C H R I S T I A N   S C H A R R N B E C K

Chr.Scharrnbeck@t-online.de
☎ 05 5 55 – 33 33 33
Helmut–Kopka–Str. 21
0 2 8 2 0 TEXas

Christian Scharrnbeck Helmut–Kopka–Str. 21 02820 TEXas

An die Programmierer,
die gleichzeitig
an LATEX

interessietrt sind.

Betr.: **Vorschlag für ein LATEX–Frontend–Konzept nach STEPPLER**

TEXas, dem 11.03.2001

Hallo LATEX–Freunde

Um die Dokumentenklasse BRIEF als Beispiel darstellen zu können, wurde der gesamte Text in der Briefform abgefaßt.

Warum soll Vergessenes, aber Gutes verloren gehen?

Inhalt

LaTeX für Einsteiger

Hinter LaTeX verbirgt sich ein Programmpaket, das neben seinem billigen Preis vor allem durch sein Schriftbild zu überzeugen weiß. Andererseits wird ihm (zu recht) nachgesagt, daß man seine Texte mehr oder weniger programmieren muß. Deshalb findet es bisher auch noch nicht die Verbreitung, die ihm eigentlich gebührt.

Vielfach wird es gern gesehen oder gar gewünscht, daß Studienarbeiten bis hin zur Promotion neben der natürlich zu fordernden inhaltlichen Güte auch ein vorzügliches und quasi standardisiertes Aussehen aufweisen sollen. Dazu wird nicht selten an Hochschulen auf LaTeX verwiesen. Diesem Verweis in Fakultäten der Informationstechnik dürften selten Schwierigkeiten entgegenstehen. Dagegen wird es an anderen Fakultäten und "anderen höheren Schulen" gewisse Probleme geben, obwohl heute fast jeder Student einen Computer zum Schreiben seiner Arbeiten verwenden dürfte. Aber für diese ist eben der Schritt vom üblichen Textprogramm zu LaTeX besonders schwer, auch lohne sich der hohe Umlernaufwand für "diese paar Mal" nicht.

Der LaTeX–Wunsch

Gäbe es ein geeignetes Frontend zu LaTeX, das fast wie ein normales Textprogramm nutzbar wäre, in dem all die für die jeweilige Schule, Hochschule bzw. Fakultät zu beachtenden Normen für den Schriftsatz... versteckt enthalten sind, dann stünden dem verbreiteten Einsatz wesentlich weniger Probleme im Wege.

Auch für LaTeX–Profis durch hohe Übersichtlichkeit bestens geeignet

Aber auch (werdende) Profis könnten, in solcher Art in LaTeX zu schreiben, wesentlich profitieren: Durch den Einsatz von Formblättern mit relavanten Schlüsselwörtern, die das Gerüst des zu formulierenden Textes vorgeben, gewänne das Textfeld auf dem Monitor eine besondere Klarheit, die es ermöglicht, sich voll auf den Inhalt zu konzentrieren.

Mit solch einem Frontend würde sich eine relativ universelle Anwendung von LaTeX vom wissenschaftlichen Buch bis zur Banküberweisung anbieten, so daß für Rountineaufgaben weitestgehend auf andere Textprogramme verzichtet werden könnte. Und dadurch steigen von Mal zu Mal die speziell benötigten Erfahrungen mit LaTeX und schließlich fände man auch hin oder wieder den (dann nicht mehr so komplizierten) Zugang zum direkten Bearbieten von *.tex–Files.

In einem solchen Frontend würden sich die routinemäßig zu benutzenden (z.T. umfangreichen) LaTeX–Befehlssequenzen hinter Schlüsselwörtern verbergen. Auf diese Weise verschwände LaTeX in den Hintergrund. Man schreibt seinen Text in das Frontend, benutzt vielleicht zusätzlich den einen oder anderen LaTeX–Befehl und gibt den (Menü)–Befehl VORSCHAU. Ohne daß man LaTeX selbst starten muß, wird der geschriebene Text, gesteuert durch die Schlüsselwörter zunächst auf seine richtige Struktur überprüft, in ein *.tex–File übersetzt, anschließend nach *.dvi compiliert und schließlich auf dem Monitor angezeigt bzw. ausgedruckt.

Konzepte als Open–Source

Über ein Programm—Konzept zu schreiben, ist unüblich. Aber im Zeitalter der OpenSource dürfte es hierzu keine Beschränkung "nach unten" geben. Ein gutes Konzept, an dem seit langem nicht mehr weitergearbeitet wird, das aber sehr leistungsfähig ist, das sollte nicht verloren gehen. Der Autor dieses Beitrages ist selbst nicht in der Lage, die Last einer solchen Programmierung auf sich zu nehmen, deshalb will er andere animieren, ein solches Projekt für LINUX zu verwirklichen.

Das existierende Frontend

Ein Frontend, das solches zu leisten im Stande ist, gibt es schon seit Anfang der 90er Jahre. Es bietet von seiner Grundkonzeption her einerseits einen sehr bequemen Zugang zu den üblichen LaTeX–Anwendungen wie Buch, Brief, Artikel usw., ist aber auch z.B. für " Alltägliches" wie das routienemäßige Ausfüllen von Überweisungen relativ leicht z.B. unter den Klassen Brief, Artikel programmierbar. Mit diesen Möglichkeiten kann es andere Textprogramme weitestgehend ersetzen. Selbstverständlich sind mit dem Frontend auch verschiedenartige Listen und Tabellen sehr leicht generierbar und Bilder (auch maßstabverändernd) in Dokumente einbindbar.

Dieses Frontend und seine zugehörige LaTeX–Umsetzung PasTeX laufen auf dem Anfang der 90er Jahre sehr beliebten Betriebssystem AmigaOS und hatten damals einen sehr hohen Stand erreicht. Das wohl wichtigste Merkmal dieses Frontend ist, daß das hier vorzustellende Paket vom User selbst bei Bedarf um weitere Features leicht erweitert bzw. an seine speziellen Bedürfnisse angepaßt werden kann.

Der Autor von TPP

Dieses Frontend entstand in den Jahren 1988 bis 1994 als Shareware. Autor ist Martin Steppler. Er nannte sein Programm TeXt Plus Professional (TPP). Die letzte Version 5.01 ist auf der Amiga–CD Meeting–Pearls–IV zu finden, die aber wohl nirgens mehr im Handel zu bekommen sein dürfte. Leider ist durch die vielen seitdem vergangenen Jahre die Verbindung zu M. Steppler abgerissen. Dennoch muß beim Aufgreifen dieses hier darzustellenden Konzeptes auch in Zukunft immer sein Name als "Erfinder" dieses Konzeptes genannt werden.

Leider beendete Steppler 1994/95 seine Arbeiten an diesem Programm bei einem Stande von schätzungsweise 95 % (einige sog. größere und kleinere Kleinigkeiten blieben unerledigt), wohl weil er mit dem Konkurs von Commodore damals für den Amiga keine Zukunft mehr sah (womit er bis heute weitgehend recht behielt) und auch weil er mehr Zeit für seine beginnende Promotion und vielleicht auch für seine Familie(?) benötigte. Auf meine seinerzeitige Frage, ob er irgendwann unter Windows weiterprogrammieren wolle, erhielt ich die Antwort, er wolle sich doch nicht selbst bestrafen!
Der Autor dieses Beitrages erhielt autorisiert 1995 die Gelegebheit, die Dokumentation zu TPP auf den letzten Stand zu bringen und mit auf der o.g. CD unterbringen zu lassen.

