

Fontdateien für T_EX

Javier Bezos*

19. September 1999

Dieser Artikel faßt, mit einigen Korrekturen, eine Reihe von fünf Beiträgen zusammen, die als Antworten auf eine Frage eines Abonnenten an die Spanish-TeX-Mailing list geschickt wurden.

1. Fonts

Dieser erste Abschnitt ist den verschiedenen Formaten, in denen Fonts definiert werden, gewidmet, d.h. den Dateien, in denen die Form der Zeichen beschrieben ist.

MetaFont

Das älteste, aber dennoch raffinierteste und perfekteste von allen im Hinblick auf typographisches Design. Es wurde von Knuth selbst geschaffen, und von ihm haben alle gelernt. Seine Hauptnachteile sind, daß nur T_EX es versteht und daß es notwendig ist, vorher die Dateien mit dem Bild, so wie es zum Drucker geschickt wird, in der korrekten Auflösung zu erzeugen (mit dem damit verbundenen Verbrauch an Speicherplatz).

- .mf** [MetaFont-Code (Text)] Enthält den MetaFont-Code, der mittels eines Programms die Form der Zeichen beschreibt. Oft benötigt ein bestimmter Font mehr als eine mf-Datei.
- .pk** [binär] Enthält die bereits erzeugten Zeichen (in verschiedenen Auflösungen: .600pk, .300pk usw. Die genaue Dateinamenerweiterung ist systemabhängig.)
- .gf** [binär] Dies ist ein Zwischenschritt in mf → gf → pk und nicht von grundlegender Wichtigkeit.

PostScript

Der Industriestandard. Jeder Font in jeder seiner Varianten Fettdruck, Kursiv ... hat nur eine Datei mit dem Zeichenprogramm für seine Zeichen; die Erzeugung der Zeichen besorgt der Drucker selbst während des Drucks. Seine Vorteile sind seine hervorragende Qualität, die Speicherplatzersparnis und die fast grenzenlose Veränderbarkeit.

- .pfa/pfb** [Text/binär] Enthält das bereits kompilierte Programm, das der Drucker benutzt. Der Dateiname entspricht nicht dem Fontnamen; der Name, der auf Betriebssystem- und Programmebene sichtbar ist, kann anders sein und ist in der Regel länger und genauer¹. Die

* jbezos@arrakis.es. Copyright 1999 Javier Bezos. All rights reserved. Distributed under the terms of the L^AT_EX Project Public License. Erhältliche Packages von diesem Autor: accents, polyglot, gloss, tensind, esindex, titlesec, titletoc. — Deutsche Übersetzung: Thomas Ruedas, ruedas@geophysik.uni-frankfurt.de, 1999. Inhaltliche Fragen bitte auf spanisch oder englisch an den Autor des Originaltextes.

¹Man kann den Fontnamen ermitteln, indem man die Datei mit einem Editor wie emacs oder Alpha öffnet; er steht üblicherweise in der ersten Zeile, und zwar mit einem vorangestellten Schrägstrich hinter /FontName

.pfa- und .pfb-Dateien unterscheiden sich in ihrer internen Form, sind aber gleichwertig. Welche benutzt werden muß, ist im Druckerhandbuch nachzusehen.

- **.afm** [Text] Lesbare Information über den Font. Sie ist nicht nötig, außer wenn wir uns eingehender mit der Manipulation von Fonts beschäftigen wollen.
- **.pfm** [binär] Eine Windows-spezifische binäre Version von afm.

TrueType

Dieses Format wurde von Apple geschaffen, um technisch unabhängig von den PostScript-Fonts von Adobe zu sein, aber seine Qualität stellte sich als unterlegen heraus, und das Ganze endete in einem Fiasko. Wenn Microsoft es nicht gekauft hätte, wäre es wahrscheinlich heute verschwunden. Sein zweiter Nachteil ist, daß es heutzutage nur in der nicht-professionellen Windows-Welt ernstgenommen wird, und daher nehmen es nur wenige T_EX-Implementationen an, darunter, als erwähnenswerte Ausnahme, pdftex.

- **.ttf** [binär] Das Pendant zu .pfa/.pfb.

