

```

#####      #####      #####
#####      #####      #####
  ##      ##      ##      ##
  ##      #####      ##      ##
  ##      #####      ##      ##
  ##      ##      ##      ##
  ##      #####      #####
  ##      #####      #####

```

```

#####      #
#          #
#  #  #  #  #  #  #  #
#  #  #  #  #  #  #
#  #  #  #  #  #  #

```

TurboEditor

TED 1.4 890906/K1

Willkommen zur TED-DEMO!

TED zeigt Ihnen in den naechsten Minuten, was er alles kann.

Sie koennen die DEMO jederzeit durch Eingabe einer beliebigen Taste anhalten bzw. fortsetzen oder mit ESC abbrechen und in gewohnter Weise mit dem Editor arbeiten.

Die DEMO laeuft ca. 20 bis 30 Minuten. Falls Sie bislang eine

Dokumentation vermisst haben - im Anschluss an die DEMO wird sie automatisch aus dem Tutorial erzeugt.

Beachten Sie aber bitte, dass diese DEMO-Version lediglich eine Minimalvariante von TED ist und damit die maximal bearbeitbare Dateigroesse bei ca. 20 bis 25 kByte (ungefaehr 1000 Textzeilen) liegt. Normalerweise stellen 200 kByte-Dateien keine Huerde fuer TED dar.

* * *

Grundsaeztliche Erlaeuterungen zu TED

TED ist ein WordStar-kompatibler Programm- und Texteditor. Sein Leistungsvermoegen entspricht dem non-document-mode von WordStar.

Man darf von TED also in Bezug auf ausgefeilte Textformatierung keine Wunderdinge erwarten.

Er ist jedoch insbesondere Programmierern eine nuetzliche Hilfe, die vor allem Wert auf schnelle und saubere Edition ihrer Programmtexte legen.

TED arbeitet schnell, da die bearbeitete Datei komplett im Hauptspeicher des Rechners gehalten wird. Es werden keine Zwischendateien auf der Diskette angelegt und verwaltet.

Die maximale Groesse einer bearbeitbaren Datei wird damit von der Groesse des fuer Nutzer zur Verfuegung stehenden Speichers bestimmt.

Die Varianten des TED-Aufrufs sind

TED - TED erfragt die zu bearbeitende Datei
TED dateiname - Die angegebene Datei wird bearbeitet.

An den Dateinamen werden die TED-Versionen fuer ADM31- oder VT100-kompatible WEGA-Terminals unterschieden:

TEDA - Version fuer WEGA/ADM31
TEDV - Version fuer WEGA/VT100

Die vorliegende DEMO-Version startet normalerweise den TUTOR. Sie koennen dies jedoch unterbinden, indem Sie die Datei TUTOR.TED loeschen oder umbenennen. Der TUTOR wird dann nicht aktiviert.

TED ist nach seinem Vorbild, der TURBO-PASCAL-EDITOR-TOOLBOX (Entwickler ???), in C geschrieben und weiterentwickelt worden und damit jetzt auf die verschiedensten Maschinen und Betriebssysteme portierbar!

Die aktuelle Version TED 1.4 ist durch Aenderung einer Praeprozessoranweisung und anschliessender Neukompilation bei gleicher Nutzeroberflaeche sowohl unter DCP als auch unter WEGA lauffaehig! Welches andere Programm kann dies schon von sich behaupten?!

TED nutzt - durch Praeprozessoranweisungen steuerbar - die Moeglichkeiten der Maschine aus, auf der er laeuft. Damit koennen z.B. schnellere Algorithmen zur Terminal- bzw. Bildschirmsteuerung oder Speicherverwaltung eingebunden bzw. ausgegliedert werden.

Da TED durch bedingte Compilation auch ganz klein gemacht werden kann, eignet er sich hervorragend zur Einbindung in groessere Projekte, die einen Editor benoetigen.

Machen Sie also von unserem Angebot Gebrauch, die TED-Quellen nachzunutzen, um den Editor optimal an Ihre Wuensche anpassen zu koennen!

* * *

Die Kommandos von TED

Allgemeine Kommandos

Bei der folgenden Beschreibung der TED-Kommandos wird zur Darstellung der CTRL-Taste das Zeichen "^" synonym verwendet. Wenn verfuegbar, werden die entsprechenden Sondertasten fuer die einzelnen Systeme (DCP bzw. WEGA/VT100) zusaetzlich angegeben.

