79 | ret 2 | Берёт неправильный адрес |

Таблица 2 — Изменения в регистрах и ячейках памяти при выполнении программы lr2.exe

| Начальные значения регистров | | | | | | | | | | | | | | | | |
|---------------------------------|---------------------------|----|----|------|----|----|-----------------------|------|----|----|--|------|----|--|----|----|
| CS | | | | DS | | | | ES | | | | SS | | | | |
| 1A0A | | | | 19F5 | | | | 19F5 | | | | 1A05 | | | | |
| Начальные значения по DS = 1A07 | | | | | | | | | | | | | | | | |
| DS | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 0000 | 00 | 00 | 00 | 00 | 00 | 00 | 26 | 25 | 24 | 23 | 1F | 20 | 21 | 22 | 46 | 50 |
| 0010 | BA | B0 | 32 | 3C | CE | C4 | FE | FF | 05 | 06 | F8 | F9 | 03 | 04 | FC | FD |
| 0020 | 07 | 08 | FA | FB | 01 | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Адрес команды | Символический код команды | | | | | | 16-ричный код команды | | | | Содержимое регистров и ячеек памяти | | | | | |
| | | | | | | | | | | | До | | | После | | |
| 0000 | PUSH DS | | | | | | 1E | | | | (SP) = 0018 Stack +0 0000 +2 0000 +4 0000 +6 0000 | | | (SP) = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000 | | |
| 0001 | SUB AX, AX | | | | | | 2BC0 | | | | (AX) = 0000 | | | (AX) = 0000 | | |
| 0003 | PUSH AX | | | | | | 50 | | | | (SP) = 0016 Stack +0 19F5 +2 0000 +4 0000 | | | (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000 | | |

| | | | | |
|------|-----------------------|--------------|--|--|
| | | | +6 0000 | +6 0000 |
| 0004 | MOV AX, 1A07 | B8071A | (AX) = 0000 | (AX) = 1A07 |
| 0007 | MOV DS, AX | 8ED8 | (DS) = 19F5 | (DS) = 1A07 |
| 0009 | MOV AX, 01F4 | B8F401 | (AX) = 1A07 | (AX) = 01F4 |
| 000C | MOV CX, AX | 8BC8 | (CX) = 0000 | (CX) = 01F4 |
| 000E | MOV BL, 24 | B324 | (BX) = 0000 | (BX) = 0024 |
| 0010 | MOV BH, CE | B7CE | (BX) = 0024 | (BX) = CE24 |
| 0012 | MOV [0002], FFCE | C7060200CEFF | DS:0000 +2 00 +3 00 | DS:0000 +2 CE +3 FF |
| 0018 | MOV BX, 0006 | BB0600 | (BX) = CE24 | (BX) = 0006 |
| 001B | MOV [0000], AX | A30000 | DS:0000 +0 00 +1 00 | DS:0000 +0 F4 +1 01 |
| 001E | MOV AL, [BX] | 8A07 | (AX) = 01F4 | (AX) = 0126 |
| 0020 | MOV AL, [BX+03] | 8A4703 | (AX) = 0126 | (AX) = 0123 |
| 0023 | MOV CX, [BX+03] | 8B4F03 | (CX) = 01F4 | (CX) = 1F23 |
| 0026 | MOV DI, 0002 | BF0200 | (DI) = 0000 | (DI) = 0002 |
| 0029 | MOV AL, [000E + DI] | 8A850E00 | (AX) = 0123 | (AX) = 01BA |
| 002D | MOV CX, [000E + DI] | 8B8D0E00 | (CX) = 1F23 | (CX) = BOBA |
| 0031 | MOV BX, 0003 | BB0300 | (BX) = 0006 | (BX) = 0003 |
| 0034 | MOV AL, [0016+BX +DI] | 8A811600 | (AX) = 01BA | (AX) = 01F9 |
| 0038 | MOV CX, [0016+BX+DI] | 8B891600 | (CX) = BOBA | (CX) = 03F9 |
| 003C | MOV AX, 1A07 | B8071A | (AX) = 01F9 | (AX) = 1A07 |
| 003F | MOV ES, AX | 8EC0 | (ES) = 19F5 | (ES) = 1A07 |
| 0041 | MOV AX, ES:[BX] | 268B07 | (AX) = 1A07 | (AX) = 00FF |
| 0044 | MOV AX, 0000 | B80000 | (AX) = 00FF | (AX) = 0000 |
| 0047 | MOV ES, AX | 8EC0 | (ES) = 1A07 | (ES) = 0000 |
| 0049 | PUSH DS | 1E | (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000 | (SP) = 0012 Stack +0 1A07 +2 0000 +4 19F5 +6 0000 |
| 004A | POP ES | 07 | (ES) = 0000 (SP) = 0012 | (ES) = 1A07 (SP) = 0014 |

