



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»

ЛАБОРАТОРНАЯ РАБОТА №2

«Команды пересылки данных»

ДИСЦИПЛИНА: «Системное программирование»

Выполнил: студент гр. ИУК4-32Б


(Подпись)

(____ Зудин Д.В. ____)
(Ф.И.О.)

Проверил:


(Подпись)

(____ Амеличева К.А. ____)
(Ф.И.О.)

Дата сдачи (защиты): 04.10.2022

Результаты сдачи (защиты):

- Балльная оценка:

5 + 2 балла

- Оценка:

защито.

Калуга, 2022 г.

Цель: практическое овладение навыками разработки программного кода на языке Ассемблер; изучение команд передачи данных; практическое освоение основных функций отладчика TD.

Постановка задачи:

1. Изучить методические указания и рекомендованную литературу;
2. Написать программу Prog_3 с помощью шаблона;
3. Задать начальные значения переменных A, B, C, D в сегменте данных в соответствии с вариантом;
4. Проследить за работой в Турбоотладчике, заполнить таблицу для строк программы с 11 по 35;
5. Написать программу Prog_4 согласно условию.

Вариант №7

Листинг программы Prog_3

```
.model small
.stack 100h
.data
    A db ?           ;Зарезервировать место
    B db ?           ;в памяти для
    C db ?           ;переменных
    D db ?           ;A, B, C, D

.code               ;Открыть сегмент кодов
start:             ;Инициализировать
    mov ax, @data
    mov ds, ax     ;сегментный регистр ds
    mov A, 7h      ;Инициализировать
    mov B, 12      ;переменные A, B, C, D
    mov C, 1Dh     ;значениями 7-ого
    mov D, 9       ;варианта
    mov al, A      ;Поместить в регистр al значение переменной A
    mov ah, B      ;Поместить в регистр ah значение переменной B
    xchg al, ah    ;Поменять местами значения регистров al и ah
    mov bx, 3E10h  ;Поместить в регистр bx число 3E10h
    mov cx, bx     ;Поместить в регистр cx значение регистра bx
    push bx        ;Поместить в стек значение регистра bx
    push cx        ;Поместить в стек значение регистра cx
    push ax        ;Поместить в стек значение регистра ax
```

```

    lea si, C      ;Загрузить в регистр SI адрес переменной C
    mov ax, si     ;Поместить в регистр ax значение регистра SI
    lea di, D      ;Загрузить в регистр DI адрес переменной D
    mov bx, di     ;Поместить в регистр bx значение регистра DI
    pop ax         ;Извлечь из стека значение и поместить его в регистр
ax
    pop cx         ;Извлечь из стека значение и поместить его в регистр
cx
    pop bx         ;Извлечь из стека значение и поместить его в регистр
bx
    mov bx, ax     ;Поместить в регистр bx значение регистра ax
    mov A, al      ;Поместить в переменную A значение регистра al
    mov B, ah      ;Поместить в переменную B значение регистра ah
    mov C, 0       ;Поместить в переменную C значение 0
    mov ax, 4c00h  ;Завершить программу
    int 21h        ;с помощью DOS
end start         ;Закрывать процедуру

```

Результат выполнения программы

Вариант №7			
№ строки	Команда Ассемблера	Машинный код	Состояние регистров
11	mov ax, @data	B8CC0A	ax=0ACC, ip=0003
12	mov ds, ax	8ED8	ds=0ACC, ip=0005
13	mov A, 7h	C606000007	ip=000A
14	mov B, 12	C60601000C	ip=000F
15	mov C, 1Dh	C60602001D	ip=0014
16	mov D, 9	C606030009	ip=0019
17	mov al, A	A00000	ax=0A07, ip=001C
18	mov ah, B	8A260100	ax=0C07, ip=0020
19	xchg al, ah	86C4	ax=070C, ip=0022
20	mov bx, 3E10h	BB103E	bx=3E10, ip=0025
21	mov cx, bx	8BCB	cx=3E10, ip=0027
22	push bx	53	sp=00FE, ip=0028
23	push cx	51	sp=00FC, ip=0029
24	push ax	50	sp=00FA, ip=002A
25	lea si, C	BE0200	si=0002, ip=002D
26	mov ax, si	8BC6	ax=0002, ip=002F

