



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

**ФАКУЛЬТЕТ ИУК «Информатика и управление»**

**КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,  
информационные технологии»**

## **ПРАКТИЧЕСКАЯ РАБОТА №3**

### **«Основные типы данных. Способы адресации»**

**ДИСЦИПЛИНА: «Системное программирование»**

Выполнил: студент гр. ИУК4-31Б

  
(подпись)

( Суриков Н.С. )  
(Ф.И.О.)

Проверил:

\_\_\_\_\_  
(подпись)

( Амеличева К.А. )  
(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

**Цель:** Практическое овладение навыками разработки программного кода на языке Ассемблер. Изучение различных способов адресации операндов. Практическое освоение основных функций отладчика TD.

**Постановка задачи:**

1. Написать программу с именем Prog\_№.asm используя различные виды режимов адресации, сделать исполняемый файл, и проследить за работой в Турбоотладчике (фрагмент программы Приложение 1).

2. Опишите в сегменте данных следующую информацию:

```
B_TAB    db    1Ah,2Bh,3Ch,4Dh,5Eh,6Fh,7Ah,8Bh
W_TAB    dw    1A2Bh,3C4Dh,5E6Fh,7A8Bh
B_TAB1   db    0Ah,8 dup(1)
W_TAB1   dw    8 dup(1)
W_TAB2   dw    11h,12h,13h,14h,15h,16h,17h,18h
```

3. На основе работы программы в таблице 1, в графы 2 и 3 зафиксировать значение операнда приемника на каждом шаге программы.

**Таблица 1**

Оператор	Операнд-приемник	
	до выполнения	После выполнения
1	2	3

4. Выполнить задание варианта (Приложение\_2). На основе работы программы в таблице 1, в графы 2 и 3 зафиксировать значение операнда приемника на каждом шаге программы.

5. Ответить на контрольные вопросы

**Листинг программы 1:**

```
1  .model small
2  .stack 100h
3  .data
4      B_TAB    db  1Ah,2Bh,3Ch,4Dh,5Eh,6Fh,7Ah,8Bh
5      W_TAB    dw  1A2Bh,3C4Dh,5E6Fh,7A8Bh
6      B_TAB1   db  0Ah,8 dup(1)
7      W_TAB1   dw  8 dup(1)
8      W_TAB2   dw  11h,12h,13h,14h,15h,16h,17h,18h
9
10 .code
11     start:
12         mov ax, @data
13         mov ds, ax
14
15     ; Непосредственная адресация
16         mov al, -3                ; Расширение знака
17         mov ax, 3                 ; Переместить значение 3 в регистр ax
18         mov B_TAB, -3             ; Переместить значение -3 в массив B_TAB
19         mov W_TAB, -3             ; Переместить значение -3 в массив W_TAB
20         mov ax, 2A1Bh             ; Переместить значение 2A1Bh в регистр ax
21
22     ; Регистровая адресация
23         mov bl, al                 ; Переместить значение регистра al в регистр bl
24         mov bh, al                 ; Переместить значение регистра al в регистр bh
25         sub ax, bx                 ; Переместить разность регистров ax и bx в регистр ax
26         sub ax, ax                 ; Переместить разность регистров ax и ax в регистр ax
27
28     ; Прямая адресация
29         mov ax, W_TAB              ; Переместить в ax 1-ый элемент W_TAB
30         mov ax, W_TAB + 3          ; Переместить в регистр ax
31         mov ax, W_TAB + 5          ; Переместить в регистр ax
```

```

32      mov al, byte ptr W_TAB + 6      ; Переместить в регистр al
33      mov al, B_TAB                  ; Переместить в al 1-ый элемент B_TAB
34      mov al, B_TAB + 2              ; Переместить в регистр al
35      mov ax, word ptr B_TAB          ; Переместить в регистр ax
36      mov es:W_TAB2 + 4, ax          ; В W_TAB2 ax
37
38      ; Косвенная адресация
39      mov bx, offset B_TAB            ; Переместить в bx адрес 1-ого элемента B_TAB
40      mov si, offset B_TAB + 1        ; Переместить в si адрес 2-ого элемента B_TAB
41      mov di, offset B_TAB + 2        ; Переместить в di адрес 3-ого элемента B_TAB
42      mov dl, [bx]                   ; Переместить в dl 1-ый элемент B_TAB
43      mov dl, [si]                   ; Переместить в dl 2-ый элемент B_TAB
44      mov dl, [di]                   ; Переместить в dl 3-ый элемент B_TAB
45      mov ax, [di]                   ; Переместить в ax 3-ий элемент B_TAB
46      mov bp, bx                     ; Переместить значение регистра bx в bp
47      mov al, [bp]                   ; Переместить в al 1-ый элемент B_TAB
48      mov al, ds:[bp]                ; Переместить в al 1-ый элемент B_TAB
49      mov al, es:[bx]                ; Переместить в dl 1-ый элемент B_TAB
50      mov ax, cs:[bx]                ; Переместить в dl 1-ый элемент B_TAB
51
52      ; Базовая адресация
53      mov ax, [bx] + 2                ;
54      mov ax, [bx] + 4                ;
55      mov ax, [bx + 2]                ;
56      mov ax, [4 + bx]                ;
57      mov ax, 2 + [bx]                ;
58      mov ax, 4 + [bx]                ;
59      mov al, [bx] + 2                ;
60      mov bp, bx                     ;
61      mov ax, [bp + 2]                ;
62      mov ax, ds:[bp] + 2            ;
63      mov ax, ss:[bx + 2]            ;
64
65      ; Индексная адресация
66      mov si, 2                      ; Загрузка индекса
67      mov ah, B_TAB[si]              ; Переместить в ah 2-ой элемент B_TAB
68      mov al, [B_TAB + si]           ; Переместить в al 2-ой элемент B_TAB
69      mov bh, [si + B_TAB]           ; Переместить в bh 2-ой элемент B_TAB
70      mov bl, [si] + B_TAB           ; Переместить в bl 2-ой элемент B_TAB
71      mov bx, es:W_TAB2[si]          ;
72      mov di, 4                      ; Загрузка индекса
73      mov bl, byte ptr es:W_TAB2[di] ;
74      mov bl, B_TAB[si]              ; Переместить в bl 2-ой элемент B_TAB
75
76      ; Базовая индексная адресация
77      mov bx, offset B_TAB            ; Загрузка базы
78      mov al, 3[bx][si]              ;
79      mov ah, [bx + 3][si]           ;
80      mov al, [bx][si + 2]           ;
81      mov ah, [bx + si + 2]          ;
82      mov bp, bx                     ;
83      mov ah, 3[bp][si]              ;
84      mov ax, ds:3[bp][si]           ;
85      mov ax, word ptr ds:2[bp][si]  ;
86
87      mov ax, 4c00h
88      int 21h
89      end start

