

Helmut Kopka

# LATEX

Band 3: Erweiterungen

**st**  
scientific tools

# Kapitel 1

## Das T<sub>E</sub>X-Gesamtsystem

Dieses Kapitel richtet sich an T<sub>E</sub>X-Systemverwalter, die das T<sub>E</sub>X-System auf ihrem Rechner zu warten, anzupassen und/oder zu ergänzen haben. Dies schließt den PC-Betreiber ein, der diese Aufgabe auf seinem Individualrechner selbst vorzunehmen hat. Dabei wird vorausgesetzt, dass auf dem betrachteten Rechner ein arbeitsfähiges T<sub>E</sub>X- und L<sup>A</sup>T<sub>E</sub>X-System verfügbar ist. [5a, Anh. F.1–F.3] enthält ausführliche Hinweise zur Installation eines T<sub>E</sub>X- und L<sup>A</sup>T<sub>E</sub>X-Grundsystems, auf die bei Bedarf zurückgegriffen werden sollte. Einige der dort gegebenen Erläuterungen werden hier zunächst zur Erinnerung wiederholt.

### 1.1 Ein T<sub>E</sub>X-Grundsystem

#### 1.1.1 Das T<sub>E</sub>X-Startsystem

Nach der Installation eines T<sub>E</sub>X-Systems stehen zwei lauffähige Programme zur Verfügung, von denen das eine den Namen `initex` und das andere den Namen `virtex` oder häufig auch nur `tex` trägt. Für die Ausführung dieser Programme wird *zwingend* ein File mit dem Namen `tex.poo` oder, wenn der Namensanhang mehr als drei Buchstaben enthalten darf, `tex.pool` benötigt. Dieses File entsteht bei der Kompilierung der ausführbaren Programme aus deren Quellenfiles<sup>1</sup>. Wurde das T<sub>E</sub>X-Paket nur in Form ausführbarer Programme geliefert, wie es bei den meisten PC-Versionen der Fall ist, so muss `tex.poo` zwingend beigelegt sein.

Die ausführbaren Programme `initex` und `virtex`, die unter DOS zusätzlich durch den Anhang `.exe` als `initex.exe` bzw. `virtex.exe` gekennzeichnet werden, erwarten das File `tex.poo` in einem speziellen Verzeichnis des sog. T<sub>E</sub>X-Filesystems (s. 1.2), das bei der Kompilierung vorgegeben oder mittels Umgebungsvariablen festgelegt werden kann. Falls der Betreiber zu Beginn das T<sub>E</sub>X-Filesystem noch nicht überschaut, so kann er das `tex.poo`-File ebenso wie die im nächsten Unterabschnitt aufgelisteten Systemfiles ins aktuelle Verzeichnis kopieren, da das aktuelle Verzeichnis bei allen ausführbaren T<sub>E</sub>X-Programmen bei der Filesuche ebenfalls durchmustert wird.

---

<sup>1</sup>Das File `tex.pool` besteht aus einer Vielzahl von Zeichenketten mit vorangestellter Längeninformation. Hiermit werden die zugehörigen Texte, z. B. die Fehlermeldungen, effizienter verwaltet, als dies standardmäßig durch den Pascal-Compiler erfolgt.

Das Programm `virtex` oder `tex` ist das T<sub>E</sub>X-Grundprogramm zur Formatierung der einzugebenden Texte. Es stellt die ca. 300 T<sub>E</sub>X-Grundbefehle bereit, die im Standardwerk von Donald E. Knuth, "The T<sub>E</sub>Xbook" [10a], mit einem \* im Indexregister gekennzeichnet sind. Im gleichen Buch werden jedoch weitere ca. 600 T<sub>E</sub>X-Befehle vorgestellt, die für den Anwender in gleicher Weise wie die Grundbefehle erscheinen. Sie werden durch sog. T<sub>E</sub>X-Makros realisiert. Makros sind T<sub>E</sub>X-Strukturen, die wie Befehle unter einem Namen bereitgestellt werden, hinter denen sich aber T<sub>E</sub>X-Ablaufolgen verbergen, die ihrerseits entweder auf einfache Zeichenketten und/oder T<sub>E</sub>X-Grundbefehle oder auf weitere Makros zurückführen. Makrodefinitionen stellen von ihrer Struktur her ein Stückchen T<sub>E</sub>X-Text dar, wie er mit jedem Editor erstellt werden kann. Beim Aufruf eines Makros wird dieses dann im Zuge der Bearbeitung gemäß seiner Definition durch den Ablauftext ersetzt und dann abgearbeitet.

Alle in "The T<sub>E</sub>Xbook" vorgestellten Zusatzbefehle werden als Makros in dem File `plain.tex` definiert. Dieses gehört damit ebenfalls zur Minimalausstattung eines nutzbaren T<sub>E</sub>X-Systems. Die Abarbeitung komplexer Makros kann zeitaufwendig sein, da sie mit jedem Aufruf aufs Neue vollständig in ihren Ersatztext umgesetzt werden und dieser dann Schritt für Schritt abgearbeitet wird.

### 1.1.2 T<sub>E</sub>X-Formatfiles

Hier kommt nun das Programm `initex` zum Tragen. Es stellt eine spezielle Version von T<sub>E</sub>X dar, mit der Makros oder ganze Makropakete *vorbearbeitet* und in *maschinenspezifischer* Form abgelegt werden. Die Abarbeitung solcher vorbereiteten Makros erfolgt schneller als die der Originaldefinitionen. Das Programm `initex` stellt gewissermaßen den T<sub>E</sub>X-Compiler zur Vorbearbeitung von T<sub>E</sub>X-Makropaketen dar. Dieser erwartet beim Aufruf die Angabe des Filenamens für das Makropaket. Trägt der Filename für das Makropaket den Anhang `.tex`, so genügt die Angabe des Grundnamens. Der Bearbeitungsaufwurf ist systemabhängig und lautet meistens:<sup>2</sup>

```
initex file_grund_name.anh oder initex file_grund_name
```

Das Bearbeitungsergebnis wird in einem sog. *Formatfile* mit dem gleichen Grundnamen und dem Anhang `.fmt` abgelegt. Für das File `plain.tex` erfolgt der Bearbeitungsaufwurf also einfach durch:

```
initex plain
```

Auf dem Bildschirm erscheint hierauf (am Beispiel meines UNIX-Systems):

```
This is TeX, Version 3.1415 (C Version 6.1) (INITEX)
(plain.tex Preloading the plain format: codes, registers, parameters,
fonts, more fonts, macros, math definitions, output routines, hyphenation
(/usr/local/lib/texmf/tex/plain/hyphen.tex))
*
```

Hierauf bleibt das Programm stehen und wartet auf eine Anwenderreaktion. Diese muss lauten: `\dump`, gefolgt von der Returnntaste, worauf die Bearbeitung mit der Bildschirmnachricht

<sup>2</sup>Bei dem auf PCs verbreiteten emT<sub>E</sub>X-Paket ist `initex` kein eigenständiges Programm, sondern erfolgt als Aufruf des ausführbaren T<sub>E</sub>X-Programms mit der Optionsangabe `/i` als

```
tex /i file_grund_name.anh oder tex /i file_grund_name.
```

```
Beginning to dump on files plain.fmt
(format=plain aktuelles Datum
```

fortgesetzt wird, gefolgt von etwas Statistik und einer Vielzahl Zeilen der Form

```
\font\TeX_name=zs_file oder
\font\preloaded=zs_file
```

Die Bearbeitung endet mit einigen weiteren statistischen Angaben über die vorbereiteten Zeichensätze und das/die verwendete(n) Trennmuster. Das Bearbeitungsergebnis wird in `plain.fmt` abgelegt. Die vorangegangenen Bildschirmmitteilungen und eine Reihe weiterer Bearbeitungsinformationen werden gleichzeitig in `plain.log` protokolliert.

Mit den Anweisungen der Form `\font\TeX_name=zs_file` werden die *metrischen* Eigenschaften der angeforderten Zeichensätze, also die *Höhe*, *Tiefe* und *Breite* ihrer Zeichen sowie deren Abstand zum nachfolgenden Zeichen, bekannt gemacht, die dann unter dem angegebenen  $\TeX$ -Befehlsnamen zur Verfügung stehen (*definierte* Zeichensätze). Die metrischen Zeichensatzinformationen werden mit den sog. `.tfm`-Files, also den Zeichensatzfiles mit dem Namensanhang `.tfm`, bereitgestellt. Mit

```
\font\tenrm=cmr10
```

wird die Information aus dem File `cmr10.tfm` für die  $\TeX$ -Bearbeitung unter dem Befehlsnamen `\tenrm` in maschinenspezifischer Form intern abgespeichert.

Mit der anderen Anweisungsgruppe der Form `\font\preloaded=zs_file` werden die metrischen Informationen der angeforderten `.tfm`-Files ebenfalls im erzeugten Formatfile maschinenspezifisch abgelegt. Sie bleiben zunächst jedoch anonym, da ihnen kein entsprechender Befehlsname zugeordnet wurde. (Der benutzte Befehlsname `\preloaded` wird in `plain.tex` später als *undefiniert* erklärt!) Der Vorteil dieser *vorgeladenen* anonymen Zeichensätze liegt darin, dass sie bei einer späteren  $\TeX$ -Bearbeitung mit expliziten `\font`-Befehlen aktiviert werden können, *ohne* dass die zugehörigen `.tfm`-Zeichensätze zum Bearbeitungszeitpunkt erneut eingelesen werden müssen.

