

РАБОТА СО СТАНДАРТНЫМИ СИМВОЛЬНЫМИ УСТРОЙСТВАМИ

Устройства можно разделить на два основных вида: символьные и блочные. Это разделение отражает способ обмена информации между устройствами. Блочными устройствами являются, например, диски и ленты, а символьными - принтеры, клавиатура, последовательный интерфейс. Символьные устройства обмениваются информацией посимвольно (по байтам), а блочные устройства обмениваются информацией большими блоками, например размером в один сектор.

КЛАВИАТУРА

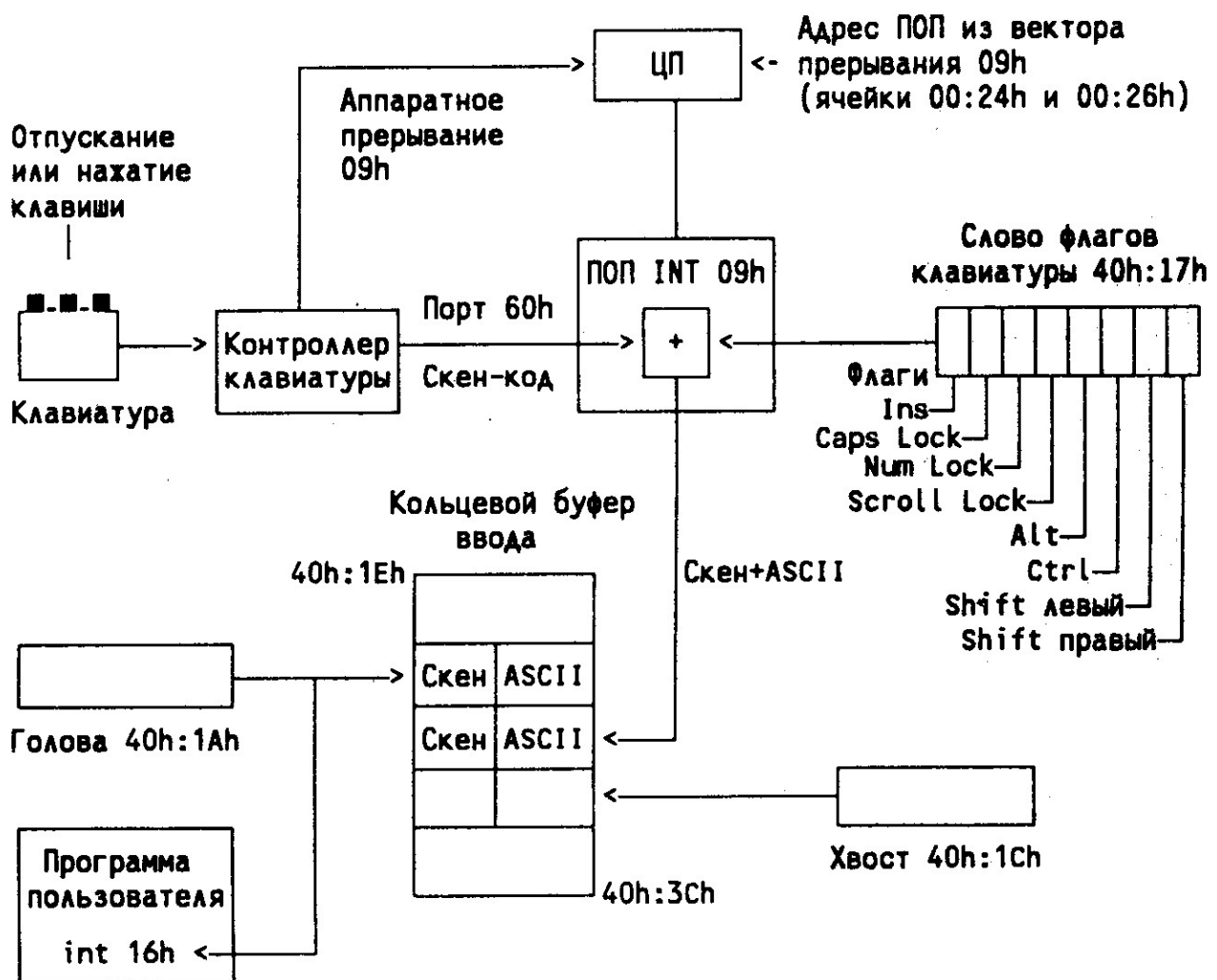
Клавиатура представляет собой отдельное устройство с микропроцессорным управлением. При включении питания клавиатура устанавливается в исходное состояние. Затем выполняется автотест, который проверяет схемы и память клавиатуры. После прохождения теста клавиатура начинает работу.

