Bernhard Walle

Schriften unter Linux

19. Januar 2005

Inhaltsverzeichnis

Über dieses Dokument 3

1 Grundlagen 3
1.1 Historische Hintergründe . . . 3
1.2 Verschiedene technische Schrifttypen . . . 4
1.3 Verschiedene Formate für Schriften . . . 5
1.4 Encoding . . . 8
1.5 X Logical Font Description (XLFD) . . . 10

2 Nützliche Programme 11
2.1 Zeichensatzumwandlung . . . 11
2.2 Anzeige von Schriften . . . 12
2.3 Schriftkonvertierung . . . 13
2.4 Vollautomatische Schrifteninstallation . . . 14
2.5 Komplette Schriftenverwaltung . . . 14

3 Neue Schriften installieren 14
3.1 Vorbereitungen . . . 15
3.2 XFree86 . . . 15
3.3 OpenOffice.org 1.0.x und StarOffice 6 . . . 18
3.4 Ghostscript . . . 18
3.5 WordPerfect . . . 19

4 Das Eurosymbol 19
4.1 Die korrekte Sprachumgebung . . . 19
4.2 Das Eurozeichen und die Zeichensätze . . . 20
4.3 Eingabe mit der Tastatur . . . 21
4.4 Das Eurosymbol in verschiedenen Anwendungen . . . 22

5 Anti-Aliasing von Bildschirmschriften 25
5.1 Allgemeines . . . 25
5.2 Konfiguration . . . 25
5.3 Unterstützung durch Anwendungsprogramme . . . 26

6 Erzeugung von PDF-Dateien 27
6.1 Allgemeine Anleitung zur PDF-Erstellung . . . 27
6.2 Einbettung von Schriften . . . 28

7 Bezugsquellen von Schriften 31
7.1 35 Standard-Postscript-Schriften . . . 31
7.2 Bildschirmschrift Terminus . . . 33

Literatur 34

Über dieses Dokument

Ursprünglich entstand dieser Artikel im Zusammenhang mit zwei Skripten, welche die Schrifteninstallation insbesondere für StarOffice 5.2 vereinfachen sollten.

Mittlerweile hat sich einiges geändert unter Linux: Der Nachfolger für StarOffice 5.2 heißt OpenOffice.org 1.0.x bzw. StarOffice 6.0 – beide basieren auf dem gleichen Code. Auf die Schriftenkatastrophe StarOffice 5.2 soll nun nicht mehr eingegangen werden.

Meine eigenen Skripten sind nun nicht mehr notwendig, deshalb habe ich sie entfernt. Vielmehr will ich – zusätzlich zu der schon vorhandenen Dokumentation – eine Anleitung bereitstellen, die die Schrifteninstallation auf aktuellen Linuxsystemen zusammenfassend beschreibt. Ich hoffe, dass mir dies hiermit gelungen ist, freue mich auf E-Mails (bernhard.walle@gmx.de) mit Kritik, Anregungen (vielleicht auch Lob) und bedanke mich abschließend bei allen Menschen, die Linux in der heutigen Form möglich gemacht haben.

Das Dokument kann in seiner jeweils aktuellen Fassung unter folgender Adresse heruntergeladen werden:

➠ http://www.bwalle.de/docs/schriften.pdf

Den LaTeX-Quellcode oder ältere Versionen verschicke ich auf Anfrage gerne per E-Mail. Die Veröffentlichung erfolgt unter den Bedigungen der GNU Free Documentation License in beliebiger Version, zu finden im Internet unter http://www.gnu.org/copyleft/fdl.html.

1 Grundlagen

1.1 Historische Hintergründe

Vor einigen Jahrzehnten waren Schriften in Zusammenhang mit Computern eher nebensächlich. Die Bildschirmschrift war fest in der Grafikkarte oder in das Terminal eingebaut, wie das heute noch bei der Konsole der Fall ist (es sei denn man verwendet einen Framebuffer).

Beim Drucker war es ähnlich: Dem Drucker wurden nur die Zeichen (bzw. Befehle für Zeilen- und Seitenumbruch oder -vorschub) selbst übermittelt, nicht aber, in welcher Form diese zu drucken waren. Die Schrift war entweder mechanisch vorgegeben (Typenrad) oder fest einprogrammiert (Nadeldrucker). Dies ist auch bei modernen Druckern noch der Fall, nur dass es wegen der geringen Zahl an eingebauten Schriften praktisch nicht mehr genutzt wird. Außerdem ist nicht gewährleistet, dass die Druckerschriften mit den Bildschirmschriften übereinstimmen.

Mit der Zeit der Textverarbeitungen kam das Bedürfnis, verschiedene Schriftarten in verschiedenen Größen zu drucken. Das Betriebssystem DOS war aber überhaupt nicht auf solche Anliegen vorbereitet. Jede Textverarbeitung entwickelte ihre eigene Lösung, d. h. eigene Druckertreiber und evtl. verschiedene Bildschirmschriften. Letztere waren beispielsweise für eine Druckvorschau nötig. Die Schriftenauswahl erfolgt immer noch über Druckerbefehle. Es konnten daher nur die im Drucker eingebauten Schriften verwendet werden.

Im Profibereich ging man einen komplett anderen Weg. (Das Textsatzsystem LaTeX und die Metafont-Schriften sollen zunächst unerwähnt bleiben.) Von Adobe wurde die Druckersprache Postscript entwickelt, in der Text und Grafiken geräteunabhängig beschrieben werden konnten. Das System gibt es immer noch, und Postscript-Drucker werden im professionellen Bereich nach wie vor eingesetzt. Neben fest eingebauten Schriften gibt es die ladbaren Schriften, die vorab in den Drucker geladen werden. Damit ist eine Vielzahl von Schriften verfügbar, die nur von dem im Drucker verfügbaren Speicher begrenzt wird.

Im Heimbereich setzte sich dieses System aber nie durch, da Postscript-Drucker wegen der eingebauten Hardware, die die Berechnungen vornimmt, zu teuer sind. Stattdessen wird hier, wenn man verschiedene Schriften einsetzt, die nicht im Drucker eingebaut sind, das Ganze im Rechner in Punkte umgesetzt, sodass faktisch ein Bild einer Seite gedruckt wird.

Dieses Prinzip wurde erst mit dem Siegeszug des Betriebssystems Windows 3.1 von Microsoft verbreitet – und mit den mitgelieferten TrueType-Schriften. Die Times New Roman wurde zur Standardschrift, was man heute deutlich erkennen kann.

Wieder einen anderen Weg ging man bei Unix-Systemen, auf denen Linux basiert. Hier druckt jedes Programm, indem es zunächst eine Postscript-Datei ausgibt. Text wird, ohne dass man es normalerweise merkt, in Postscript umgewandelt, bevor es gedruckt wird. Wer keinen Postscript-Drucker hat, der benutzt eben Ghostscript, das Postscript-Befehle in verschiedene Ausgabeformate umrechnen kann, auch in verschiedene Druckersprachen. Das System ist unglaublich flexibel, da ein einheitliches Format existiert, mit dem Datenaustausch ermöglicht wird.

1.2 Verschiedene technische Schrifttypen

1.2.1 Bitmap-Schriften

Früher gab es nur Bitmap-Schriften. Reine Bildschirmschriften sind immer Bitmap-Schriften, und die im Drucker eingebauten in der Regel ebenfalls.