Alle durch `plain.tex` angeforderten `.tfm`-Zeichensatzfiles müssen zum Zeitpunkt des `initex`-Aufrufs verfügbar sein. Sie gehören damit ebenfalls zur Minimalausstattung eines  $\TeX$ -Systems. Für ein Standard-`plain.tex` sind dies

<code>cmbx5.tfm</code>	<code>cmr5.tfm</code>	<code>cmmi5.tfm</code>	<code>cmsy5.tfm</code>
<code>cmbx7.tfm</code>	<code>cmr7.tfm</code>	<code>cmmi7.tfm</code>	<code>cmsy7.tfm</code>
<code>cmbx10.tfm</code>	<code>cmr10.tfm</code>	<code>cmmi10.tfm</code>	<code>cmsy10.tfm</code>
<code>cmex10.tfm</code>	<code>cmtt10.tfm</code>	<code>cmti10.tfm</code>	<code>cmsl10.tfm</code>

für die definierten Zeichensätze und zusätzlich

<code>cmbx6.tfm</code>	<code>cmr6.tfm</code>	<code>cmmi6.tfm</code>	<code>cmsy6.tfm</code>	<code>cmti7.tfm</code>
<code>cmbx8.tfm</code>	<code>cmr8.tfm</code>	<code>cmmi8.tfm</code>	<code>cmsy8.tfm</code>	<code>cmti8.tfm</code>
<code>cmbx9.tfm</code>	<code>cmr9.tfm</code>	<code>cmmi9.tfm</code>	<code>cmsy9.tfm</code>	<code>cmti9.tfm</code> ,
<code>cmss10.tfm</code>	<code>cmssi10.tfm</code>	<code>cmsl8.tfm</code>	<code>cmtt8.tfm</code>	<code>cmsl10.tfm</code>
<code>cmssq8.tfm</code>	<code>cmssqi8.tfm</code>	<code>cmsl9.tfm</code>	<code>cmtt9.tfm</code>	

sowie

<code>cmssbx10.tfm</code>	<code>cmcsc10.tfm</code>	<code>cmuib10.tfm</code>	<code>cmbsy10.tfm</code>	<code>cmu10.tfm</code>
<code>cmdunh10.tfm</code>	<code>manfnt.tfm</code>			

für die vorgeladenen Zeichensätze. Die letzte Gruppe enthält diejenigen Zeichensätze, denen in `plain.tex` der Bildschirmausgabebefehl `\message{more fonts}` vorangeht.

Kurz vor Ende des Files `plain.tex` findet man den Befehl `\input hyphen`. Bei der `initex`-Bearbeitung wird an dieser Stelle nach dem File `hyphen.tex` gesucht und dieses, wenn es gefunden wird, eingelesen. Es enthält das US-englische Trennverzeichnis, das somit Bestandteil des Formatfiles `plain.fmt` wird. Wird `hyphen.tex` in `UShyphen.tex` umbenannt, so bleibt die Suche nach `hyphen.tex` erfolglos, und `initex` bleibt mit der Fehlermeldung

```
! I can't find file 'hyphen'
1.1211 \input hyphen
Please type another input file name:
```

stehen und wartet auf eine entsprechende Tastatureingabe. Mit der Eingabe von `UShyphen`, gefolgt von der Return Taste, wird das umbenannte File `UShyphen.tex` eingelesen, die `initex`-Bearbeitung fortgesetzt und mit der abschließenden Eingabe `\dump` (Return) wie vorher mit der Ausgabe von `plain.fmt` beendet.

Die Abfrage zur Eingabe des Trennmusterfiles kann aber auch mit einem anderen Filenamens beantwortet werden. Die deutschen Trennmusterfiles wird in unter den Namen `dehyph.tex` (alte Rechtschreibung) bzw. `dehyphn.tex` (neue Rechtschreibung) bereitgestellt. Mit der Antwort `dehyph` bzw. `dehyphn` und der Abschlusseingabe `\dump` wird ein Formatfile `plain.fmt` erzeugt, das die entsprechende deutsche Trennmusterliste enthält.

Wird der vorstehende `initex`-Aufruf zweimal vorgenommen und die Frage nach dem einzulesenden Trennmusterfile einmal mit `dehyphn` und ein anderes Mal mit `UShyphen` beantwortet, so wird das Bearbeitungsergebnis beide Male unter dem Namen `plain.fmt` abgelegt, womit das zweite Ergebnisfile das erste überschreibt. Um dies zu vermeiden, muss das erste Ergebnisfile vorab umbenannt werden, z. B. als `gplain.fmt`. Die beiden Ergebnisfiles `gplain.fmt` und `plain.fmt` können dann in Verbindung mit `virtex` zur Bearbeitung deutscher bzw. englischer Texte genutzt werden.

T<sub>E</sub>X-Versionen ab 3.0 können im Prinzip bis zu 255 verschiedene Trennmusterfiles in *einem* Formatfile einbinden. Wird mit dem Editor ein kleines File mit dem Inhalt

```
\message{= Loading hyphenation patterns:}
\chardef\l@USenglish=\language
\chardef\l@english=\l@USenglish %% british english as "dialect"
\input USshyphen

\newlanguage\l@german \language=\l@german
\chardef\l@austrian=\l@german %% austrian as german dialect
\input dehyphn

\newlanguage\l@french \language=\l@french
\input fhyp

%% Default hyphenation pattern: USenglish
\language=\l@USenglish \lefthyphenmin=2 \righthyphenmin=3
\endinput
```

unter dem Namen `hyphen.tex` abgelegt, so wird dieses bei der `initex`-Bearbeitung an der Stelle des Befehls `\input hyphen` eingelesen. Es liest seinerseits dann nacheinander die US-englischen, deutschen und französischen Trennmusterfiles `USshyphen.tex`, `dehyphn.tex`

und `fhyph.tex` ein und verknüpft sie mit den internen Werten 0, 1 und 2 für den Sprachschalter `\language`. Die alternative Einbindung von `dehyph.tex` mit dem Trennmusterfile zur alten deutschen Rechtschreibung bedarf keiner zusätzlichen Erläuterung. Zur Einbindung beider deutscher Trennmusterfiles verweise ich bei Bedarf auf [5a, F.1.1 und F.2.1].

Achtung: Die ausführbaren Programme `initex` und `virtex` richten einen internen Pufferspeicher zur Aufnahme der Trennmuster ein, dessen Größe durch die interne Konstante `trie_size` bei den T<sub>E</sub>X-Standardprogrammen mit 8000 vorgegeben wird. Dieser Wert ist zur Aufnahme mehrerer Trennmusterlisten zu klein und muss vor der Kompilierung vergrößert werden, falls die Pufferspeicher nicht zur Laufzeit abgeändert werden können. Letzteres ist z. B. bei den T<sub>E</sub>X-Programmen aus dem emT<sub>E</sub>X-Paket der Fall. Bei den ausführbaren Programmen aus dem UNIX-Installationspaket wird `trie_size=30000` eingestellt, was zur Aufnahme der vorgeschlagenen dreisprachigen Trennmusterfiles ausreicht.

### 1.1.3 Formatfile-Einbindung

Das eigentliche T<sub>E</sub>X-Textbearbeitungsprogramm `virtex` bzw. `virtex.exe` verlangt beim Aufruf neben dem Namen des zu bearbeitenden Textfiles auch die Angabe des zu verwendenden Formatfiles. Das Formatfile wird hierbei durch ein vorangestelltes `&`-Zeichen gekennzeichnet (das unter UNIX durch einen nochmals vorangestellten Rückstrich `\` zu maskieren ist). Die Aufrufsyntax lautet damit:

```
virtex &format_file text_file bzw. virtex \&format_file text_file (UNIX)
```

Wurden mit `plain.fmt` und `gplain.fmt` zwei Plain-Formatfiles mit dem US-englischen Original- bzw. dem deutschen Trennmusterfile erzeugt, so können englische Textfiles mit dem Aufruf `virtex &plain text_file` und deutsche Textfiles mit dem Aufruf `virtex &gplain text_file` sachgerecht bearbeitet werden. Erzeugt man unter DOS zwei kleine Befehlsdateien mit den Namen `tex.bat` bzw. `gtex.bat` und dem Inhalt

```
virtex &plain %1 bzw. virtex &gplain %1
```

dann kann der Aufruf vereinfacht mit

```
tex text_file für englische Texte bzw.
gtex text_file für deutsche Texte
```

erfolgen. Wurde ein Formatfile `plain.fmt` mit mehrsprachigen Trennmustern erstellt, so lautet der Aufruf zunächst einfach `virtex &plain text_file` oder erfolgt mit einer entsprechenden Befehlsdatei `tex.bat` als `tex text_file`. Die Aktivierung des erforderlichen sprachspezifischen Trennmusters erfolgt dann mit der zugehörigen Schaltereinstellung `\language=n` innerhalb des Textfiles `text_file` selbst.

Auch der Aufruf zur L<sup>A</sup>T<sub>E</sub>X-Bearbeitung ist nichts anderes als `virtex &latex text_file`, bei dem nun ein Formatfile `latex.fmt` zugefügt wird, das aus der `initex`-Bearbeitung des L<sup>A</sup>T<sub>E</sub>X-Quellenfiles `latex.ltx` gewonnen wird. Bei dieser `initex`-Bearbeitung liest das Ausgangsquellenfile `latex.ltx` bei der Formaterstellung weitere Files mit dem Anhang `.ltx` ein, worauf in 2.1.2 genauer eingegangen wird. In gleicher Weise können Formatfiles für andere Makropakete, z. B. `amstex.tex`, erzeugt und anschließend mit `virtex` durch Aufrufe der Form ‘`virtex &makro_pak text_file`’ aktiviert werden, z. B. als ‘`virtex &amstex text_file`’ zur  $\mathcal{A}\mathcal{M}\mathcal{S}$ -T<sub>E</sub>X-Bearbeitung von `text_file`.

Der `virtex`-Aufruf mit der Einbindung des Formatfiles `plain.fmt` wird im T<sub>E</sub>X-Sprachgebrauch häufig als PLAIN-T<sub>E</sub>X bezeichnet. Dies ist in sich schlüssig, da sich hinter den Ausdrücken der L<sup>A</sup>T<sub>E</sub>X- oder A<sub>M</sub>S-T<sub>E</sub>X-Bearbeitung ebenfalls nur der `virtex`-Aufruf mit der Einbindung der `latex.fmt`- bzw. `amstex.fmt`-Formatfiles verbirgt.