Mit Eingabe von <^J> (Taste CTRL und J zusammen betaeligen) bietet TED kurze Hilfestellung an. Die Eingabe von <ESC> schaltet zurueck.

Beispiel: Zweimalige Eingabe von <^J> und <ESC>.

Zunaechst beachten Sie bitte die Statuszeile. TED zeigt dort an, in welcher Zeile und Spalte des Textes er sich momentan befindet.

Ebenso wird der Editorschalter "Einfuegen" in seiner momentanen Stellung angezeigt. Dabei bedeutet

Insert : Eingabezeichen werden ab Cursorposition in den Text eingefuegt;
Overwrite : Eingabezeichen ueberschreiben den momentanen Text ab Cursorposition.

Die Umschaltung zwischen beiden Modi erfolgt mit Eingabe von <^V>.

Beispiel: Viermalige Eingabe von <^V>. Beachten Sie die Statuszeile.

Mit dem Einrueckmodus (Editorschalter "Einruecken") wird die Gestaltung strukturierter Programmquelltexte erleichtert: Wird "Indent" angezeigt, beginnt der Quelltext der neuen Zeile auf der Hoehe der Einrueckung der vorhergehenden Zeile.

Indent : automatischer Einrueckmodus ist eingeschaltet;
Unindent : automatischer Einrueckmodus ist ausgeschaltet.

Die Umschaltung zwischen beiden Modi erfolgt mit Eingabe von <^OI>.

Beispiel: Umschaltung in den Indent-Mode, Schreiben von fuenf fiktiven Quelltextzeilen mit automatischer Einrueckung und Rueckschaltung in den Unindent-Mode.

```
if (x==y) {  
  a=b<5 ? y : z;  
  z=3;  
  y=4;  
}
```

Natuerlich laesst sich derselbe Effekt auch mit Eingabe von Tabulatoren - <^I> oder Sondertaste -->| (DCP) bzw. Pfeil rechts (WEGA) - erreichen. Die Tabulatorsprungweite (Standard 8 Spalten) kann mit <^OT> gefolgt von einer Ziffer zwischen 1 und 8 eingestellt werden.

Beispiel: Umschaltung der Tabulatorsprungweite auf 4 und zurueck auf 8 Spalten.

Wird bei der Texteingabe der rechte Rand erreicht, erfolgt ein horizzontaler soeben gesehenen Form.

In der Statuszeile wird abschliessend der Name der momentan bearbeiteten Datei (mit Laufwerks- und Pfadbezeichner) angezeigt. Wurde auch auf Nachfrage beim Start kein Dateiname angegeben, so heisst die momentane Editordatei automatisch "NONAME".

* * *

Kommandos zur Cursorbewegung

Die Kommandos zur Cursorbewegung sind, wie alle WordStar-Kommandos, Kombinationen der CTRL-Taste mit einem Buchstaben. Einige Rechner bieten zusaetzliche Unterstuetzung durch einen Cursorstastenblock. Diese wird von TED, was DCP betrifft, voll genutzt.

Unter WEGA ist die Sache nicht ganz so einfach. Im verbreiteten ADM31-Terminalmode z.B. liefern die Cursorstasten "Scan-Codes", die mit gaengigen WordStar-Kommandos kollidieren. Zur Wahrung der WordStar-Kompatibilitaet wurde im ADM31-Modus auf die Cursorstastenunterstuetzung verzichtet.

Bei VT100-kompatiblen WEGA-Terminals koennen Cursor- und Funktionstasten

verwendet werden. Neuere Terminals sind mit der Funktionstaste <F1> zwischen beiden Modi umschaltbar.

<^S> : Cursor ein Zeichen nach links (DCP/VT100: CURSOR LEFT).
<^D> : Cursor ein Zeichen nach rechts (DCP/VT100: CURSOR RIGHT).
Zeilengrenzen werden ggf. ueberschritten.

Beispiel: Cursor nach links und rechts.

<^E> : Cursor eine Zeile hoch (DCP/VT100: CURSOR UP).
<^X> : Cursor eine Zeile runter (DCP/VT100: CURSOR DOWN).

Beispiel: Cursor hoch und runter.

<^W> : Bildschirm eine Zeile nach unten rollen.
<^Z> : Bildschirm eine Zeile nach oben rollen.
Cursorposition bleibt erhalten.

Beispiel: Bildschirm runter und hoch rollen.

<^A> : Cursor ein Wort nach links (DCP: CTRL-LEFT; VT100: |<--).
<^F> : Cursor ein Wort nach rechts (DCP: CTRL-RIGHT; VT100: -->|).