Auf Mac-Systemen sind die Dateinamenerweiterungen nicht notwendig und daher nicht immer vorhanden. Obwohl dieses System die Fonts so akzeptiert, wie sie auf anderen Systemen verwendet werden, wird PostScript in einem Format mit dem Namen `llwf` bevorzugt.

Wir kommen nun zur ersten „Goldenen Regel“:

Um einen Font zu benutzen, muß man eine Datei mit diesem Font im Format `pk`, `ttf` oder `pfa/pfb` haben.

Die einzige Ausnahme ist `pk`, das sich mit MetaFont aus den `mf`-Dateien erzeugen läßt. (MetaFont wird, abgesehen von einigen wenigen Ausnahmefällen, immer mit T_EX mitgeliefert.) Ohne einen Font kann man schlicht und ergreifend nicht weitermachen.

2. Die Metriken

Grunddateien

Dieser Abschnitt beschäftigt sich mit der einzigen Datei, die T_EX braucht, um ein Dokument zu erstellen.

T_EX weiß nichts über Zeichen: das einzige, was es interessiert, sind ihre Maße, um ihnen den entsprechenden Platz auf der Seite zur Verfügung zu stellen; ihre Form, ihre Natur oder sogar die Stelle (Datei), wo sie sich befinden, sind ihm vollkommen gleichgültig. In gewisser Weise ist jedes Zeichen in einem Dokument nicht mehr als ein Befehl, der T_EX mitteilt: „reserviere den Platz, den der aktuelle Font für meinen [ASCII-]Code angibt“².

- **.tfm** [binär] Dies ist die Namensweiterung der Dateien, die die Information über die Maße der Zeichen eines Fonts – zusammen mit einigen anderen Angaben – enthalten. Ohne die zugehörige `.tfm`-Datei kann T_EX nicht fortfahren und gibt eine Fehlermeldung, daß es die Metrik nicht findet.

²Die Fontdateien mit einer Erweiterung, die mit `o` beginnt, sind von einer T_EX-Variante namens Omega: `opl`, `ovf`, `ofm` usw.

- .pl [Text] Das für Menschen lesbare Pendant einer tfm-Datei. Beide können auf einfache Weise mit tftopl und pltotf ineinander umgewandelt werden, aber ihr Inhalt kann etwas obskur aussehen. Es gibt eine Erweiterung von .pl namens .vp1, auf die wir im folgenden Abschnitt zurückkommen werden, aber normalerweise werden sie nicht benutzt.
- .afm [Text] Diese Datei haben wir schon im ersten Abschnitt gesehen: Es ist eine Beschreibung im Stil von .pl, die zu den PostScript-Fonts gehört. Mit dem Tool afm2tfm kann man eine .tfm-Datei erzeugen.
- .mf [Text] Bei der Erzeugung einer MetaFont-Schrift werden automatisch die benötigten tfms generiert.

Bei den TrueType-Schriften gibt es kein Äquivalent zu pl und afm, so daß man entweder eine afm-Datei einer gleichwertigen PostScript-Schrift sucht oder ein Tool wie ttf2pk, ttf2mf oder die kommerziellen Programme von Y&Y benutzt.

LaTeX-Dateien

Um einen Befehl zu erzeugen, der die tfm-Datei wechselt, muß man `\font\name=schriftart` verwenden; z.B.: `\font\rm=Times`, unter der Annahme, daß es eine Datei `Times.tfm` gibt. Aber diese Art, Fonts zu definieren, ist sehr primitiv und nicht vorteilhaft, da jedem Schriftschnitt (fett, kursiv usw.) und jeder Schriftgröße ein anderer Bezeichner `\name` entspricht.

LaTeX 2_ε führt ein erheblich vollständigeres (und komplexeres) System ein, das die Wahl der Schrift mittels ihrer verschiedenen Merkmale Schriftart, Form (*shape*), Größe usw. gestattet.

- .fd [LaTeX] Diese Datei enthält die Definitionen einer Schriftart, die eine bestimmte Metrik einem bestimmten Schriftschnitt zuordnet. Natürlich bedarf es mehrerer Definitionen, um verschiedene Kombinationen abzudecken. Betrachten wir ein paar Linien von `ot1ptm.fd`, z.B.:

```
\DeclareFontShape{OT1}{ptm}{m}{n}{<-> ptmr7t}{}
\DeclareFontShape{OT1}{ptm}{m}{it}{<-> ptmri7i}{}

```

Dies legt *nur* die Fonts fest, damit man sie benutzen kann, wenn man will, aber es ändert nicht die Definition von `\textrm`, `\textit` usw. Tatsächlich aber braucht man sie noch nicht einmal selbst zu laden: LaTeX lädt sie automatisch, wenn einer von ihnen gebraucht wird (sogar mitten im Dokument).