### 1.1.4 Formatfile-Erweiterungen

Soll ein bestehendes Formatfile `format.fmt` um ein weiteres Makropaket, das mit dem File `extra_makr.tex` bereitgestellt wird, erweitert werden, so kann dies mit einem `initex`-Aufruf der Form

```
initex &format extra_makr
```

geschehen. Bei dieser `initex`-Bearbeitung wird das vorhandene Formatfile `format.tex` vorab eingelesen und mit den Makrodefinitionen aus `extra_makr.tex` verknüpft. Das Bearbeitungsergebnis wird in einem neuen Formatfile mit dem Namen `extra_makr.fmt` abgelegt. Wurde z. B. ein Standardformatfile `plain.fmt` mit dem US-englischen Trennverzeichnis eingerichtet, so könnte mit der Bereitstellung eines Files `mplain.tex` und dem Inhalt, wie er auf S. 4 mit dem dort vorgeschlagenen `hyphen.tex` mit Ausnahme der vierten Zeile `'\input UShyphen'` entspricht, ein mehrsprachiges Formatfile `mplain.fmt` schneller mit dem Aufruf `'initex &plain mplain'` erzeugt werden.

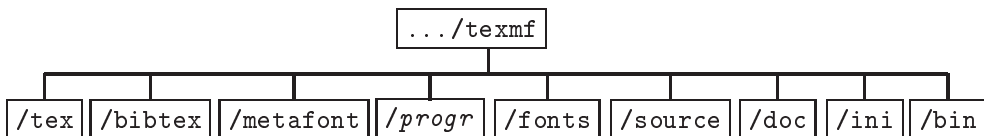
## 1.2 Das T<sub>E</sub>X-Filesystem

Die ausführbaren T<sub>E</sub>X-Programme erwarten die angeforderten Files unter bestimmten Pfad- und Verzeichnisnamen, die das sog. T<sub>E</sub>X-Filesystem bilden. Dieses wird bei der Kompilierung der Quellenprogramme vorbestimmt und kann bei den meisten Betriebssystemen mit sog. Umgebungsvariablen abgeändert werden. Soweit die ausführbaren Programme beim Anwender durch eigene Kompilierung erstellt werden, kann das geforderte Filesystem nach den Vorstellungen des Anwenders durch Editieren der zugehörigen Makefiles vorgenommen werden. Dies ist z. B. bei der Installation eines T<sub>E</sub>X-Systems unter UNIX der Fall.

Werden die ausführbaren Programme als fertige binäre Files geliefert, wie das bei den meisten PC-T<sub>E</sub>X-Systemen der Fall ist, dann muss das eingebaute Filesystem übernommen oder mit dem Setzen expliziter Umgebungsvariablen an die Wünsche des Anwenders angepasst werden. Dabei hängt es von der Lieferquelle ab, welche Files mit welchen Umgebungsvariablen zugeordnet werden können. Dies führt meistens zu unterschiedlichen T<sub>E</sub>X-Filesystemen für die verschiedenen Herkunftsquellen.

Die internationale T<sub>E</sub>X-Users-Group (TUG) hat deshalb einen Arbeitskreis eingerichtet, der einen Vorschlag für ein einheitliches T<sub>E</sub>X-Filesystem erarbeiten soll. Ein solcher wurde Mitte Juni 1995 unter dem Kürzel TDS (T<sub>E</sub>X Directory Structure, Vers. 0.98) vorgestellt und zuletzt am 21. April 1999 mit der Version 0.9996 verbessert. Nach diesem Vorschlag soll das Ausgangsverzeichnis für das T<sub>E</sub>X-Gesamtsystem den Namen `.../texmf` tragen. Die Einbindung dieses T<sub>E</sub>X-Ausgangsverzeichnisses in das gesamte Rechner-Filesystem bleibt dem Systemverwalter überlassen. Unter UNIX ist es traditionell `/usr/local/lib/texmf` und unter DOS vermutlich ein Hauptverzeichnis eines Laufwerks, z. B. `D:\texmf`.

Die erste Ebene für das T<sub>E</sub>X-Eingabeverzeichnis `.../texmf` soll dann mindestens aus den Unterverzeichnissen



bestehen. Die Namen sind teilweise selbsterklärend. Die meisten dieser Unterverzeichnisse sind ihrerseits untergliedert. Diese Eingangsebene und ihre weitere Untergliederung wird in den nachfolgenden Unterabschnitten vorgestellt.

Bei dieser Vorstellung verwende ich die UNIX-Syntax für die Verzeichnisgliederung. Sie ist, bis auf das Eingangsverzeichnis, weitgehend mit der MS-DOS-Syntax identisch, wenn der Schrägstrich / aus UNIX unter DOS durch den Rückstrich (Backslash) ersetzt wird. Der UNIX-Angabe `/usr/local/lib/texmf/source` könnte unter MS-DOS z. B. `d:\texmf\source` entsprechen. Die VMS-Syntax auf VAX-Rechnern unterscheidet sich zwar formal von der einfachen UNIX- oder DOS-Verzeichnisstruktur, stimmt jedoch mit diesen in der Logik der Verzeichnishierarchie überein, und die entsprechende Zuordnung wird dem VMS-Nutzer nicht schwerfallen.

UNIX erlaubt Filenamen mit einer Namenslänge von bis zu 256 Zeichen (Buchstaben, Ziffern und einige Sonderzeichen wie `_` und `-`), wobei der Namensanhang mehrfach untergliedert sein darf. Dies macht unter UNIX weitgehend selbstbeschreibende Filenamen möglich. MS-DOS beschränkt Filenamen auf die Gliederung *grund.name.anh*, wobei die maximale Länge für den Grundnamen acht und für den Anhang drei Zeichen beträgt. Der TDS-Vorschlag für das T<sub>E</sub>X-Filesystem übernimmt die MS-DOS-Einschränkungen für die Verzeichnisnamen, um eine weitgehend einheitliche Namensgebung auf allen Rechnerplattformen zu ermöglichen.

Nach dem vorstehenden Vorschlag für ein T<sub>E</sub>X-Filesystem sollten zukünftig alle T<sub>E</sub>X-Systeme strukturiert werden, was bei neueren T<sub>E</sub>X-Installationen meistens auch der Fall ist. Frühere T<sub>E</sub>X-Installationen waren dagegen nach den Vorstellungen des jeweiligen Quellenbetreuers untergliedert. Dies führte, je nach Herkunftsquelle, zu recht unterschiedlichen Verzeichnisstrukturen. Der T<sub>E</sub>X-Betreuer für ein Rechenzentrum mit mehreren unterschiedlichen Rechnern und Betriebssystemen sollte den TDS-Vorschlag für alle seine Plattformen zu übernehmen. Die Wartung der T<sub>E</sub>X-Systeme wird dadurch mit Sicherheit einfacher, auf jeden Fall übersichtlicher.

Für ein T<sub>E</sub>X-Filesystem nach dem TDS-Vorschlag ist es hilfreich, wenn die ausführbaren Programme eine rekursive Suchstrategie für die angeforderten Files erlauben. Darunter ist Folgendes zu verstehen: Die Suche nach einem angeforderten File beginnt in einem (oder mehreren) bei der Installation vorbestimmten oder explizit mit Umgebungsvariablen erklärten Eingangsverzeichnis(en). Wird es dort nicht gefunden, wird die Suche in den darunter liegenden Unterverzeichnissen der ersten Ebene fortgesetzt. Bleibt die Suche auch hier erfolglos, so wird die nächste Unterverzeichnisebene durchmustert usw.

Die rekursive Suchstrategie kann für alle ausführbaren Programme einer T<sub>E</sub>X-Installation unter UNIX vorausgesetzt werden, zumindest wenn diese auf das von DANTE e. V. gepflegte Installationspaket `dante-tex.tar` oder auf dessen Original `web2c` von Karl Berry zurückgreifen. Dies gilt gleichermaßen für die DVI-Treiberpakete `dante-dviflj.tar` und `dante-dvips.tar`. Ebenso können die ausführbaren Programme aus dem emT<sub>E</sub>X-Paket, einschließlich aller beigefügten DVI-Treiber, mit einer rekursiven Suchstrategie ausgestattet werden.

Die rekursive Suchstrategie für angeforderte Files aus einem nach dem TDS-Vorschlag gegliederten Filesystem ist nicht zwingend erforderlich. Ohne diese müssen jedoch alle bei einer Filesuche zu durchmusternden Verzeichnisse und Unterverzeichnisse entweder bereits bei der Installation oder mit entsprechenden Umgebungsvariablen explizit angegeben werden. Dies erschwert die Erweiterung des T<sub>E</sub>X-Filesystems, weil damit entweder die Installation mit der entsprechenden Erweiterung wiederholt oder die zugehörigen Umgebungsvariablen entsprechend erweitert werden müssen.

### 1.2.1 Format- und Basisfiles

Alle mit `initex` bearbeiteten Makropakete, also die sog. Formatfiles mit dem Namensanhang `.fmt`, sollten im Eingangsverzeichnis `.../texmf/ini` abgelegt werden. Bei der Erstellung von multilingualen Formatfiles gemäß 1.1.2, S. 4 wird sich dies häufig auf die beiden Formatfiles `plain.fmt` und `latex.fmt` beschränken, evtl. ergänzt durch `amstex.fmt` und `texinfo.fmt`. Dies gilt zumindest nach der Einführung von  $\LaTeX 2_{\epsilon}$ , dank derer die bisherigen unterschiedlichen  $\LaTeX$ -Formatfiles für diverse  $\LaTeX$ -Variationen durch nur ein einheitliches  $\LaTeX 2_{\epsilon}$ -Formatfile abgelöst wurden.