Beispiel: Cursor wortweise nach links und rechts.

<^R> : Cursor eine Bildschirmseite zurueck (DCP: PGUP; VT100: F13).
<^C> : Cursor eine Bildschirmseite vor (DCP: PGDN; VT100: F14).

Beispiel: Cursor seitenweise zurueck und wieder vor.

* * *

Kommandos zum Loeschen und Einfuegen

<^G> : Zeichen auf Cursorposition loeschen.

Beispiel: Gehe an den Anfang von "loeschen", loesche die ersten 3 Zeichen und stelle sie danach wieder her.

<^H> : Zeichen links vom Cursor loeschen (DCP/VT100: DEL).

Beispiel: Gehe an das Ende von "loeschen", loesche die letzten 3 Zeichen und stelle sie danach wieder her.

<^T> : Wort rechts vom Cursor loeschen.

Beispiel: Gehe an den Anfang von "Gehe", loesche die naechsten 3 Woerter und stelle Zeile wieder her.

<^Y> : Cursorzeile loeschen.

Beispiel: Zeile schreiben und wieder loeschen.

<^N> : Zeile einfuegen. Ab der aktuellen Cursorposition wird eine Leerzeile eingefuegt. Wenn der Cursor sich nicht am Zeilenanfang befindet, wird die Zeile an dieser Stelle getrennt. Der Cursor verbleibt in der getrennten Zeile.

Beispiel: Zeile schreiben, Wort zurueck und trennen.

Diese Zeile wird vor dem letzten Wort getrennt.

* * *

Das ^Q-Menue

Mit der Eingabe von ^Q gefolgt von einem Buchstaben werden weitere Kommandos zur Cursorpositionierung, fuer Suche und Austausch von Strings und Zusatzfunktionen zugaenglich.

- <^QR> : Cursor an den Dateianfang (DCP: CTRL-PGUP; VT100: Taste Pfeil nach links/oben).
- <^QC> : Cursor an das Dateiende (DCP: CTRL-PGDN; VT100: F9).
- Beispiel: Cursor an den Dateianfang und zurueck an das Dateiende.
- <^QE> : Cursor an den Anfang der Bildschirmseite (DCP: CTRL-HOME).
- <^QX> : Cursor an das Ende der Bildschirmseite (DCP: CTRL-END).
- Beispiel: Cursor an Seitenanfang und Seitenende.
- <^QS> : Cursor an den Anfang der Zeile (DCP: HOME; VT100: F7).
- <^QD> : Cursor an das Ende der Zeile (DCP: END; VT100: F12).
- Beispiel: Cursor an Zeilenanfang und Zeilenende.
- <^QZ> : Sprung zur Zeile xxxxx.
- Beispiel: Zum Dateianfang, danach zur Zeile 50, anschliessend zurueck an das Dateiende gehen.
- <^Qn> : Sprung zur Marke n (n=0...9). Diese Funktion wird im ^K-Menue naeher erlaeutert.
- <^QY> : Loeschen einer Zeile von Cursorposition bis Zeilenende.
- Beispiel: Schreiben einer Zeile, 3 Woerter zurueckgehen und bis Zeilenende loeschen, Zeile wiederherstellen.
- Von dieser Zeile werden die letzten drei Woerter geloescht.
- <^QL> : Wiederherstellen einer manipulierten Zeile. Die Zeile darf nicht bereits komplett geloescht worden sein.
- Beispiel: Schreiben einer Zeile, Loeschen einzelner Woerter und wiederherstellen des alten Inhalts.
- Diese Zeile wird manipuliert und anschliessend wiederhergestellt.
- <^QF> : Suche eines Strings von max. 32 Zeichen Laenge ab Cursorposition vor- oder rueckwaerts und Setzen des Cursors auf die Fundstelle. CTRL-Zeichen koennen nicht gesucht werden.
- Optionen: u - Egal, ob Gross- oder Kleinschreibung;

w - nur ganze Woerter;
b - rueckwaerts ab Cursorposition.

Optionen werden in beliebiger Reihenfolge und ohne Trennzeichen eingegeben.

Mit Eingabe von <^L> wird der vorangegangene Suchlauf ab Cursorposition wiederholt. Bei der Eingabe von Suchstrings und Optionen werden mit <^R> die letzten Eintraege uebernommen.

Beispiel: Zum Dateianfang gehen, String "Optionen" als Wort suchen, wiederholen ab Fundstelle, zum Ende gehen.