Für mehr Information darüber, wie man .fd-Dateien definiert kann man `fntguide.tex` heranziehen. Die .fdd-Dateien erfüllen dieselbe Aufgabe wie die dtx-Dateien: es sind viele .fd-Dateien, die man mit einer .ins-Datei „auspacken“ muß.

- .sty [LaTeX] Nun gut, diese Erweiterung betrifft nicht die Fonts ... Es gibt einige Packages, die dem „Ja, ich will“ dienen und die Definitionen von `\textrm`, `\textit` usw. ändern, um einige bestimmte Fonts zu wählen. Der interessanteste ist `times`, aber es gibt auch `palatino`, `newcent`, `helvet` usw. Für die Erstellung neuer Packages oder die Änderung von Fonts in einem Dokument verweise ich wiederum auf `fntguide.tex`.

3. Druck

Reale Fonts

In diesem dritten Teil werde ich über die Dateien sprechen, die zum Ausdrucken einer .dvi-Datei notwendig sind. Die Art der Implementation variiert hier von System zu System, wenngleich

die Mehrheit im großen und ganzen dem hier angegebenen Muster folgt und `dvips` entspricht. Eine erwähnenswerte Ausnahme ist Y&Y, die keine virtuellen Fonts zuläßt und einem vollkommen anderen Schema folgt. Textures weist ebenfalls wichtige Unterschiede auf.

In den ersten beiden Abschnitten habe ich gesagt, daß zur Benutzung eines Fonts eine `.tfm`-Datei zusammen mit der entsprechenden Datei zum Erzeugen der Zeichen (`pk`, `ttf`, `pfa`, `pfb`) verfügbar sein muß. Jeder Datei der ersten Sorte entspricht eine andere der zweiten. Im Fall einer `pk`-Datei sind ihr Name und der ihrer `tfm`-Datei gleich (bis auf die Erweiterung natürlich). Wenn wir also `cmr10.tfm` haben, sucht `dvips cmr10.pk` mit den passenden Maßen (wenn es vorher keine PS- [PostScript-] Version findet). Mit PS ist das dagegen nicht so, sei es nicht, daß man einen Bezug zwischen dem `tfm`-Namen und der Schriftart in der Datei `psfonts.map` herstellt, wie z.B.:

```
ptmb8r Times-Bold
```

wodurch festgelegt würde, daß die Zeichen der Metrik `ptmb8r` mit dem PS-Font namens Times-Bold zu drucken sind.

Bei den `pk` stehen die Zeichen fest und nehmen daher auf allen Systemen die gleichen Positionen ein; bei den PostScript-Schriften stehen sie dagegen nicht fest, vielmehr ist jedes Zeichen durch einen Formnamen gekennzeichnet, dem (über eine zusätzliche Datei) ein Code frei zugeordnet werden kann. Die `dvi`-Datei enthält nur Information über die Metrik und den Code, aber sie weiß nichts über das zu diesem Code gehörende Zeichen: einem gegebenen Code entsprechen auf verschiedenen Systemen verschiedene Zeichen. So ist z.B. der Code 150 unter Unix ein Halbkreis (Akzent), unter Windows ein Halbgeviertstrich (*endash*) und unter Mac ein `n` mit Tilde (`ñ`). Stellt Euch das Chaos vor, das man da anrichten kann!