Das Pendant zu den Formatfiles und ihrem Erzeugungsprogramm `initex` sind bei einem METAFONT-System die Basisfiles mit dem Anhang `.bas` sowie das Erzeugungsprogramm `inimf`. Für Details zur Erstellung der Basisfiles verweise ich auf [5a, F.1.6] sowie ergänzend für  $\emTeX$ -Betreiber auf Abschnitt F.2.3.2 im gleichen Buch. Für die allermeisten METAFONT-Anwendungen reicht die Erstellung von `plain.bas` und `cmbase.bas`. Auch die Basisfiles sind im Unterverzeichnis `.../texmf/ini` abzulegen.

In dieses Verzeichnis gehören schließlich noch die  $\TeX$ - und METAFONT-Poolfiles, also die Files `tex.pool` und `mf.pool`, deren Namensanhänge unter MS-DOS auf drei Buchstaben und damit auf `.poo` verkürzt sind.

Kommen  $\TeX$  und METAFONT beim Anwender sowohl in einer Standard- als auch in einer BIG-Version zur Anwendung, dann empfiehlt es sich, das Eingangsverzeichnis `.../texmf/ini` entsprechend zu untergliedern, z. B. in `.../texmf/ini/std` und `.../texmf/ini/big`, und die Format-, Basis- und Poolfiles der Standard- und Big-Versionen in den zugeordneten Unterverzeichnissen abzulegen.

Die Einrichtung des Verzeichnisses `./ini` in der Eingangsebene `.../texmf` entspricht einer früheren Empfehlung der TDS-Arbeitsgruppe. In der letzten Version 0.999 wurde sie wieder zurückgenommen. Falls die aktuelle Empfehlung nun so zu verstehen ist, die verschiedenen Format-, Basis- und Poolfiles in vertieften Verästelungen des  $\TeX$ -Dateienbaums unterzubringen, so erscheint mir der frühere explizite Vorschlag für das Eingangsverzeichnis `./ini` konsequenter, da er gleichzeitig die herausgehobene Bedeutung der Format-, Basis- und Poolfiles für die  $\TeX$ - bzw. METAFONT-Bearbeitung unterstreicht. Die Einrichtung dieses Sonderverzeichnisses entspricht auch der `Makefile`-Installationshilfe für ein  $\TeX$ -Filesystem unter UNIX, falls diese nicht manuell abgeändert wird.

### 1.2.2 $\TeX$ -Makropakete

Das Eingangsverzeichnis `.../texmf/tex` ist zur Aufnahme aller Makropakete gedacht, die bei einem  $\TeX$ -Aufruf evtl. zusätzlich eingebunden werden. Da zu einem  $\TeX$ -Programmaufruf stets die Angabe des zu verwendenden Formatfiles gehört (s. 1.1.3), ist das Eingangsverzeichnis `./tex` weiter zu untergliedern, und zwar wenigstens in Unterverzeichnisse, deren Namen den Grundnamen der erstellten Formatfiles entsprechen.

Für die Mehrzahl der  $\LaTeX$ -Anwender sind dies wenigstens die Unterverzeichnisse `./plain` und `./latex`. Als weiteres paralleles Unterverzeichnis wird häufig `./amstex` benötigt, insbesondere bei Anwendern mit umfangreichen mathematischen Formatierungsaufgaben. Die formatspezifischen Unterverzeichnisse sollten weiter untergliedert werden, und zwar in die Subunterverzeichnisse `./base`, `./misc`, `./local` und bei Bedarf in `./<zus_pakete>` für weitere Ergänzungen. Das nachstehende Diagramm demonstriert die empfohlene Untergliederung des Eingangsverzeichnisses `.../texmf/tex`.

Unter `./tex/plain/base` sind diejenigen Makropakete abzulegen, die zur Grundausstattung eines PLAIN-T<sub>E</sub>X-Pakets gehören und *nur* im Zusammenhang mit dem `plain.fmt`-Formatfile verwendet werden. Dies sind die `.tex`-Files, die man auf den T<sub>E</sub>X-Fileservern unter `/tex-archive/macros/plain/base` findet, mit Ausnahme der dortigen Files `null.tex` und `hyphen.tex`.

In gleicher Weise sind unter `./tex/latex/base` alle `.cls-`, `.clo-`, `.def-`, `.fd-`, `.sty-` und `.ltx`-Files abzulegen, die nach dem Entpacken der L<sup>A</sup>T<sub>E</sub>X-Originalfiles aus dem Archiv `/tex-archive/macros/latex/base` entstehen. Für Erstellungsdetails verweise ich auf 2.1.1 und [5a, F.1.3].

Das jeweilige Unterverzeichnis `./misc` für die verschiedenen Formatunterverzeichnisse ist für solche Makropakete gedacht, die jeweils nur aus *einem* File bestehen und nicht bereits im Unterverzeichnis `./base` enthalten sind. Schließlich sind die Unterverzeichnisse mit dem Namen `./local` für lokale Konfigurationsfiles aus dem jeweiligen Formatverzeichnis vorgesehen. Mit weiteren Unterverzeichnissen `./<zus_paket>` können weitere Makropakete geordnet werden. Auf den öffentlichen Fileservern findet man solche in großer Vielzahl, z. B. unter dem Eingangsverzeichnis `/tex-archive/macros` in weiteren Unterverzeichnissen wie `./plain/contrib`, `./latex/packages`, `./latex/contrib` u. a.

Das Eingangsverzeichnis `./generic` unterhalb von `.../texmf/tex` ist für solche Makropakete gedacht, die von mehreren Formaten benutzt werden. Ebenso gehören hierhin diejenigen Makropakete, die bei der Erstellung von verschiedenen Formatfiles gemeinsam genutzt werden. Die hierunter vorgeschlagene Gliederung ist mit `./hyphen` zur Aufnahme der sprachspezifischen Trennmusterfiles gedacht und mit `./misc` für sonstige Makropakete wie `null.tex` aus dem PLAIN-T<sub>E</sub>X-Grundpaket, das für alle Formate als Eingabereaktion auf die Fehlermeldung `! I can't find file '...' Please type another input file name:` für eine zielgerichtete Programmbeendigung genutzt werden kann.

### 1.2.3 METAFONT-Makropakete

Das Eingangsverzeichnis `.../texmf/metafont` ist zur Aufnahme des METAFONT-Makrofilesystems vorgesehen. Es ist seinerseits untergliedert in `./base`, `./misc` und `./local`. Unter `./base` sind alle `.mf`-Files abzulegen, die zum METAFONT-Grundsystem gehören. Das sind insbesondere die Files `plain.mf`, `cmbase.mf`, `expr.mf`, `grayf.mf` und `slant.mf` sowie die Beispielfiles aus [10b], wie `io.mf`, `manfnt.mf`, `test.mf` u. ä. Nicht hierher gehören die METAFONT-Quellenfiles für die diversen Zeichensätze. Diese sind im Zeichensatz-Filesystem einzurichten (s. u.).

Unter dem Unterverzeichnis `./misc` sind sonstige METAFONT-Makropakete abzulegen, die jeweils nur aus einem File bestehen und nicht bereits in `./base` auftreten. Typische Vertreter sind hierfür die Files `modes.mf` und `null.mf`. Ersteres findet man auf den T<sub>E</sub>X-Fileservern unter `/tex-archive/fonts/ams/amfonts/sources`. Es stellt die Einstellvorgaben für eine Vielzahl von Druckern bereit. Das andere File `null.mf` ist ein Leerfile, das bei einem METAFONT-Aufruf als Anwenderantwort zur Programmbeendigung genutzt werden kann, falls die Suche nach einem METAFONT-Makrofile erfolglos blieb.

Das Unterverzeichnis `./local` ist, seinem Namen entsprechend, für lokale METAFONT-Ergänzungen gedacht, falls solche existieren. Mit dieser relativ einfachen Filestruktur ist das METAFONT-Filesystem bereits vollständig beschrieben.

## 1.2.4 Das Zeichensatz-Filesystem

Das Eingangsverzeichnis `.../texmf/fonts` enthält das am tiefsten untergliederte T<sub>E</sub>X-Filesystem, das die ganz unterschiedlichen Typen der unter T<sub>E</sub>X möglichen Zeichensatzfiles aufnimmt. Die TDS-Arbeitsgruppe empfiehlt hierfür die nachstehende Unterstruktur:

```
./fonts/
  <file_typ>/
    <drucker_typ>/
      <herkunft>/
        <zs_quelle>/
          dpi <nnn>/
```

Die oberste Ebene *file\_typ* kennzeichnet hierbei die verschiedenen Zeichensatz-Filetypen. Vorgeschlagen werden hierfür als Minimalausstattung die Eingangsverzeichnisse `./tEX` für die T<sub>E</sub>X-Zeichensatz-Metrikfiles, `./pk` für die gepackten Drucker-Zeichensatzfiles, `./source` für die METAFONT-Zeichensatz-Quellenfiles und `./vf` zur Aufnahme der sog. *virtuellen* Zeichensätze (s. 1.3.3 und [5b, 4.2.4]).

Die METAFONT-Bearbeitung der Zeichensatz-Quellenfiles *zs\_name.mf* erzeugt bekanntlich Druckerzeichensätze in einem speziellen Format, das durch den Namensanhang `.gf` gekennzeichnet wird. Das Umwandlungsprogramm `gftopk` erzeugt hieraus die gepackten Druckerzeichensätze, erkennbar an einem `pk` im Namensanhang, die die meisten Druckertreiber dann verwenden. Kommt beim Anwender ein Druckertreiber zur Anwendung, der das `.gf`-Format erwartet, so sind die erzeugten `.gf`-Files ebenfalls abzulegen. Die TDS-Arbeitsgruppe schlägt hierfür als weiteres Eingangsverzeichnis `./gf` für *file\_typ* vor. Bei den meisten Installationen wird dieses Eingangsverzeichnis nicht benötigt, da die `.gf`-Files nach Erzeugung der äquivalenten `.pk`-Files gewöhnlich gelöscht werden.