<^QA> : Suche eines Strings von max. 32 Zeichen Laenge ab Cursorposition und Tausch gegen einen anderen String von max. 32 Zeichen Laenge. Der Cursor wird auf die Austauschposition gesetzt. CTRL-Zeichen koennen nicht getauscht werden.

Optionen: u - Egal, ob Gross- oder Kleinschreibung;
w - nur ganze Woerter;
b - rueckwaerts suchen;
g - in der gesamten Datei;
n - unterdruecke Abfrage vor dem Tausch.

Optionen werden in beliebiger Reihenfolge und ohne Trennzeichen eingegeben.

Mit Eingabe von <^L> wird der vorangegangene Suchlauf wiederholt. Bei der Eingabe von Suchstrings und Optionen werden mit <^R> die letzten Eintraege uebernommen. Die Eingabe eines beliebigen Zeichens in den Modi "gbn" oder "gn" unterdrueckt die Anzeige aller Austauschpositionen, <^U> bricht den Austauschvorgang ab.

Beispiel: Suche des Strings "Optionen" als Wort in der gesamten Datei und Austausch gegen "Optionen" mit vorheriger Rueckfrage. Der Ruecktausch erfolgt beginnend ab Dateijende ohne Rueckfrage.

<^QP> : Diese Zusatzfunktion wird im Anschluss an die Funktionen des ^K-Menues erlaeutert.

* * *

Das ^K-Menue

Das ^K-Menue beinhaltet Kommandos zum Lesen, Schreiben, Loeschen, Verschieben und Kopieren von Bloecken und Dateien. Ein Block ist ein von zwei explizit zu setzenden Markierungen (Blockanfangs- und Blockendemarke) geklammerter Bereich der bearbeiteten Datei.

- <^KB> : Fuer eine der nachfolgenden Loesch-, Schreib-, Verschiebe- und Kopieroperationen wird die Blockanfangsmarke auf die aktuelle Cursorposition gesetzt.
- <^KB> : Fuer eine der nachfolgenden Loesch-, Schreib-, Verschiebe- und Kopieroperationen wird die Blockendemarke auf die aktuelle Cursorposition gesetzt.

Die Blockmarken sind nicht sichtbar!

Beispiel: Zu Zeile 4 gehen, Blockanfangsmarke setzen, zu Zeile 14 gehen, Blockendemarke setzen.

- <^KW> : Ablegen eines Blocks auf Diskette.

Beispiel: Der oben geklammerte Block wird als "TED.DOC" geschrieben.k

- <^KY> : Loeschen eines Blocks ohne Rueckfrage!

Beispiel: Der oben geklammerte Block wird geloescht.

- <^KR> : Einlesen eines Blocks (einer Datei) von Diskette.

Beispiel: Einlesen der Datei "TED.DOC" ab der aktuellen Cursorposition.

- <^KV> : Verschieben eines Blocks zur Cursorposition. Der Block wird auf seiner vorherigen Position geloescht.

- <^KC> : Kopieren eines Blocks zur Cursorposition. Der Block bleibt auf seiner vorherigen Position erhalten.

Beispiel: Verschieben des eingelesenen Blocks an seine alte Position ab Zeile 4. Dazu 10 Zeilen hoch gehen und Blockendemarke setzen, nochmals 10 Zeilen hoeher Blockanfangsmarke setzen, zu Zeile 4 gehen und Block verschieben.

<^KP> : Ausgabe eines Blocks auf den Drucker. Die Ausgabe erfolgt auf folgende Devicefiles:

```
WEGA: File "/dev/lp"
DCP:  File "PRN"
```

Diese Zuweisungen sind nur durch entsprechende Generierung und Neukompilation von TED aenderbar. Sollten sie nicht mit den in Ihrem System gegebenen Zuweisungen uebereinstimmen, so ist eine Ausgabe auf den Drucker auch ueber die ^KW-Funktion unter Angabe Ihres Devicefiles moeglich.

Beispiel: /

<^Kn> : Setze Textmarke n (n=0...9) auf die aktuelle Cursorposition. Ein Wiederauffinden der somit markierten Textstelle ist ueber die Funktion ^Qn moeglich.

Beispiel: Gehe an Dateianfang, setze Textmarke 5, gehe an Dateiende und ueber ^Q5 wieder an den Anfang.

<^KS> : Sichern der aktuellen Datei und Verbleib im Editor.