```

## Результат выполнения

Оператор	Операнд-приёмник	
	До выполнения	После выполнения

mov ax, @data	0000	11DE
mov ds, ax	11C2	11DE
mov al, -3	DE	FD
mov ax, 3	11FD	0003
mov B_TAB, -3	1A	FD
mov W_TAB, -3	1A2B	FFFD
mov ax, 2A1Bh	0003	2A1B
mov bl, al	00	1B
mov bh, al	00	1B
sub ax, bx	2A1B	0F00
sub ax, ax	0F00	0000
mov ax, W_TAB	0000	FFFD
mov ax, W_TAB + 3	FFFD	6F3C
mov ax, W_TAB + 5	6F3C	8B5E
mov al, byte ptr W_TAB + 6	5E	8B
mov al, B_TAB	8B	FD
mov al, B_TAB + 2	FD	3C
mov ax, word ptr B_TAB	8B3C	2BFD
mov es:W_TAB2 + 4, ax	48	FD
mov bx, offset B_TAB	1B1B	0000
mov si, offset B_TAB + 1	0000	0001
mov di, offset B_TAB + 2	0000	0002
mov dl, [bx]	00	FD
mov dl, [si]	FD	2B
mov dl, [di]	2B	3C
mov ax, [di]	2BFD	4D3C
mov bp, bx	0000	0000

mov al, [bp]	3C	00
mov al, ds:[bp]	00	FD
mov al, es:[bx]	FD	CD
mov ax, cs:[bx]	4DCD	D3B8
mov ax, [bx] + 2	D3B8	4D3C
mov ax, [bx] + 4	4D3C	6F5E
mov ax, [bx + 2]	6F5E	4D3C
mov ax, [4 + bx]	4D3C	6F5E
mov ax, 2 + [bx]	6F5E	4D3C
mov ax, 4 + [bx]	4D3C	6F5E
mov al, [bx] + 2	5E	3C
mov bp, bx	0000	0000
mov ax, [bp + 2]	6F3C	06FE
mov ax, ds:[bp] + 2	06FE	4D3C
mov ax, ss:[bx + 2]	4D3C	06FE
mov si, 2	0001	0002
mov ah, B_TAB[si]	06	3C
mov al, [B_TAB + si]	FE	3C
mov bh, [si + B_TAB]	00	3C
mov bl, [si] + B_TAB	00	3C
mov bx, es:W_TAB2[si]	3C3C	ADFF
mov di, 4	0002	0004
mov bl, byte ptr es:W_TAB2[di]	FF	FD
mov bl, B_TAB[si]	FD	3C
mov bx, offset B_TAB	AD3C	0000
mov al, 3[bx][si]	3C	6F
mov ah, [bx + 3][si]	3C	6F

mov al, [bx][si + 2]	6F	5E
mov ah, [bx + si + 2]	6F	5E
mov bp, bx	0000	0000
mov ah, 3[bp][si]	5E	00
mov ax, ds:3[bp][si]	005E	7A6F
mov ax, word ptr ds:2[bp][si]	7A6F	6F5E
mov ax, 4c00h	6F5E	4C00

