

# Лекция 16. Системный сервис для ввода с клавиатуры. Преобразование символьных кодов в числовые

- Контроллер клавиатуры создает однобайтный «скан-код» клавиши.

Скан-код содержит № строки и № столбца клавиатуры, где находится клавиша

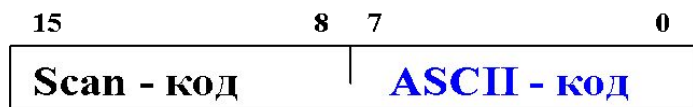


- По прерыванию от контроллера начинает исполняться программа-драйвер клавиатуры. Считывает скан-код из контроллера, определяет ASCII-код символа и формирует двухбайтный код нажатия
- Код нажатия клавиши доступен программно через функции системного сервиса прерываний INT 21h и INT 16h.

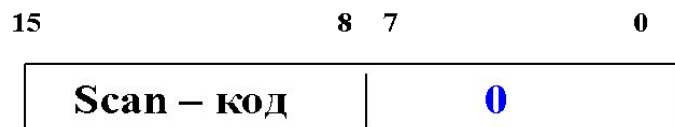
# Формат кода нажатия клавиши

---

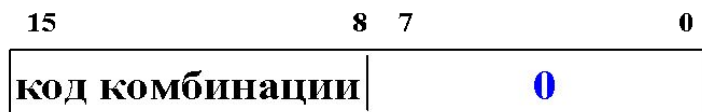
а) для алфавитно-цифровых клавиш



б) для управляющих и функциональных клавиш



в) для «зарезервированных комбинаций» клавиш



**Код комбинации – никогда не пересекается с множеством скан - кодов.**

---

# Чтение нажатия с клавиатуры (функции 1 и 8 прерывания 21h)

---

Функции 1 и 8 ожидают нажатия клавиши.

Если нажата символьная клавиша, функция 1 отображает символ на экране, функция 8 – не отображает.

Выход функций: AL – младший байт кода нажатия.

Для символьной клавиши – это ASCII-код.

Пример: Чтение одного символа с клавиатуры с отображением

```
mov ah, 1
```

```
int 21h
```

**Замечание:** Однократный вызов функции дает младший байт кода нажатия. Для чтения его старшего байта надо повторно вызвать функцию.

Пример: Прочитать нажатие клавиши. Если это символьная клавиша, записать в память ее Ascii-код, если управляющая/ функциональная – Scan –код

---

```
Ascii    db  ?  
Scan     db  ?
```

```
. . . . .  
    mov ah, 8  
    int 21h      ; AL ← младший байт кода нажатия  
    cmp al, 0  
    jz  short m2  ; на m2, если была функ/упр.клавиша  
    mov ds: Ascii, al  
    jmp short m3
```

*; повторный вызов функции для получения Scan-кода*

```
m2:  mov ah, 8  
     int 21h      ; AL ← старший байт кода нажатия  
     mov ds: Scan, al  
m3:  . . . . .
```

## Чтение с клавиатуры строки символов (функция 0Ah прерывания 21h)

---

Функция ожидает ввод строки с клавиатуры. Пока не нажата «Enter», строка ввода может редактироваться

Вход: DS:DX – адрес области памяти для размещения строки

**!!** Требования к области ввода и ее размеру:

- 1-й байт области должен содержать максимальное количество планируемых нажатий, включая «Enter» («план нажатий»)
- 2-й байт резервировать под количество фактически введенных символов («факт нажатий»), его запишет сама функция
- остальные байты - под символы, согласно «плану нажатий»

Выход :

- во 2-й байт области ввода будет записано число фактически введенных символов (без Enter)
  - с 3-го байта области будут размещены ASCII коды введенных символов и код 0Dh (возврат каретки)
-

Пример: Организовать ввод с клавиатуры строки символов длиной не более 5 символов

---

```
Buffer    db  6, 7 dup(0) ; область для ввода. План нажатий - 6 и еще 7
           ;байт для размещения кодов
```

```
      . . .
mov    ah, 0Ah
lea    dx, Buffer
int    21h
```

Отладочный пример:

Если были нажаты символы: ' 2 3 4 ' и затем **Enter**, область ввода после вызова сервиса должна будет содержать следующие коды:

```
ds:Buffer  06 03 32 33 34 0D 00 00
```

Пример: Ввести с клавиатуры строку символов (не более 80 символов) и определить в ней количество символов '0'

Размещение данных в памяти и регистрах:

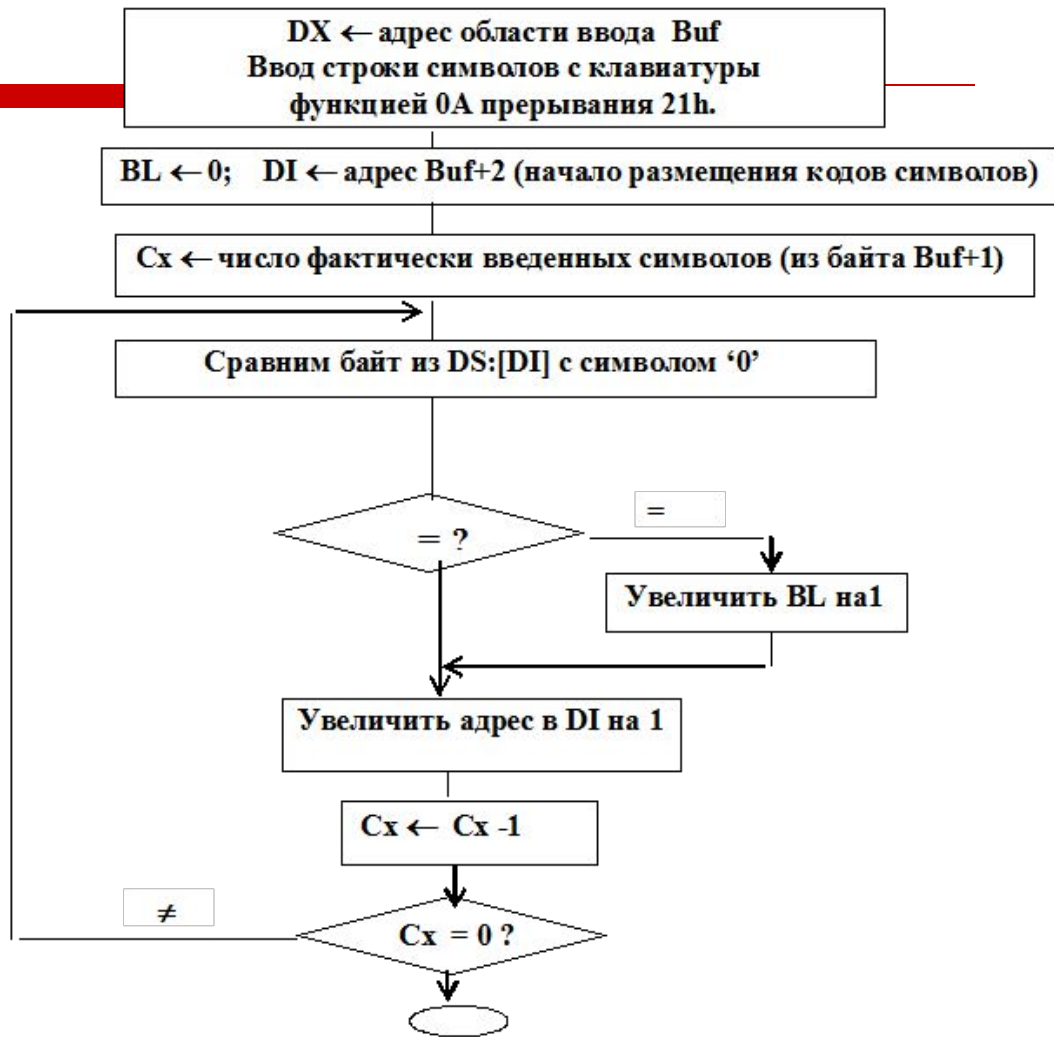
**Ds:** Buf – адрес области ввода ( 83 байта);

**DI** –косвенный внутрисегм. адрес символа

**CX** – счетчик циклов

**BL** - количество символов '0'

Выделение памяти под область ввода в исх.тексте:  
Buf db 81, 82 dup (?)



# Преобразование символьной строки после ввода с клавиатуры в числовые коды

---

## 1. Строка символьного 2-го представления кода

Пример:

Массив символов после ввода (hex): 30 30 31 31 31 30 31 30h ( “00111010” )

Желаемый результат: получить числовой байт **3A** в регистре AL  
(00111010)

Алгоритм:

$AL \leftarrow 0$

Цикл для каждого байта массива символов:

- сравниваем байт с кодом 31h (“1”)
- если равен,  $AL \leftarrow AL \vee 00000001b$
- сдвиг AL влево на 1 бит



---

## 2. Строка символьного 16-го представления кода

### Пример:

Символьные коды после ввода с клавиатуры: 33 41h (“3A”)

Желаемый результат: числовой байт 3A

### Алгоритм (на примере):

- Из кодов символов сделать коды цифр: 03 0A
- В старшем байте поменять местами тетрады: 30 0A
- Логическое ИЛИ между старшим и младшим байтом:

$$\begin{array}{r} 30 \\ \vee 0A \\ \hline 3A \end{array}$$

---

### 3. Строка символьного 10-го представления кода

#### Пример:

Символьные коды после ввода: **2D 32 35 35h** ( “-255” )

Желаемый результат: числовой 2-байтный код **FF01**

#### Алгоритм (на примере):

- Преобразуем байты символов в цифры: **02 05 05h**
- Вычислим двухбайтный числовой код, исходя из веса каждой цифры:  **$02 * 100 + 05 * 10 + 05$**  . Получится **00 FFh**
- Проверим символ знака. Поскольку был “-” , получим числовой код с обратным знаком:

$$00\ 00 - 00FF = \mathbf{FF\ 01}$$