`dvips` nutzt aus, daß die Codes bei PS frei zugeordnet werden können und verteilt die Codes in einer Windows-ähnlichen Weise, aber nicht, weil man glaubt, daß das so besser ist, sondern weil Windows von TrueType abhängt, das gewisse Beschränkungen aufweist. In Wirklichkeit ist die vollständige Zeile von `psfonts.map` von oben:

```
ptmb8r Times-Bold "TeXBase1Encoding ReEncodeFont" <8r.enc
```

Die Datei `8r.enc` legt fest, daß der Code 150 dem Zeichen namens Halbgeviertstrich entspricht. Mit anderen Worten, wenn in einer `dvi`-Datei die Metrik `ptmb8r` erscheint, verwendet `dvips` die Schriftart Times-Bold mit den Zeichen in der in `8r.enc` angegebenen Zuordnung. (Anmerkung. Jede Plattform hat ihre voreingestellte Zuordnung, die auch andere Programme wie Word, WP usw. benutzen. Mit PS aber kann man sie ändern und sogar auf mehr als 256 Zeichen zugreifen. Mit TrueType ist dies nicht möglich, und wenn die Datei mehr als 256 Zeichen enthält, sind nicht alle verfügbar.)

Virtuelle Fonts

Bis jetzt haben wir von PS, `ttf`- und `pk`-Fonts, die die Beschreibung der Zeichenform enthalten, gesprochen. Nichtsdestotrotz gibt es eine andere Möglichkeit namens virtueller Font, die keine Beschreibung des „Bildes“ enthält, sondern von anderen, „echten“ Fonts „parasitiert“. Dies wird mittels einer neuen Datei bewerkstelligt:

- **vf** [binär] Enthält Information darüber, welcher Font wirklich für jedes Zeichen zu benutzen ist.
- **vp1** [Text] Dies ist das Äquivalent zu den `.pl`-Dateien im vorigen Abschnitt, aber mit einer wichtigen Änderung: `.vp` erhalten wir normalerweise über einen Font, von dem wir seine Metriken extrahiert haben. Mit `.vp1` kehren wir normalerweise den Vorgang um: wir erzeugen einen maßgeschneiderten Font, indem wir eine `.vp1`-Datei schreiben und eine `.vf`-Datei mit dem Tool `vptovf` generieren. Diese Schritte vollzieht man nur, wenn man die Fonts manipulieren will, sie sind nicht für den Durchschnittsanwender gedacht. Ein in $\text{T}_{\text{E}}\text{X}$ geschriebenes

Tool namens `fontinst` ist bei diesen Aufgaben sehr hilfreich, wenn man hier weiterforschen will. Ein anderes Tool ist `afm2tfm`, das schneller und einfacher zu bedienen ist, aber keine `fd`-Dateien erzeugt; andererseits läßt sich dies leicht manuell machen.

Normalerweise werden die `vf`-Dateien zur Reorganisation der Zeichen gemäß den verschiedenen Systemen verwendet, aber sie können auch zu anderen Zwecken wie der Nachahmung von Kapitälchen nach Großbuchstaben oder der Erzeugung von von nicht existierenden Zeichen benutzt werden.

Wenn `dvips` in der `dvi`-Datei auf ein `tfm` eines Fonts stößt, von dem eine `vf`-Datei vorhanden ist, ersetzt es dieses durch das `tfm`, das von der `vf`-Datei angezeigt wird und beginnt von vorn. Wenn es die `vf`-Datei nicht findet, fährt es fort, indem es nachsieht, ob es für das `tfm` einen Eintrag in `psfonts.map` und demnach einen PS-Font gibt. Wenn letzteres auch fehlschlägt, sucht es schließlich einen `pk`-Font. Z.B. sucht es bei `ptmr8t.tfm` ein `vf`; es wird gefunden und liefert die Zeichen `ptmr8t.tfm` zurück. Für dieses sucht es aufs Neue ein `vf`, aber da keines vorhanden ist, sucht es in `psfonts.map` und findet dort einen „echten“ Font `ptmr8r`, den es PS Times-Roman nennt.

Wie verhält es sich mit den `dvi`-Viewern? Normalerweise funktionieren sie auf die gleiche Art wie `dvips`, nur daß sie üblicherweise nicht `psfonts.map` verstehen (sie haben keinen PS-Interpreter) und daher irgendeine Alternative benutzen. Da die Viewer für eine spezifische Plattform entwickelt werden, kennen sie bereits den Code, der einem jeden Zeichen entspricht, und daher wird statt der Übertragung vom Namen auf einen Code direkt Code auf Code übertragen. In jedem Fall muß betont werden, daß die Prozesse beim Betrachten und beim Drucken *unterschiedlich sind* und daher unterschiedliche Dateien erfordern können und meist auch tatsächlich erfordern. Das Handbuch jeder Implementation erklärt, wie der jeweilige `dvi`-Viewer zu konfigurieren ist.