Der Nachteil von Bitmap-Schriften ist, dass man für jede Schriftgröße einen eigenen Font benötigt. Für moderne Textgestaltung sind diese daher weniger geeignet, da man in der Wahl der Schriftgrößen eingeschränkt wird. Letztlich existiert also für jedes einzelne Zeichen ein Punktemuster, ein kleines Bildchen.

Der Vorteil besteht aber darin, dass diese Schriften handoptimiert werden können und oft bei kleinen Größen besser lesbar sind als skalierbare Varianten. Wirkliche Vorteile ergeben sich daher nur bei der Bildschirmanzeige, wo mit kleinen Auflösungen gearbeitet werden muss. Das Argument, dass keine Rechenleistung erforderlich ist, um die Schrift zu berechnen (sie existiert ja schon), fällt bei heutigen PCs kaum mehr ins Gewicht, wohl aber bei mobilen Geräten wie Handys oder PDAs.

1.2.2 Skalierbare Schriften

Skalierbare Schriften liegen in einem Vektorformat vor. Sie können beliebig skaliert (vergrößert/verkleinert) werden. Das geschieht aber nicht linear, was die Lesbarkeit deutlich erhöht, aber auch das Ganze erheblich schwieriger macht.

Neben einigen Bitmapschriften, die beim Betriebssystem fest dabei sind (auch bei Windows), verwendet man heute nur noch skalierbare Schriften. Zusätzliche Schriften auf CD-ROMs oder im Internet gibt es praktisch nur in skalierbarer Form. Beim Druck mit den entsprechenden hohen Auflösungen haben skalierbare Schriften eigentlich nur Vorteile, wenn man davon absieht, dass zur Berechnung Rechenleistung erforderlich ist.

1.3 Verschiedene Formate für Schriften

Auf die Bitmap-Formate will ich hier nicht näher eingehen, da sie im Prinzip uninteressant sind. Normalerweise installiert man auch keine zusätzlichen Bitmapschriften, aber Ausnahmen bestätigen die Regel und so beschreibe ich im Abschnitt 7.2 die Installation der Schrift *Terminus*.

Nur soviel: Bitmapschriften haben unter Linux (genauer: X11, d. h. der graphischen Oberfläche von Linux) die Erweiterung bdf oder pcf. Die meisten Systeme können auch mit gepackten Fonts umgehen, um Speicherplatz zu sparen; daraus resultiert dann bdf.gz bzw. pcf.gz.

Detaillierte Informationen über die Schriftformate finden Sie unter [2], [3] (OpenType) und unter [4] (verschiedene Formate von Adobe).

1.3.1 Das Type1-Format

Das erste skalierbare Format, das den Durchbruch für diese Art von Schriften schaffte, war das von Adobe für die Verwendung in Postscript entwickelte Type1-Format. Es ermöglicht es, die Schriften sowohl in den Drucker zu laden und auszudrucken als auch am Bildschirm anzuzeigen. Damit ist das Type1-Format universell einsetzbar. Auf Unix-Systemen ist Type1 Standard, bei Windows muss man zusätzlich den *Adobe Type Manager*[1] installieren. Beim klassischen MacOS ist die Installation des *ATM* empfehlenswert.[2] In Windows 2000 und XP ist eine Unterstützung für Type1 bereits integriert.

Eine Type1-Schrift besteht aus mehreren Dateien: Die pfa- oder pfb-Datei enthält die tatsächliche Schrift, wobei pfa *(Printer Font ASCII)* in Rohform und pfb *(Printer Font Binary)* in komprimierter und evtl. auch verschlüsselter Form die tatsächliche Schrift enthalten. Daneben werden eine pfm- *(Printer Font Metric)* oder afm-Datei *(Adobe Font Metric)* mitgeliefert, welche die so genannten Metriken enthalten, d. h. Angaben über Schriftname, Neigungswinkel und Platzbedarf der einzelnen Buchstaben. Die pfm-Dateien werden nur für Windowssysteme gebraucht und können bei Linux gelöscht werden. Die afm-Dateien werden bei Unix-Systemen verwendet und enthalten Klartext. Die benötigten Informationen können daher aus den Dateien ziemlich einfach ausgelesen werden.

Im Profibereich hat sich Type1 als Standard etabliert. Der Hauptgrund liegt wohl in der Postscript-Integration. Außerdem gibt es im Type1-Format eine größere Auswahl an Profischriften, welche sich u. a. durch echte Kapitälchen und Ligaturen auszeichnen. Der Grund besteht einerseits darin, dass Adobe selbst ein großer Schriftenanbieter ist, zum anderen ermöglicht nur das Type1-Format

[1] Die Light-Version kann unter http://www.adobe.com/products/atmlight/main.html kostenlos heruntergeladen werden.

[2] Type1-Schriften für MacOS enthalten zusätzlich Bitmapschriften, die für die Bildschirmanzeige verwendet werden und die skalierbaren Varianten für Postscript-Drucker. Das Betriebssystem selbst kann aber die skalierten Schriften nicht verarbeiten, was bei Vergrößerungen zu unschönen Bildschirmschriften führt.

dem Anbieter, die Schriftschnitte zu verschlüsseln. Somit kann sich der Hersteller vor billigen aber trotzdem hochwertigen Kopien schützen.

Schließlich gibt es für Type1 eine größere Anzahl von Software zur Erstellung, sodass die auf dem Markt befindlichen TrueType-Schriften oft nur eine Konvertierung aus dem Type1-Format sind, was einen Qualitätsnachteil zur Folge hat.

1.3.2 Type3-Schriften

Type3 ist eine Weiterentwicklung des Type1-Formats. Das Besondere ist, dass sowohl Schriften im Vektorformat als auch Schriften im Bitmapformat enthalten sein können. In der Praxis wird Type3 oft dazu verwendet, Bitmapschriften in Postscript oder PDF einzubetten. Solche Schriften entstehen standardmäßig z. B. bei der Arbeit mit LaTeX, indem die Schriften vom DVI-Treiber *(dvips)* auf die passende Auflösung (die über einen Parameter oder über die Konfigurationsdatei eingestellt wird) gerastert werden.

Unter Linux haben diese Schriften sonst keine Bedeutung, da XFree86 dieses Schriftformat nicht unterstützt.

1.3.3 TrueType

TrueType wurde von Apple und Microsoft gemeinsam entwickelt, um das Adobe-Schriftmonopol zu brechen, d. h. vor allem, um sich Lizenzgebühren zu sparen. Letztlich hat dies dazu geführt, dass das Type1-Format offen gelegt wurde.

TrueType ist heute das am meisten verbreitete Format. Unter Windows kann man seit der Version 3.1 die Schriften ohne irgendwelche Zusatzprogramme anzeigen und ausdrucken.

Die Vorteile liegen vor allem in der großen Verbreitung. Technisch gesehen sind TrueType-Schriften in einem Verfahren gespeichert, das weniger Rechenarbeit erfordert, um die Schriften zu verwenden. In der Praxis spielt das bei den heutigen Rechenleistungen von Prozessoren aber kaum eine Rolle.