| | | | | |
|------|--------------------|----------|---|---|
| | | | Stack +0 1A07 +2 0000 +4 19F5 +6 0000 | Stack +0 0000 +2 19F5 +4 0000 +6 0000 |
| 004B | MOV CX, ES:[BX-01] | 268B4FFF | (CX) = 03F9 | (CX) = FFCE |
| 004F | XCNG AX, CX | 91 | (AX) = 0000 (CX) = FFCE | (AX) = FFCE (CX) = 0000 |
| 0050 | MOV DI, 0002 | BF0200 | (DI) = 0002 | (DI) = 0002 |
| 0053 | MOV ES:[BX+DI], AX | 268901 | ES:0000 +5 00 +6 26 | ES:0000 +5 CE +6 FF |
| 0056 | MOV BP, SP | 8BEC | (BP) = 0000 | (BP) = 0014 |
| 0058 | PUSH [0000] | FF360000 | (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000 | (SP) = 0012 Stack +0 01F4 +2 0000 +4 19F5 +6 0000 |
| 005C | PUSH [0002] | FF360200 | (SP) = 0012 Stack +0 01F4 +2 0000 +4 19F5 +6 0000 | (SP) = 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5 |
| 0060 | MOV BP, SP | 8BEC | (BP) = 0014 | (BP) = 0010 |
| 0062 | MOV DX, [BP+02] | 8B5602 | (DX) = 0000 | (DX) = 01F4 |
| 0065 | RET Far 0002 | CA0200 | (CS) = 1A0A (SP) = 0010 Stack +0 FFCE +2 01F4 +4 0000 +6 19F5 | (CS) = 01F4 (SP) = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000 |
| FFCE | ... | ... | ... | ... |

Выводы.

Были разобраны и изучены режимы адресации и формирования исполнительного адреса на языке Ассемблера.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr2.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 38,37,36,35,31,32,33,34
vec2 DB 70,80,-70,-80,50,60,-50,-60
matr DB -2,-1,5,6,-8,-7,3,4,-4,-3,7,8,-6,-5,1,2
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

```

ДИАГНОСТИЧЕСКИЕ СООБЩЕНИЯ

Название файла: lr2.lst

Microsoft (R) Macro Assembler Version 5.10
21:59:57

10/4/22

Page

1-1

```
= 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          26 25 24 23 1F 20  vec1 DB 38,37,36,35,31,32,33,34
        21 22
000E          46 50 BA B0 32 3C  vec2 DB 70,80,-70,-80,50,60,-50,-60
        CE C4
0016          FE FF 05 06 F8 F9  matr DB -2,-1,5,6,-8,-7,3,4,-4,-3,7,8,-6,-
5,1,2
        03 04 FC FD 07 08
        FA FB 01 02

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
                ; ИЛИ 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]
                   ; Базированная адресация
0020 8A 47 03       mov al,[bx]+3
0023 8B 4F 03       mov cx,3[bx]
                   ; Индексная адресация
Microsoft (R) Macro Assembler Version 5.10
21:59:57

```

10/4/22

Page

1-2

```

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.ASM(48): warning A4031: Operand types must match
                   ; Адресация с базирование
                   ; 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.ASM(52): warning A4031: Operand types must match
                   ;mov ax,matr[bx*4][di]
                   ; ПРОВЕРКА РЕЖИМОВ АДРЕСА
                   ; ИИ С УЧЕТОМ СЕГМЕНТОВ
                   ; Переопределение сегмент
                   a
                   ; ----- вариант 1
003C B8 ---- R       mov ax, SEG vec2
003F 8E C0            mov es, ax
0041 26: 8B 07        mov ax, es:[bx]
0044 B8 0000          mov ax, 0
                   ; ----- вариант 2
0047 8E C0            mov es, ax
0049 1E               push ds
004A 07               pop es
004B 26: 8B 4F FF     mov cx, es:[bx-1]
004F 91               xchg cx,ax
                   ; ----- вариант 3
0050 BF 0002          mov di,ind
0053 26: 89 01        mov es:[bx+di],ax
                   ; ----- вариант 4
0056 8B EC            mov bp,sp
                   ;mov ax,matr[bp+bx]
                   ;mov ax,matr[bp+di+si]
                   ; Использование сегмента
                   ; тека
0058 FF 36 0000 R     push mem1
005C FF 36 0002 R     push mem2
0060 8B EC            mov bp,sp
0062 8B 56 02         mov dx,[bp]+2
0065 CA 0002          ret 2
0068                  Main ENDP

```

```

0068                      CODE ENDS
                          END Main
Microsoft (R) Macro Assembler Version 5.10
21:59:57

```

10/4/22

Symbols-1

Segments and Groups:

| | N a m e | Length | Align | Combine | Class |
|--------|-----------|--------|-------|---------|-------|
| ASTACK | | 0018 | PARA | STACK | |
| CODE | | 0068 | 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 = |
| 0068 | | | | |
| 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 | |
| @VERSION | | TEXT | 510 | |

```

82 Source Lines
82 Total Lines
19 Symbols

```

47842 + 459418 Bytes symbol space free

```

2 Warning Errors
0 Severe Errors

```