4. Namen und Kodierungen

Im vorigen Beitrag sagten wir, daß `dvips` die Konversion des von \TeX verwendeten Namens (sagen wir, `ptmb8r.tfm`) in den realen der Schriftart besorgt. Aber: was bedeutet dieser stumpfsinnige Name, den \TeX benutzt? Warum nimmt man nicht gleich den Namen Times-Bold? Der Grund ist vor allem Kompatibilität. Mit dieser Methode stellen wir sicher, daß ein Font, der in einem Dokument benutzt wird, auf jedem beliebigen System gleich identifiziert wird. Textures bezeichnet z.B. Times-Bold als `TimesB.tfm`, und die so erzeugten `dvi`-Dateien sind unter anderen Implementationen unlesbar.

Der Name geht aus einer Reihe von Konventionen hervor, die eine eindeutige Identifikation eines Fonts mit nur acht Zeichen (dank unseres alten Bekannten MS-DOS) erlauben. Das erste Zeichen ist der Eigentümer der Schriftart: `p` (Adobe), `e` (Apple), `l` (Linotype), `i` (ITC) usw.

Die nächsten beiden bezeichnen die Familie: `hv` (Helvetica), `bk` (Bookman), `ge` (Geneva), `ur` (Centaur), `ag` (Avant Garde), `zc` (ZapfChancery), `tm` (Times), `ut` (Utopia), `cr` (Courier), `pl` (Palatino) usw.

Das vierte Zeichen ist immer der Schriftschnitt: `r` (Text), `b` (Fettdruck).

Danach kommen die „Varianten“: `i` (kursiv), `c` (Kapitälchen) usw. Hierauf folgt die Kodierung, die ich gleich erkläre. Wenn keine dieser Varianten auf den Font paßt und die Kodierung unbekannt oder nicht standardmäßig ist, ergänzt man ein `r` zum Auffüllen (vielleicht ein *horror vacui* desjenigen, der sich diese Namen ausgedacht hat?). Am Ende fügt man die Breite an, wenn es nicht die normale Schrift ist, bei der man nichts ergänzt: `n` (eng), `x` (weit) usw. Wenn der resultierende Name mit `rr` endet, verkürzt man dies auf ein `r`.

Mit dieser Zusammenfassung und nach Heranziehung einiger `.fd`-Dateien läßt sich ihre Bedeutung etwas besser verstehen. Beispielsweise ist `phvrrn` Helvetica von Adobe, Text, unbekannte Kodierung und `eng; pplri7tx` ist Palatino von Adobe, Text, kursiv, mit der Kodierung `7t` und weit; die vier Grundvarianten von Times (T1) sind `ptmr8t`, `ptmri8t`, `ptmb8t` und `ptmbi8t`. Da dieses Schema noch nicht kompliziert genug zu sein schien, werden bei einigen Schriftarten andere Zeichen benutzt: z.B. sind die entsprechenden Varianten von AvantGarde `pagk8t`, `pagko8t`, `pagd8t` bzw. `pagdo8t`, ohne daß es hier irgendeine Regel gäbe, die uns hilft³.

Nun werde ich etwas ausführlicher die Kodierungsbezeichnung anhand von konkreten Beispielen für Times-Bold erklären⁴.

8r Dies ist die Grundschriftart. `ptmr8r` ist z.B. der reale Font, dessen entsprechende PostScript-Datei mit den Zeichen in der oben beschriebenen Weise benannt wird, d.h., die dieser Kodierung für Times-Bold zugeordneten Dateien sind `8rptm.fd`, `ptmb8r.tfm` und Times-Bold (oder wie die PS-Datei auf dem jeweiligen System gerade heißt, mit der Erweiterung `.pfa` oder `.pfb`, außer unter MacOS). Sie dürfen nicht direkt benutzt werden, und selbstverständlich gibt es keine `vf`-Datei, weil der Font real ist.

7t Dieser Font ist abgeleitet und hat nur Zeichen mit einem Code kleiner als 128 in der Zuordnung, die im *TeXbook* beschrieben ist. In \LaTeX sind sie als `OT1` bekannt. Seine Dateien sind `ot1ptm.fd`, `ptmb7t.tfm` und `ptmb7t.vf`. Mit ihm stellt \TeX die Akzente durch Übereinandersetzen von Akzent und Buchstabe her.