Werden beim Anwender auch PostScript-Schriften genutzt, dann wird als weiteres Eingangsverzeichnis für *file\_typ* `./afm` (Adobe Font Metrics) benötigt. Diese speziellen Zeichensatz-Metrikfiles stehen auf den T<sub>E</sub>X-Fileservern für alle erdenklichen PostScript-Schriften frei zur Verfügung. Das Umwandlungsprogramm `afm2tEX` von Tomas Rokicki oder das T<sub>E</sub>X-Makropaket `fontinst.tex` von Alan Jeffrey erzeugt hieraus die zugehörigen `.tEX`-Metrikfiles, die für eine T<sub>E</sub>X- oder L<sup>A</sup>T<sub>E</sub>X-Bearbeitung zur Anwendung kommen.

Neben den in einem PostScript-Drucker fest eingebauten PostScript-Schriften existieren weitere PostScript-Schriften als sog. programmierte Schriften (Soft-Zeichensätze). Die PostScript-Programmiersprache stellt hierfür u. a. ein spezielles Beschreibungsformat unter dem Namen `type1` bereit. Auf den T<sub>E</sub>X-Fileservern findet man einige wenige frei verfügbare `type1`-kodierte Zeichensätze, z. B. für die Courier- und Utopia-Schriften. Die meisten der erhältlichen `type1`-PostScript-Zeichensätze sind dagegen Lizenzprodukte, die gekauft werden müssen und nur unter Einhaltung der Lizenzvorgaben genutzt werden dürfen. Kommen solche `type1`-Zeichensätze zur Anwendung, dann sind sie in dem zusätzlichen Eingangsverzeichnis `./type1` abzulegen.

Werden beim Anwender verschiedene Druckertypen verwendet, so sind die Eingangsverzeichnisse `./gf` und `./pk` in druckerspezifische Unterverzeichnisse *drucker\_typ* zu untergliedern, für die der Anwender eigene charakterisierende Verzeichnisnamen wählen kann. Bei allen anderen Zeichensatz-Eingangsverzeichnissen entfällt diese Unterebene.

Die nächste Unterebene *herkunft*, die bei allen Zeichensatz-Filetypen zur Anwendung kommt, kennzeichnet die Herkunft. Die TDS-Arbeitsgruppe empfiehlt, für alle frei verfügbaren bzw. verteilbaren Schriften das Unterverzeichnis `./public` bereitzuhalten, das seinerseits in die Zeichensatzquellen *zs\_quelle*, wie `./cm`, `./dc`, `./pandora` u. a., untergliedert ist. Weitere Verzeichnisnamen zur Kennzeichnung der Herkunft können sein `./ams` für die Zeichensätze der  $\mathcal{A}\mathcal{M}\mathcal{S}$  (American Mathematical Society), `./adobe` für PostScript-Schriften der Firma Adobe Systems Inc. u. a. Auch diese Verzeichnisse sind für die verschiedenen Quel-

lengruppen zu untergliedern, z. B. in `./euler`, `./symbols`, `./extracm` und `./cyrillic` für die  $\mathcal{A}\mathcal{M}\mathcal{S}$ -Schriften oder `./times`, `./helvetic`, `./newcent` usw. für die Adobe-PostScript-Schriften.

Die beiden letzten Gliederungsebenen *herkunft* und *zs\_quelle* treten unter allen Eingangsverzeichnissen für die verschiedenen Filetypen *file\_typ* auf. Die Herkunftskennzeichnung `./adobe` kann selbst beim Filetyp `./pk` erforderlich werden, nämlich dann, wenn mit dem Umwandlungsprogramm `ps2pk` von Piet Tutelaers (s. [5b, 4.4.1]) Typ-1-kodierte Adobe-Zeichensätze in äquivalente `./pk`-kodierte Zeichensätze für diskrete Vergrößerungsstufen umgewandelt werden. Umgekehrt könnte der Unterverzeichniszweig `./public/cm` auch unter dem Eingangsverzeichnis `./type1` auftreten, nämlich dann, wenn die T<sub>E</sub>X-cm-Schriften auch in Typ-1-kodierter Form vorliegen. Man findet sie in dieser Form auf den öffentlichen T<sub>E</sub>X-Fileservern unter `/tex-archive/fonts/cm/ps-type1`.

Bei den Eingangsverzeichnissen `./gf` und `./pk` erscheint als unterste Ebene zusätzlich noch `./dpi $n$ n`/, womit die fiktive Druckerauflösung entsprechend dem gewählten Skalierungsfaktor beim METAFONT-Bearbeitungsaufwurf untergliedert wird. Für einen 600 dpi-Drucker enthält das Unterverzeichnis `./dpi600` die `./gf`- bzw. `./pk`-Zeichensätze in ihrer Entwurfsgröße und das Unterverzeichnis `dpi720` diese dann um den Faktor 1.2 vergrößert.

## 1.2.5 Das Quellenfile-Eingangsverzeichnis

Das Eingangsverzeichnis `.../texmf/source` ist zur Aufnahme der originären Quellenfiles vorgesehen. Ich benutze es gleichzeitig als vorläufiges Installationsverzeichnis, indem ich die von mir gewünschten Dateien vom DANTE-Fileserver in gepackter und/oder archivierter Form zunächst in dieses Verzeichnis kopiere, z. B. das ganze T<sub>E</sub>X-Installationspaket für UNIX `dante-tex.tar.Z`, hier unter dem Anfangsverzeichnis `./tex.src/`. Ebenso habe ich das L<sup>A</sup>T<sub>E</sub>X-Grundsystem aus `/tex-archive/macros/latex/base` in `./zip`-gepackter Form hier unter `./latex.src/` als `base.zip` abgelegt und parallel dazu auch die L<sup>A</sup>T<sub>E</sub>X-Ergänzungen aus `/tex-archive/macros/latex/packages` als `packages.zip` kopiert. Entsprechendes gilt für alle sonstigen von mir vom DANTE-Fileserver angeforderten Dateiteilbäume.

Dazu habe ich mir unterhalb des Eingangsverzeichnisses `.../texmf/source` zunächst entsprechende Unterverzeichnisse eingerichtet, deren Namen alle mit dem Anhang `./src` gekennzeichnet sind, wie die vorab genannten Unterverzeichnisse `./tex.src/` und `./latex.src/`. Weitere Beispiele für bei mir eingerichtete Anfangsverzeichnisse für Quellenverzeichnisbäume sind `./dvips.src/`, `./makeindx.src/` u. ä.

Mit dem Entpacken und/oder Entarchivieren entstehen unterhalb der einzelnen Anfangsverzeichnisse die ursprünglichen Dateienstrukturen mit ihren eigenen Verzeichnisbäumen. Diese müssen regelmäßig gewartet werden, um unnötige Plattenspeicherbelegungen zu vermeiden. Viele Files aus den entpackten und/oder entarchivierten Dateien werden für ein arbeitsfähiges T<sub>E</sub>X-System in anderen Teilen des Filesystems, z. B. unterhalb der bereits vorgestellten Eingangsverzeichnisse `.../texmf/tex/`, `.../texmf/metafont/`, `.../texmf/fonts/` u. a., erwartet. Werden sie dorthin lediglich kopiert, so können sie anschließend zwar ordnungsgemäß genutzt werden, bleiben aber zusätzlich im Quellenverzeichnis erhalten.

Um eine Vorstellung von der Größenordnung zu erhalten, um die es hierbei zum Teil gehen kann, demonstriere ich dies am Beispiel des T<sub>E</sub>X-Installationspakets für UNIX. Das übertragene archivierte und komprimierte Paket `dante-tex.tar.Z` belegt anfänglich gut 13 Mbyte.

Nach dem Entkomprimieren wächst es als `dante-tex.tar` auf gut 37 Mbyte. Mit dem Entarchivieren belegt das wiederhergestellte Original-Dateiensystem weitere 38 Mbyte. Bei der Aufbereitung der Programm-Quellenfiles mittels der beigefügten Konfigurations- und Installationshilfen (`configure.sh` und `Makefile`) wächst das Original-Dateiensystem dann bereits auf mehr als 60 Mbyte. Von den bei der Aufbereitung entstehenden zusätzlichen Files werden nur die ausführbaren Programme, die Format- und Basisfiles sowie die UNIX-Manual-Dokumentationsfiles benötigt.

Bei der Kompilierung der Original-Quellenfiles entstehen viele hundert Zwischenfiles im Objektcode, die anschließend nicht wieder benötigt werden. Zum Glück enthalten die meisten beigefügten Installationshilfen eine Säuberungsoption, so dass mit dem Aufruf `'make clean'` o. ä. das Gesamtsystem von temporär angelegten Zwischenfiles befreit werden kann.

Nach der erfolgreichen Installation und Aufbereitung von `dante-text.tar.Z` unterhalb des Quellenverzeichnisses `.../texmf/source/tex.src` habe ich alle Files, die das arbeitsfähige T<sub>E</sub>X- und METAFONT-System in anderen Zielverzeichnissen erwartet, dorthin verschoben. Die ausführbaren Zusatzprogramme (zusätzlich zu den Systemprogrammen) erwartet mein UNIX-System unter `/usr/local/bin` bzw. unter `/usr/local/gnu/bin`. Die für das UNIX-T<sub>E</sub>X-System entstandenen ausführbaren Programme habe ich bei mir in das erstere Unterverzeichnis verschoben. Zusätzliche UNIX-Manualdokumentation erwartet mein System unter `/usr/local/man`. Dorthin habe ich die erzeugten T<sub>E</sub>X-UNIX-Manualfiles verschoben. Weitere Dokumentationsfiles, die zum UNIX-T<sub>E</sub>X-System gehören, habe ich in geeignete Unterverzeichnisse unterhalb von `.../texmf/doc (s.u)` verschoben.