27	lea di, D	BF0300	di=0003, ip=0032
28	mov bx, di	8BDF	bx=0003, ip=0034
29	pop ax	58	ax=070C, sp=00FC, ip=0035
30	pop cx	59	sp=00FE, ip=0036
31	pop bx	5B	bx=3E10, sp=0100, ip=0037
32	mov bx, ax	8BD8	bx=070C, ip=0039
33	mov A, al	A20000	ip=003C
34	mov B, ah	88260100	ip=0040
35	mov C, 0	C606020000	ip=0045

Листинг программы Prog_4

```

title prg_3                      ;Lab2, Zudin, IUK4
.model small                     ;Количество сегментов в программе - 2
.stack 200h                      ;Размер стека - 512б
.data                            ;Сегмент данных
    day    db 25                 ;День рождения - 25
    qwer   dw 4321h              ;Число qwer
    month  dw 01                 ;Месяц рождения - январь
    year   dw 7D3h               ;Год рождения - 2003
    min    db 75                 ;Минимальная оценка - 75
    max    db 100                ;Максимальная оценка - 100
    age     db 19                ;Возраст - 19
    mas_1  db 01 dup(19)         ;Массив длиной 01, каждый символ = 19
    mas_2  dw 75 dup(100 dup(?)) ;Двумерный массив, число строк - min,
    столбцов - max
    mes     db 'Zudin', '$'      ;Фамилия

.code                            ;Сегмент кода
start:                          ;Инициализировать
    mov ax, @data                ;
    mov ds, ax                  ;сегментный регистр ds
    mov di, year                 ;Переместить значение переменной year в регистр
DI
    mov al, day                  ;Переместить значение переменной day в регистр
al
    mov cx, month                ;Переместить значение переменной month в регистр
cx

```

```

mov es, month          ;Переместить значение переменной month в регистр
es
mov ax, word ptr day   ;Переместить значение преобразованной переменной
day в регистр ax
push ax                ;Поместить в стек значение регистра ax
push cx                ;Поместить в стек значение регистра cx

mov ax, qwer           ;Переместить значение переменной qwer в регистр
ax
mov month, ax          ;Переместить значение регистра ax в переменную
month
mov qwer, es           ;Переместить значение регистра es в переменную
qwer

lea si, day            ;Поместить в регистр SI адрес переменной day
;mov cs, ds            Сегментный регистр может быть только приёмником!
pop dx                ;Извлечь из стека значение и переместить его в
регистр dx
pop cx                ;Извлечь из стека значение и поместить его в
регистр cx
mov ah, 09h           ;Вывод строки на экран
mov dx, offset mes     ;Поиск адреса первого символа строки
int 21h               ;Прерывание DOS
mov ax, 4c00h          ;Завершить программу
int 21h               ;с помощью DOS
end start              ;Закрыть процедуру

```

Результат выполнения программы

Вариант №7			
№ строки	Команда Ассемблера	Машинный код	Состояние регистров
18	mov ax, @data	B8CB0A	ax=0ACB, ip=0003
19	mov ds, ax	8ED8	ds=0ACB, ip=0005
20	mov di, year	8B3E0500	di=07D3, ip=0009
21	mov al, day	A00000	al=19, ip=000C
22	mov cx, month	8B0E0300	cx=0001, ip=0010
23	mov es, month	8E060300	es=0001, ip=0014
24	mov ax, word ptr day	A10000	ax=2119, ip=0017
25	push ax	50	sp=01FE, ip=0018

26	push cx	51	sp=01FC, ip=0019
27	mov ax, qwer	A10100	ax=4321, ip=001C
28	mov month, ax	A30300	ip=001F
29	mov qwer, es	8C060100	ip=0023
30	lea si, day	BE0000	ip=0026
31	pop dx	5A	dx=0001, sp=01FE, ip=0027
32	pop cx	59	cx=2119, sp=0200, ip=0028
33	mov ah, 09h	B409	ah=09, ip=002A
34	mov dx, offset mes	BAA33A	dx=3AA3, ip=002D