Noch einmal begegnete ich Dr. Steppler: Gemäß einer ZdF–Meldung im vergangenen Herbst in heute erhielt er als erster in Deutschland einen Greencard–Mitarbeiter. Auch dadurch ergibt sich die " Vermutung", daß seine Programme nicht beliebige Straßenware sind.

TPP versus LyX

LyX zeichnet sich dadurch aus, daß von Anfang an ein Schriftbild entsteht, das dem späteren Ergebnis mehr oder weniger nahe kommt. Andererseits zwingt es den Schreiber dabei zu einem diesem Endergebnis nahen Schreibstil: Es gibt z.B. nur einen einfachen Zwischenraum, auch wenn die SpaceTaste mehrfach gedrückt wird; die Zeilen müssen in der Regel voll ausgeschrieben werden, die in LyX ähnlichen Schlüsselwörter verschwinden beim Schreiben des Textes und werden durch deren Entsprechungen ersetzt...

Mit TPP kann man, so wie es LaTeX gestattet, "völlig verquer" schreiben, wodurch im Moment des Schreibens durch den dadurch entstehenden Flattersatz, durch die dadurch meist verkürzten Zeilen und z.B. auch vergrößerten Absatzabstände die Übersichtlichkeit im unmittelbaren Schreibprozeß erhöht wird. Man kann vor das Schlüsselwort, das die Dokumentenklasse angibt (Brief, Artikel, Buch...) beliebige Kommentare und auch zum späteren Hineinkopieren bereitzustellende Schlüsselwörter (natürlich nicht die für die Dokumentenklasse!) und auch orginale LaTeX–Befehle (und auch –Sequenzen) ablegen. Bei der Übersetzung des Geschriebenen nach *.tex wird das Schlüsselwort für die Dokumentenklasse gesucht und erst dort beginnt die Umsetzung! Im Ausgangstext werden allein die Schlüsselwörter (z.B. in weiß auf augenschonendem grau oder auch grün(?)) hervorhebend geschrieben.

Das Programmpaket aus TPP und PasTEX arbeitet sehr schnell, da es durchweg in sauberem C geschrieben wurde. Mit meinem 25 MHz AMIGA mit nur 13 MB RAM bin ich mindestens ebenso schnell wie unter Linux auf einem 500 MHz PC und 128 MB RAM. Dieses vorliegende 19seitige Dokument braucht auf diesem AMIGA 28 Sekunden vom Befehl VORSCHAU bis zur dvi– Darstellung!

Das Frontend ist eine Mischung aus einem selbständigen Editor (z.B. mit bis zu 512 Zeichen erweiterbaren Zeilenlängen und einer optionalen Zeilennummerierung beim Druck) und einem Textprogramm (aus dem Jahre ca. 1990), das sich als besonders schreibfreundlich darstellt. Aber dazu später.

Das Konzept des Frontend

Der Programmautor bietet LaTeX–üblich Dateien für die verschiedenen Dokumentklassen an wie BRIEF, ARTIKEL, BUCH usw. Läd man die gewünschte Datei, so stellt sie sich in Form eines Formulares mit einer Reihe sofort verständlicher Schlüsselwörter dar. Der Autor nennt es Formblatt.

Hinter jedem Schlüsselwort steht (in einer Parse–Datei versteckt) eine LaTeX–Befehlssequenz, die später an dessen Stelle tritt. Dateien mit den einzelnen Schlüsselwörtern für jede Dokumentenklasse sind im ASCII–Format im Parse–Verzeichnis abgelegt. Diese Dateien bestehen aus einem unveränderlichen Teil und bieten damit die Möglichkeit, entsprechend der vorab eingestellten Sprache (englisch oder deutsch) das Formblatt in jeweils der einen dieser Sprachen zu benutzen. Das heißt, daß TPP die deutschen Schlüsselwörter zunächst ins englische übersetzt, ehe es in der o.g. *.parse–Datei die Übersetzung in die LaTeX–Sequenz sucht. Dieser unveränderliche Teil läßt sich durch eigene Schlüsselwörter beliebig(!) erweitern, hier aber nur in einer Sprache. Die Form der zusätzlich wählbaren Schlüsselwörter ist völlig frei.

Hier ein ParseDateiAbschnitt unter der Dokumentenklasse BRIEF:

```
% Brief
% <insert> Hoehe des verwendeten Zeichensatzes (<11pt,> <12pt,>)
% begin of letter
% <insert> height of character set to be used (<11pt,> <12pt,>)
### Letter
\documentstyle[<insert>german]{tppdinbrief}
\begin{document}
###

% Geschaeftsbrief
% <insert> Hoehe des verwendeten Zeichensatzes (<11pt,> <12pt,>)
% Businessletter
% <insert> height of character set to be used (<11pt,> <12pt,>)
### Businessletter
\documentstyle[<insert>german]{tppgeschaeftsbrief}
\begin{document}
###

% <insert> Unterschrift
% <insert> signature
### Signature
\signature{<insert>}
###

(documentstyle entspricht noch dem Stand von ca. 1990 mit \LaTeX 2.09)
```

In Brief folgen zwei Kommentarzeilen in deutsch, zwei in englisch. Relevant für das Parsen ist der Ausdruck ### Letter (mit dem englischen Schüsselwort), dem die zugehörigen Befehle folgen. Die Befehlssequenz wird abgeschlossen mit ###.   insert ermöglicht die an dieser Stelle notwendige Auswahl der Eingabewerte.

Die beiden anderen Schlüsselwörter sind selbsterklärend.

Während bei anderen Schlüsselwörtern für insert der Wert im Textverlauf einzutragen ist, werden die Daten für die Schlüsselwörter für die Dokumentklassen beim Start der Übersetzung/Compilierung in einem Auswahlfenster neben anderen konfigurierenden Parametern abgefragt bzw. bestätigt.

In gleicher Weise sind alle anderen Schlüsselwort–Entsprechungen aufgebaut. Am Ende dieser zweisprachigen Liste steht eine darauf hinweisende Bemerkung. Danach kann der User beliebig viele eigene einsprachige(!) Schlüsselwort–Entsprechungen nach dem gleichen Muster generieren. Sie stehen sofort zur Verfügung.

Durch Variation der Schlüsselwörter kann sich jeder neue Formblätter erstellen und immer zur Verfügung haben. Vom Autor sind allein unter Brief 36 zweisprachige Schlüsselwörter vorgegeben (u.a. für Textzentrierungen, für die Briefkopf–, Listen– und Tabellengestaltungen usw.) Für meine speziellen Anwendungen habe ich diese Datei um weitere 30 einsprachige erweitert. Alle diese Schlüsselwörter lassen sich aus einem menügesteuertem Fenster auswählen oder direkt selbst in weiß schreiben.