Работой клавиатуры управляет специальная электронная схема – контроллер клавиатуры. В его функции входит распознавание нажатой клавиши и помещение закрепленного за ней кода в свой выходной регистр (порт), обычно с номером 60h. В ходе работы клавиатура непрерывно проверяет, есть ли изменения в состоянии клавиш. При каждом нажатии клавиши по интерфейсу посылается однобайтовый код, который равен позиционному номеру клавиши - этот код называется позиционным или скэн-кодом. При освобождении клавиши к ее номеру добавляется 80h и полученный позиционный код тоже посылается по интерфейсу. Таким образом, каждой клавише соответствуют два позиционных кода - код нажатия и код освобождения. Если клавиша остается нажатой дольше некоторого времени, ее позиционный код начинает посылаться по интерфейсу через 0,1сек до освобождения клавиши.

Нажатие, а также отпускание любой клавиши вызывает сигнал аппаратного (внешнего) прерывания, заставляющий процессор прервать выполняемую программу и перейти на программу обработки прерывания от клавиатуры.

Процессор вместе с сигналом прерывания получает еще и номер вектора прерывания – 09h.

Программа INT 09h, получив управление в результате прерывания от клавиатуры, считывает из порта 60h скен-код и анализирует его значение. Если скен-код принадлежит одной из управляющих клавиш, и, к тому же, представляет собой код нажатия, в байте статуса клавиатуры устанавливается бит (флаг), соответствующий нажатой клавише. При нажатии любой клавиши программа INT 09h считывает из порта 60h ее скэн-код нажатия и по таблице трансляции скен-кодов в коды ASCII формирует двухбайтовый код, старший байт которого



содержит скен-код, а младший -код ASCII. При этом если скен-код характеризует клавишу, то код ASCII определяет закрепленный за ней символ. Поскольку за каждой клавишей закреплено, как правило, не менее двух символов ("a" и "A", "1" и "!", "-" и "_" и т.д.), то каждому скен-коду соответствуют, как минимум, два кода ASCII. В процессе трансляции программа INT 09h анализирует состояние

байтов статуса, так что если нажата, например, клавиша Q (скен-код 10h, код ASCII буквы Q - 51h, а буквы q - 71h), то формируется двухбайтовый код 1071h, но если клавиша Q нажата при нажатой клавише <Shift> (смена регистра), то результат трансляции составит 1051h. Тот же код 1051h получится, если при нажатии клавиши Q был включен режим <Caps Lock> (заглавные буквы), однако при включенном режиме <CapsLock> и нажатой клавише <Shift> образуется код 1071h, поскольку в такой ситуации клавиша <Shift> на время нажатия переводит клавиатуру в режим нижнего регистра (строчные буквы).

После передачи одного позиционного кода, клавиатура ожидает от компьютера подтверждения его готовности принять следующий код. Если до получения сигнала подтверждения нажаты другие клавиши, клавиатура записывает их позиционные коды в буфер.