Das entkomprimierte Archiv `dante-tex.tar` habe ich anschließend wieder komprimiert, auf eine DAT-Kassette kopiert und dann auf meiner Festplatte gelöscht. Im bereinigten Quellenverzeichnis `.../texmf/source/tex-src` habe ich letztlich nur die WEB-Quellenfiles und ihre Systemanpassungen, die originären C-Programmfiles mit den zugehörigen Include-Dateien und die beigefügten Installationsdateien behalten. Diese belegen knapp fünf Mbyte gegenüber den während der Installation belegten 100 Mbyte. In gleicher Weise verfare ich mit allen Ausgangspaketen des DANTE-Fileservers.

## 1.2.6 Das Dokumentations-Filesystem

Viele Pakete auf den T<sub>E</sub>X-Fileservern enthalten beigefügte Dokumentationsfiles oder können solche eigenständig erstellen und/oder aufbereiten. Der Vorschlag der TDS-Gruppe für das T<sub>E</sub>X-Filesystem enthält hierfür ein eigenes Eingangsverzeichnis `.../texmf/doc`.

Dieses Eingangsverzeichnis wird unterschiedlich tief untergliedert. Als erste Ebene werden Unterverzeichnisse auftreten, die weitgehend der Eingangsebene unterhalb von `.../texmf (s. S. 7)` entsprechen und teilweise wie diese weiter untergliedert sind. Unterhalb von `.../texmf/doc/tex` habe ich bei mir `./plain/`, `./latex/`, `./amstex/`, `./generic/` u. a. eingerichtet. `./latex` enthält dann weiter die Unterverzeichnisse `./base/`, `./packages/` und `./contrib/`. Unterhalb von `./base/` habe ich die L<sup>A</sup>T<sub>E</sub>X-Dokumentationsfiles `usrguide.tex`, `clsguide.tex`, `cnfguide.tex`, `ftnguide.tex`, `modguide.tex` und alle `ltnews*.tex`-Files abgelegt und mit L<sup>A</sup>T<sub>E</sub>X bearbeitet. Die entstandenen `.dvi`-Files wurden gleichermaßen hier abgelegt und können bei Bedarf als Preview durchmustert oder über den Druckertreiber ausgegeben werden. Sie stellen Einführungsmanuale dar, die für alle L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-Anwender von Nutzen sind.

Das L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-Grundpaket enthält das File `source2e.tex`. Dieses habe ich im Originalverzeichnis mit L<sup>A</sup>T<sub>E</sub>X bearbeitet und das entstandene File `source2e.dvi` dann ebenfalls nach

.../texmf/doc/tex/latex/base verschoben. Sein Ausdruck erzeugt die 525-seitige Programmdokumentation des gesamten L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-Kerns.

Unter .../texmf/doc/tex/plain habe ich das File `texbook.tex` abgelegt, das man auf den öffentlichen T<sub>E</sub>X-Fileservern unter `/tex-archive/systems/knuth/tex` findet. Entsprechend habe ich das File `mfbook.tex` aus `/tex-archive/systems/knuth/mf` nach .../texmf/doc/metafont/plain kopiert. Diese beiden Files enthalten jeweils den Eingabetext für die beiden Bücher von Donald E. Knuth, ‘The T<sub>E</sub>Xbook’ [10a] und ‘The METAFONTbook’ [10b], die man nach einer T<sub>E</sub>X-Bearbeitung auf dem eigenen Drucker ausgeben könnte. Mir ist nicht ganz klar, welche Copyright-Einschränkungen hierbei zu beachten sind.<sup>3</sup> Nach der in der Fußnote wiedergegebenen, sehr restriktiven Bedingung scheint selbst ein privater einmaliger Ausdruck einen Copyright-Verstoß darzustellen. Andererseits frage ich mich, was den Autor bewogen haben mag, die originären Erzeugungsfiles dem T<sub>E</sub>X-System beizufügen, wenn selbst ihr Preview verboten sein sollte.

Weitere Unterverzeichnisse unter .../texmf/doc, wie `./bibtex/`, `./makeindx/`, `./fonts/` u. a. sind zur Aufnahme von Dokumentationsfiles aus diesen Paketen gedacht. Je nach Umfang der Dokumentationen möge der Anwender oder Systembetreuer diese Eingangsverzeichnis ggf. eigenständig sachgerecht untergliedern.

### 1.2.7 Das BIBT<sub>E</sub>X-Filesystem

Das Eingangsverzeichnis .../texmf/bibtex ist zur Aufnahme der Files gedacht, die das Programm BIBT<sub>E</sub>X zur Erstellung von Literaturverzeichnissen aus Bibliographie-Datenbanken benötigt. Es sollte dem nachfolgenden Diagramm entsprechend untergliedert werden:

```
./bibtex/
  bib/
    base/
    local/
    misc/
  bst/
    base/
    local/
    misc/
```

Das erste Unterverzeichnis `./bib/` dient zur Aufnahme von Bibliographie-Datenbanken, die durch den Anhang `.bib` gekennzeichnet sind. Das zweite Unterverzeichnis `./bst` nimmt die Stilfiles für das Programm BIBT<sub>E</sub>X auf, die den Anhang `.bst` tragen.

Diese beiden Eingangsverzeichnisse sind jeweils für sich in die Unterverzeichnisse `./base/`, `./local` und `./misc` gegliedert. `./base` nimmt die Bibliographie-Datenbanken bzw. die Stilfiles auf, die dem BIBT<sub>E</sub>X-Programmpaket beigelegt sind. `./local` dient zur Aufnahme von Bibliographie-Datenbanken bzw. BIBT<sub>E</sub>X-Stilfiles, die beim Anwender lokal erstellt wurden.

Das jeweils dritte Unterverzeichnis `./misc` ist zur Aufnahme von Bibliographie-Datenbanken bzw. BIBT<sub>E</sub>X-Stilfiles gedacht, die weder unter `./base` noch unter `./local` einzugliedern sind.

Die Erstellung von eigenen Bibliographie-Datenbanken wurde bereits [5a, Anh. B] beschrieben. Dieser Band enthält mit Abschnitt 6.8.2 Hinweise, wie weitere BIBT<sub>E</sub>X-Stilfiles zu erstellen sind, um Literaturverzeichnisse nach den Forderungen des Anwenders oder seines Auftraggebers zu gestalten. Die dem BIBT<sub>E</sub>X-Originalpaket beigelegte Dokumentation ist, entsprechend dem vorangegangenen Unterabschnitt, unter .../texmf/doc/bibtex abzuliegen.

---

<sup>3</sup>Die Copyright-Vermerke in den Originalbüchern enthalten die Angabe “No part of this publication may be reproduced, stored in retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publishers.”

## 1.2.8 Weitere Verzeichnisstrukturen des T<sub>E</sub>X-Filesystems

Das Diagramm für die Eingangsebene des T<sub>E</sub>X-Filesystems auf S. 7 enthält weitere Eingangsverzeichnisse, die dort symbolisch mit `.../texmf/progr/` angeführt werden. Hiermit sind in sich abgeschlossene zusätzliche Programmpakete einer T<sub>E</sub>X-Installation gemeint, z. B. `makeindx`, `dvips` u. a. In diesen Verzeichnissen sollen diejenigen Files aus den genannten Paketen abgelegt werden, die deren ausführbare Programme bei der Bearbeitung zusätzlich einlesen. Dies sind im Wesentlichen Konfigurations- und sonstige Datenfiles. Nicht hierher gehören die diesen Paketen zugeordneten Quellen- und Dokumentationsfiles, da diese entsprechend 1.2.5 und 1.2.6 im Quellen- bzw. Dokumentations-Filesystem abzulegen sind.

Die Entscheidung, welche Files hier abzulegen sind, kann der Anwender oder Systemverwalter am ehesten treffen, wenn er meiner Installationspraxis folgt, wie sie in 1.2.5 für `dante-tex.tar` mit `./tex.src` beschrieben wurde. Das Ergebnis wird sein, dass für das Eingangsverzeichnis `./texmf/makeindx` kaum Files aus dem MakeIndex-Installationspaket übrig bleiben. Zwar enthält das Originalpaket einige Stilfiles wie `makeidx.sty` und `showidx.sty`. Diese gehören als L<sup>A</sup>T<sub>E</sub>X-Stilfiles jedoch in das Verzeichnis `.../texmf/tex/latex/misc`. Andererseits entstehen bei der Installation von L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> die speziellen MakeIndex-Stilfiles `gind.ist` und `gglo.ist`, die umgekehrt unter `.../texmf/makeindx/` abzulegen sind, ebenso wie evtl. weitere MakeIndex-Stilfiles, über die der Anwender oder sein Rechenzentrum verfügt.

Das Eingangsverzeichnis für die DVI-Treiber sollte den Namen `.../texmf/dviware/` tragen, das dann für die verschiedenen Treiber wie `dvips`, `dvilj`, `xview` u. ä. in entsprechende Unterverzeichnisse gleichen Namens zu untergliedern ist. Kommt nur ein Druckertreiber zur Anwendung, dann könnte `./dviware` entfallen und durch das Treiberverzeichnis ersetzt werden. Ich rate davon ab, da neben dem Druckertreiber ein DVI-Treiber als Previewer verfügbar sein sollte, womit `./dviware` bereits in zwei Unterverzeichnisse aufzugliedern ist.

## 1.2.9 Die ausführbaren T<sub>E</sub>X-Programme

Der TDS-Vorschlag für die Eingangsebene des T<sub>E</sub>X-Filesystems enthält als letztes Eingangsverzeichnis `.../texmf/bin`, das zur Aufnahme der ausführbaren Programme und evtl. Befehlsdateien gedacht ist. In den Erläuterungen zum TDS-Vorschlag wird es als Alternative empfohlen, falls beim benutzten Betriebssystem die ausführbaren binären Programme nicht in einem anderen Verzeichnis vorgehalten werden.