Es gibt noch eine zweite Parse–Datei, in der alle mit der Tastatur möglichen Sonderzeichen aufgelistet und so übersetzbar sind, daß diese orginal im *.dvi–Ausdruck erhalten bleiben. Das hat aber zur Folge, daß insbesondere der Backslash nicht für LaTeX–Kommandos zur Verfügung steht. Steppler zog sich damit aus der Klemme, in dem er alle orginalen LaTeX–Befehle, für die es keine Schlüsselwörter gibt, mit den beiden Schlüsselwörtern TeX und Ende einrahmen läßt. Das entspricht äußerlich der farblichen Darstellung der LaTeX–Befehle in LyX.

Dieses Einrahmen ist aber, falls es häufig notwendig ist, sehr unbequem und führt ggf. auch zur Verschlechterung der Übersichtlichkeit des Dokumentes. Deshalb wurden der Backslash und weitere relevante Zeichen wie % und $ aus der Parseliste der Sonderzeichen eliminiert. Auf diese Weise können in den Text an beliebiger/zulässiger Stelle die orginalen LaTeX–Befehle eingegeben werden. Andereseits müßten dann aber diese eliminierten Zeichen (\,%,$ usw.) mit besonderen LaTeX–Konstruktionen dargestellt werden. Aber: Steppler stellte über das o.g. Auswahlfenster, das beim Start der Übersetzung/Compilierung erscheint, zwei alternative Sonderzeichentabellen zur Wahl, so daß man sich für diese Fälle eine geeignete Ausweichmöglichkeit schaffen kann.

Hier ein bearbeiteter Ausschnitt aus einer der beiden Sonderzeichentabellen:

```
% ASCII 35-38
%   #   {\#}          % gesperrt am 13.03.01
%   $   {\$}          % gesperrt am 13.03.01
%   %   {\%}          % gesperrt am 13.03.01
    &   {\&}

% ASCII 43, 60-62
    +   $+$
%   <   $<$           % gesperrt am 13.03.01
    =   $=$
%   >   $>$           % gesperrt am 13.03.01

% ASCII 92, 94-95
%   \   $\backslash$ % gesperrt am 17.03.97
    ^   \^{}          % gesperrt am 05.01.98, aufgehoben 05.03.01
```

```
    _   \_{}         % gesperrt am 05.01.98, aufgehoben 05.03.01

% ASCII 123-126
%   {   $\{${}       % gesperrt am 17.03.97
    |   $\vert$
%   }   $\}${}       % gesperrt am 17.03.97
    ~   \~{}
```

Das Formblatt Brief

Jeder selbst zu schreibende Brief besitzt im Briefkopf konstante Informationen, die vorab als normale Texte in das (nunmehr eigene) Formblatt aufgenommen werden können/sollten. Das orginale Formblatt unterscheidet sich nur dadurch, daß (für den Anfänger) statt der konkreten Eintragungen z.B. nur "Absendername, Vorname, Anschrift, hier steht der Brieftext" usw. steht.

Wollte ich jetzt eines der Formblätter hier auf dem Monitor des Lesers abbilden lassen, bekäme ich Probleme: Weiße Schrift auf weißem Untergrund ist nicht darstellbar. Damit kommen wir zu einem Nachteil der Schlüsselwortdarstellung in TPP:

Um die Schlüsselwörter farbig (weiß) darstellen zu können, muß im ANSI– (oder einem gleichwertigen) –Format geschrieben werden. Um andersfarbig schreiben zu können, werden im Textfile unsichtbare Steuercodes abgelegt, nämlich vor dem Schlüsselwort: umschalten nach Farbe weiss und danach zurückschalten nach Farbe schwarz. Diese unsichtbaren ANSI–Codes nehmen aber in der Datei Speicherplatz für mehrere Zeichen ein. Wenn man nun (insbesondere in komplizierteren Konstruktionen wie z.B. Listen) Änderungen von Schlüsselwörtern vornimmt, dann kann es vorkommen, daß schon bei der Überprüfung des Urtextes auf Integrität TPP eine Fehlermeldung mit Sprung zur Fehlerstelle meldet. Dann ist nämlich ein Teil eines Steuercodes nicht gelöscht worden und man muß nun versuchen, durch eine, besonders am Zeilenanfang " intelligente", Bearbeitung diesen Fehler zu beseitigen.

Eine andere ähnlich auffällige Kennzeichnung aber ohne versteckte Steuercodes wäre daher diskutabel. Diese bietet sich an, wenn in normalem Schwarz geschriebene Schlüsselwörter mit z.B. ||| eingerahmt würden, die so wohl kaum in einem Text vorkommen würden. Diese Darstellungsform soll hier angewandt werden. (Dadurch gelingt es hier außerdem, ohne zu linkende Zusatzdateien auszukommen.)

Es folgt der Abdruck meines entsprechend veränderten Formblattes Brief, siehe unten. Da ein AutoSave (mit beliebig einstellbaren Zeiten) vorhanden ist, steht vor ||| Brief ||| die für mich wichtige Erinnerung, sofort das noch unausgefüllte Formblatt an die vorgesehene Stelle abzuspeichern, sonst erhält eben das Formblatt den Brieftext aufgedrückt. — Außerdem ist die oft benutzte Befehlssequenz für bitte wenden zum sofortigen Kopieren bereitgelegt.

Der Briefkopf unterliegt der freien Gestaltungsmöglichkeit. TPP liefert aber eine annehmbare Form mit: Sengerling[1990] => dinbrief, variiert von Steppler[1991] zu tppdinbrief. Letztere verwende ich mit kleinen Änderungen. Dieser Text, den Sie gerade lesen, entspricht diesem tppdinbrief–Format. (Über eMail–Anforderung schicke ich gern diese Datei zu.)

Erläuterungen zu Brief

Die zur Ligaturentrennung verwendeten "trichterförmigen" Doppelstriche eignen sich auch recht gut zur kleinschrittigen Abstandsgestaltung und somit zur Kompensierung von verbliebenen Schönheitsfehlern im Programm. (Es muß an entprechender Stelle im tppdinbrief.sty die picture–Gestaltung verändert werden) Die Leerzeile nach dem Absendernamen (1.Zeile) bewirkt, daß der Anschriftentext insgesamt unter den Strich unter dem Absendernamen gelangt. Sonst würde (in diesem Beispiel) die eMail–Anschrift über dem Strich zu stehen kommen (s. Seite 1 oben).

wasyfont generiert das Telefon–Bildchen. Die vielen # erinnern mich, die Eintragung des Datums mit Ctrl+D nicht zu vergessen; ||| Heute ||| dagegen zeigt an Folgetagen immer wieder (oft unerwünscht) das aktuelle Datum an.