Буфер устроен как циклическая очередь, работающая по принципу первый вошел - первый ушел. Он занимает непрерывную область адресов памяти. Два указателя хранят позиции головы и хвоста строки символов, находящихся в буфере в текущий момент. Новые нажатия клавиш сохраняются в позициях, следующих за хвостом, т.е. в более старших адресах памяти и соответственно обновляется указатель хвоста буфера.

40:1A	1C	1E	20	22	24	26	28	2A	2C	2E	30	32	34	36	38	3A	3C
28	34						b	u	f	f	e	r					

40:1A	1C	1E	20	22	24	26	28	2A	2C	2E	30	32	34	36	38	3A	3C
38	24	f	e	r											b	u	f

В то время как указатель на голову установлен на первый введенный символ, указатель на хвост установлен на позицию за последним введенным символом. Когда оба указателя равны, буфер пуст. Чтобы разрешить ввод 15 символов, требуется 16-я пустая позиция, два байта которой всегда содержат код возврата каретки (Dh) и скэн-код клавиши Enter.

Эта пустая позиция непосредственно предшествует голове строки символов. 32 байта буфера начинаются с адреса 0040:001E. Указатели на голову и хвост расположены по адресам 0040:001A и 0040:001C соответственно. Хотя под указатели отведено два байта, используется только младший байт. Значения

указателей меняются от 30 (1E) до 60 (3C), что соответствует позициям в области данных BIOS.

Имеются два типа кодов символов: коды ASCII и расширенные коды.

Коды ASCII - это байтовые числа, соответствующие расширенному набору кодов ASCII для IBM. Этот набор включает в себя обычные символы пишущей машинки, ряд специальных букв и символов псевдографики и 32 управляющих кода, которые обычно используются для передачи команд периферийным устройствам.

Расширенные коды присвоены клавишам или комбинациям клавиш, которые не имеют представляющего их символа ASCII, таким, как функциональные клавиши или комбинации с клавишей Alt. Расширенные коды имеют длину два байта, причем первый байт всегда 0, позволяющий программе определить, принадлежит ли данный код набору ASCII или расширенному набору. Второй байт - номер расширенного кода.

Программы, работающие с расширенными кодами ASCII, должны, считав из кольцевого буфера ввода младший байт и убедившись, что он равен нулю, считать далее и старший байт (скен-код).

Широкое использование в компьютерах интерактивных средств потребовало расширения возможностей ввода с клавиатуры управляющей информации, которую программа должна легко отличать от вводимого текста. С этой целью расширенные коды ASCII генерируются не только с функциональными клавишами и клавишами управления курсором, но и всеми алфавитно-цифровыми клавишами, если они нажимаются вместе с клавишей <Alt>. Расширенные коды ASCII генерируются также при нажатии клавиш <Alt>, <Ctrl>, <Shift> одновременно с функциональными клавишами <F1>...<F10>. В этом случае в старший байт расширенного кода ASCII помещается уже не скен-код клавиши, а некоторый код, специально назначенный этой комбинации клавиш.

Два байта, расположенные в ячейках памяти 0040:0017 и 0040:0018, содержат биты, отражающие статус клавиш-переключателей следующим образом:

Байт 0040:0017h								байт 0040:0018h									
БИТЫ				КЛАВИША				БИТЫ				КЛАВИША					
7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0	
X	INS (1=активна)	X	INS нажата
.	X	Caps Lock (1=активна)	.	X	Caps Lock нажата
.	.	X	Nun Lock (1=активна)	.	.	X	Nun Lock нажата
.	.	.	X	Scroll Lock (1=активна)	.	.	.	X	Scroll Lock нажата
.	.	.	.	X	.	.	.	Alt 1=нажата)	X	.	.	.	Ctrl-NumLock (режим паузы включен)
.	X	.	.	Ctrl(1=нажата)	X	.	.	
.	X	.	Левая Shift (1=нажата)	X	.	
.	X	ПраваяShift (1=нажата)	X	

Прерывание клавиатуры немедленно обновляет эти биты статуса как только будет нажата одна из клавиш переключателей, даже если не было считано ни одного символа из буфера клавиатуры. Это верно и для клавиши Ins, которая единственная из этих 8 клавиш помещает код в буфер.