Ein weiterer Vorteil soll ebenfalls angesprochen werden: Bei kleinen Schriften in niedriger Auflösung wird bei TrueType ein Verfahren namens Hinting eingesetzt, das hier eine bessere Qualität ermöglicht. Beim Druck spielt das keine Rolle, da im Vergleich zum Bildschirm sehr hohe Auflösungen verwendet werden: Während am Bildschirm in der Regel eine Auflösung von 96 dpi oder 72 dpi (je nach Betriebssystem) verwendet wird, ist bei modernen Laser- oder Tintenstrahldruckern 600 dpi die untere Grenze.

1.3.4 Type42-Schriften

Diese Schriften sind identisch mit TrueType-Schriften, enthalten jedoch einen Header, der es erlaubt, dass sie direkt von Postscript-Druckern gedruckt werden können. Ghostscript z. B. kann direkt mit TrueType umgehen, für einige Postscript-Drucker kann es aber notwendig sein, Type42 zu erzeugen.

1.3.5 OpenType

OpenType ist das neueste Schriftenformat. Es resultiert aus einer Gemeinschaftsentwicklung von Adobe und Microsoft. OpenType-Schriften können sowohl auf Postscript als auch auf TrueType basieren. Erstere erhalten die Endung otf, letztere ttf.

Der Hauptvorteil besteht darin, dass für Windows und MacOS die gleichen Schriftdateien verwendet werden können. Außerdem besitzt OpenType eine verbesserte Unterstützung für Sprachen mit nicht-lateinischen Schriftzeichen. Windows 2000 und XP besitzen eine eingebaute Unterstützung für OpenType. Ein Linux-Programm, das mit diesen Schriften umgehen kann, ist mir nicht bekannt.

Weitere Informationen über OpenType finden Sie unter [3].

1.3.6 Metafont

Metafont wurde 1977 zusammen mit dem Textsatzsystem TEX von Donald E. Knuth entwickelt. Damals bot es die erste Möglichkeit, skalierbare Schriften auf Computern zu nutzen. Metafont hat eine sehr hohe Qualität. Die Skalierung erfolgt sehr gut der Größe und Lesbarkeit angepasst.

Allerdings hat Metafont einen großen Nachteil: Die Berechnung dauert sehr lange. Deshalb werden bei Metafont ähnlich wie bei einem Compiler zuerst feste Bitmapfonts erzeugt und diese werden dann angezeigt. In der Praxis werden die Fonts bei Bedarf generiert. Deshalb dauert auf einem neu installierten TEX-System die Anzeige von DVI-Dateien auch recht lange. Dieses Verhalten erfordert einen relativ großen Platzbedarf (der heute aber nicht mehr das Problem wäre), dafür ist die Anzeigegeschwindigkeit dann hoch.

Durch diese Vorgehensweise sind die generierten Schriften nicht mehr geräteunabhängig. Die Berechnung der Fonts aus dem Quellcode im Ausgabegerät – wie z. B. einem Drucker – wäre aber wegen der hohen Rechenleistung nicht möglich.

Unter anderem deshalb konnten sich diese Schriften nicht als Standard wie Type1 oder TrueType etablieren, sie werden aber im Zusammenhang mit TEX immer noch verwendet. Ein weiterer Grund dürfte auch sein, dass Metafont und TEX freie Software sind und kein großes Unternehmen an der Vermarktung beteiligt war. Die umständliche Bedienung tat ein übriges, wodurch sich TEX und Metafont nur im naturwissenschaftlichen Bereich durchsetzen konnten, v. a. wegen des ausgezeichneten Formelsatzes.

Es existieren nur sehr wenige Schriften im Metafont-Format. Die Tatsache, dass heutige TEX-Systeme problemlos mit Type1-Schriften umgehen können, hat dazu geführt, dass dieses Format kaum noch verwendet wird.

1.3.7 Welches Format ist nun wirklich das beste?

Jedes Format hat unumstritten seine Vorteile. Aber die Qualitätsvor- und -nachteile spielen im Heimbereich gar keine Rolle, da die verwendeten Ausgabegeräte sie ohnehin nie ausnützen können.

Die Unterschiede liegen in der Handhabung und in der besseren Unterstützung des einen oder anderen Formates auf dem jeweiligen System. Verwenden Sie also das Format, das auf Ihrer Plattform besser unterstützt wird oder in dem Ihre vorhandenen Schriften vorliegen.

Unter Linux ist meinem subjektivem Empfinden nach die Druckqualität bei Type1-Schriften immer noch etwas besser.[3] Bei der Bildschirmanzeige sieht es aber umgekehrt aus: Hier sind TrueType-Schriften v. a. bei geringen Schriftgrößen deutlich besser lesbar. Deshalb empfehle ich, die sog. Webschriftarten (zumindest Arial, Verdana und Times New Roman) auf dem System zu installieren. Die Bezugsquellen finden Sie im Abschnitt 7.

1.4 Encoding

Das Encoding bezeichnet den Zeichensatz der Schrift. Dabei ist es durchaus möglich, dass eine Schriftdatei mehrere Encodings enthält. Zum Beispiel enthalten die mit Windows mitgelieferten Schriftarten sehr viele Encodings, um die Anzeige von verschiedenen Sprachen zu ermöglichen.

1.4.1 Übersicht gängiger Zeichensätze nach ISO

Tabelle 1 zeigt eine Übersicht gängiger Zeichensätze nach ISO.

Die ersten 127 Zeichen sind bei allen ISO-Zeichensätzen identisch (ASCII-Zeichensatz), für Unicode wird angenommen, dass UTF-8 verwendet wird – eine Beschreibung folgt später. Die Belegung entnimmt man der *ascii(7)*-Manpage.

Für deutschsprachige Texte können die Zeichensätze Latin 1 (Westeuropäisch), Latin 2 (Osteuropäisch), Latin 5 (Türkisch) und Latin 9 (Westeuropäisch mit Euro) verwendet werden. Üblich sind aber der Latin-1-Zeichensatz und in Zukunft zunehmend der Latin-9-Zeichensatz. Natürlich kann auch Unicode verwendet werden. Die Belegung ennimmt man den jeweiligen Manpages, sofern sie installiert sind, z. B. *iso-8859-1(7)*[4] oder *iso-8859-15(7)*.

Die unter Windows verwendeten Zeichensätze stellen Erweiterungen der ISO-Zeichensätze dar. Im Wesentlichen wurden einige Sonderzeichen wie das Promillezeichen, typographische Anführungszeichen, das Eurozeichen sowie im Fall des Latin-1-Zeichensatzes einige Zeichen, die im Französischen und Finnischen verwendet werden, hinzugefügt. Diese Zeichensätze tragen spezielle Namen, z. B. heißt die Latin-1-Erweiterung *cp1252* (von den unter DOS verwendeten *Codepages*).

Der Grund, warum in den ISO-8859-Zeichensätzen zwischen 0x7F und 0xA0 eine Lücke gelassen wurde und daher weniger Zeichen aufgenommen werden konnten, liegt in der 7-Bit-Kompatibiliät begründet. Lässt man von 0x80 (in cp1252 das Eurozeichen) das achte Bit weg, liegt das Nullzeichen 0x00 vor. Dies ist ein Steuerzeichen, das z. B. in der Programmiersprache C dazu verwendet wird, das Ende von Zeichenketten zu markieren. Es könnte daher bei alten Systemen vorkommen, dass ab dem Eurozeichen alles abgeschnitten wird.