Alles andere dürfte verständlich sein. Ein Abdruck des daraus generierten tex–Files mit ein wenig Text ist anschließend zu sehen.

```
                          #######################       \hspace{3.4cm}
                          # Sofort abspeichern! #
                          #######################

\vspace*{\fill}  \hfill b.w.

||| Brief |||

||| Absender |||
C h r i s t i a n \/ \/  S c h a r r n b e c k

Helmut--Kopka--Str. 21
0 2 8 2 0 \/ \/ \TeX as \/
\input wasyfont \phone   \/ 03 3 33 -- 55 55 55   \/ \/
Chr.Scharrnbeck@t--online.de  \/ \/  \/

||| Retouradresse |||
 Christian Scharrnbeck
  Helmut--Kopka--Str. 21
  02820 \TeX as

||| Datum |||
\TeX as, den  #################

||| Adresse |||
{\bf
               }

||| Betreff |||
{\bf                        }

||| Anrede |||
{\bf Hallo          ,}

||| Beginn  |||
\setlength{\baselineskip}{2.5ex  plus0.5ex  minus0.35ex}
% Dient der weitgehenden Vermeidung von Schusterjungen und Hurenkindern.
\pagestyle{empty}    % ab 3 Seiten entfernen!!!!

Hier steht der Brieftext.

||| Ende |||
||| Schluss |||
Mit freundlichen Gr\"{u}{\ss}en

||| Anlagen |||
Anlage:

||| Verteiler |||
Verteiler:

||| PS |||
\underline{P.S.:}
```

Hier steht das um einen Text erweiterte tex–File:

```
\documentstyle[11pt,german]{tppdinbrief}
\begin{document}

\def\UserName{C h r i s t i a n \/ \/  S c h a r r n b e c k}
\def\UserAddressLineOne{}
\def\UserAddressLineTwo{Helmut--Kopka--Str. 21}
\def\UserAddressLineThree{0 2 8 2 0 \/ \/ \TeX as \/}
\def\UserAddressLineFour{\input wasyfont \phone\/05 5 55 -- 55 55 55 \/\/}
\def\UserAddressLineFive{Chr.Scharrnbeck@t--online.de  \/ \/  \/}

\Retouradresse{ Christian Scharrnbeck\\
Helmut--Kopka--Str. 21\\
  02820 \TeX as}

\Fenster
\Datum{\TeX as, dem 15.03.01}

\begin{letter} {{\bf An die\\
Programmierer\\
die an einem Frontend\\
f\"{u}r \LaTeX interessiert sind.  }\\
~ }

\Betreff{{\bf  Das SteppMarkieren in TPP}}

\opening{\vspace{-1.5cm} {\bf Hallo \LaTeX--Freunde,}}
\setlength{\baselineskip}{2.5ex  plus0.5ex  minus0.35ex}
% Dient der weitgehenden Vermeidung von Schusterjungen und Hurenkindern.
\pagestyle{empty} % ab 3 Seiten entfernen!!!!

In einer Reihe von TextProgrammen wird eine (ungewollte) TextMarkierung
schon durch ein "H\"{a}ndeZittern"\ oder ein Klicken an einer
"diesbez\"{u}glich empfindlichen Stelle"\ ausgel\"{o}st. Nicht so in
TPP. Das sog. SteppMarkieren geschieht dadurch, da{\ss} mit gedr\"{u}ckter linker
MausTaste in MarkierungsRichtung gefahren wird. Erst nach einem "
Stepp"\ von sechs Zeichen springt die Markierung an. Jetzt kann man,
wenn man das (selten!) will, zur\"{u}ckgehen bis auf ein Zeichen oder gar in
negative Richtung. In der Regel f\"{u}hrt man aber seine Maus beim Markieren
in SchreibRichtung.

Es gibt drei durch Buttons einstellbare Markierungsarten:
\begin{deflist}{  Zeilex}

\item[    Wort]
 \''{U}ber einen beliebigen Textbereich kann zeichengenau markiert
werden.
\item[    Zeile]
 In vertikaler Richtung betr\"{a}gt hier der SteppSchritt vier
Zeilen. Es k\"{o}nnen beliebig viele Zeilen markiert werden. So markierte
Zeilen k\"{o}nnen dann auch formatiert werden: z.B. Flattersatz, Blocksatz,
Einr\"{u}ckungen.
\item[    Block]
 Es ist ein beliebig gro{\ss}es Rechteck markierbar, z.B. f\"{u}r das
rationelle Einf\"{u}gen von senkrecht untereinanderstehenden Trennzeichen f\"{u}r
eine Tabelle.
```

```
\end{deflist}

\Anlagen{\underline{Anlage:}}

\Verteiler{\underline{Verteiler:}}

\closing{Good Bytes\\
~ }

\ps{{\footnotesize \underline{P.S.:}
Es ist ein g\"{u}nstiger Zufall, da{\ss} dieses Markieren von \sc Steppler \rm
stammt. {\sc SteppMarkieren} m\"{o}chte ich heute und hier daf\"{u}r vorschlagen.
Vielleicht ist aber auch anderswo schon jemand auf diesen naheliegenden
Begriff gekommen.}
}
\end{letter}
\end{document}
```

Listengestaltung

Wie oben schon ausgeführt, können die Schlüsselwörter aus einem menüaufgerufenem Fenster in den Text kopiert oder auch selbst nach Umschaltung in die Schlüsselwort–Farbe selbst geschrieben werden. Auch ein Kopieren von beliebiger Stelle ist möglich.

Zur Konstruktion z.B. von Listen wird das Schlüsselwort LISTE angeboten. Beim Hereinkopieren über das Menüfenster wird an den Anfang der Zeile mit dem Cursor ausgegeben (Der Folgetext wird entsprechend verschoben.):

		Liste		
		Punkt		
		Eingabe		
		LEnde		

Die beiden Worte/Zeilen PUNKT und EINGABE kopiert man dann in der benötigten Anzahl. Im Kopiermodus BLOCK kann man diese beiden Schlüsselwörter auch aus einer zu erweiternden Liste heraus leicht weiterkopieren. Hinter ||| Liste ||| lassen sich so viele, in der dvi–Darstellung nicht erscheinende, Zeichen einsetzen, wie der EINGABE–Text eingerückt werden soll. — Zusätzlich gibt es ||| NListe ||| (ohne ||| Punkt |||), damit lassen sich numerische Listen zusammenstellen.

Wir bauen eine Tabelle

Für die Einbindung von Tabellen und Grafiken gibt es eigene " Schlüsselwort–Fenster". In ihnen kann man z.B. für Tabellen alle struktuellen Eingaben machen, wie z.B. Breite der Tabelle, welche Umrandungs- bzw. Zwischenlinien werden gewünscht, die Eingabe der Spaltenüberschriften und die Ausrichtung der Spaltentexte. Dazu kommen optional noch ein Tabellentitel und ein Untertitel. Alle diese Eingaben erscheinen dann als Schlüsselwörter zusammen mit den eingegebenen Texten. Man kann aber auch alle oder nur einige Schlüsselwörter und zugehörige Textergänzungen unter ||| Tabelle ||| selbst schreiben. Diese erscheinen dann beim Aufruf wiederum im o.g. Fenster. Alle Änderungen ergänzen sich somit wechselseitig. Die jeweilige Spaltenbreite generiert TPP selbst!

Es liegt in der Natur der Sache, daß bei dieser Tabellenkonstruktion Kompromisse eingegangen werden müssen. Aber bei der Gestaltung verschiedenster Tabellen kam TPP meinen Wünschen immer relativ nahe und ich brauchte keine reine LaTeX–Konstruktion.