Прерывание клавиатуры проверяет состояние статусных битов каждый раз перед тем как интерпретировать нажатые клавиши.

В прерывании клавиатуры можно выделить три основных шага:

1. Прочитать скэн-код и послать клавиатуре подтверждающий сигнал.
2. Преобразовать скэн-код в номер кода или в установку регистра статуса клавиш-переключателей.
3. Поместить код клавиши в буфер клавиатуры.

В момент вызова прерывания скэн-код будет находиться в порте А (адрес 60H). Поэтому сначала надо этот код прочитать и сохранить в стеке. Затем используется порт В (адрес 61H), чтобы быстро послать сигнал подтверждения микропроцессору клавиатуры. Надо просто установить бит 7 в единицу, а затем сразу вернуть его в ноль. Бит 6 порта В управляет сигналом часов клавиатуры. Он всегда должен быть установлен в 1, иначе клавиатура будет выключена.

Сначала скэн-код анализируется, была ли клавиша нажата или отпущена. Все коды освобождения отбрасываются, за исключением клавиш переключателей, для которых делаются соответствующие изменения в байтах их статуса. С другой стороны, все коды нажатия отбрасываются. При этом снова могут изменяться

байты статуса клавиш-переключателей. В случае символьных кодов нужно проверять байты статуса, чтобы определить, например, что скэн-код соответствует нижнему или верхнему регистру.

После того, как введенный символ идентифицирован, процедура ввода с клавиатуры должна найти соответствующий код ASCII или расширенный код. В общем случае скэн-коды сопоставляются с элементами таблицы данных, которая анализируется командой XLAT.

После этого коды символов должны быть помещены в буфер клавиатуры. Процедура должна сначала проверить, имеется ли в буфере место для следующего символа. Новые символы вставляются в ячейки буфера с более старшими адресами, а когда достигнута верхняя граница, следующий символ переносится в нижний конец буфера. Когда буфер полон, указатель на его хвост на 2 меньше указателя на голову, за исключением ситуации, когда указатель на голову равен 30(1E). В этом случае буфер полон, когда указатель на его хвост равен 60(3C).

Для вставки символа в буфер надо поместить его в позицию, на которую указывает хвост буфера и затем увеличить указатель хвоста на 2; если указатель хвоста был равен (3C), то его надо изменить на значение (1E).

Существует несколько способов проверки, был ли ввод с клавиатуры.

1. Когда значения указателей на голову и на хвост буфера равны, буфер пуст. Надо просто сравнить содержимое ячеек памяти 0040:001A и 0040:001C.

```
mov AX,0           ; устанавливаем добавочный сегмент
mov ES,AX          ; на начало памяти
mov AL,ES:[41Ah]   ; берем указатель на голову
mov AH,ES:[41Ch]   ; указатель на хвост
cmp AH,AL          ; сравниваем их
jne vvod           ; если не равны - есть символ, уходим на ввод символа
```

2. Функция 0Bh прерывания 21h возвращает значение 0FFh в регистре AL, когда буфер клавиатуры содержит хотя бы один символ.

```
mov AH,0Bh        ; номер функции
int 21h           ; вызываем прерывание
cmp AL,0FFh       ; сравниваем с 0FFh
je vvod           ; переход, если буфер не пуст
```

3. Функция 1 прерывания BIOS 16h предоставляет ту же возможность, но кроме того, она показывает, какой символ в буфере. Флаг нуля ZF устанавливается, если буфер пуст, и сбрасывается, если в буфере есть символ. Копия символа из