Ich bin auf meinem PC dem TDS-Vorschlag gefolgt, indem ich das emT<sub>E</sub>X-Paket entsprechend diesem Vorschlag neu gegliedert habe. Entsprechend erscheint beim Setzen der PATH-Variablen innerhalb von `autoexec.bat` bzw. `config.sys` die Angabe `...;c:\texmf\bin;...`

Auf meiner UNIX-Workstation werden lokale ausführbare Programme traditionell unter `/usr/local/bin` und `/usr/local/gnu/bin` erwartet, während sonstige lokale Dateien unter dem Eingang `/usr/local/lib` mit ihren eigenen Verzeichnisbäumen eingerichtet werden. Hier erschien mir ein Verzeichnisname `./bin` unterhalb eines Oberverzeichnisses `./lib` als ein Syntax-Stilbruch. Auf meiner UNIX-Workstation wurden deshalb alle ausführbaren Programme aus dem T<sub>E</sub>X-Gesamtsystem, einschließlich etwaiger Shell-Skripten, unter `/tex/local/bin` abgelegt. Das Verzeichnis `./bin` entfällt hier im T<sub>E</sub>X-Filesystem.


## 1.3 Die Zeichensatz-Filetypen

Auf Grund gelegentlicher Leseranfragen und Problembeschreibungen auf den  $\text{T}_{\text{E}}\text{X}$ -Diskussionslisten habe ich den Eindruck gewonnen, dass manchen  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Anwendern die verschiedenen Zeichensatz-Filetypen nicht ganz klar sind. Hier folgt deshalb noch einmal eine Beschreibung der verschiedenen  $\text{T}_{\text{E}}\text{X}$ -Zeichensatz-Filetypen.

### 1.3.1 Die Zeichensatz-Metrikfiles

Zur Bearbeitung eines Textes, bei dem verschiedene Schriften zur Anwendung kommen, greift  $\text{T}_{\text{E}}\text{X}$  während der Formatierung *nur* auf die sog. Zeichensatz-Metrikfiles zurück. Das sind diejenigen Zeichensatzfiles, die durch den Anhang `.tfm` ( $\text{T}_{\text{E}}\text{X}$  Font Metric) gekennzeichnet sind.

Die `.tfm`-Files enthalten für jedes Zeichen des betrachteten Zeichensatzes dessen Abmessungen, also seine Höhe, Tiefe und Breite. Höhe und Tiefe eines Zeichens kennzeichnet die Abmessung, um die sich das Zeichen oberhalb bzw. unterhalb seiner Grundlinie erstreckt. Die Breite steht für die fiktive Weite des Zeichens, so dass mehrere Zeichen mit diesen Weiten nebeneinander gestellt werden, ohne sich zu berühren.

Über den grafischen Inhalt der Zeichen enthält das `.tfm`-File keine weitere Information. Für die Textformatierung werden die einzelnen Zeichen als leere Kästchen betrachtet, die mit ihren natürlichen Abmessungen nebeneinander gestellt werden. Die Eingabe ‘ganz typisch’ erscheint bei der  $\text{T}_{\text{E}}\text{X}$ -Bearbeitung intern nur als .

Die einzelnen Zeichen werden normalerweise mit ihren natürlichen Weiten, wie sie im `.tfm`-File angegeben sind, nebeneinander gestellt. Für bestimmte Buchstabenkombinationen wie `fi` oder `ffl` werden diese durch ein jeweils eigenes Gemeinschaftszeichen als Ligatur und damit als ‘fi’ und ‘ffl’ gesetzt, wohingegen die nebeneinander gestellten Einzelzeichen als ‘fi’ bzw. ‘ffl’ erscheinen würden.

Bei anderen Zeichenkombinationen werden die Einzelzeichen nicht unter Verwendung ihrer natürlichen Weiten, sondern durch Unter- oder Überschneidungen näher oder weiter zusammengefügt, wie bei ‘AV’ statt ‘AV’ deutlich zu erkennen ist. Die `.tfm`-Files enthalten für alle Zeichenkombinationen, die durch Ligaturen ersetzt oder mit Unterschneidungen nebeneinander gestellt werden, eine sog. Ligatur- und Kerningliste. Treten solche Zeichenkombinationen im Text auf, so berücksichtigt  $\text{T}_{\text{E}}\text{X}$  die Angaben aus diesen Listen.

Die `.tfm`-Files enthalten weiterhin noch Informationen, die für alle Zeichen des Zeichensatzes gemeinsam gelten. Hierzu gehört die Entwurfsgröße des Zeichensatzes, die Angabe der Neigung bei geneigten Schriften, die Angabe des Sollwertes für den normalen Wortabstand sowie für dessen maximalen Dehn- oder Schrumpfwert, die Höhe des kleinen ‘x’ sowie den Zusatzabstand, der nach einem Punkt als Satzende zusätzlich eingefügt wird.

Mathematische Zeichensätze enthalten noch einige allgemeine Zusatzinformationen, wie den Betrag der Verschiebung von Zähler und Nenner eines Bruchs sowie den Betrag der Hoch- und Tiefstellungen bei Exponenten und Indizes.

Schließlich enthalten die `.tfm`-Files bei geneigten Schriften für jedes einzelne Zeichen noch den Betrag der sog. Italic-Korrektur, also den Zusatzabstand, der nach Übergang von einer geneigten auf eine aufrechte Schrift zugefügt werden sollte.

Die `.tfm`-Files enthalten diese Informationen in sehr kompakter Form. Es handelt sich teilweise um Bitfelder, die kürzer als ein Byte sind und kompakt nebeneinander stehen.  $\text{T}_{\text{E}}\text{X}$  kennt das interne Format der `.tfm`-Daten und vermag es zu verarbeiten. Mit dem Editor

können .*tfm*-Files als binäre Files nicht gelesen werden. Man könnte sie als oktalen oder hexadezimalen Dump ausgeben, doch verlangt dessen Interpretation vertiefte Kenntnisse über das Format der .*tfm*-Datenstrukturen.

Zum T<sub>E</sub>X-Gesamtpaket gehört das Umwandlungsprogramm *tfptop*, mit dem .*tfm*-Files in eine lesbare und leichter interpretierbare Form umgewandelt werden können. Dieses Umwandlungsprogramm (und sein Rückwandelungs Pendant *ptotf*) wird in Anhang B.3.4 vorgestellt.

Zeichensätze sollen für eine T<sub>E</sub>X-Bearbeitung häufig vergrößert oder verkleinert werden. Hierzu wird kein weiteres .*tfm*-File benötigt. T<sub>E</sub>X verwendet die .*tfm*-Files nur für die Zeichensätze in der Entwurfsgröße. Die Vergrößerung bzw. Verkleinerung nimmt T<sub>E</sub>X selbst vor, indem alle Zeichensatzabmessungen aus dem .*tfm*-File mit dem verlangten Skalierungsfaktor einfach multipliziert werden.

Trotz des relativ großen Informationsgehalts sind die .*tfm*-Files wegen der sehr kompakten Datenformate erstaunlich klein. Typische Werte für die T<sub>E</sub>X-cm-Standardschriften liegen bei 1000–1500 Byte. Selbst bei den erweiterten dc-Schriften mit 256 Zeichen pro Zeichensatz sind Filegrößen mit mehr als drei Kbyte die Ausnahme.

Demzufolge werden die .*tfm*-Files für eine große Zahl von Zeichensätzen bereits bei der Installation eingerichtet und nicht erst bei Bedarf mit METAFONT generiert. .*tfm*-Files für einige Hundert Zeichensätze sind keine Seltenheit. Ihre Gliederung wurde in 1.2.4 vorgestellt. Auf den öffentlichen T<sub>E</sub>X-Fileservern findet man die fertigen .*tfm*-Files für alle nur erdenklichen Schriften. Man kann sie sich von dort in Paketgruppen kopieren und im Zeichensatzfilessystem unter `.../texmf/fonts/tfm` aufteilen, auch wenn viele von ihnen kaum zur Anwendung kommen. Wegen ihrer Kompaktheit hält sich der Plattenspeicherbedarf in Grenzen. Die vorsorgliche Installation dieser Files verlangt vermutlich weit weniger Zeitaufwand, als sie erst bei Bedarf jeweils einzeln zu generieren oder zu beschaffen und erst dann in das T<sub>E</sub>X-Filesystem einzubinden.

### 1.3.2 Druckerzeichensätze

Für die Druckausgabe oder das Preview werden die grafischen Muster für die Zeichen des Zeichensatzes benötigt. Die interne Beschreibung der grafischen Muster durch sog. Pixel oder Bitmaps wurde bereits in Band 1 dieser Buchserie [5a] in den dort enthaltenen Anhängen C.3.7 und C.3.8 ausführlich beschrieben, auf die bei Bedarf zurückgegriffen werden sollte.

Die Zeichensatzfiles für die Druckerzeichensätze im gepackten Bitmap-Format werden durch den Namensanhang *.pk* oder, wenn der Namensanhang mehr als drei Zeichen zulässt, durch *.nnn.pk* gekennzeichnet, wobei *nnn* für eine mehrstellige Zahl steht, die die *fiktive* Auflösung widerspiegelt. Während für die T<sub>E</sub>X-Bearbeitung die dort benötigten .*tfm*-Metrikfiles für jeden Zeichensatz nur einmal vorgehalten werden, unabhängig davon, mit welcher Skalierung (Vergrößerung oder Verkleinerung) und auf welchem Ausgabegerät die Ausgabe erfolgen soll, gilt das für die Druckerzeichensätze nicht.

Für jede Skalierungsstufe eines Zeichensatzes ist ein eigenes *.pk*-Zeichensatzfile zu generieren, da die zugehörigen grafischen Muster, also ihre Bit- oder Pixelmaps stark von der Skalierungsstufe abhängen und nicht einfach aus den Bitmustern der Zeichensätze in ihrer Entwurfsgröße abgeleitet werden können. Es ist gerade die Aufgabe des METAFONT-Programms, für jede Skalierungsstufe das zugehörige Pixelmuster optimal zu erstellen.