Unverzichtbar halte ich dabei den in PasTeX vorhanden Maßstab (cm, Punkte, Inch) mit dem hier eine zu breite oder zu schmale Tabelle schnell korrigiert werden kann. Ein Beispiel:

Meine Rechner — Lang ist es her!					
Nr.	**Typ**	**Name**	**Hersteller**	**Speicher**	**Jahr**
1.	Kleinrechner	???	Sömmerda	???	ca. 1965
2.	DEC PDP 7	KSR 4100	Zeiß, Jena	4 kB	ca. 1971/4
2a.	DEC PDP 7	KSR 4100	Zeiß, Jena	8/16 kB	ca. 1975
3.	DEC PDP 11	???	Zeiß, Jena	16 kB ?	ca. 1985
4.	DEC PDP 11	???	Philips	32 kB ?	ca. 1985
5.	Amiga	A2000	Commodore	1 MB/7MHz	1991
6.	dto. Amiga	Erweitg.	Commodore	13MB/25Mhz	1996
6a	dto. Amiga	defekt			1999
	Wechsele	**nie den**	**Monitor**	**unter**	**Strom!!**
7.	PentiumIII	Presario	Compaq	124 MB/500MHz	1999
				Win/Amiga/Lin	
heute	dto. Amiga	defekt,	für 3.LT.	13 MB sind	reichlich

Nr. 2 bis 4 liefen bis zur Wende

Offensichtlich ist die vorgegebene Breite von 15 cm zu schmal gewesen. Mit dem Meßsystem wird eine fehlende "Strichlänge" von 1,8 cm ermittelt. Die Tabellenbreite wird korrigiert, und jetzt erscheint sie auch schön zentriert:

Meine Rechner — Lang ist es her!					
Nr.	**Typ**	**Name**	**Hersteller**	**Speicher**	**Jahr**
1.	Kleinrechner	???	Sömmerda	???	ca. 1965
2.	DEC PDP 7	KSR 4100	Zeiß, Jena	4 kB	ca. 1971/4
2a.	DEC PDP 7	KSR 4100	Zeiß, Jena	8/16 kB	ca. 1975
3.	DEC PDP 11	???	Zeiß, Jena	16 kB ?	ca. 1985
4.	DEC PDP 11	???	Philips	32 kB ?	ca. 1985
5.	Amiga	A2000	Commodore	1 MB/7MHz	1991
6.	dto. Amiga	Erweitg.	Commodore	13MB/25Mhz	1996
6a	dto. Amiga	defekt			1999
	Wechsele	**nie**	**St**ecker	**unter**	**Strom!!**
7.	PentiumIII	Presario	Compaq	124 MB/500MHz	1999
				Win/Amiga/Lin	
heute	dto. Amiga	defekt,	für 3.LT.	13 MB sind	reichlich

Nr. 2 bis 4 liefen bis zur Wende

Der Urtext zur letzten Tabelle sieht so aus (Zur Erhöhung der Übersichtlichkeit wurden Leerzeilen eingefügt.):

```
||| Tabelle |||

||| Titel   |||
```

```
\large \bf Meine Rechner --- Lang ist es her!

|||  Untertitel |||
\footnotesize Nr. 2 bis 4 liefen bis zur Wende

||| Breite |||
16.800cm

||| Vertikal |||
||| NurNamen |||
||| Rahmen   |||
||| Spalten  |||

||| TabZentriert |||  % ab hier folgen die Spalten\"{u}berschriften
\bf Nr.
||| TabLinks |||
\bf Typ
||| TabLinks |||
\bf Name
||| TabLinks |||
\bf Hersteller
||| TabZentriert |||
\bf Speicher
||| TabLinks |||
\bf Jahr

||| Eintr\"{a}ge  |||
1. \ Kleinrechner\ ???     \  S\"{o}mmerda  \???           \  ca. 1965
2. \ DEC PDP 7   \ KSR 4100\  Zei{\ss}, Jena\4 kB          \  ca. 1971/4
2a.\ DEC PDP 7   \ KSR 4100\  Zei{\ss}, Jena\8/16 kB       \  ca. 1975
3. \ DEC PDP 11  \ ???     \  Zei{\ss}, Jena\16 kB ?       \  ca. 1985
4. \ DEC PDP 11  \ ???     \  Philips   \32 kB ?       \  ca. 1985
5. \ Amiga       \ A2000   \  Commodore \1 MB/7MHz     \  1991
6. \ dto. Amiga  \ Erweitg.\  Commodore \13MB/25Mhz    \  1996
6a \ dto. Amiga  \ defekt  \            \              \  1999
   \{\bf Wechsele}\{\bf nie}\{\bf S}tecker\{\bf unter}\{\bf Strom!!}
7. \ PentiumIII  \ Presario\  Compaq    \124 MB/500MHz \  1999
   \             \         \            \Win/Amiga/Lin \
heute\ dto. Amiga\ defekt, \  f\"{u}r 3.LT. \13 MB sind    \  reichlich

||| Ende |||          % Ende der Tabelle
```

Zu den einzugehenden Kompromissen zählt, daß z.B. Fettdruck in den Spaltentexten nur durch im Editor erzeugten Fettdruck (ANSI) erreicht werden kann, da ja, wie zu sehen, die Spalten mit Backslashs getrennt werden. Aber VERBATIM bringt das nicht geschriebene \bf wieder hervor!

Grafiken einbinden

In ähnlicher Weise können fremde (auch farbige) Grafiken gestaltet und eingebunden werden. In der Dokumentation zu TPP ist eine Fraktal–Grafik in zwei verschiedengroßen Maßstäben eingearbeitet.

Andere Dokumentenklassen

Im Analogieschluß kann man sich die Gestaltung der anderen Klassen mit Hilfe von Schlüsselwörtern gut vorstellen. Beispielsweise ist die Dokumentation zu TPP mit BUCH geschrieben worden: Umfang 80 Seiten, Titelblatt, Inhalts– und Stichwortverzeichnis, Fußnoten, alles in der von LaTeX gewohnten druckfertigen Art und Weise.

Freie Formblatt–Gestaltung

Als Beispiel für eine freie Gestaltung von Formblättern soll das Ausfüllen von Banküberweisungen herangezogen werden.

Hier das Formblatt mit verbatim:

```
     {\bf \underline{\"{U} b e r w e i s u n g   d r u c k e n}}

     {\bf \underline{L\"{o}cher auf Durchschlag nicht zukleben!}}


||| Brief |||
||| Absender |||
||| Adresse |||
||| Beginn |||
\sloppy
\thispagestyle{empty}


\vspace*{-19mm}  % Anpassung zwischen Formular und Drucker

                    % von |                     bis |
 ||| Empf\"{a}nger: |||       Linux--Installateure         ||| Ende |||

 ||| Empf-KontoNr.: |||   111 111 11                   ||| Ende |||

 ||| Empf-BLZ:  |||       870 560 00                   ||| Ende |||

                    % von |                     bis |
 ||| an Bank:  |||        SuSE--Freebank               ||| Ende |||

 ||| DM/EUR:  |||         DEM                          ||| Ende |||

 ||| Betrag:  |||       129.00                         ||| Ende |||

                    % von |                     bis |
 ||| VerwendgZeile1: |||  BelegNr:SuSE 7.1,Nr.: 12345  ||| Ende |||

                    % von |                     bis |
 ||| VerwendgZeile2: |||  vom 10.10.00                 ||| Ende |||


 ||| Konto-Inhaber: |||   Bit Byter                    ||| Ende |||

 ||| Konto-Nummer: |||    8888888888                   ||| Ende |||

 ||| Ende |||
```

