

```

1 Program Tastint2;
2 (*****
3 (**          Tastint2 8.6.1998          **)
4 (**          Abfangen des Hardware-Interrupts 9          **)
5 (** - Demonstration der Folgen einer Nichtfreigabe des Interrupt-          **)
6 (**   Controllers          **)
7 (**          **)
8 (** - Ausgabe des Inhalts des Tastaturpuffers bzw. der niederen 7 Bits **)
9 (**          **)
10 (** - Aufruf der alten Tastaturroutine nur dann, wenn die Taste Nr. 25 **)
11 (**   gedrückt wurde (Zeichen "p")          **)
12 (**          **)
13 (*****
14 {$M 1024,0,256} { Begrenzung des Speicherplatzbedarfs für das residente
15                  Programm }
16
17 Uses DOS, CRT;
18
19 Type T_Prozedur = Procedure; { Prozedurtyp, ist im Grunde genommen ein Zeiger, der
20                               auf den ersten Speicherplatz einer Prozedur im
21                               Codesegment zeigt }
22
23 VAR  pAlter_KBD_Intr      : Pointer;
24      Alter_KBD_Intr      : T_Prozedur ABSOLUTE pAlter_KBD_Intr;
25                               { Alte Keyboard-Interrupt-Prozedur im ROM-BIOS
26                               Diese Prozedur kann zunächst nicht über einen Namen aufgerufen werden,
27                               da sie keine Pascalprozedur ist. Sie beginnt am absoluten Spei-
28                               cherplatz pAlter_KBD_Intr. Dieser Speicherplatz muß bei der
29                               Programminitialisierung erst ermittelt werden. Damit läßt sich nun
30                               die ROM-Prozedur mit dem Namen Alter_KBD_Intr wie eine normale
31                               Pascal-Prozedur aufrufen }
32
33      pAlte_Exit_Prozedur : Pointer; { Beim Verlassen des Pascalprogramms, ob regulär oder infolge
34                                       eines Abbruchs durch einen Laufzeitfehler wird standard-
35                                       mäßig eine Prozedur aufgerufen, die am Speicherplatz ExitProc
36                                       beginnt. Der Zeiger ExitProc ist in Pascal definiert.
37                                       Sollen über den Standardausstieg hinaus weitere Aktionen
38                                       durchgeführt werden, wie zum Beispiel das Restaurieren
39                                       "verbogener" Interruptvektoren, so kann der Zeiger ExitProc
40                                       auf die eigene Prozedur gerichtet werden.}
41
42 Var Zei      : Char;
43
44 Procedure KBDNeu; Interrupt;
45 Var Taste    : Byte;
46 Begin
47     Taste := Port[$60]; { Auslesen des Tastaturpuffers }
48     Begin
49         Gotoxy(10,15); Clreol; { Zeile 15 ab Spalte 15 löschen }
50         Write(' Taste Nr.',Taste AND $7F); { Höchstwertiges Bit ausblenden }
51         if Taste < 128 then { Oberstes Bit ist Flag, ob Taste }
52             Write(' gedrückt') { drückt (Flag = 0) oder losgelassen }
53         else { ist (Flag = 1) }
54             Write(' losgelassen');
55
56         Gotoxy(10,16); Clreol;
57         Write(' Tastaturpufferinhalt: ',Taste);
58
59         Gotoxy(10,18);
60         Write(' Falls die Zeile "port[$20] := $20] auskommentiert ist,');
61         Gotoxy(10,19);
62         Write(' dann war das war die letzte Aktion in diesem Programm! ');
63         Gotoxy(10,20);
64         Write(' Der Interrupt-Controller wird nicht mehr freigegeben!');

```

```

64         Gotoxy(10,22);
65         Write(' Bei Druck auf die Taste "p" wird die alte BIOS-Routine aufgerufen,');
66         Gotoxy(10,23);
67         Write(' die das Zeichen dem Pascalprogramm übergibt. ')
68     End;
69     { Hier wird nun nicht mehr die alte Tastaturroutine aufgerufen, die von
70       sich aus den Interruptcontroller wieder freigibt. Der Rechner reagiert daher nicht mehr und
71       muß durch Hardware Reset wieder gestartet werden }
72     if taste = 25 then { Taste mit Buchstaben p gedrückt!}
73     Begin
74         ASM
75         pushf
76         END;
77         alter_KBD_intr;
78     End
79     Else
80         Port[$20] := $20; { Interrupt Controller freigeben! Zum Test bitte
81                           durch Kommentarklammern unwirksam machen}
82
83 end;
84
85
86
87 Procedure Programm_Exit; far;
88 {+-----+
89 | Diese Prozedur wird beim Verlassen des Programms aufgerufen, also auch
90 | beim Verlassen durch einen Laufzeitfehler oder durch <Ctrl><C>.
91 | Dabei wird die alte Interrupt-Prozedur wiederhergestellt und der Inter-
92 | rupt-Controller wieder freigegeben, falls der Abbruch durch <Ctrl><C>
93 | erfolgte. In diesem Fall wird nämlich die alte Interrupt-Prozedur, die
94 | den Interrupt-Controller freigibt nicht mehr aufgerufen.
95 +-----+}
96 Begin
97     ExitProc := pAlte_Exit_Prozedur; { Vektor auf alte Standard-Exit-Prozedur
98                                     belegen }
99     SetIntVec($9,pAlter_KBD_Intr); { Interruptvektor Nr. 9 = Zeiger auf alte
100                                   Serviceroutine des Tastaturinterrupts wieder
101                                   herstellen }
102     Port[$20] := $20; { Interrupt Controller sicherheitshalber
103                       freigeben }
104 End;
105
106 (*-----*)
107
108 Begin (*Hauptprogramm*)
109     GetIntVec($9,pAlter_KBD_Intr); { Vektor auf die alte Tastaturinterrupt-Serviceroutine in der
110                                     absoluten Prozedurvariablen Alter_KBD_intr retten.
111                                     Damit kann die alte Prozedur mit diesem Bezeichner
112                                     aufgerufen werden }
113     Setintvec($9, @KBDNeu); { Vektor Nr. 9 auf eigene Routine richten }
114
115     pAlte_Exit_Prozedur := ExitProc; { Exit-Handler installieren }
116     ExitProc := @Programm_Exit;
117
118
119     ClrScr;
120     Textcolor(green);
121     Highvideo;
122     Writeln('***** Test der neuen Tasturoutine *****');
123     Normvideo;
124     Writeln;
125     Writeln('Druck auf beliebige Taste außer der Taste mit dem Buchstaben p ');
126     Writeln('läßt die Tasternummer auf dem Bildschirm erscheinen');

```

```
127   Writeln('In der nächsten Zeile ist der Inhalt des Tastaturpuffers zu sehen.');
```

```
128   Writeln('Das oberste Bit ist offensichtlich 1, wenn die Taste losgelassen wird.');
```

```
129   Writeln;
```

```
130   Writeln('Ende      :   Bitte p drücken');
```

```
131
```

```
132   Zei := Readkey;    { Diese Prozedur, die eine Tasteneingabe verarbeitet,
```

```
133                     muß nur einmal aufgerufen werden, da sie nur in Aktion tritt,
```

```
134                     wenn die neue Interruptprozedur die alte aufruft. Dies ist
```

```
135                     dann der Fall, wenn die Taste "p" gerückt wird, mit der das
```

```
136                     Programm beendet werden soll.}
```

```
137
```

```
138   { Alter KBD-Interruptvektor wird von Pascal mit PROGRAMM_EXIT restauriert }
```

```
139 end.
```

```
140
```

```
141
```