8t Diese Schriftart ist abgeleitet und enthält die Zeichen in der Verteilung gemäß der Vereinbarung, die auf einer \TeX -Konferenz in Cork getroffen wurde und in der die Codes über 127 mit Buchstaben mit diakritischen Zeichen belegt sind. In \LaTeX laufen sie unter der Bezeichnung `T1`. Ihre Dateien sind `t1ptm.fd`, `ptmb8t.tfm` und `ptmb8t.vf`. Mit diesem Font existieren die Akzente und andere Buchstaben für eine ganze Menge Sprachen tatsächlich im Zeichensatz. Unter diesen befindet sich entgegen anderslautenden Behauptungen *nicht* das Katalanische (es fehlt das *ele geminada*⁵).

8c Aber wenn `8t` (= `T1`) nur Buchstaben mit diakritischen Zeichen hat, wo bleiben dann die Zeichen, die sich normalerweise auf diesen Positionen befinden, wie etwa Promille, gewisse hochgestellte Zeichen, der senkrechte Strich usw.? `8c` ist diesen Symbolen und anderen, mittels einer `vp1`-Datei aus ihnen gebildeten, gewidmet. In \LaTeX bezeichnet man sie als `TS1`. Ihre Dateien sind `ts1ptm.fd`, `ptmb8c.tfm` und `ptmb8c.vf`. Sie müssen mit Hilfe des `textcomp`-Packages benutzt werden.

Man beachte, daß es für jede Kodierung/Familie eine andere `fd`-Datei gibt und daß diese Konventionen für PostScript-Schriften gelten. Die EC-Fonts, wie auch andere, folgen einem anderen Kriterium.

Um eine bestimmte Schriftart zu benutzen, können wir die in vorangegangenen Abschnitten beschriebenen Werkzeuge verwenden und die Arbeit selbst machen; aber es gibt eine lange Reihe von Dateien, die unter dem Namen `PSNFSS` zusammengefaßt sind und uns die wichtigsten zur Verfügung stellen. Für andere Fonts sucht man entweder im CTAN, oder wir müssen sie selber mit Hilfe von `fontinst` machen⁶. Wenn wir, sagen wir, Garamond benutzen wollen, besteht der

³Vertrauen wir darauf, daß eines Tages dieses Namenssystem abgeschafft – wenig wahrscheinlich – oder wenigstens vereinfacht wird, denn es ist ein wahres Chaos.

⁴Die Entsprechungen zwischen diesen Kodierungen und den von \LaTeX benutzten sind nicht vollkommen, da sie sich auf verschiedene Aufgaben beziehen. Die von \LaTeX wissen nichts über Ligaturen, die innerhalb der Fonts erzeugt werden; die von den Fonts wissen nichts über die Konstruktion von Akzenten, die in \TeX zusammengesetzt werden.

⁵Diese Bemerkung bezieht sich auf einen Sonderbuchstaben des Katalanischen, der in der `T1`-Kodierung fehlt; das Originaldokument ist auf spanisch geschrieben und hat seinen Ursprung in Beiträgen für die Spanish- \TeX -Newsgroup (Anm. des Übersetzers).

⁶Es kann von Interesse sein, die Perl `VfInst`-Anleitungen zu Rate zu ziehen.

kürzeste Weg darin, alle Dateien zu suchen, in deren Namen `pgm` vorkommt und dann Einträge in `psfonts.map` zu ergänzen, indem man die von anderen Fonts als Vorlage nimmt.

5. Die \LaTeX -Schnittstelle

Inputenc.sty, latin1.def, ansinew.def, applemac.def, usw.