Практическое задание

Листинг программы Prog_5

```
.model small          ;Количество сегментов - 2
.stack 100h           ;Размер стека - 256б
.data                 ;Сегмент данных

    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

.code                 ;Сегмент кода
start:

    mov ax, @data     ;
    mov ds, ax        ;

    ;Непосредственная адресация
    mov al, -3         ;Расширение знака
    mov ax, 3          ;Переместить значение 3 в регистр ax
    mov B_TAB, -3      ;Переместить значение -3 в массив B_TAB
    mov W_TAB, -3      ;Переместить значение -3 в массив W_TAB
    mov ax, 2A1Bh      ;Переместить значение 2A1Bh в регистр ax

    ;Регистровая адресация
    mov bl, al         ;Переместить значение регистра al в регистр bl
    mov bh, al         ;Переместить значение регистра al в регистр bh
```

```

sub ax, bx          ;Переместить разность регистров ax и bx в регистр ax
sub ax, ax          ;Переместить разность регистров ax и ax в регистр ax

;Прямая адресация
mov ax, W_TAB        ;Переместить в ax 1-ый элемент W_TAB
mov ax, W_TAB + 3     ;Переместить в регистр ax
mov ax, W_TAB + 5     ;Переместить в регистр ax
mov al, byte ptr W_TAB + 6 ;Переместить в регистр al
mov al, B_TAB        ;Переместить в al 1-ый элемент B_TAB
mov al, B_TAB + 2     ;Переместить в регистр al
mov ax, word ptr B_TAB ;Переместить в регистр ax
mov es:W_TAB2 + 4, ax ;В W_TAB2 ax

;Косвенная адресация
mov bx, offset B_TAB  ;Переместить в bx адрес 1-ого элемента
B_TAB
mov si, offset B_TAB + 1 ;Переместить в si адрес 2-ого элемента
B_TAB
mov di, offset B_TAB + 2 ;Переместить в di адрес 3-ого элемента
B_TAB
mov dl, [bx]           ;Переместить в dl 1-ый элемент B_TAB
mov dl, [si]           ;Переместить в dl 2-ый элемент B_TAB
mov dl, [di]           ;Переместить в dl 3-ый элемент B_TAB
mov ax, [di]           ;Переместить в ax 3-ий элемент B_TAB
mov bp, bx             ;Переместить значение регистра bx в bp
mov al, [bp]           ;Переместить в al 1-ый элемент B_TAB
mov al, ds:[bp]        ;Переместить в al 1-ый элемент B_TAB
mov al, es:[bx]        ;Переместить в dl 1-ый элемент B_TAB
mov ax, cs:[bx]        ;Переместить в dl 1-ый элемент B_TAB

;Базовая адресация
mov ax, [bx] + 2       ;
mov ax, [bx] + 4       ;
mov ax, [bx + 2]       ;
mov ax, [4 + bx]       ;
mov ax, 2 + [bx]       ;
mov ax, 4 + [bx]       ;
mov al, [bx] + 2       ;
mov bp, bx             ;
mov ax, [bp + 2]       ;
mov ax, ds:[bp] + 2    ;
mov ax, ss:[bx + 2]    ;

```

```

;Индексная адресация
mov si, 2 ;Загрузка индекса
mov ah, B_TAB[si] ;Переместить в ah 2-ой элемент B_TAB
mov al, [B_TAB + si] ;Переместить в al 2-ой элемент B_TAB
mov bh, [si + B_TAB] ;Переместить в bh 2-ой элемент B_TAB
mov bl, [si] + B_TAB ;Переместить в bl 2-ой элемент B_TAB
mov bx, es:W_TAB2[si] ;
mov di, 4 ;Загрузка индекса
mov bl, byte ptr es:W_TAB2[di] ;
mov bl, B_TAB[si] ;Переместить в bl 2-ой элемент B_TAB

;Базовая индексная адресация
mov bx, offset B_TAB ;Загрузка базы
mov al, 3[bx][si] ;
mov ah, [bx + 3][si] ;
mov al, [bx][si + 2] ;
mov ah, [bx + si + 2] ;
mov bp, bx ;
mov ah, 3[bp][si] ;
mov ax, ds:3[bp][si] ;
mov ax, word ptr ds:2[bp][si] ;

mov ax, 4c00h ;
int 21h ;
end start

```

Результат выполнения

Оператор	Операнд-приёмник	
	До выполнения	После выполнения
mov ax, @data	0000	0AD3
mov ds, ax	0AB7	0AD3
mov al, -3	D3	FD
mov ax, 3	0AFD	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