буфера помещается в регистр AX, но символ из буфера не удаляется. В AL возвращается ASCII код символа. Если он равен нулю, то это расширенный код и в AH пересылается номер кода.

```

mov  AH,1          ; номер функции
int  16h           ; проверка наличия символа
jz   no_char        ; переход, если буфер пуст (ZF=1)
                        ; проверяем, какой символ
cmp  AL,0           ; это расширенный код?
je   ext_code       ; если да, переход

```

Выполняемая программа, желая получить код нажатой клавиши, должна вызвать прерывание INT 16h, которое активизирует драйвер клавиатуры BIOS. Драйвер считывает из кольцевого буфера содержимое ячейки, адрес которой находится в головном указателе, и увеличивает этот адрес на 2. Таким образом, программный запрос на ввод с клавиатуры фактически выполняет прием кода не с клавиатуры, а из кольцевого буфера.

Если компьютер находится в пассивном состоянии ожидания команд DOS с клавиатуры, то за состоянием кольцевого буфера ввода следит командный процессор COMMAND.COM. Как только в буфере появляется код символа, командный процессор с помощью соответствующих системных программ переносит его в свой внутренний буфер командной строки, очищая при этом кольцевой буфер ввода, а также выводит символ на экран, организуя режим эхо-контроля. При получении кода клавиши <Enter> (0Dh) командный процессор предполагает, что ввод команды закончен, анализирует содержимое своего буфера и приступает к выполнению введенной команды. При этом командный процессор работает практически лишь с младшими половинами двухбайтовых кодов символов, т.е. с кодами ASCII.

Если компьютер выполняет какую-либо программу, ведущую диалог с оператором, то, ввод данных с клавиатуры (а точнее - из кольцевого буфера ввода) и вывод их на экран с целью эхо-контроля организует эта программа, обращаясь непосредственно к драйверу BIOS (INT 16h) или к соответствующей функции DOS (INT 21h). Может случиться, однако, что выполняемой программе не требуется ввод с клавиатуры, а оператор нажал какие-то клавиши. В этом случае вводимые символы накапливаются (с помощью программы INT 09h) в

кольцевом буфере ввода и, естественно, не отображаются на экране. Так можно ввести до 15 символов. Когда программа завершится, управление будет передано COMMAND.COM, который сразу же обнаружит наличие символов в кольцевом буфере, извлечет их оттуда и отобразит на экране. Такой ввод с клавиатуры называют вводом с упреждением.

ПРОЦЕДУРА ОБРАБОТКИ "Ctrl-Break"

Когда вводится комбинация "Ctrl-Break", прерывание клавиатуры устанавливает флаг, указывающий, что должна быть выполнена процедура обработки "Ctrl-Break". Управление передается процедуре только в тот момент, когда программа обращается к функции DOS, способной распознать этот флаг. Обычно только стандартные функции ввода-вывода могут распознать этот флаг. Но поместив строку Break=on в файл CONFIG.SYS возникает ситуация, когда обращение к любой функции DOS приведет к вызову процедуры обработки Ctrl-Break. При этом выполнение программы будет немного замедлено.

Процедура обработки Ctrl-Break дает возможность завершить программу в любой момент времени. Когда функция DOS распознает статус Ctrl-Break, управление передается процедуре, на которую указывает вектор прерывания 23H. DOS применяет эту процедуру для завершения работающей программы. Эта процедура должна быть программируемой, чтобы перед завершением программы могли быть выполнены все критические операции.

Может потребоваться выравнивание стека, для того, чтобы регистр SP указывал на второе слово от вершины перед выполнением завершающей инструкции RET. векторы прерываний, измененные программой должны быть восстановлены, а все открытые устройства ввода-вывода - закрыты. Если были запрещены прерывания, то надо разрешить их. Все это должно обеспечить машине возможность нормально работать со следующей программой после завершения программы по Ctrl-Break. Если сделать процедуру обработки Ctrl-Break состоящей из одной инструкции IRET, это запретит завершение программы таким способом.