

Schnelle Taste



Für das Warten auf eine Taste ist in Turbo Pascal standardmäßig die Funktion *KeyPressed* zuständig. Da diese Funktion auf den BIOS-Interrupt 16h zurückgreift, ist sie relativ langsam. Der direkte Zugriff auf den Tastaturpuffer bereitet unter Turbo Pascal jedoch keine Probleme, es bietet sich also an, die Abfrage einer kleinen Inline-Funktion zu übertragen. Die Assembler-Funktion *FastKeyPressed* von **Guido Janzen** aus Paderborn greift direkt auf den Tastaturpuffer zu, um zu überprüfen, ob sich darin ein Tastencode befindet.

Wie Sie vielleicht wissen, ist der Tastaturpuffer als Ringpuffer angelegt. Zwei Zeiger im BIOS-Variablenbereich zeigen auf das erste und letzte Zeichen dieses Puffers. Weisen beide Zeiger den gleichen Wert auf (das heißt, beide Zeiger deuten auf dasselbe Zeichen), dann ist der Puffer leer. Andernfalls befindet sich mindestens ein Zeichen im Tastaturpuffer. Die Inline-Funktion *FastKeyPressed* (Listing) überprüft die Zeiger und stellt so fest, ob ein Zeichen in den Tastaturpuffer übertragen wurde.

Die ersten beiden mov-Befehle dienen dazu, die Segmentadresse 0001h in das ES-Register zu laden. Über diese Adresse läßt sich der BIOS-Variablenbereich adressieren. Anschließend wird der Zeiger auf das erste Zeichen (Offsetadresse 40Ah) in das DX-Register geladen und mit dem Zeiger auf das letzte Zeichen (Offsetadresse 40Ch) verglichen. Unterscheiden sich die beiden Werte, so bedeutet dies, daß eine Taste gedrückt wurde. Als Resultat wird im AX-Register der Wert 1 (wahr) übergeben. Erfolgte dagegen kein Tastendruck (beide Zeigerinhalte sind identisch), verringert die letzte Programmzeile den Wert im AX-Register um eins, so daß die Funktion den Wert 0 (falsch) zurückliefert. Zwar ändert die Funktion die Inhalte der Register AX, DX und ES, doch stellt dies kein Problem dar, da Turbo Pascal lediglich erwartet, daß sich die Inhalte der Register BP, SP, DS und SS nicht ändern. Vergleicht man die Rechenzeiten der Funktionen *KeyPressed* und *FastKeyPressed* (zum Beispiel mit Hilfe des Turbo Profilers), so ergibt sich eine Geschwindigkeitssteigerung um den Faktor 10. Glauben Sie daß man den Vorgang noch weiter beschleunigen kann? Wenn nicht, dann lassen Sie sich überzeugen. Turbo Pascal ordnet den Booleschen Zu ständen *wahr* und *falsch* normalerweise die Werte 1 und 0 zu erkennt aber bei Abfragen alle Werte ungleich Null als *wahr* an. Mit anderen Worten: Im AX-Register muß keine 1 stehen, damit der Inhalt als *wahr* erkannt wird. Sie können sich daher in der Assembleroutine einen direkten Vergleich sparen, indem Sie diesen durch eine Exklusiv-Oder-Verknüpfung ersetzen. Die Inline-Funktion *EvenFasterKeyPressed* (Listing) zeigt Ihnen die Umsetzung. Das Resultat spricht für sich: Auf diese Weise lassen sich noch einmal ungefähr 20 Prozent Ausführungszeit einsparen.

```
function FastKeyPressed : boolean;
inline ( $B8/$01/$00/      { mov AX,0001h }
         $8E/$C0/          { mov ES,AX   }
         $26/$8B/$16/$0A/$04/ { mov DX,word PTR ES:[40Ah] }
         $26/$3B/$16/$0C/$04/ { cmp DX,word PTR ES:[40Ch] }
         $75/$01/          { jne 01h   }
         $48 );            { dec AX    }

function EvenFasterKeyPressed : boolean;
inline ( $31/$C0/          { xor AX,AX  }
         $8E/$C0/          { mov ES,AX  }
         $26/$A1/$1A/$04/   { mov AX,word PTR ES:[41Ah] }
         $26/$33/$06/$1C/$04 ); { xor AX,word PTR ES:[41Ch] }
```

Listing: die "turboschnellen" Eingabefunktionen als Alternative zu *KeyPressed*