Листинг программы Prog_6

```

.model small ;Количество сегментов - 2
.stack 100h ;Размер стека - 256 б
.data ;Сегмент данных
    arr dw 1A2Bh, 3C4Dh, 5E6Fh, 7A8Bh, 2ABCh, 1BCDh, 2FEFh, 7A9Bh, 4DABh,
5ECAh ;Массив слов из 10 элементов
    fio db "Zudin Daniil Vasilevich", "$" ;ФИО
    _const dw 09C5h ;День и месяц рождения - 2501
    _name db "Danechka", "$" ;Уменьшительно-ласкательная форма имени -
Данечка

.code ;Сегмент кода
start:
    mov ax, @data ;Поместить адрес сегмента данных
    mov ds, ax ;в регистр ds

    mov ax, arr + 4 ;Поместить в регистр ax 2-ой элемент arr (прямая
адресация)

```

```

    mov bx, offset arr      ;Поместить в регистр bx адрес 0-ого элемента arr
    mov bx, [bx] + 10 ;Поместить в регистр bx значение 5-ого элемента arr
(базовая адресация)

    mov si, offset arr      ;Поместить в регистр si адрес 0-ого элемента arr
    mov cx, [si]            ;Поместить в регистр cx значение 0-ого элемента
arr (косвенная адресация)

    mov si, 2               ;Определяем значение индекса
    mov dx, arr[si]         ;Поместить в регистр dx значение 1-ого элемента
arr (индексная адресация)

    mov si, ax              ;Поместить в регистр si значение 2-ого элемента
arr (регистровая адресация)

    mov di, arr             ;Поместить в регистр di значение 0-ого элемента
arr (прямая адресация)

    mov bp, di              ;Поместить в регистр bp значение 0-ого элемента
arr (регистровая адресация)

    mov sp, arr + 6        ;Поместить в регистр sp значение 3-его элемента arr
(прямая адресация)

    lea bx, fio             ;Поместить в регистр bx адрес буквы Z
    lea cx, 6[bx]           ;Поместить в регистр cx адрес буквы D
    lea dx, 13[bx]         ;Поместить в регистр dx адрес буквы V
    mov al, byte ptr fio + 12 ;Поместить в регистр al значение буквы l

    mov ds:[09C5h], 7D3h;Поместить в регистр ds по адресу 09C5 значение year

    mov cx, _const          ;Поместить в регистр cx значение переменной cnst

    lea bx, _name           ;Поместить в регистр bx адрес переменной _name
(адрес первой буквы)

    lea bx, _name + 8       ;Поместить в регистр bx адрес следующего за
_name элемента

    mov [bx], byte ptr '*' ;Поместить по адресу в регистре bx символ *
    mov 1[bx], byte ptr ' ' ;Поместить по адресу в регистре bx + 1 символ ' '
    mov 2[bx], 2501         ;Поместить по адресу в регистре bx + 2 число 2501

```

```

mov 4[bx], 7D3h      ;Поместить по адресу в регистре bx + 4 число
2003

mov ax, 4C00h        ;Завершение
int 21h              ;программы
end start             ;Закреть программу

```

Результат выполнения программы

Оператор	Операнд-приёмник	
	До выполнения	После выполнения
mov ax, @data	0000	0ACD
mov ds, ax	0AB7	0ACD
mov ax, arr + 4	0ACD	5E6F
mov bx, offset arr	0000	0000
mov bx, [bx] + 10	0000	1BCD
mov si, offset arr	0000	0000
mov cx, [si]	0000	1A2B
mov si, 2	0000	0002
mov dx, arr[si]	0000	3C4D
mov si, ax	0002	5E6F
mov di, arr	0000	1A2B
mov bp, di	0000	1A2B
mov sp, arr + 6	0100	7A8B
lea bx, fio	1BCD	0014
lea cx, 6[bx]	1A2B	001A
lea dx, 13[bx]	3C4D	0021
mov al, byte ptr fio + 12	6F	20
mov ds:[09C5h], 7D3h	0000	D307
mov cx, _const	001A	09C5
lea bx, _name	0014	002E
lea bx, _name + 8	002E	0036
mov [bx], byte ptr '*'	00	2A
mov 1[bx], byte ptr ' '	00	20
mov 2[bx], 2501	0000	0125

mov 4[bx], 7D3h	0000	D307
mov ax, 4C00h	5E20	4C00

Выводы:

В ходе работы были сформированы практические навыки разработки программного кода на языке Ассемблер; изучены команды передачи данных; освоены основные функции отладчика TD.