[3] Die Ursache dafür scheint bei Ghostscript zu liegen. Unter http://www.dcs.ed.ac.uk/home/jec/programs/xfsft/renderers.html wird die Qualität von verschiedenen Programmen verglichen, die TrueType- und Type1-Schriften anzeigen können, verglichen.

[4] Manpages ruft man über das Kommando `man [Sektion] [Name]` auf, d. h. in diesem Fall `man 7 iso-8859-1`. Wird keine Sektion angegeben, so wird die erste Manual Page angezeigt, die gefunden wird (von Sektion 1 aufsteigend).

Zeichensatz	Sprachengruppe	Sprachen
ISO 8859-1 ISO Latin 1	Westeuropäische Sprachen	Afrikaans, Albanisch, Baskisch, Dänisch, Holländisch, Englisch, Faröerisch, Deutsch, Isländisch, Italienisch, Lateinisch, Luxemburgisch, Norwegisch, Portugiesisch, Spanisch und Schwedisch
ISO 8859-2 ISO Latin 2	Osteuropäische Sprachen	Albanisch, Kroatisch, Tschechisch, Englisch, Deutsch, Ungarisch, Lateinisch, Polnisch, Rumänisch (mit Einschränkungen), Slowakisch, Slowenisch und Sorbisch
ISO 8859-3 ISO Latin 3	Südosteuropäi-sche und sonstige Sprachen	Englisch, Esperanto, Deutsch, Italienisch, Lateinisch, Maltesisch und Portugiesisch
ISO 8859-4 ISO Latin 4	Skandinavische und baltische Sprachen	Dänisch, Englisch, Estnisch, Finnisch, Deutsch, Lateinisch, Lettisch, Litauisch, Norwegisch, Sámi, Slowenisch und Schwedisch
ISO 8859-5	Kyrillisch	
ISO 8859-6	Arabisch	
ISO 8859-7	Griechisch	
ISO 8859-8	Hebräisch	
ISO 8859-9 ISO Latin 5	Westeuropäische Sprachen	Albanisch, Baskisch, Dänisch, Holländisch, Englisch, Deutsch, Italienisch, Lateinisch, Luxemburgisch, Norwegisch, Portugiesisch, Slowakisch, Spanisch, Schwedisch und Türkisch
ISO 8859-10 ISO Latin 6	Skandinavische und baltische Sprachen	Dänisch, Englisch, Estnisch, Faröerisch, Finnisch, Deutsch, Isländisch, Lateinisch, Litauisch, Norwegisch, Sámi, Slowenisch und Schwedisch
ISO 8859-11	Thailändisch	
ISO 8859-13 ISO Latin 7	Baltische Sprachen	Englisch, Estnisch, Finnisch, Lateinisch, Lettisch, Litauisch, und Norwegisch
ISO 8859-14 ISO Latin 8	Sonstige Sprachen	Albanisch, Baskisch, Dänisch, Englisch, Deutsch, Gälisch, Italienisch, Lateinisch, Luxemburgisch, Norwegisch, Portugiesisch, Spanisch, Schwedisch und Walisisch
ISO 8859-15 ISO Latin 9	Westeuropäische Sprachen (Euro)	Afrikaans, Albanisch, Baskisch, Dänisch, Holländisch, Englisch, Estnisch, Faröerisch, Finnisch, Französisch, Deutsch, Isländisch, Italienisch, Lateinisch, Luxemburgisch, Norwegisch, Portugiesisch, Spanisch und Schwedisch
ISO 10646	alle	Unicode-Zeichensatz

Tabelle 1: Zeichensätze nach ISO

Für symbolische Schriften, die auf den ersten 127 Zeichen keine ASCII-Zeichen haben, verwendet man das Encoding `adobe-fontspecific`. Eine Kennzeichnung ist u. a. deshalb nötig, weil das Anwendungsprogramm wissen muss, ob es sich um eine echte Textschrift oder um eine Symbolschrift handelt. In OpenOffice.org hängt beispielsweise die Rechtschreibprüfung oder die automatische Schriftvorschau davon ab.

Weitere Informationen über die verschiedenen Zeichensätze mit Zeichenabbildungen finden Sie in [6]; ein Programm zum Umwandeln der Zeichensätze in Abschnitt 2.1.

1.4.2 Unicode – die Lösung aller Probleme?

In einer Welt, in der immer mehr global kommuniziert wird, immer mehr Schriftzeichen benötigt werden und immer mehr Betriebssysteme nebeneinander verwendet werden, sind die alten Zeichensätze letztlich ein Flickwerk aus alten Zeiten. Die Lösung ist es, einen Zeichensatz für alle Sprachen zu schaffen, der auf allen Systemen unterstützt wird. Das genau ist der Zweck von Unicode.

Nach ISO 10646 ist ein Wort danach 32 Bit lang. Damit ergeben sich theoretisch 2^{32} = 4 294 967 296 Zeichen. Neben Schriftzeichen für alle möglichen Sprachen enthält Unicode auch Rahmenzeichen, die für Terminals benötigt werden oder typographische Anführungszeichen, die bisher auf Unix-Systemen nicht oder nur über Umwege zur Verfügung standen.

Dieses System wirft aber auch neue Probleme auf: Würde man tatsächlich dieses Verfahren mit 32-Bit-Wörtern in der Praxis verwenden, so würde jede Textdatei (und so auch jede Mail) viermal so groß und mit bisherigen Programmen, die kein Unicode beherrschen, nicht lesbar sein.

Deshalb hat man ein Verfahren namens UTF-8 entwickelt, das folgendermaßen vorgeht: Reine ASCII-Zeichen bleiben unverändert. Erst das achte Bit markiert, dass es sich um Multibyte-Zeichen handelt. Und dann werden je nach Bedarf weitere Bytes belegt. Das genaue Verfahren kann unter *utf8(7)* nachgelesen werden.

Somit geht man sehr platzsparend um und ermöglicht zumindest bei reinen ASCII-Zeichen Abwärtskompatibilität. Dies ist insbesondere auf Unix-Systemen wichtig, wo reine Textdateien als Konfigurationsdateien eine viel größere Bedeutung als auf anderen Systemen haben (wie z. B. Windows, wo schon immer stärker auf Binärformate gesetzt wurde). Außerdem geht man bei E-Mails kein Risiko ein, dass die gesamte Nachricht bloß wegen ein paar Umlauten nicht gelesen werden kann.

In der Praxis existieren übrigens verschieden lange Unicode-Varianten und zwar eine lange mit 32 Bit (in der ISO-Norm mit UCS4 bezeichnet) und eine kurze mit 16 Bit (UCS2). Von der kurzen Variante kann in die lange leicht konvertiert werden, indem einfach die zusätzlichen Bits mit Nullen aufgefüllt werden. Außerdem gibt es bis jetzt noch keine Zeichen in UCS4, die nicht in UCS2 enthalten wären. Gängige Unicode-Implementierungen wie beispielsweise die in Java verwenden deshalb auch nur 16 Bit.

1.5 X Logical Font Description (XLFD)

Schriften werden unter X11 mit sog. XLFD-Strings eindeutig bezeichnet. Eine solche Zeichenkette hat folgendes Format:

-Herst.-Familie-Gewicht-Neigung-Breite-Stil-Pixel-Punkt-X-Y-Abst.-Durchschn.-Registr.-Kodierung

Dabei haben die einzelnen Felder folgende Bedeutung:

Hersteller: Bezeichnet die Herstellerfirma der Schrift, z. B. `adobe` oder `b&h`.

