

## **Прерывания. Функции DOS.**

Если на компьютере произошло событие, которое требует вмешательства процессора, на процессор поступает сигнал, предписывающий ему временно прервать исполнение программы и разобраться с данным событием.

Существует много видов *прерываний*:

- сигналы от аппаратуры,
- необычные ситуации в работе самого процессора,
- запросы программ на ту или иную операцию (прием сигнала от клавиатуры и т.п.),
- и другие.

Каждое прерывание имеет *уникальный номер (от 0 до 255)*, который однозначно связан с конкретным блоком команд (процедурой) в операционной системе, обслуживающим данное прерывание.

Начальные адреса блоков команд обработки прерываний записываются в самые первые ячейки памяти; эти адреса попадают сюда при загрузке ОС в оперативную память.

*Идея прерывания* - получив сигнал прерывания с определенным номером, процессор предпринимает следующие действия:

1. прекращает выполнение текущей программы и запоминает в специальных ячейках (регистрах) все ее параметры, необходимые для возобновления работы этой программы;
2. используя номер прерывания, запускает соответствующий блок команд в операционной системе (процедуру обработки данного прерывания).

После обработки прерывания возобновляется прерванная программа с того места, на котором ее работа была прервана.

## Вызов прерываний

**INT i8**

Эта команда вызывает искусственное, насильственное прерывание с номером i8, а это означает, что начнет работать та процедура ОС, начальный адрес которой записан в элементе вектора прерываний с номером i8. После выполнения этого прерывания будет выполнена следующая команда программы.

В состав ОС входит много процедур обработки прерываний и для них не хватает допустимых номеров прерываний. Поэтому программы обработки прерываний объединяются в группы с тем, чтобы процедуры из одной группы вызывались по прерыванию с одним и тем же номером.

Процедуры одной группы принято называть функциями соответствующего прерывания. Чтобы различать их, перед выполнением команды INT в регистр АН записывают номер нужной функции:

```
mov  ah, <номер функции>
INT  <номер прерывания>
```

## Некоторые функции прерывания 21h

**21h** - диспетчер запросов к функциям DOS

### *Завершение программы*

завершив все свои действия, программа обязана вернуть управление ОС, чтобы пользователь мог продолжить работу на компьютере.

```
mov    al, <код завершения>
mov    ah, 4ch
INT    21h
```

### *Вывод на экран (в текстовом режиме)*

#### 1. Вывод одного символа

```
mov    dl, <код выводимого символа>
mov    ah, 2
INT    21h
```

*Специальные символы:*

7 - звуковой сигнал;

8 - возвращает курсор на одну позицию влево, если только он не был в самой левой позиции строки;

9 - табуляция (перемещает курсор вправо на ближайшую позицию, кратную 8)

10 - перемещает курсор в следующую строку экрана, оставаясь в той же колонке;

13 - устанавливает курсор на начало текущей строки.

Вывод подряд символов с кодами 13 и 10 - перевод курсора на начало следующей строки.

#### 2. Вывод строки

```
ds : dx := начальный адрес строки
mov    ah, 9
INT    21h
```

*В конце строки должен находиться символ '\$', который служит признаком конца строки и сам не выводится.*

**Ввод с клавиатуры**

## 1. очистка буфера клавиатуры

```

mov  al, 0
mov  ah, 0ch
INT  21h

```

## 2. ввод с клавиатуры и редактирование набранного текста

```

ds : dx := адрес буфера для записи введенной строки
mov  ah, 0Ah
INT  21h

```

DOSSEG

.MODEL SMALL

.STACK 100h

.DATA

Message DB 'Ïðèááò!',13,10','\$'

.CODE

```

mov  ax,@Data
mov  ds,ax          ; óñòáííáèòü ðááèñòð DS òáèèì
                    ; íáðàçîì, ÷òíáú íí óéàçúââë
                    ; íà ñáâíáíò ááííüð
mov  ah,9           ; ôóíéöëý DOS áúâíâà ñòðíèè
mov  dx,OFFSET Message ; ññûèèà íà ñííáúáíèâ "Ïðèááò!"
int  21h           ; áúâáñòè "Ïðèááò!" íà ýèðáí

mov  ah,4ch        ; ôóíéöëý DOS çáâáðøáíèý ïðíãðàííú
int  21h          ; çáâáðøèòü ïðíãðàííó

```

END

## **Некоторые функции прерывания 10h**

### **10h - вызов функций драйвера видеотерминала**

#### ***Установить положение курсора***

Вход:

AH = 2

DH = <номер строки>

DL = <номер позиции>

BH = <номер страницы>

#### ***Получить положение и размер курсора***

Вход:

AH = 3

BH = <номер страницы>

Выход:

CH = <начальная строка курсора>

CL = <конечная строка курсора>

DH = <номер строки положения>

DL = <номер позиции положения>

#### ***Установить цвет фона***

Вход:

AH = 10h

AL = 0

BL = 0

BH = <цвет>

#### ***Вывести пиксель***

Вход:

AH = 0Ch

BH = <номер страницы>

DX, CX - координаты (X, Y)

AH = <цвет>

**Программа**

определяет положение курсора, а затем изменяет его

DOSSEG

.MODEL SMALL

.STACK 100h

.DATA

Message DB 'Привет!',13,10,'\$'

.CODE

mov ax,@Data

mov ds,ax

mov ah,9

mov dx,OFFSET Message ; ссылка на сообщение "Привет!"

int 21h ; вывести "Привет!" на экран

mov ah,3 ; определить положение курсора

int 10h

add dl,10 ;

mov ah,2 ; изменить положение курсора

int 10h

mov ah,9

mov dx,OFFSET Message ; ссылка на сообщение "Привет!"

int 21h ; вывести "Привет!" на экран

mov ah,4ch

int 21h

END

**Программа**

рисует прямую горизонтальную линию длиной 200 пиксель

DOSSEG

.MODEL SMALL

.STACK 100h

.DATA

X dw 200; начальные координаты по X

Y dw 100; начальные координаты по Y

Xt dw 0 ; рабочая ячейка

Xsize dw 200; длина линии

Xcd db 14 ; цвет

.CODE

mov ax,@Data

mov ds,ax

mov ah,0 ; очистка экрана

mov al,12h

int 10h

mov ax,X

mov Xt,ax

mov cx,Xsize

@1: push cx

mov ah,0ch ; рисование пикселя

mov al,Xcd

mov bh,0

mov cx,Xt

mov dx,Y

int 10h

inc Xt

pop cx

loop @1

mov ah,0 ; ожидание нажатия клавиши

int 16h

mov ah,0 ; установка текстового видеорежима

mov al,3

int 10h

mov ah,4ch

int 21h

END

**Программа**

очищает часть экрана

```

DOSSEG
.MODEL SMALL
.STACK 100h
.DATA
.CODE
    mov  ax,@Data
    mov  ds,ax

    mov  ah,6           ; очистка части экрана
    mov  al,0           ; ch,cl - dh,dl
    mov  ch,0
    mov  cl,0
    mov  dh,10
    mov  dl,10
    int  10h

    mov  ah,4ch
    int  21h
END

```

**Пример***;Программа считывает символ с клавиатуры и**;в зависимости от значения выдает одно или другое сообщение*

```

DOSSEG
.MODEL SMALL
.STACK 100h
.DATA
TimePrompt          DB 'Это время после полудня? (ДА/НЕТ) - [Y/N]',13,10,'$'
GoodMorningMessage DB 13,10,'Доброе утро!',13,10,'$'
GoodAfternoonMessage DB 13,10,'Здравствуйте!',13,10,'$'
.CODE
    mov  ax,@Data
    mov  ds,ax

    mov  al,0
    mov  ah,0Ch
    int  21h
    mov  dx,OFFSET TimePrompt ; ссылка на сообщение-запрос

    mov  ah,9           ; функция DOS вывода строки
    int  21h           ; получить ответ из одного символа

    mov  ah,0
    int  16h

```



```
    cmp    al,'Y'      ; указано время после полудня (прописная буква Y)
    jz     IsAfternoon ; да, время указано после полудня

    cmp    al,'y'      ; указано время после полудня (строчная буква y)
    jnz    IsMorning  ; нет, время указано до полудня
IsAfternoon:
    mov    dx,OFFSET GoodAfternoonMessage ; указывает на
                                           ; приветствие "Здравствуйте"
    jmp    DisplayGreeting
IsMorning:
    mov    dx,OFFSET GoodMorningMessage ; указывает на
                                           ; приветствие "Доброе утро"
DisplayGreeting:
    mov    ah,9        ; функция DOS вывода сообщения
    int    21h        ; вывести соответствующее сообщение

    mov    ah,4ch      ; функция DOS завершения программы
    int    21h        ; завершить программу
END
```

**Пример**

*; Выделение из числа X последней цифры и вывод ее на экран*

DOSSEG

.MODEL SMALL

.STACK 100h

.DATA

X DB 21

.CODE

mov ax,@Data

mov ds,ax

mov dl,10 ; запись в dl = 10

mov al,X ; запись в ax = X

mov ah,0

div dl ; выделение последней цифры - ah

add ah,48 ; получение кода цифры в ah

mov dl,ah ; вывод цифры на экран

mov ah,2

int 21h

mov ah,4ch ; функция DOS завершения программы

int 21h

END