

```

1 Program Tastint3;
2 (*****
3 (***)          Tastint2 8.6.1998/12.5.2006          (***)
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 (***) - Restaurieren des alten Interrupt-Vektors wird NICHT in der (***)
15 (***) Schleife EXITPROC, sondern am Programmende erledigt. (***)
16 (***) Dann wird das Programm durch Division durch Null zum Absturz ge- (***)
17 (***) bracht.                                       (***)
18 (***) Beobachten sie die Folgen!!!!!!             (***)
19 (***)                                                (***)
20 (***) Achtung!!! Um in Windows die Folgen zu sehen, müssen Sie das (***)
21 (***) übersetzte Programm Tastint3.exe in einem DOS-Fenster (***)
22 (***) starten.                                     (***)
23 (***) Windows ist nämlich so freundlich, diesen Fehler (***)
24 (***) in irgendeiner Weise mit dem Pascal-System abzufangen. (***)
25 (***)                                                (***)
26 (***)                                                (***)
27 (***)                                                (***)
28 (*****
29 {$M 1024,0,256} { Begrenzung des Speicherplatzbedarfs für das residente
30 Programm }
31
32 Uses DOS, CRT;
33
34 Type T_Prozedur = Procedure; { Prozedurtyp, ist im Grunde genommen ein Zeiger, der
35                               auf den ersten Speicherplatz einer Prozedur im
36                               Codesegment zeigt }
37
38 VAR  pAlter_KBD_Intr      : Pointer;
39       Alter_KBD_Intr      : T_Prozedur ABSOLUTE pAlter_KBD_Intr;
40                               { Alte Keyboard-Interrupt-Prozedur im ROM-BIOS
41                               Diese Prozedur kann zunächst nicht über einen Namen aufgerufen werden,
42                               da sie keine Pascalprozedur ist. Sie beginnt am absoluten Spei-
43                               cherplatz pAlter_KBD_Intr. Dieser Speicherplatz muß bei der
44                               Programminitialisierung erst ermittelt werden. Damit läßt sich nun
45                               die ROM-Prozedur mit dem Namen Alter_KBD_Intr wie eine normale
46                               Pascal-Prozedur aufrufen }
47
48     pAlte_Exit_Prozedur : Pointer; { Beim Verlassen des Pascalprogramms, ob regulär oder infolge
49                                     eines Abbruchs durch einen Laufzeitfehler wird standard-
50                                     mäßig eine Prozedur aufgerufen, die am Speicherplatz ExitProc
51                                     beginnt. Der Zeiger ExitProc ist in Pascal definiert.
52                                     Sollen über den Standardausstieg hinaus weitere Aktionen
53                                     durchgeführt werden, wie zum Beispiel das Restaurieren
54                                     "verbogener" Interruptvektoren, so kann der Zeiger ExitProc
55                                     auf die eigene Prozedur gerichtet werden.}
56
57 Var Zei      : Char;
58     Absturz   : real; { künstlichen Absturz erzeugen }
59
60 Procedure KBDNeu; Interrupt;
61 Var Taste    : Byte;
62 Begin
63     Taste := Port[$60]; { Auslesen des Tastaturpuffers }
64     Begin

```

```

64      Gotoxy(10,15); Clreol;           { Zeile 15 ab Spalte 15 löschen }
65      Write(' Taste Nr.',Taste AND $7F); { Höchstwertiges Bit ausblenden }
66      if Taste < 128 then              { Oberstes Bit ist Flag, ob Taste }
67          Write(' gedrückt')          { drückt (Flag = 0) oder losgelassen }
68      else                             { ist (Flag = 1) }
69          Write(' losgelassen');
70
71      Gotoxy(10,16); Clreol;
72      Write(' Tastaturpufferinhalt: ',Taste);
73
74      Gotoxy(10,18);
75      Write(' Falls die Zeile "port[$20] := $20] auskommentiert ist,');
76      Gotoxy(10,19);
77      Write(' dann war das war die letzte Aktion in diesem Programm! ');
78      Gotoxy(10,20);
79      Write(' Der Interrupt-Controller wird nicht mehr freigegeben!');
80      Gotoxy(10,22);
81      Write(' Bei Druck auf die Taste "p" wird die alte BIOS-Routine aufgerufen,');
82      Gotoxy(10,23);
83      Write(' die das Zeichen dem Pascalprogramm übergibt.')
```

End;

```

85  { Hier wird nun nicht mehr die alte Tastaturroutine aufgerufen, die von
86    sich aus den Interruptcontroller wieder freigibt. Der Rechner reagiert daher nicht mehr und
87    muß durch Hardware Reset wieder gestartet werden }
88  if taste = 25 then { Taste mit Buchstaben p gedrückt!}
89      Begin
90          ASM
91              pushf
92          END;
93          alter_KBD_intr;
94      End
95  Else
96      Port[$20] := $20; { Interrupt Controller freigeben! Zum Test bitte
97                        durch Kommentarklammern unwirksam machen}
98
99  end;
100
101
102
103
104  (*-----*)
105
106  Begin (*Hauptprogramm*)
107      GetIntVec($9,pAlter_KBD_Intr); { Vektor auf die alte Tastaturinterrupt-Serviceroutine in der
108                                     absoluten Prozedurvariablen Alter_KBD_intr retten.
109                                     Damit kann die alte Prozedur mit diesem Bezeichner
110                                     aufgerufen werden }
111      Setintvec($9, @KBDNeu);        { Vektor Nr. 9 auf eigene Routine richten }
112
113
114      ClrScr;
115      Textcolor(green);
116      Highvideo;
117      Writeln('*****      Test der neuen Tasturoutine      *****');
118      Normvideo;
119      Writeln;
120      Writeln('Druck auf beliebige Taste außer der Taste mit dem Buchstaben p ');
121      Writeln('läßt die Tastennummer auf dem Bildschirm erscheinen');
122      Writeln('In der nächsten Zeile ist der Inhalt des Tastaturpuffers zu sehen.');
```

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

```

124  Writeln;
125  Writeln('Ende      :   Bitte p drücken');
```

```
127     Zei := Readkey;    { Diese Prozedur, die eine Tasteneingabe verarbeitet,  
128                        muß nur einmal aufgerufen werden, da sie nur in Aktion tritt,  
129                        wenn die neue Interruptprozedur die alte aufruft. Dies ist  
130                        dann der Fall, wenn die Taste "p" gerückt wird, mit der das  
131                        Programm beendet werden soll.}  
132  
133  
134  
135     Absturz := 0;  
136     Absturz := 1/Absturz;  
137  
138     SetIntVec($9,pAlter_KBD_Intr);    { Interruptvektor Nr. 9 = Zeiger auf alte  
139                                       Serviceroutine des Tastaturinterrupts wieder  
140                                       herstellen }  
141  
142     end.  
143  
144
```