Familie: Name der Schriftenfamilie, z. B. `helvetica` oder `times`.

Gewicht: Beschreibung der Schriftdicke in Worten, z. B. `light`, `medium`, `bold` etc.

Neigung: Beschreibung der Neigung. Hierbei werden Abkürzungen verwendet: `r` (Roman, keine Neigung); `i` (Italic, Kursiv); `o` (Oblique, schräg); `ri` (Reverse italic, umgekehrt kursiv); `ro` (Reverse oblique, umgekehrt schräg); `ot` (sonstiges); *`Zahl`* (für skalierte Schrift).

Breite: Angabe für die Breite in Worten, z. B. `normal`, `condensed`, `narrow`, `double wide`.

Stil: Zusätzliche Stilbeschreibung, z. B. `serif`, `sans serif`, `informal`, `decorated`. Meistens bleibt dieses Feld leer.

Pixel: Höhe der Zeichen in Pixeln. Somit ist die tatsächlich auf dem Monitor dargestellte Größe abhängig von der verwendeten Bildschirmauflösung.

Punkt: Höhe der Zeichen in Punkt. Hierbei handelt es sich eine vom Betriebssystem errechnete Angabe die idealerweise der tatsächlich dargestellten Punktgröße (72 Punkt entsprechen einem Zoll)

X: Auflösung in der Horizontalen (Einheit: dpi).

Y: Auflösung in der Vertikalen (Einheit: dpi).

Abst.: Gibt an, ob es sich um eine Proportionalschrift (`p`) oder um eine Festbreitenschrift (`m`) wie z. B. Courier handelt.

Durchschn.: Dieses Feld gibt die zehnfache durchschnittliche Breite der Zeichen in Pixeln an.

Registr.: Name der Zeichensatzgruppe, z. B. `iso8859` oder `iso10646`.

Kodierung: Rest des Zeichensatznamens, z. B. `1` oder `15`.

2 Nützliche Programme

2.1 Zeichensatzumwandlung

Das unter Linux oder auch anderen Unix-Systemen gängige Umwandlungsprogramm heißt *recode*. Es wird folgendermaßen aufgerufen:

```
$ recode [Quellzeichensatz]..[Zielzeichensatz] [Dateiname]
```

Dann wird die Umwandlung vorgenommen. *recode* beherrscht sehr viele Zeichensätze. Eine Liste aller verfügbaren Zeichensätze kann man sich mit dem Befehl `recode -l` ausgeben lassen. Ein Beispiel zur Umwandlung von ISO 8859-1 in UTF-8 würde folgendermaßen aussehen:

```
$ latin1..utf-8 text.txt
```

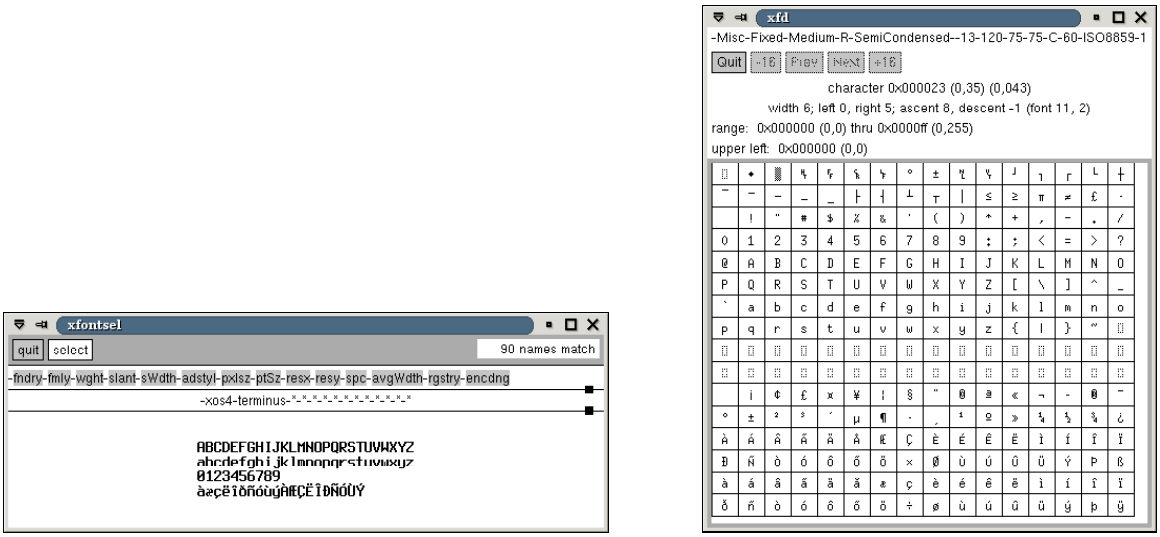

Abbildung 1: Die Programme *xfontsel* und *xfd*

2.2 Anzeige von Schriften

2.2.1 X11-Standardprogramme

X11 bringt einige Standardtools mit, u. a. um vorhandene Schriften anzuzeigen. Diese Programme sollten standardmäßig installiert sein, mindenstens aber im Lieferumfang Ihrer Linuxdistribution enthalten sein.

xlsfonts Zur Ausgabe einer Liste aller verfügbaren Schriften dient das Programm *xlsfonts*. Es macht im allgemeinen wenig Sinn, sich einfach die Schriften ausgeben zu lassen; vielmehr muss der Ausgabestrom mit Tools wie *grep* nach bestimmten Schriften durchsucht werden.

xfontsel Um sich die Schriften tatsächlich am Bildschirm anzeigen zu lassen, gibt es zunächst das Programm *xfontsel*. Es ermöglicht zum einen die Anzeige aller unter X installierten Schriften, zum anderen eine Auswahl als X-Logical-Font-Description-String (siehe 1.5). Damit kann man dann eine solche Schrift auswählen und gleich in die Zwischenablage die richtige Beschreibung kopieren.

xfd Dann gibt es noch das Programm *xfd*, das die Anzeige alle vorhandenen Zeichen einer Schrift ermöglicht. Aufgerufen wird das Programm `xfd -fn [Schrift]`, wobei bei `[Schrift]` eine XLFD erwartet wird (siehe 1.5).

Abbildung 1 zeigt einen Screenshot der beiden Programme.

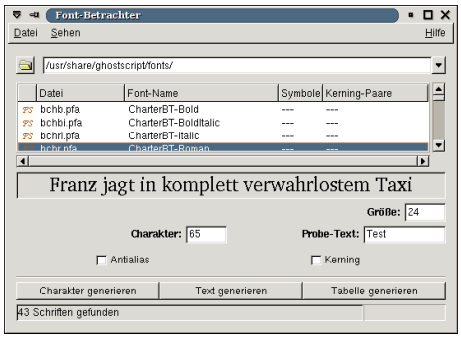

Abbildung 2: Das Programm *gfontview*

2.2.2 gfontview

Alle oben vorgestellten Programme setzen voraus, dass die Schriften bereits installiert sind. Sie besitzen zudem einen äußerst geringen Funktionsumfang. Dagegen setzt *gfontview* keine vorherige Installation der Schriften voraus, sondern kann einfach alle Fonts in einem bestimmten Verzeichnis anzeigen. Darüber hinaus lassen sich alle möglichen Informationen über die Schrift anzeigen, Übersichtsblätter drucken, Umwandlungen in Type42 vornehmen u. v. m.