In einem der vorigen Abschnitte sagte ich, daß in gewisser Weise jedes Zeichen in einem Dokument ein Befehl ist, der \TeX mitteilt: „reserviere den Platz, den der aktuelle Font für meinen Code angibt“. In Wirklichkeit erlaubt \TeX , jedem Zeichen eine beliebige Definition zuzuordnen. Das Package `inputenc` macht sich dies zunutze, um á – oder genauer gesagt, den Code, der á entspricht – als `\'a` zu definieren; dies ist alles, was es tut. Dank dem, was man sich für eine spezifische Plattform überlegt hat, stellt der Prozessor, den man für das Schreiben des Dokuments verwendet, den Bezug $225 \leftrightarrow \acute{a}$ her, und so sehen wir es auf dem Bildschirm. \TeX aber weiß nichts von alledem, und daher nimmt sich `inputenc` dessen mit der Zeile

```
\DeclareInputText{225}{\@tabacckludge\'a}
```

in der Datei `latin1.def` an. (Anmerkung: `\@tabacckludge\' = \'`.) Auf einem Mac entspricht dieser Code einem vertikal zentrierten Punkt, und daher lautet in `applemac.def` diese Zeile:

```
\DeclareInputText{225}{\textperiodcentered}
```

während es eine andere gibt, die

```
\DeclareInputText{135}{\@tabacckludge\'a}
```

ist. Der Befehl

```
\usepackage[latin1]{inputenc}
```

lädt `inputenc` und die Definitionsdatei `latin1.def`. (Für Windows 9x ist es `ansinew.def`.) Sind diese Definitionen einmal gemacht, braucht \TeX nicht mehr über die Plattform zu wissen.

Fontenc.sty, ot1enc.def, t1enc.def, ts1enc.def, usw.

Diese Dateien setzen den umgekehrten Vorgang um: sie stellen den Bezug zwischen dem Buchstaben á und dem Code her, der ihm in der verwendeten Kodierung entspricht, die ihrerseits von der verwendeten `tfm`-Datei abhängt. So enthält `t1enc.def` eine Zeile, die

```
\DeclareTextComposite{\'}{T1}{a}{225}
```

lautet, weil dies die Position ist, die das „Bild“ des Buchstabens á in den Dateien der T1-Kodierung einnimmt, wie etwa `ptmr8t` (man erinnere sich, daß `8t` in Fonts = `T1` in \LaTeX ist), `phvr8t`, `pcrr8t` usw. (Und auch in denen der EC-Familie, auch wenn `8r` nicht in ihrem Namen vorkommt.) In diesem konkreten Beispiel befindet sich á in der `8t`-Kodierung an der gleichen Stelle wie in `latin1`, da die `Cork`-Kodierung eine Modifikation von ISO Latin 1 ist.

Dagegen enthält `ot1enc.def` *keine* gleichwertige Zeile und stellt keine Beziehung her. Dies zeigt an, daß das Bild für diesen Buchstaben bei Fonts mit dieser Kodierung (`ptmr7t`, `phvr7t` ... und die der `cm`-Familie) nicht zur Verfügung steht, und bewirkt, daß \LaTeX das á mittels eines \TeX -Primitives aus dem Akzent und dem Buchstaben als einzelnen Zeichen konstruiert.

6. Leben und Werk eines á

In diesem letzten Teil werden wir sehen, was mit dem folgenden Dokument von der Kompilierung bis zum Druck geschieht:

```
\documentclass{article}
\usepackage[latin1]{inputenc}
\usepackage[t1]{fontenc}
\usepackage{times}
\begin{document}
\textbf{á}
\end{document}
```

Über `times` ist bereits im zweiten Abschnitt gesprochen worden, es besagt (aufgepaßt: es besagt *nur*), daß die Familie für die Serifenschrift `ptm` ist.

Wir können schon sagen, was passiert, wenn der Befehl

```
\textbf{á}
```

erreicht wird. `\textbf` legt fest, daß der Schriftschnitt des Buchstabens fett ist und teilt \TeX mit, daß die folgenden Zeichen in der `dvi`-Datei zu der Metrik gehören, die irgendein `.fd` der Kombination `T1/ptm/b/n` zuweist, worin `n` die Form ist (`n` ist normal, `it` ist kursiv, `sc` ist Kapitälchen usw.). Die `.fd`-Dateien werden nach dem Paar Kodierung/Familie benannt, so daß \LaTeX die Datei `t1ptm.fd` sucht und liest.

In dieser befindet sich die Zeile

```
\DeclareFontShape{T1}{ptm}{b}{n}{<-> ptmb8t}{}

```

So weiß \LaTeX schon, daß die Metrik `ptmb8t.tfm` ist. \TeX lädt die Daten von `ptmb8t` in seinen Speicher und schreibt in die `dvi`-Datei einen Befehl, der besagt, daß die nächsten Zeichen von diesem Font sind.