Die zugehörigen Schlüsselwörter sind wie folgt formuliert:

```
% Es folgt \"{U}berweissung ###########

% Empfaenger:
### Empf\"{a}nger:
\mbox{\hspace*{-2cm}<insert>}
###

% Empf-KontoNr.:
### Empf-KontoNr.:
\vspace{2mm}
\mbox{\hspace*{-2cm}\hspace*{-3pt}<insert>}
###

% Empf-BLZ
### Empf-BLZ:
\vspace*{-7mm}
\mbox{\hspace*{75mm}<insert>}
###

% an Bank:
### an Bank:
\vspace*{1.5mm}
\mbox{\hspace*{-2cm}\hspace*{-3pt}<insert>}
###

% DM/EUR
### DM/EUR:
\vspace*{2mm}
\mbox{\hspace*{3.6cm}<insert>}
###

% Betrag
### Betrag:
\vspace*{-7mm}
\mbox{\hspace*{5.5cm}<insert>}
###

% VerwendgZeile1:
### VerwendgZeile1:
\vspace*{1mm}
\mbox{\hspace*{-2cm}<insert>}
###

% VerwendgZeile2:
### VerwendgZeile2:
\vspace*{2.5mm}
\mbox{\hspace*{-2cm}\hspace*{-3pt}<insert>}

###
% KontoInhaber:
### Konto-Inhaber:
\vspace*{2.5mm}
\mbox{\hspace*{-2cm}\hspace*{-3pt}<insert>}
###

% KontoNummen:
### Konto-Nummer:
\vspace*{1.0mm}
\mbox{\hspace*{-2cm}\hspace*{-3pt}<insert>}
###
```

Brief als "Universalvorlage"

Wie oben zu sehen, wurde das Überweisungsformular unter Brief ausgefüllt. Unter TPP ist es möglich, mit den für Brief zwingend notwendigen Schlüsselwörtern

		Brief		
		Absender		
		Adresse		
		Beginn		

Beliebige Textgestaltung

||| Ende |||

andere Anwendungsgebiete zu erschließen. Da aus den im Abschnitt Autor genannten Gründen die Klasse Artikel damals nicht mehr ausreichend feriggestellt wurde, kann man z.B. auch für das Schreiben von Artikeln auf Brief ausweichen.

Bequeme Auswahl der LaTeX–Varianten

Mit dem Menübefehl Vorschau öffnet sich ein Fenster, in dem bestimmte Eingaben für die Compilierung in das LaTeX–Format vorgenommen oder bestätigt werden können:

1. Einstellungsmöglichkeit für HOffset und VOffset in den Maßeinheiten cm, inch, Picas, Punkt und Didot
2. Auswahl der normalsize–Schriftgröße: 10, 11 und 12 pt
3. Soll während der Compilierung ein Log–Fendster geöffnet werden?
4. Soll dieses Log–Fenster sofort wieder geschlossen werden? Durch die Eingabe eines Zeichens kann der Vorgang angehalten und mit Backspace wieder fortgeführt werden.
5. Soll die sog. Datumsfunktion angewendet werden: Beim Start von Vorschau werden die DatumEintragungen der *.tpp–, *.tex– und *.dvi–Dateien miteinander verglichen. Die Bearbeitung beginnt immer beim File mit dem jüngsten Datum, bzw. wenn dieses die *.dvi–Datei trägt, wird sofort letztere angezeigt. Dadurch kann bei umfangreicheren Texten ggf. viel Zeit eingespart werden. Dieser Ihnen hier vorliegende gesamte Text (19 Seiten) wird am oben beschriebenen Amiga in 28 Sekunden neu von A bis Z bearbeitet und angezeigt, aber er erscheint ohne Bearbeitung schon nach 5 Sekunden am Monitor.
6. Man kann wählen, ob man einen Text nur nach .tex übersetzt haben will, oder ob man eine vorliegende *.tex–Datei nach *.dvi compilieren möchte oder eben im Normalfall beides nacheinander ablaufen soll.
7. Es gibt eine Auswahlmöglichkeit für verschiedene Drucker... und ob sie in Draft drucken sollen.
8. Die Compilierung kann optimiert erfolgen.
9. Man kann für die Log–Files zu Showdvi und Printdvi Adressen eingeben.
10. Man kann die Grafikbearbeitung durch die Wahl geeigneter Dateien steuern.
11. Ähnliches gilt für die Index–Gestaltung.
12. Für jede der 5 verschiedenen Dokumentenklassen kann man vorab jeweils zwei (beliebige) Pfade (Dateien) für die Compilierung vorab eingeben, z.B. für die Bearbeitung in deutsch oder englisch oder die Auswahl eines "Big–Systems" und später über Buttons auswählen. (In einer neuen Umsetzung z.B. nach LINUX könnten damit auch verschiedene LaTeX–Varianten nur durch Mausklick miteinander verglichen werden.)
13. Es stehen zwei über Buttons auswählbare Dateien zur Sonderzeichen–Bearbeitung zur Verfügung.

14. **Für den Ausdruck** stehen zusätzlich zur Verfügung:
    (a) Auswahl der auszudruckenden Seiten von ... bis. Als Einstellung für ALLES AUSDRUCKEN, was die Regel ist, kann z.B. gewählt werden: Von –1000 bis 1000 oder noch größere Zahlen.
    (b) Steht ein PostscriptDrucker zur Verfügung?
    (c) Wieviele Kopien sollen erstellt werden?
    (d) Soll von der ersten zur letzten Seite oder umgekehrt gedruckt werden?
15. Alle diese Einstellungen lassen sich zusätzlich in die eben benutzte Config.Datei nur durch einen Menübefehl dauerhafzt abspeichern.

Fehlende Tastaturzeichen

Mein Amiga ist seit einiger Zeit defekt. Ich mußte deshalb mittels einer Emulation (UAE) auf einen PC umsteigen. Verschiedene Sonderzeichen befinden sich nun an anderen Stellen als auf der PC–Tastatur angegeben. Und einige Zeichen fehlen: Die beiden für LaTeX so wichtigen geschweiften Klammern und das "Dach" für die Exponentendarstellung.

Da die Schlüsselwörter nur zur Übersetzung nach *.tex eingesetzt werden und diese fehlenden Zeichen orginal auch erst dort in der richtigen Darstellung benötigt werden, lassen sich entsprechende Schlüsselwörter für diese fehlenden Zeichen generieren:

```
% Quadrat
### 2
$ ^2 $
###

% Kubus
### 3
$ ^3 $
###

% Klammerauf
### Ka
{
###

% Klammerzu
### Kz
}
###
```

Eines fehlt noch:

Der TPP–Editor soll sehr schreibfreundlich sein

Die wichtigsten (noch nicht erwähnten) TPP–Features werden im folgenden dargestellt. Weiterhin sind auch einige Features, die das gut gelungene PasTeX darüberhinaus zur hohen Leistungsfähigkeit diese Paketes beiträgt, beschrieben:

<u>TPP</u>

SteppMarkieren: Nicht schon durch Händezittern, sondern "abgeblockt" allein mit der linken Maustaste: Erst ein vorausgehender STEPP(LER)schritt mit der linken Maustaste von 6 Zeichen läßt den Mauszeiger zum Markierer für ein bis beliebig viele Zeichen bzw. Zeilen werden.