Das Programm ist auf jeden Fall einen Blick wert und eignet sich vorzüglich, um beispielsweise von einer Schriften-CD-ROM einige Schriften auszuwählen, die dann installiert werden sollen. Die Abbildung 2 zeigt das Programm bei der Arbeit. Die Projekthomepage, von der das Programm heruntergeladen werden kann, befindet sich unter:

➠ http://gfontview.sourceforge.net

2.3 Schriftkonvertierung

Beachten Sie, dass Formatumwandlungen bei Schriften immer mit einem Qualitätsverlust verbunden sind. Führen Sie deshalb eine Konvertierung nur dann durch, wenn es keine andere Möglichkeit gibt, die Schrift zu nutzen!

2.3.1 Von TrueType nach Type1

Um Schriften von TrueType nach Type1 umzuwandeln, existiert das Programm *ttf2pt1*. Eine Bezugsquelle:

➠ http://ttf2pt1.sourceforge.net

Aufgerufen wird das Programm mit: `ttf2pt1 -b -l [Sprache] [Font.ttf] [Font]`.

Dabei steht `[Sprache]` für das Encoding bei der Umwandlung. Für eine Auswahl dient der Parameter `-h`. Ohne Angabe des Encodings wird ISO Latin 1 erzeugt, siehe auch Abschnitt 1.4.

2.3.2 Von BDF nach Type1

Um BDF-Schriften (X11-Bitmapfonts) nach Type1 zu konvertieren, liefert Ghostscript das Tool *bdftops* mit. Die Syntax lautet:

```
$ bdftops BDF_filename [AFM_file1_name ...] gsf_filename fontname
            UniqueID [XUID] [encodingname]
```

Eine genauere Beschreibung finden Sie in der Datei Fonts.htm, die sich doc-Verzeichnis von Ghostscript befindet, z. B. /usr/share/ghostscript/7.05/doc.

2.4 Vollautomatische Schrifteninstallation

Das bekannteste Programm, das verspricht, Schriften automatisch in allen möglichen Programmen (Ghostscript, StarOffice, X11) zu installieren, heißt *kfontinst* und ist im KDE-Desktop enthalten.

Ich habe das Programm nie verwendet, kenne aber dennoch einige Erfahrungsberichte aus Mailinglisten, die z. T. nicht sehr positiv waren. Manchmal wurde die komplette Schrifteninstallation zerschossen, d. h. Schriften, die vorher bereits in StarOffice funktionierten, gingen nicht mehr. Deshalb empfehle ich, von solchen Programmen die Finger zu lassen und stattdessen die wenigen Schritte von Hand durchzuführen.

2.5 Komplette Schriftenverwaltung

Wenn Sie sehr viele Schriften besitzen möchten Sie möglicherweise eine komfortable Verwaltung Ihrer Schriften mit Vorschaufunktion, Klassifizierung usw. Unter

➠ http://www.gesindel.de

finden Sie das Programm *Fontlinge*, das genau dies tut: Es benennt Schriftdateien in einen lesbaren Namen um und bietet ein komfortables Web-Frontend zur Verwaltung und Ansicht.

Obwohl ich die Software bisher noch nicht getestet habe – so viele Schriften habe ich nicht, dass sich der Aufwand lohnt – sehen die Screenshots doch sehr vielversprechend aus.

3 Neue Schriften installieren

Zwar existieren Programme, die eine automatische Schrifteninstallation mit einer graphischen Oberfläche ermöglichen, siehe 2.4. Dennoch empfehle ich aus den genannten Gründen, eine Installation von Hand mit Kommandozeilentools durchzuführen. Dies erfordert nur wenig Mehraufwand, ist aber sicherer und ermöglicht es, bei Problemen einfacher einzugreifen.

Grundsätzlich ist das Verfahren für Type1- und TrueType-Schriften ähnlich. Deshalb erfolgt keine getrennte Beschreibung.

3.1 Vorbereitungen

Zunächst muss ein Verzeichnis erstellt werden, in das die Schriftdateien später kopiert werden können. Zum Beispiel könnte man /usr/X11R6/lib/X11/fonts/truetype oder /usr/share/fonts/corel verwenden.

Ich empfehle, bei größerer Anzahl von Schriften mehrere Verzeichnisse zu verwenden, z. B. alphabetisch sortiert oder nach Hersteller/Herkunft. Dies erleichtert Übersicht und Fehlersuche. Außerdem empfehle ich, keine eigenen Schriften in Verzeichnisse zu installieren, in denen sich bereits Schriften befinden. Damit lassen sich leichter Backups anfertigen, und bei einer Neuinstallation müssen nur noch die entsprechenden Verzeichniss kopiert und in X11 eingebunden werden.

Um die Arbeit zu vereinfachen, sollten Sie für Type1- und TrueType-Schriften getrennte Verzeichnisse verwenden. Zwar ist es ohne weiteres möglich, solche Schriften in ein gemeinsames Verzeichnis zu kopieren, es werden aber getrennte Tools benötigt, um die Übersichtsdatei zu erzeugen (dazu später mehr), und somit ist es einfacher, getrennte Verzeichnisse zu verwenden.

Bei TrueType-Schriften kopieren Sie in dieses Verzeichnis alle ttf-Dateien, bei Type1-Schriften alle pfb- oder pfa- sowie afm-Dateien. Achten Sie darauf, dass zu jeder Schriftdatei (also pfb/pfa) eine bis auf die Endung gleichnamige afm-Datei exisitert.

3.2 XFree86

Dieses Verfahren funktioniert nur mit XFree86 ab Version 4.0. Auf frühere Versionen werde ich nicht eingehen. Stellen Sie sicher, dass Sie eine solche Version verwenden. Benutzen Sie dazu den Befehl `X -version`.

Nun muss zunächst eine Art Inhaltsverzeichnis erstellt werden. Diese Übersichtsdatei heißt fonts.dir und ist folgendermaßen aufgebaut:

- In der ersten Zeile steht eine Zahl, die die Anzahl der Schriften angibt. Normalerweise handelt es sich dabei einfach um die Anzahl der Zeilen der Datei minus eins.
- Danach folgt für jede Schrift mindestens eine Zeile. Soll eine Schrift X11 in mehreren Encodings zur Verfügung gestellt werden, so sind mehrere Zeilen erforderlich. Natürlich ist für jede einzelne Schriftdatei, also für jeden Schriftschnitt (Fett, Kursiv), ein Eintrag erforderlich.
- Am Zeilenanfang steht die Schriftdatei, z. B. arialb.ttf oder denv.pfb. Nach einem Leerzeichen folgt ein XLFD-String (siehe 1.5).