Danach kommt das `á`, das ein als `\'a` definierter Befehl ist. Im einzelnen ist `\'` ein Befehl, der das `a` nimmt und untersucht, ob es in der aktuellen Kodierung (`T1`) einen Code für `\'a` gibt. Diesen Code gibt es, es ist die 225, und so reserviert \TeX in der `dvi`-Datei den Platz, den `ptmb8t.tfm` für den Code 225 angibt.

Damit ist die Kompilierung des Dokuments bereits beendet, und wir wollen es ausdrucken. `Dvips` liest die `dvi`-Datei und stößt auf den Metrikwechsel nach `ptmb8t`. Zuerst ermittelt es, ob ein `ptmb8t.vf` gibt; das ist der Fall, und `dvips` speichert seine gesamte Information für später. Danach fährt es fort zu untersuchen, was folgt.

Das nächste, was es findet, ist ein Zeichen mit dem Code 225. Es sucht dieses Zeichen in der Information, die es gespeichert hat, und findet heraus, daß es „parasitiert“ ist und es sich in Wirklichkeit um das Zeichen 225 von `ptmb8r` handelt. (Wiederum stimmen die Codes überein, aber das muß nicht unbedingt so sein.) Da es kein `ptmb8r.vf` gibt, sucht es in `psfonts.map`, stellt fest, daß der reale Font `Times-Bold` ist und schreibt daher in die PostScript-Datei, die es generiert, einen Befehl, um zu dieser Schriftart zu wechseln und ein Zeichen mit dem Code 225 zu ergänzen⁷.

Nun haben wir schon die `ps`-Datei erzeugt, die wir ausdrucken werden. Da `ptmb8r` der Kodierung in der Datei `8r.enc` folgt, stellt der Drucker oder `Ghostview` fest, daß 225 das Zeichen namens `aacute` ist. Das einzige, was noch zu tun bleibt, ist, die Daten des Bildes von `aacute` in `Times-Bold` zu nehmen und den Drucker anzuweisen, das physische Bild herzustellen. (Oft laufen die Erzeugung einer PostScript-Datei und ihr Ausdruck automatisch im Hintergrund ab.)

⁷Die tatsächliche Reihenfolge, in der `dvips` vorgeht, ist nicht ganz so wie ich sage, aber es ist „so als ob“.

7. Fonts in fünf Minuten

Die scharfen Beobachter werden bemerkt haben, daß dieses Dokument eine etwas engere Version von Courier hat. Es läßt sich recht schnell eine Variante erstellen, wenn wir nicht allzu anspruchsvoll sind, nicht die erzeugten tfm-Dateien weitergeben und auch nicht andere Sonderzeichen als Vokale mit Akzenten oder n's mit Tilde verwenden. Dazu habe ich einfach die afm-Dateien, die es auf dem CTAN gibt, kopiert und afm2tfm in der folgenden Weise benutzt:

```
afm2tfm pcr8a.afm -e .8 -p 8r.enc pcr8rn.tfm
afm2tfm pcrb8a.afm -e .8 -p 8r.enc pcrb8rn.tfm
```

In psfonts.map habe ich folgendes ergänzt:

```
pcr8rn Courier " .8 ExtendFont TeXBase1Encoding ReEncodeFont " <8r.enc
pcrb8rn Courier-Bold " .8 ExtendFont TeXBase1Encoding ReEncodeFont " <8r.enc
```

und in der Datei meines dvi-Viewers (OzTeX), in der die Fonts sind:

```
pcr8rn - "Courier" Mac8r.enc
pcrb8rn - "Courier" Mac8r.enc b
```

Dank des glücklichen Umstandes, daß die Buchstaben mit Akzent in 8r und 8t auf den gleichen Positionen sitzen, habe ich mich darauf beschränkt, die ganze Datei t1pcr.def in den Vorspann zu kopieren und folgendes zu ändern:

```
\DeclareFontShape{T1}{pcr}{m}{n}{
  <-> pcr8rn
}{}
\DeclareFontShape{T1}{pcr}{b}{n}{
  <-> pcrb8rn
}{}
```

Das ist ziemlicher Pfusch, funktioniert im Spanischen aber sehr gut. Wäre doch die vollständige und ordnungsgemäße Installation nicht so kompliziert.