Markieren:
Wort = von 1 bis beliebig viele Buchstaben, Zeilen, Abschnitte
Zeile = beliebig viele Zeilen
mit *Block* = beliebig großes Rechteck, bes. wichtig bei Tabellen– u.ä. Arbeiten
gesteuert durch einen "dreistufigen" Button

Kopieren: Jedes markierte Feld kann mit Doppelklick (mehrfach) kopiert werden. Danach ist die Löschung der Markierung durch einen Tasten– oder Menübefehl nötig.

Löschen: Die DEL– / ENTF–Taste löscht nur auf der Zeile mit dem Cursor: kein Zeilensprung!

Cursor: Auf einer (teilweise) leeren (noch unbeschriebenen) Zeile kann der Cursor bis zum Zeilenende bewegt werden.

Scrollbar– (Scrollbalken–)Länge ist immer proportional zum Gesamtumfang! Das gilt für das gesamte Betriebssystem AMIGAOS

Blöcke und ganze Textfiles aus beliebigen Dateien können an jeder Stelle eingefügt werden.

Augenschonung: Die Hintergrundfarbe des TPP–Screens läßt sich variieren und auf z.B. angenehmes Grau einstellen. Auch in PasTEX möglich.

Configs: Alle konfigurierenden Einstellungen (Farben, Schriften, LATEX–Pfade, Auswahlfenster, Sreenverhalten, Zeilenlänge, AutoSave usw.) lassen sich in TPP und ähnlich auch in PasTEX " ohne Textarbeit" über das Menü abspeichern bzw. laden. Man kann beliebig viele Config–Varianten abspeichern. Über entsprechende Startscripte läßt sich jede Variante von TPP ohne Umwege starten. Meine Configs sind:

1. *ASCII* u.a. für Programmtexte, Configs usw.
2. *ANSI* u.a. für Direktausdrucke mit den Zeichensätzen des Druckers
3. *TEX* für LATEX
4. Heute: Config. für den 3. LinuxTag mit voreingestelltem Dateiauswahlfenster, das nur die heute benötigten Files im *.tpp–Format anzeigt.

Fenster: TPP stellt bis zu 10 Textfenster übereinander (und gestaffelt) zur Verfügung. Sie lassen sich einzeln ikonifizieren und so während der Nutzung gut verwalten.

Autosave: TPP besitzt ein Autosave–System mit für jedes der 10 Textfenster beliebig einstellbaren Zeiten.

Tastatur: Es lassen sich übers Menü für alle denkbaren Tastenkombinationen spezielle zusätzlich programmierbare Funktionen vereinbaren und ggf. zu Macros zusammenfassen. Für die Textdarstellung auf dem Screen lassen sich viele Fonts auswählen, insbesondere um auf dem sichtbaren Screen mit unterschiedlich langen Zeilen arbeiten zu können.

Tabulator: TPP besitzt oben eine schmale vertikale Leiste mit je einem Feld für jedes Zeichen einer Zeile. Dort hinein kann man mit der linken Maustaste Tabulatorstopps in beliebigen Abständen setzen und so auch wieder löschen. Z.B. für eine übersichtliche Tabellenkonstruktion oder das Schreiben von Programmtexten ist dies sehr vorteilhaft. Alle TabStopps lassen sich mit einem einzigen ARexx–Befehl (AmigaScriptSprache) löschen.

Drucken: Für den Ausdruck z.B. von Programmscripten innerhalb des Editors kann eine Zeilennumerierung eingeschaltet werden.

Undo: Eine einstufige UndoFunktion ist vorhanden.

Silbentrennung: TPP besitzt eine für LATEX abschaltbare Silbentrennung.

Suchen: Such– und Ersetzfunktionen (vor und zurück) sind vorhanden.

Backspace: Beim Rücktransport von Textteilen auf die vorhergehende Zeile bleibt in der Regel ein Zwischenraum. Bei automatisch(!) getrennten Wörtern verschwindet der Teilungsstrich.

Einrücken: Wenn unter TPP eine Zeile eingerückt wird, springt der Cursor nach dem nächsten Zeilenwechsel unter den 1. Buchstaben der vorhergehenden Zeile (abschaltbar).

Hervorheben: Die in TPP durch ANSI–Fett– oder Schrägdruck bzw. Unterstreichung hervorgehobenen Textteile werden ohne zusätzliche LaTeX–Befehle nach .dvi transponiert. Natürlich sind alternativ auch die entsprechenden LaTeX–Befehle möglich.

Änderungen: Wird ein Text auf dem Monitor verändert, wird dies deutlich sichtbar im Fenstertitel angezeigt.

Rechtschreibkontrolle: Diese wäre, wäre der AMIGA am Leben geblieben, irgendwann hinzugefügt worden. Ein Konkurrent zu TPP hatte sie schon um 1990!

Dateien lassen sich von TPP aus löschen oder umbenennen. Auch lassen sich von hier aus beliebige neue Verzeichnisse erstellen.

Datenzugriff: In jeder Config.Datei läßt sich das Verzeichniss festlegen, zu dem beim Laden oder Sichern sofort zugegriffen werden soll (s. auch Configs, Punkt 4).

Laden/Speichern 1: Das FileAuswahlFenster des jewiligen TPP–Fensters merkt sich dauerhaft, von wo aus es gestartet wurde.

Laden/Speichern 2: In den Einstellungen zu TPP läßt sich unkompliziert festlegen, welche Dateien mit welchen Dateierweiterungen im Requester zum LADEN bzw. SPEICHERN ALS angezeigt oder nicht angezeigt werden sollen, z.B.: *.log, *.aux, *.bak ... Dadurch kann die Übersichtlichtkeit und Arbeitsgeschwindigigkeit erhöht werden. Im AmigaOS existiert dazu ein umfangreiches Joker–System.

Mauszeiger: Vom AmigaOS aus ist es möglich, den Mauszeiger nach einer vorgegebenen Zeit (z.B. 2 sec) zu verstecken. Bei der Berührung der Maus oder der Tastatur ist er sofort wieder da. Deshalb kann der Mauszeiger gut sichtbar ausgebildet sein und verdeckt dennoch keine Textteile. Auch läßt er sich wie jedes Icon selbst neu zeichnen.

fehlende Zeichen: Bei der Emulation des Amiga auf dem PC durch UAE finden sich die Zeichen für *Dach* = Zeichen für Potenzieren und die sehr wichtigen geschweiften Klammern nicht auf der Tastatur. Aber mit Hilfe von TPP lassen sie sich als Schlüssselwörter generieren und in den *.tex–Text richtig einsetzen.

linuxtauglich? Es scheint durchaus sinnvoll, sich das eine oder andere vom (neuesten) AmigaOS für LINUX abzuschauen. Beachte auch: Viele Amigaprogramme sind schon seit langem **textsensitiv**.