Natürlich gibt es Programme, die solche Dateien selbstständig erzeugen:

3.2.1 Übersichtsdatei für TrueType-Schriften erzeugen

Die Übersichtsdatei kann bei neueren Versionen von XFree86 (meines Wissens ab 4.3.0) mit dem Programm *mkfontscale* erzeugt werden. Das Programm kann auch unabhängig installiert werden. Die Dateien kann man sich entweder im CVS[5]-Baum von XFree86 besorgen oder auf meiner Homepage als Snapshot für alle, die mit CVS weniger vertraut sind:

➠ http://www.bwalle.de/docs/mkfontscale-20030817.tar.gz

Zur Installation sind folgende Schitte erfolderlich:

```
$ tar xvfz mkfontscale-20030817.tar.gz
$ cd mkfontscale
$ xmkmf
$ make
$ su -c "make install"
```

Es wird eine neuere Version der FreeType-Bibliothek benötigt. Das Paket inkl. dem Developement-Paket müssen installiert sein, ansonsten kompiliert das Programm nicht und es wird eine Fehlermeldung ausgegeben. Nun kann man mit dem Befehl

```
$ mkfontscale
```

im entsprechenden Schriftenverzeichnis eine Indexdatei erzeugen.

Ein anderes Programm, das die gleiche Aufgabe erfüllt, heißt *ttmkfdir*. Es wurde von Jörg Pommnitz entwickelt. Es befindet sich entweder im Lieferumfang Ihrer Linux-Distribution oder kann unter folgender Adresse heruntergeladen werden:

➠ http://www.joerg-pommnitz.de/TrueType/ttmkfdir.tar.gz

Da das Programm länger nicht mehr weiterentwickelt wurde, enthält jede Distribution ihre eigene Version. Das ist auch der Grund, warum man es nicht mehr verwenden sollte, falls *mkfontscale* zur Verfügung steht.

Um die Übersichtsdatei zu erzeugen, wechselt man einfach in das Verzeichnis und führt folgendes Kommando aus:

```
$ ttmkfdir -o fonts.dir
$ ln -s fonts.dir fonts.scale
```

Normalerweise sollte das Programm ohne Fehlermeldungen ausgeführt werden. Der letzte Befehl (Anlegen des symbolischen Links) muss nur einmal ausgeführt werden.

[5] Versionsverwaltungssystem, das Entwickler zum Verwalten ihrer Quellcodedateien verwenden. Neben CVS gibt es noch andere Systeme, CVS ist aber das populärste.

3.2.2 Übersichtsdatei für Type1-Schriften erzeugen

Auch für Type1-Schriften eignet sich das oben vorgestellte Tool *mkfontscale*. Somit ist es ohne Probleme möglich, beide Schriftformate in einem Verzeichnis zu mischen. Trotzdem empfehle ich eine Trennung wegen der besseren Übersicht.

Ansonsten eignet sich noch das Programm *type1inst* von JAMES MACNICOL. Es befindet sich entweder im Lieferumfang Ihrer Linux-Distribution oder kann unter folgender URL heruntergeladen werden:

➠ http://www.ibiblio.org/pub/Linux/X11/xutils/type1inst-0.6.1.tar.gz

Zum Erzeugen der fonts.dir und fonts.scale führen Sie folgendes Kommando aus:

```
$ type1inst -nogs
```

Nachdem die Dateien fonts.dir und fonts.scale erzeugt wurden, müssen X11 die neuen Schriften bekannt gemacht werden.

Haben Sie die Schriften in ein Verzeichnis kopiert, das bereits als `FontPath` in der X11-Konfigurationsdatei eingetragen war, muss X nur die Schriftenverzeichnisse neu einlesen. Dazu reicht es, den Befehl `xset fp rehash` auszuführen.

Neue Verzeichnisse kann man X11 vorübergehend mit dem Kommando `xset +fp [Pfad]` bekannt machen; genauso kann man sie mit `xset -fp [Pfad]` wieder entfernen.

Prüfen Sie mit dem Programm *xfontsel*, ob die Schriften ordnungsgemäß angezeigt werden. Ist dies der Fall, können Sie den Pfad dauerhaft in der X11-Konfigurationsdatei – in der Regel /etc/X11/XF86Config – hinzufügen. Dazu tragen Sie folgende Zeile im Abschnitt `Files` ein:

```
FontPath      "/usr/share/fonts/userfonts"
```

Natürlich müssen Sie hier das Verzeichnis eintragen, in das Sie vorher die Schriftdateien kopiert haben und in dem sich die Übersichtsdateien fonts.dir und fonts.scale befinden.

Nun sollten die neuen Schriften in den meisten X-Programmen zur Bildschirmanzeige zur Verfügung stehen, beispielsweise in Mozilla/Netscape, in KDE-Programmen oder auch in GIMP.

Ausnahmen sind eigentlich nur Programme, die vorwiegend auf das Ausdrucken optimiert sind und daher nur Schriften zur Auswahl anbieten, die auch ausgedruckt werden können. Da unter Linux keine einheitliche Verbindung von Bildschirmanzeige und Ausdruck existiert, existieren verschiedene Lösungen. Einige Programme (wie StarOffice 5.2 oder ApplixWare) setzen daher eine zusätzliche Installation der Schriften in der betreffenden Anwendung voraus.

3.3 OpenOffice.org 1.0.x und StarOffice 6

Grundsätzlich unterscheidet OpenOffice.org zwischen Schriften, die als Datei vorliegen, und Schriften, die nur über X11 angezeigt werden können. Um zu drucken, benötigt das Programm die Schrift als Datei (sei es TrueType oder Type1), außer es handelt sich um eine Schrift, die im Drucker eingebaut ist. Schriften, die nicht gedruckt werden können, werden im Schriftauswahldialog der Textverarbeitung nicht angeboten – hingegen schon bei der Erstellung einer Bildschirmpräsentation.

OpenOffice.org sucht in folgenden Quellen nach Schriftdateien, und zwar in der angegebenen Reihenfolge:

1. Im Verzeichnis /usr/X11R6/lib/X11/fonts/Type1.
2. Ausgabe der Kommandos `/usr/sbin/chkfontpath` und `chkfontpath`.
3. Pfade, die von der Funktion `XGetFontPath()` zurückgegeben werden. Dies sind alle Pfade, die in der X11-Konfigurationsdatei als `FontPath` eingetragen sind.
4. Verzeichnisse, die durch die Umgebungsvariable `SAL_FONTPATH_PRIVATE` angegeben werden. Standardmäßig wird diese durch das soffice-Startskript auf /usr/X11R6/lib/X11/fonts/truetype gesetzt.

Konkret bedeutet dies, dass OpenOffice.org in allen Verzeichnissen nach Schriftdateien sucht, die in der X11-Konfigurationsdatei angegeben wurde. Somit reicht es, die Schriften auf die oben angegebene Art und Weise in XFree86 hinzuzufügen.

Sollen bestimmte Schriften ausschließlich in OpenOffice.org installiert werden, so bietet das Programm *spadmin („OpenOffice.org Druckerverwaltung")* entsprechende Optionen an.

Das Programm erlaubt außerdem eine Schriftenersetzung. Durch Aktivieren dieser Option wird z. B. Arial durch Helvetica ersetzt. Dies erlaubt es, am Bildschirm die Arial zu verwenden, wodurch eine bessere Bildschirmdarstellung erreicht wird, und trotzdem Helvetica zu drucken, welche eine bessere Druckqualität und kleinere Dateien liefert. (Helvetica gehört zu den Standard-Postscript-Schriften, die in jedem Drucker und auch in Ghostscript integriert sind.) Dies ist möglich, weil beide Schriften die gleichen Metriken verwenden können. Ein interessanter Artikel zu „Helvetica vs. Arial" finden Sie unter [11].

Weitere Informationen über die Handhabung von Schriften in OpenOffice.org finden sich in [10].