## ПРИЛОЖЕНИЕ 2

1. Задан одномерный массив, состоящий из 10 элементов (4,7,0,5,1,2,9,3,6,8)
2. Заполнить массив числовыми константами. Размер элементов массива для четных номеров.
3. Разместить элементы массива в регистры общего назначения, используя различные способы адресации, по следующей схеме:
  - в AX – элемент массива, номер которого соответствует первой цифре Вашего дня рождения,
  - в BX – элемент массива, номер которого соответствует второй цифре Вашего дня рождения,
  - в CX – элемент массива, номер которого соответствует первой цифре Вашего месяца рождения,
  - в DX – элемент массива, номер которого соответствует второй цифре Вашего месяца рождения,
  - в SI – элемент массива, номер которого соответствует первой цифре Вашего года рождения,
  - в DI – элемент массива, номер которого соответствует второй цифре Вашего года рождения,
  - в BP – элемент массива, номер которого соответствует третьей цифре Вашего года рождения, в SP – элемент массива, номер которого соответствует четвертой цифре Вашего года рождения.
4. Назначить переменной fio Вашу фамилию, имя, отчество. Определить физические адреса заглавных букв. Разместить в регистр AL среднюю букву ФИО (значение округлить до целого).
5. По адресу равному дню и месяцу Вашего рождения (например, 23 февраля – 2302) занести год Вашего рождения, представив его как шестнадцатеричное число.
6. Определить переменную const, присвоив ей день и месяц Вашего рождения в формате описанном выше. Разместить это значение в регистре CX.
7. Поместить в переменную name уменьшительно-ласкательную форму Вашего имени. Определить адрес name.

## Листинг программы 2:

```

1 .model small
2 .stack 100h
3 .data
4     arr    dw 4, 7, 0, 5, 1, 2, 9, 3, 6, 8      ; Массив из 10 элементов
5     fio    db "Surikov Nikita Sergeevich", "$" ; ФИО

```

```

6      _const dw 0A29H                ; День и месяц рождения - 2601
7      _name  db "Nikitushka", "$"    ; Уменьшительно-ласкательная форма имени
8  .code
9      start:
10     mov ax, @data
11     mov ds, ax
12
13     mov ax, arr + 4                 ; Поместить в регистр ax 2-ой элемент arr (прямая адресация)
14
15     mov bx, offset arr              ; Поместить в регистр bx адрес 0-ого элемента arr
16     mov bx, [bx] + 12              ; Поместить в регистр bx значение 6-ого элемента arr
(базовая адресация)
17
18     mov si, offset arr              ; Поместить в регистр si адрес 0-ого элемента arr
19     mov cx, [si]                   ; Поместить в регистр cx значение 0-ого элемента arr
(косвенная адресация)
20
21     mov si, 2                      ; Определяем значение индекса
22     mov dx, arr[si]                ; Поместить в регистр dx значение 1-ого элемента arr
(индексная адресация)
23
24     mov si, ax                     ; Поместить в регистр si значение 2-ого элемента arr
(регистровая адресация)
25
26     mov di, arr                    ; Поместить в регистр di значение 0-ого элемента arr (прямая
адресация)
27
28     mov bp, di                     ; Поместить в регистр bp значение 0-ого элемента arr
(регистровая адресация)
29
30     mov sp, arr + 10                ; Поместить в регистр sp значение 5-его элемента arr (прямая
адресация)
31
32     lea bx, fio                     ; Поместить в регистр bx адрес буквы S
33     lea cx, 8[bx]                   ; Поместить в регистр cx адрес буквы N
34     lea dx, 15[bx]                  ; Поместить в регистр dx адрес буквы S
35     mov al, byte ptr fio + 12       ; Поместить в регистр al значение буквы t
36
37     mov ds:[0A29H], 7D5H            ; Поместить в регистр ds по адресу 0A29H значение year
38
39     mov cx, _const                  ; Поместить в регистр cx значение переменной _const
40
41     lea bx, _name                   ; Поместить в регистр bx адрес переменной _name (адрес
первой буквы)
42
43     mov ax, 4C00h
44     int 21h
45 end start

```

## Результат выполнения

Оператор	Операнд-приёмник	
	До выполнения	После выполнения
mov ax, @data	0000	11D7
mov ds, ax	11C2	11D7
mov ax, arr + 4	11D7	0000
mov bx, offset arr	0000	0000

mov bx, [bx] + 12	0000	0009
mov si, offset arr	0000	0000
mov cx, [si]	0000	0004
mov si, 2	0000	0002
mov dx, arr[si]	0000	0007
mov si, ax	0002	0000
mov di, arr	0000	0004
mov bp, di	0000	0004
mov sp, arr + 10	0100	0002
lea bx, fio	0009	0014
lea cx, 8[bx]	0004	001C
lea dx, 15[bx]	0007	0023
mov al, byte ptr fio + 12	00	74
mov ds:[0A29H], 7D5H	0000	11D7
mov cx, _const	001C	0A29
lea bx, _name	0014	0030
mov ax, 4C00h	0074	4C00

**Вывод:** в ходе выполнения практического задания были получены навыки разработки программного кода на языке Ассемблер, изучены различные способы адресации операндов, освоены основные функции отладчика TD.