Ebenso verlangen Drucker unterschiedlicher Auflösung ihre jeweils eigenen Drucker-Zeichensatzfiles. Selbst für Drucker gleicher Auflösung, aber unterschiedlicher Herkunft

kann es notwendig sein, jeweils druckerspezifische Zeichensatzfiles zu generieren. Trotz gleicher Auflösung können sich die Einzelpixel bei Druckern verschiedener Herkunft in ihrem Schwärzungsgrad deutlich unterscheiden. Abschnitt 8.1.3 aus Band 2 dieser Buchserie beschreibt die Details. Zum METAFONT-Gesamtpaket gehört das File modes .mf, das für eine große Zahl von Druckern die speziellen Einstellvorgaben enthält. Findet man dort für Drucker gleicher Auflösung, aber unterschiedlicher Herkunft unterschiedliche Einstellvorgaben, dann sollten für die zugehörigen Drucker die erforderlichen Druckerzeichensätze separat erstellt werden.

Der TDS-Vorschlag für das Zeichensatzfilesystem enthält deshalb unter der Eingangsebene `.../texmf/fonts/pk/` als nächste Ebene die Untergliederung nach Druckertypen `<drucker_typ>/` und als unterste Ebene zusätzlich nochmals eine Untergliederung nach den verschiedenen Skalierungsstufen `dpi<nnn>/`.

Nach diesen Bemerkungen wird es verständlich, dass für die endgültige Druckausgabe meist sehr viel mehr .pk-Files benötigt werden, als an .tfm-Files zur  $\TeX$ -Bearbeitung vorzuhalten sind. Die .pk-Zeichensatzfiles sind überdies deutlich größer als die .tfm-Zeichensatzfiles. Deshalb wird von einer Voraberstellung für die .pk-Zeichensatzfiles meistens abgesehen, evtl. mit Ausnahme der aus Erfahrung abgeschätzten, am häufigsten verwendeten Druckerzeichensätze.

Zusätzliche Drucker-Zeichensatzfiles werden erst bei ihrem tatsächlichen Bedarf generiert und unter dem zugehörigen Zielverzeichnis gemäß 1.2.4 abgelegt. Es gibt DVI-Treiber, wie `dvips` von Tomas Rokicki oder die Treiber aus dem  $\text{em}\TeX$ -Paket von Eberhard Mattes, die dies automatisch erledigen. Hier muss lediglich bei der Installation die Struktur des Zeichensatz-Filesystems mitgeteilt oder mit einem Konfigurationsfile bekannt gemacht werden. Entsprechende Hinweise sind der jeweiligen Installationsdokumentation zu entnehmen.

Das METAFONT-Programm erstellt aus den Zeichensatzquellenfiles zunächst Druckerzeichensätze in einem speziellen Format, das durch den Anhang .gf oder .nngf gekennzeichnet wird. Das Umwandlungsprogramm `gftopk`, das Bestandteil eines METAFONT-Installationspakets ist, erstellt aus diesen .gf-Files die endgültigen .pk-Files. Da alle mir bekannten DVI-Treiber das .pk-Format verarbeiten können, werden die .gf-Files meistens nur als temporäre Files erzeugt und nach der endgültigen Umwandlung in ihre .pk-Äquivalente wieder gelöscht. Dies ist sinnvoll, da die .gf-Zeichensatzfiles deutlich größer – oft mehr als doppelt so groß – als die äquivalenten .pk-Files sind.

Auch die .pk-Files sind in einem kompakten Format binär kodiert, so dass sie mit einem Texteditor nicht eingesehen werden können. Es gibt jedoch das Aufbereitungsprogramm `pktype`, mit dem .pk-Files in eine sichtbare und leicht interpretierbare Form umgewandelt werden können. `pktype` wird in diesem Buch in Anhang B.4.6 vorgestellt und erläutert.

### 1.3.3 Virtuelle Zeichensätze

Ab  $\TeX$  3.0 gibt es die Möglichkeit der Verwendung *virtueller* Zeichensätze. Zeichensatzfiles für virtuelle Zeichensätze sind durch den Anhang .vf gekennzeichnet. Zu jedem virtuellen Zeichensatz gibt es ein zugeordnetes .tfm-Metrikfile, das sich durch nichts von den .tfm-Files für die *realen* Zeichensätze unterscheidet. Bei der  $\TeX$ -Bearbeitung entnimmt  $\TeX$  dem Metrikfile an den von  $\TeX$  vorausgesetzten Positionen für die Zeichensatzanordnung diese Zeichen und behandelt sie mit ihren dort angegebenen Abmessungen.

Gleichnamige Druckerzeichensätze, genauer gleichnamige .pk-Zeichensatzfiles, fehlen dagegen. Stößt der DVI-Treiber bei einem angeforderten Druckerzeichensatz auf diese Si-

tuation, so wird ersatzweise nach einem File mit dem angeforderten Grundnamen und dem Anhang `.vf` gesucht. Dieser sog. virtuelle Zeichensatz enthält nun seinerseits die Hinweise darauf, aus welchem realen Zeichensatz das angeforderte Zeichen zu entnehmen und wo es dort positioniert ist. Dies macht es möglich, reale Zeichensätze mit einer anderen Zeichenbelegung oder einer Kombination aus mehreren Zeichensätzen unter einem einheitlichen Namen anzufordern.

Die Nutzung und Ausgabe von PostScript-Schriften für eine T<sub>E</sub>X-Bearbeitung und ihre anschließende Druckausgabe erfolgen über entsprechende virtuelle Zeichensätze. PostScript-Zeichensätze haben gegenüber den cm-T<sub>E</sub>X-Standardschriften einen teilweise unterschiedlichen Zeichenvorrat, wobei die Positionierung der Zeichen eines PostScript-Zeichensatzes teilweise von derjenigen der cm-Zeichensätze abweicht.

Soll z. B. ein Text mit dem Ergänzungspaket `times.sty` bearbeitet werden, so werden die bisherigen cm-Schriften durch die PostScript-Schriften der Times-Roman-, Helvetica- und Courier-Familie ersetzt. Bei der T<sub>E</sub>X-Bearbeitung werden dabei für die angeforderten PostScript-Schriften die zugehörigen `.tfm`-Files eingelesen, die sich von den entsprechenden `.tfm`-Files der cm-Zeichensätze nur durch die Maßangaben für die einzelnen Zeichen, nicht aber durch ihren sonstigen internen Aufbau unterscheiden. Für die aufrechte Times-Roman-Standardschrift lautet der Filename für das zugehörige Metrikfile `ptmr7t.tfm`.

Bei den cm-Textschriften stehen auf den Positionen `"00–"0a` (`"z` steht für die hexadezimale Zahlenangabe) die griechischen Großbuchstaben  $\Gamma$   $\Delta$   $\Theta$   $\Lambda$   $\Xi$   $\Pi$   $\Sigma$   $\Upsilon$   $\Phi$   $\Psi$  und  $\Omega$ . Hierauf folgen auf den Positionen `"0b–"0f` die Ligaturen `ff`, `fi`, `fl`, `ffi` und `ffl`. Das  $\beta$  und die skandinavischen Sonderbuchstaben  $\ae$ ,  $\o$ ,  $\mathcal{A}$ ,  $\mathcal{E}$  und  $\mathcal{O}$  stehen innerhalb der cm-Zeichensätze an den Plätzen `"19–"1f`. Bei der T<sub>E</sub>X-Bearbeitung werden diese Zeichen an den gewohnten Stellen erwartet und mit ihren geänderten PostScript-Abmessungen berücksichtigt.

Innerhalb der PostScript-Standardzeichensätze treten griechische Großbuchstaben nicht auf. Als Ligaturen existieren nur `fi` und `fl`. Die Zeichen  $\beta$ ,  $\ae$ ,  $\o$ ,  $\mathcal{A}$ ,  $\mathcal{E}$  und  $\mathcal{O}$  stehen hier an den Stellen `"fb`, `"f0`, `"fa`, `"f9`, `"e0`, `"ea` und `"e9`. Für die Druckausgabe wird nun zunächst das virtuelle File `ptmr7t.vf` (für die aufrechte Times-Roman-Standardschrift) eingelesen. Dieses enthält nun an der Position `"19` den Hinweis, dass das dort von T<sub>E</sub>X erwartete Zeichen  $\beta$  der PostScript-Times-Roman-Standardschrift zu entnehmen ist, sowie die Information, dass es dort auf Platz `"fb` steht. Entsprechendes gilt für die skandinavischen Sonderzeichen.

Wird von T<sub>E</sub>X die Ligatur `ff` angefordert, die bei den cm-Textschriften auf Platz `"0b` steht, so enthält das virtuelle Zeichensatzfile an dieser Stelle den Hinweis, dass die Ligatur `ff` durch zwei nebeneinander gestellte `f`'s aus dem PostScript-Zeichensatz zu ersetzen ist, wobei die beiden `f`'s jedoch näher zusammengedrückt werden, als es ihrer natürlichen Zeichenbreite entspricht. Auf diese Weise wird die fehlende Ligatur näherungsweise nachgebildet.

Wird einer der oben aufgelisteten griechischen Großbuchstaben angefordert, so enthält das virtuelle File `ptmr7t.vf` an den Originalpositionen den Hinweis, dass das angeforderte Zeichen aus einem anderen PostScript-Zeichensatz, der mit `psyr0` (für Symbol) gekennzeichnet ist, zu entnehmen ist, sowie die Information darüber, an welchen Positionen es sich dort befindet. Für weitere Hinweise zu den virtuellen Files verweise ich auf die Abschnitte 4.2.4, 4.2.5 und 4.2.7 in Band 2 [5b] dieser Buchserie.

Auch die virtuellen Zeichensatzfiles enthalten ihre Information in sehr kompakter binärer Form, so dass sie mit einem Text-Editor nicht eingesehen werden können. Mit dem Programm `vftovp` können `.vf`-Files für eine lesbare und leicht interpretierbare Form aufbereitet werden. Dieses Programm wird in Anhang B.3.5 dieses Buches näher vorgestellt.