LaTeX

Prinzip: Für LaTeX benutzt TPP ladbare Vorlagen (Schablonen/Formblätter in der Art auszufüllender Fragebogen) mit andersfarbig geschriebenen Schlüsselwörtern, zwischen die der erforderliche Text zu schreiben ist und die während der Übersetzung zum .tex–File für die spätere Struktur des Dokumentes sorgen. Werden vorsorglich z.B. in der Briefschablone die ständig gleichbleibenden Eintragungen (Absender, Retouradresse, Befehle für Fettdruck usw.) vorgenommen, steht immer ein fertiger Briefkopf zur Verfügung.
Wird ein Schlüsselwort gelöscht, verschwindet die damit verbundene Funktion. Schlüsselwörter ohne Texteintrag sind erlaubt.
Die Schlüsselwörter für Brief sind: **Brief, Absender,** Retouradresse, **Adresse,** Datum mit optional Heute (auch im Text), Betreff, Anrede, **Beginn, Ende,** Schlußformel, Anlagen, Verteiler, P.S. Viele andere sind zusätzlich möglich wie z.B. Liste, NListe, Tabelle, Ihr Zeichen, Telefon, Durchwahl usw. usw.
Allein die oben fett gedruckten Schlüsselwörter sind für die Dokumentenklasse BRIEF zwingend. Wird hierbei nur zwischen Beginn und Ende ein Text eingefügt, dann kann man auf die Schnelle eine Notiz bzw. einen formlosen mehrseitigen (auch zweispaltigen) Artikel schreiben. Auch solche Vorlagen halte ich mir bereit.

Brief: Es können automatisch ein individueller (variierbarer) Briefkopf, eine Fensteranschrift und zugehörige Falzmarken erzeugt werden. Es gibt weitere Vorlagen für Geschäftsbriefe, Briefe nach DIN, Serienbriefe. — Im (persönlichen) Brief wird mit dem Schlüsselwort HEUTE der Monatsname voll ausgeschrieben (tppdinbrief.sty), im Geschäftsbrief (tppgeschaeftsbrief.sty) dagegen wird nur die Monatszahl angegeben. Mit der Tastenkombination Ctrl–d wird unter TPP dauerhaft das heutige Datum im Format dd.mm.jj eingegeben. (Beachte bitte die Entstehungszeit dieser Programme, da gab es noch LaTeX2.09!)

Schreibweise: Mit dem Editor kann unter Einhaltung der LaTeX–Regeln völlig " verquer" geschrieben werden (= Schreibfreundlichkeit im Urtext, kein Formzwang). Mit dem %–Zeichen können in beliebigen Zeilen Kommentare eingefügt werden, die nicht ins .dvi–Dokument gelangen.
Vor dem Schlüsselwort BRIEF kann sich beliebiger (kommentierender) Text befinden. Auch er gelangt nicht mit ins Dokument. Das ist unter bestimmten Voraussetzungen auch hinter dem Dokumentende möglich.
Mit einem Menübefehl läßt sich die Zeilenlänge, an der in TPP umgebrochen werden soll, beliebig einstellen. Gelangt durch Änderung dieser Einstellung Text hinter diese Grenze, dann geht er bei der Compilierung nach LaTeX nicht verloren.

ShowDVI: Zum Lesen eines *.dvi–Textes haben die folgenden Tasten in PasTeX die angegebenen Funktionen. [An dieser Stelle dieses Briefes wird das Schlüsselwort *Einrückung* verwendet, definiert im tppdinbrief.sty und dort variierbar.]

die **4 Cursortasten** bewegen den Text entsprechend.

die **Spacetaste** bewegt den Text screenweise
nach unten, danach Seitenwechsel

die **Minustaste** bringt die vorhergehende Seite
zurück, letzte Zeilen

f zeigt die ganze Seite, zurück wieder mit f.

l zeigt die Textfeldgrenzen, zurück wieder mit l (Overflow!).

o schaltet die Scrollbar ein und aus.

m schaltet den Maßstab ein und aus.

Auflösung: Unter *showdvi* sind Auflösungen zwischen 44 und 360 dpi möglich. Die Standardauflösung (screenfüllend 800x600) beträgt 100 dpi.

Liste: Die Einrücktiefe bei Listen wird durch die Zahl der Buchstaben, die hinter das Schlüsselwort LISTE geschrieben werden, festgelegt.

Korrekturmodus: Meist wird wohl das Korrekturlesen im .dvi–File vorgenommen werden. Mit der ESC–Taste hat man sofort TPP im Vordergrund, zurück geht es mit F10. Schon während der Compilierungszeit kann man weiterlesen, wenn man statt dessen die Sreens TPP bzw. ShowDVI nach unten oder oben verschiebt. Durch das Multitasking des Amiga kann man während des Compilierens z.B. den alten showdvi–Text beliebig bewegen. . .

Quellen: PasTeX und TPP gibt es in Amigakreisen auf der CD–ROM MEETING PEARLS IV, die aber wohl kaum noch zu beschaffen ist. Bei Interesse bitte ich um eine entsprechende eMail–Nachricht.

Lizenzfragen: Ob durch meine Vorschläge und deren Weiterverwendung bestehende (Shareware–)Lizenzen verletzt werden, sollten die zukünftigen Programmierer prüfen und beachten. Von TPP gibt es eine freie Version mit einem sog. Nerv-Requester. Sie befindet sich auf der o.g. CD.

Schlußbemerkung

Ich bin der Meinung, und hoffentlich wird sie von dem einen oder anderen auch geteilt, daß die Konzepte von TPP und PasTEX überaus gut sind und nicht verloren gehen sollten. Dabei ist immer zu bedenken, daß die Entstehungszeit dieses Paketes nur von 1988 bis 1995 reicht! Welchen Stand hatten damals andere Systeme…?

Vielleicht gelingt es auf diesem Wege die beiden Autoren Martin Steppler und Georg Hessmann (zu letzterem Programmpaket aus Zeit– und Platzgründen fast nichts geschrieben wurde) wieder ausfindig zu machen und ggf. ihre Mithilfe oder ihren Rat für eine Implementation nach LINUX zu gewinnen. Zumindest verdienen die Autoren dieser Programme diese Veröffentlichung als ein kleines Denkmal.

Andererseits hoffe ich, daß sich der eine oder andere LINUX–Programmierer findet, der dieses Konzept aufgreift und erfolgreich umsetzen kann. Der Autor diese Beitrages ist gern bereit, weitere Auskünfte zu geben. Aus Altersgründen ist er nicht mehr in der Lage, sich der Last einer solchen Programmierung auszusetzen.

Meine eMail: Chr.Scharrnbeck@t-online.de

Zum Schluß eine Absurdität: Ich kann bis heute ohne TPP nicht direkt mit LATEX bzw. PasTEX arbeiten: Da müßte ich erst ein Weilchen die Doks studieren. Alles was ich seit ca. 1994 zu schreiben hatte und in Zukunft habe (Briefe, Artikel, die bearbeitete Dok zu TPP, Tabellen, Listen usw.) schreibe ich unter TEXt Plus Professional/PasTEX.