Министерство науки и высшего образования РФ

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №7

по дисциплине «ЭВМ и переферийные устройства»

на тему «Работа с клавиатурой»

Выполнили: студенты группы 22ВВП1

Беляев Д. И.

Демин М. С.

Сергунов М. Р.

Приняли:

Никишин К. И.

Патунин Д. В.

Пенза 2025

**Цель работы**

Изучить организацию ввода данных с клавиатуры

**Задание**

Задание1: Разработать и отладить программу ввода в память ASCII и скан-кодов нажатой клавиши с использованием функции 0 прерывания 16Н BIOS.Определить с помощью программы значения ASCII кодов фамилии студента, набранной английскими буквами.

Задание2: Разработать и отладить программы ввода с клавиатуры с эхо-выводом на видеотерминал, используя функцию 0 прерывания 16Н и функции 0 и 0EH прерывания 10Н BIOS.

Задание3: Разработать и отладить программы подсчета "средней" даты дня рождения членов учебной бригады. Даты вводятся после вывода приглашения (фамилии) на экран в режиме эхо-печати как показано ниже (красным выделена вводимая информация):

IVANOB: 03.11.92

PETROVA: 13.06.90

birthday: 08.08.91

При программировании ввода с клавиатуры предусмотреть предварительную очистку буфера клавиатура, используя функции 1 прерывания 16Н BIOS. При вычислениях использовать округления в меньшую сторону для четных бригад и в большую - для нечетных.

**Ход работы**

*Задание 1*

Листинг:

stseg segment para stack

dw 64 dup(?)

stseg ends

dseg segment para

; Task 1: keyboard input

task1Msg1 db 13,10,'Press keys to enter your surname (English).',13,10,'Press Enter to finish.$'

task1Msg2 db 13,10,'ASCII codes: ','$'

task1Msg3 db 13,10,'Scan codes: ','$'

asciiBuf db 50 dup(0)

scanBuf db 50 dup(0)

task1\_count dw 0

dseg ends

cseg segment para

assume cs:cseg, ds:dseg, ss:stseg

main proc far

push ds

xor ax,ax

push ax

mov ax,dseg

mov ds,ax

;---------------------------- ЗАДАНИЕ 1 ----------------------------

xor si,si ; Индекс для буферов

; Вывод приглашения

mov dx, OFFSET task1Msg1

mov ah,09h

int 21h

; Чтение клавиш

ReadLoop:

mov ah,00h

int 16h ; AH=scan, AL=ASCII

cmp al,13 ; Enter?

je ShowResults

mov asciiBuf[si],al

mov scanBuf[si],ah

inc si

cmp si,50

jb ReadLoop

ShowResults:

mov task1\_count,si

; Вывод ASCII

mov dx, OFFSET task1Msg2

mov ah,09h

int 21h

mov cx,task1\_count

xor si,si

ShowAscii:

mov dl,asciiBuf[si]

mov ah,02h

int 21h

inc si

loop ShowAscii

; Вывод Scan-кодов

mov dx, OFFSET task1Msg3

mov ah,09h

int 21h

mov cx,task1\_count

xor si,si

ShowScan:

mov al,scanBuf[si]

call PrintHex

mov dl,' '

mov ah,02h

int 21h

inc si

loop ShowScan

; Ожидание любой клавиши перед выходом

mov ah,00h

int 16h

; Завершение программы

mov ax,4C00h

int 21h

;----------------------- ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ --------------------

PrintHex PROC

push ax

push cx

mov ah,0

mov cl,16

div cl

call PrintDigit

mov al,ah

call PrintDigit

pop cx

pop ax

ret

PrintHex ENDP

PrintDigit PROC

push ax

cmp al,10

jb Digit

add al,7

Digit: add al,'0'

mov dl,al

mov ah,02h

int 21h

pop ax

ret

PrintDigit ENDP

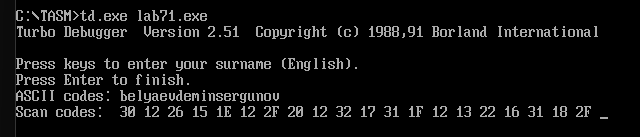
main endp

cseg ends

end main

Блок-схема программы:

Результат работы программы:

****

*Задание 2*

Листинг:

stseg segment para stack

dw 64 dup(?)

stseg ends

dseg segment para

task2Msg db 13,10,'Enter text (echo input, Esc to finish): ','$'

dseg ends

cseg segment para

assume cs:cseg, ds:dseg, ss:stseg

main proc far

push ds

xor ax,ax

push ax

mov ax,dseg

mov ds,ax

;---------------------------- ЗАДАНИЕ 2 ----------------------------

; Вывод приглашения

mov dx, OFFSET task2Msg

mov ah,09h

int 21h

; Цикл эхо-ввода

InputLoop:

mov ah,00h ; Чтение клавиши

int 16h ; AH = scan code, AL = ASCII

cmp al,27 ; Проверка на Esc

je ExitProg

mov ah,0Eh ; Вывод символа

int 10h

jmp InputLoop

;----------------------------- ВЫХОД -------------------------------

ExitProg:

; Ожидание любой клавиши

mov dx, OFFSET task2Msg ; Произвольный адрес для примера

mov ah,08h

int 21h

mov ax,4C00h

int 21h

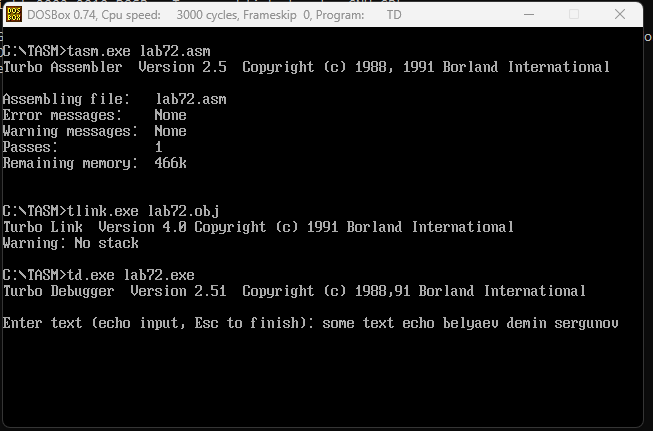
main endp

cseg ends

end main

Блок-схема программы:

Результат работы программы:

****

*Задание 3*

Листинг:

stseg segment para stack

dw 256 dup(?)

stseg ends

dseg segment para

prompt1 db 'IVANOV: $'

prompt2 db 'PETROVA: $'

days db 2 dup(0)

months db 2 dup(0)

years dw 2 dup(0)

date\_buffer db 11, 0, 11 dup(0)

result\_str db 'birthday: $'

birthday\_day db '00.00.0000$'

year\_str db '0000$'

newline db 0Dh, 0Ah, '$'

count dw 0

dseg ends

cseg segment para

assume cs:cseg, ds:dseg, ss:stseg

task proc far

push ds

xor ax, ax

push ax

mov ax, dseg

mov ds, ax

; Ввод первой даты

mov dx, offset prompt1

mov ah, 09h

int 21h

call input\_date

call parse\_date

; Ввод второй даты

mov dx, offset prompt2

mov ah, 09h

int 21h

call input\_date

call parse\_date

; Вычисление результата

call compute\_average

; Вывод результата

mov dx, offset result\_str

mov ah, 09h

int 21h

mov dx, offset birthday\_day

mov ah, 09h

int 21h

; Ожидание нажатия клавиши

mov ah, 08h

int 21h

ret

task endp

input\_date proc near

call clear\_keyboard

mov dx, offset date\_buffer

mov ah, 0Ah

int 21h

mov dx, offset newline

mov ah, 09h

int 21h

ret

input\_date endp

parse\_date proc near

mov di, [count]

mov si, offset date\_buffer + 2

; Парсинг дня (дд)

call atoi\_2

mov days[di], al

add si, 3 ; пропустить точку

; Парсинг месяца (мм)

call atoi\_2

mov months[di], al

add si, 3 ; пропустить точку

; Парсинг года (гггг)

call atoi\_4

shl di, 1

mov years[di], ax

shr di, 1

inc word ptr [count]

ret

parse\_date endp

clear\_keyboard proc near

push ax

clear\_loop:

mov ah, 01h

int 16h

jz buffer\_empty

mov ah, 00h

int 16h

jmp clear\_loop

buffer\_empty:

pop ax

ret

clear\_keyboard endp

atoi\_2 proc near

xor ax, ax

mov al, [si] ; первая цифра

sub al, '0'

mov bx, 10

mul bx ; AX = AL \* 10

mov bl, [si+1] ; вторая цифра

sub bl, '0'

add al, bl

ret

atoi\_2 endp

atoi\_4 proc near

xor ax, ax

mov cx, 4

atoi4\_loop:

mov bx, 10

mul bx ; AX = AX \* 10

mov bl, [si]

sub bl, '0'

xor bh, bh

add ax, bx

inc si

loop atoi4\_loop

ret

atoi\_4 endp

compute\_average proc near

; Вычисление среднего года

mov ax, years[0]

add ax, years[2]

shr ax, 1

mov bx, ax ; сохраняем средний год в BX

; Очистка year\_str перед форматированием

mov di, offset year\_str

mov cx, 4

clear\_year\_str:

mov byte ptr [di], '0'

inc di

loop clear\_year\_str

; Восстанавливаем значение AX из BX перед форматированием

mov ax, bx

call format\_4\_digits

; Установка ES для корректного копирования

mov ax, ds

mov es, ax

; Копирование отформатированного года из year\_str в birthday\_day (начиная с позиции 6)

mov si, offset year\_str

mov di, offset birthday\_day+6

mov cx, 4

rep movsb

; Вычисление среднего дня

xor ax, ax

mov al, days[0]

add al, days[1]

shr ax, 1

call format\_2\_digits

mov [birthday\_day], ah

mov [birthday\_day+1], al

; Вычисление среднего месяца

xor ax, ax

mov al, months[0]

add al, months[1]

shr ax, 1

call format\_2\_digits

mov [birthday\_day+3], ah

mov [birthday\_day+4], al

ret

compute\_average endp

format\_2\_digits proc near

mov bl, 10

div bl ; делим AX на 10, остаток в AH

add ax, '00' ; преобразование в ASCII

xchg al, ah ; меняем порядок цифр

ret

format\_2\_digits endp

format\_4\_digits proc near

mov si, offset year\_str + 3 ; заполняем с конца

mov cx, 4

mov bx, 10

format4\_loop:

xor dx, dx

div bx ; деление: DX:AX / BX

add dl, '0'

mov [si], dl

dec si

loop format4\_loop

ret

format\_4\_digits endp

cseg ends

end task

Блок-схема программы:

Результат работы программы:



**Вывод**

Изучили принцип вывода алфавитно-цифровой информации на видеотерминал.