<^KD> :

<^KX> : Sichern der aktuellen Datei und Rueckkehr zum System. Vor dem Abspeichern wird aus der eventuell bereits vorhandenen aktuellen Datei eine Back-Up-Datei erzeugt, deren Name folgendermassen gebildet wird:

```
WEGA:   Dateiname 'datei1'      -> 'datei1.bak'
        'datei1.v12.s' -> 'datei1.v12.bak'

DCP:   Dateiname 'DATEI1'      -> 'DATEI1.BAK'
        'DATEI1.V12'  -> 'DATEI1.BAK'
```

Eine alte "*.bak"-Datei geht verloren.

Beispiel: Sichern der aktuellen Datei als "TED.DOC" und Verbleib im Editor.

<^KQ> : Abbruch der Bearbeitung nach Rueckfrage und Verlassen des Editors ohne Sicherung der aktuellen Datei!

Beispiel: Abbruch wird angefordert, Rueckfrage jedoch negativ beantwortet.

* * *

So, damit waeren bis auf eine Funktion alle derzeitigen Moeglichkeiten von TED dargestellt.

Bei der abschliessend kurz zu demonstrierenden Funktion handelt es sich um das zusaetzlich in das ^Q-Menue aufgenommene "Datei-Pick-Up", erreichbar ueber <^QP>.

Sinn der Datei-Pick-Up-Funktion ist es, bei der quasiparallelen Arbeit mit mehreren Textdateien (z.B. ein Programm bestehend aus mehreren Quelltextmodulen) ein schnelles "Hin- und Herschalten" zwischen den einzelnen Dateien zu ermoeglichen.

Dazu fuehrt TED eine sogenannte "Pick-Liste", in der bis zu acht Dateien gleichzeitig verwaltet werden koennen. TED merkt sich beim "Swappen" von Dateien die momentane Cursorposition und alle Block- und Textmarken, so dass ein schnelles Wiederauffinden von Textstellen gesichert ist. Suchstrings und Optionen bleiben beim Umschalten erhalten.

Bei jedem ersten Aufruf der ^QP-Funktion versucht TED, die Pick-Liste aus einer Datei "*.pck" zu initialisieren (* = Name der aktuellen Arbeitsdatei, die beim Start des Editors angegeben wurde). Die Datei "*.pck" wird beim Verlassen von TED automatisch aktualisiert.

Ein kleines Beispiel: Wir haben jetzt auf jeden Fall zwei Textdateien auf der Diskette - die Tutor-Kommandodatei TUTOR.TED und die vorhin mit <^KS> erzeugte Datei TED.DOC. Unsere aktuelle Datei ist TED.DOC. Wir wollen die Datei TUTOR.TED einladen und geben dazu <^QP> ein.

Da die Pick-Liste noch leer ist, erscheint als einziger Wahlpunkt in der Statuszeile "Load New File", den wir mit <RET> selektieren. Die anschliessende Frage "Save Current File" beantworten wir mit "Y-es". Damit wird die momentane Editordatei TED.DOC aktualisiert.

TUTOR.TED wird gelesen. Wir blaettern ein wenig darin herum und rufen danach erneut die Pick-Funktion auf. Diesmal enthaelt die Pick-Liste als erstes Element die letzte Datei, naemlich TED.DOC. Wir koennten sie sofort mit <RET> selektieren, wollen jedoch noch sehen, ob noch andere Eintraege in der Liste verfuegbar sind. Geblaettert wird mit

den Cursorstasten UP und DOWN, bzw. <^E> und <^X>.

Schliesslich entscheiden wir uns doch fuer TED.DOC. Die Frage "Save Current File" beantworten wir diesmal mit "N-o", denn an TUTOR.TED sind ja keine Aenderungen vorgenommen worden.

Nachdem TED.DOC eingelesen wurde, befinden wir uns wieder an der Stelle, an der wir auf TUTOR.TED umgeschaltet hatten.

Und nun dieses komplexe Beispiel in zusammenhaengender Abfolge.

* * *

Damit waeren wir am Ende der TED-DEMO.
Danke fuer Ihre Geduld.

Der aktuelle Editortext wird unter dem Namen TED.DOC gesichert.
So koennen Sie jederzeit in Ruhe nachlesen, falls es Ihnen eben zu schnell war.

Mit allen weiteren Problemen, Wuenschen und Fragen zu TED wenden Sie sich bitte an

Dr.-Ing. Roland Klabunde

Hochschule fuer Seefahrt Warnemuende/Wustrow
Sektion Schiffsfuehrung, WB FE

WARNEMUENDE

2530

Tel.: Rostock / 57 287

* * *