

1 Program shift_shift;

2 {
 Задача. Обработка кода сканирования.
 Узнать, нажаты ли Shift-Shift.

 Решение. Сергей Митрофанов for SGP,
 Центр НИТ "Северная Звезда",
 08.05.03, 09:30
}

9 {
 При нажатии какой-либо клавиши микропроцессор клавиатуры
 вырабатывает код, который преобразуется ВР в код сканирования.
 Каждая клавиша имеет свой уникальный код сканирования, он не
 связан с изображенным на клавише символом. Код сканирования
 является основой алгоритмов управления клавиатурой.

 Символьные клавиши (буквы, цифры) возвращают при нажатии
 значение, которое принято называть просто кодом.

 Управляющие клавиши (функциональные, перемещения курсора, Insert,
 Delete и др.) при нажатии возвращают два значения, причем
 первое из них равно 0 (это и есть признак управляющей клавиши),
 а второе - фиксированная величина.

Клавиши переключатели (Ctrl, Alt, Shift, NumLock, CapsLock)
в строго определенной памяти BIOS меняют биты байтов, в
зависимости от того, какая клавиша-переключатель нажата.
См.: Румянцев Д., Монастырский Л. Путь программиста, М.:
"Издательский дом ИНФРА-М", 2000, с. 384-385.

В модуле Crt управлением клавиатурой занимаются функции:
 ReadKey,
 KeyPressed
и процедуры
 Read,
 ReadLn.

ReadKey считывает символ с клавиатуры и возвращает char.
Чтение не сопровождается отображением символа на экране.

KeyPressed возвращает булевское значение.
True - если была нажата какая-либо клавиша,
False - если нажатия на клавишу не было.

Связь между ReadKey и KeyPressed.

 Если нажатие клавиши произошло и KeyPressed возвращает
True, то считывание по ReadKey происходит немедленно и не
приводит к ожиданию нажатия.

Если KeyPressed возвращает False, то действие ReadKey сводится к ожиданию нажатия.

Структура возвращаемых клавиатурой значений

Функция	Клавиша символьная	Клавиша управляющая	Клавиша сдвига
KeyPressed	True	True	False
ReadKey	код	#0 + код	-

см. Попов В.Б. Turbo Pascal для школьников, М.: Финансы и статистика, 2002, с. 484-485

Обработка нажатия клавиш сдвига

Клавиши сдвига возвращают коды, которые не распознаются средствами ВР, и поэтому их применение представляет определенную сложность.

Для того, чтобы обнаружить нажатие клавиш сдвига, нужно просмотреть содержимое двух последовательных байтов памяти, которые расположены в области переменных BIOS. Каждый бит из этих двух байтов отвечает на нажатие какой-либо одной ОДНОЙ клавиши сдвига. В связи с особым статусом прерываний, обрабатываемых через BIOS, в области памяти, начиная с адреса \$0000:\$0400, располагаются переменные BIOS, которые дублируют все наиболее важные характеристики, получаемые с помощью этих прерываний. Работа всех прерываний BIOS невозможна без этой области памяти, поэтому какая-либо некорректная запись в эту область может нарушить работоспособность машины!

Если, например, нужно обработать нажатие клавиши NumLock, то нужно будет просмотреть 3 bit байта памяти с абсолютным адресом \$0000:\$0417, который соответствует этой клавише.

Более точно: информация о состоянии клавиш-переключателей хранится в двух ячейках по адресам 0040:0017 и 0040:0018. В нижележащей таблице описано, каким образом состояние битов в этих ячейках отражает статус клавиш сдвига и других клавиш-переключателей.

Статус клавиш-переключателей

Адрес	Бит	Клавиша	Бит = 1
-------	-----	---------	---------

0040:0017	7	Insert	Режим вставки включен
	6	Caps Lock	Режим Caps Lock включен
	5	Num Lock	Режим Num Lock включен
	4	Scroll Lock	Режим Scroll Lock включен
	3	Alt	Клавиша Alt нажата
	2	Ctrl	Клавиша Ctrl нажата
	1	Left Shift	Клавиша левый Shift нажата
	0	Right Shift	Клавиша правый Shift нажата
0040:0018	7	Insert	Клавиша Insert нажата
	6	Caps Lock	Клавиша Caps Lock нажата
	5	Num Lock	Клавиша Num Lock нажата
	4	Scroll Lock	Клавиша Scroll Lock нажата
	3	Ctrl-NumLock	Режим Ctrl-NumLock включен

}

```

96  Uses Crt;
97  Const Esc = #27;
98  Var byte_p: byte; { содержимое байта клавиши-переключателя }
99  Begin
100     ClrScr;
101     repeat
102     {
        опросим ячейку памяти, в которой хранится байт, содержащий
        информацию о состоянии клавиш-переключателей:
        Insert, CapsLock, NumLock, Scroll Lock, Alt, Ctrl,
        Left Shift, Right Shift.
    }
108     byte_p := MEM [$0040:$0017];
109     if byte_p and 1 = 1
110     then
111         WriteLn ('Нажат правый Shift');
112     if byte_p and 2 = 2
113     then
114         WriteLn ('Нажат левый Shift');
115     if byte_p and 4 = 4
116     then
117         WriteLn ('Нажат Ctrl');
118     if byte_p and 8 = 8
119     then
120         WriteLn ('Нажта Alt');
121     if byte_p and 16 = 16

```

```

122         then
123             WriteLn ('Режим Scroll Lock включен');

124     if byte_p and 32 = 32
125         then
126             WriteLn ('Режим Num Lock включен');

127     if byte_p and 64 = 64
128         then
129             WriteLn ('Режим Caps Lock включен');

130     if byte_p and 128 = 128
131         then
132             WriteLn ('Режим вставки включен');

133     Delay (10000);

134     {
        Цикл будет работать до тех пор, пока не произойдет
        одновременное
        нажатие на клавиши Left Shift и Right Shift.
        Когда же такое событие произойдет, и оба Shift будут нажаты
        одновременно, то 0-й и 1-й биты байта byte_p будут установлены
        в единицу. Если все остальные клавиши будут отпущены, то
        переменная byte_p будет равна двоичному числу
            0000 0011,
        то есть десятичному 3.

        Примечание.
        Чтобы закончить этот цикл при одновременном нажатии на левый и
        правый Shift при любых комбинациях других нажатых клавиш, надо
        будет в условие окончания цикла поставить следующее выражение:
            byte_p and 3 = 3.
    }
150     until byte_p = 3;

151     WriteLn ('Вы нажали Shift-Shift');

152     ReadLn;
153     End.

```

Listing данной задачи опубликован в сети Internet по адресу
<http://www.Best-Listing.ru/color-1-task-595.html>

Sergey Mitrofanov, 28.08.14, 20:52

E-mail: infostar@mail.ru

© <http://www.Best-Listing.ru/>, 2006–2014