3.4 Ghostscript

Normalerweise ist es nicht nötig, die Schriften Ghostscript bekannt zu machen. Das Anwendungsprogramm sollte die Schriftinformationen in die Postscript-Datei einbetten. Trotzdem kann es einige Anwendungsfälle geben, wo es dennoch sinnvoll ist, dass die Schriften in der Ghostscript-Konfigurationsdatei eingetragen werden.

Die Schriften werden alle in eine Datei mit dem Namen Fontmap.GS eingetragen, die sich normalerweise im Verzeichnis /usr/share/ghostscript/6.53/lib befindet. Der genaue Pfad kann von der verwendeten Version bzw. von den Installationsoptionen abhängen. Verwenden Sie *locate* oder *find*, um die Datei zu finden. Hier zwei Beispiele für einen korrekten Eintrag:

```
/Arial-Italic  (ariali.ttf)  ;
/OrigGarmndBT  (/usr/X11R6/lib/X11/fonts/truetype/tt0863m_.ttf) ;
```

Ghostscript verarbeitet sowohl ttf- als auch pfa- und pfb-Dateien und benötigt keine Metriken.

Das Tool *type1inst* kann auch die Fontmap-Datei aktualisieren, wenn es ohne die Option `-nogs` aufgerufen wird.

3.5 WordPerfect

Unter [9] finden Sie Informationen über die Schrifteninstallation unter WordPerfect 8 für Linux. Da ich das Programm selbst nicht verwende und die Verbreitung mittlerweile sehr gering sein dürfte, soll dieser Hinweis genügen.

4 Das Eurosymbol

Seit 1. Januar 2001 ist er da: der Euro. Und mit ihm wird sich auch das Eurozeichen zunehmend verbreiten. Es ist eigentlich kein Schriftzeichen sondern ein Logo. Konkret bedeutet dies, dass nach den Richtlinien der Europäischen Kommission keine Anpassung an die verwendete Schrift erfolgt, sondern dass es immer in seiner ursprünglichen Form verwendet werden sollte.[6] Trotzdem liefert aber praktisch jede neuere Schrift eine angepasste Form mit.

Mittlerweile scheint sich die EU-Kommission damit abgefunden zu haben, dass angepasste Versionen des Eurozeichens verwendet werden. In den Dokumenten "Notes on the €uro Symbol – its generation and printing"[7] und "The current state of Euro sign implementation and standardisation"[8] verwendet die EU selbst angepasste Versionen und es ist auch die Aussage zu finden, dass gegen angepasste Versionen im alltäglichen Umgang (d. h. in „nicht-offiziellen" Dokumenten) nichts einzuwenden ist.

Obwohl die aktuellen Linux-Distributionen eigentlich schon perfekt mit dem Euro umgehen können sollten, beschreibe ich hier trotzdem die Einzelheiten beim Umgang mit dem Eurosymbol. Zum einen gibt es immer noch Probleme, zum anderen hilft Grundwissen bei der Problembehebung.

4.1 Die korrekte Sprachumgebung

In Windows werden über die Systemsteuerung die Ländereinstellungen, z. B. die Sprache, der Dezimaltrenner oder das Währungssymbol, gesetzt. Unter Linux geschieht dies über Umgebungsvariablen.

Normalerweise wird nur die `LANG`-Variable entsprechend gesetzt. Um die Umgebungsvariable dauerhaft zu setzen trägt man die Befehle am besten in die Datei ~/.profile (nur für einen Benutzer) oder /etc/profile (systemweit) ein. Einige Distributionen erlauben, die Sprachumgebung komfortabel über

[6] Unter http://europa.eu.int/euro/html/dossiers/00203/html/index-DE.html finden Sie die offiziellen Konstruktionshinweise für das Eurosymbol.
[7] http://europa.eu.int/ISPO/y2keuro/docs/euronotes.pdf
[8] http://europa.eu.int/ISPO/y2keuro/docs/eurosign.doc

ein Tool auszuwähen. Bei SuSE Linux kann man dazu entweder direkt YaST2 verwenden oder die Datei /etc/profile.local editieren. Bei Debian kann die Einstellung mit dem Befehl

```
$ dpkg-reconfigure locales
```

vorgenommen werden. Je nach Konfiguration erscheint dann ein graphischer oder ein Konsolendialog, wo die Auswahl vorgenommen werden kann. Wenn die Einstellung nur für einen einzelnen Benutzer vorgenommen werden soll, muss aber immer der Weg „von Hand“ genommen werden.

Für eine normale deutschprachige Umgebung empfiehlt sich:

```
$ export LANG="de_DE@euro"
```

Für eine englische Umgebung, wo trotzdem der Euro verwendet werden soll, empfiehlt sich:

```
$ export LANG="en_EN"
$ export LC_MONETARY="de_DE@euro"
$ export LC_CTYPE="de_DE@euro"
```

Weitere Informationen findet man in der Manual Page *locale(1)*. Eine sehr gute Übersicht zum Thema Internationalisierung mit Schwerpunkt Unicode und asiatische Schriften findet man unter [15].

4.2 Das Eurozeichen und die Zeichensätze

Als das Eurozeichen entworfen wurde, begann zunächst Adobe, es in seine Schriften einzubauen.[9] Als Platz in der Schrift wurde die Position 0x80 (dezimal 128) gewählt, da diese noch frei war. Microsoft zog in seinen Schriften nach.

Unter Unix-Betriebssystemen, wo der Zeichensatz Latin 1 verbreitet war, war dies nicht so einfach möglich, da in diesem Bereich eine Lücke herrscht, siehe auch 1.4. Man beschloss daher, einen neuen Zeichensatz ISO 8859-15 zu schaffen, und darin das allgemeine Währungssymbol (¤), das sowieso keine große Verbreitung und Verwendung fand, durch das Eurosymbol zu ersetzen. Die Stelle im Zeichensatz ist 0xA4 (dezimal 164).

Durch dieses unterschiedliche Vorgehen sind Probleme vorprogrammiert: Streng genommen existiert das Symbol in dem verbreiteten Zeichensatz Latin 1 überhaupt nicht. Der Datenaustausch über verschiedene Plattformen hinweg wird so zur Einbahnstraße. Viele Mailprogramme kennen den neuen Zeichensatz Latin 9 überhaupt nicht, sodass das Eurozeichen nicht entsprechend kodiert wird. Stellen Sie sich also auf ein unlesbares Zeichen von Windows-Mailclients ein und umgekehrt.

Deshalb sollte man in solchen Fällen bis auf weiteres auf das Eurozeichen verzichten – die Abkürung *EUR* tut es auch und ist im Bankbereich, wo auch das Dollarzeichen nicht existiert, sowieso Standard.

[9] Adobe stellt unter http://www.adobe.com/type/eurofont.html kostenlose Schriften zum Download zur Verfügung, die ausschließlich das Eurozeichen in verschiedenen Stilen beinhalten. Die Schrift *EuroSans* beinhaltet das offizielle Eurozeichen nach den Richtlinien der Europäischen Kommission.

Im Fließtext finde ich es ohnehin stilistisch besser, das Wort *Euro* auszuschreiben. Das stört den Textfluss nicht so wie ein Sonderzeichen oder gar wie ein Symbol.

Bei gedruckten Dokumenten hingegen sieht es schön aus, spart Platz (z. B. in Preisübersichten) und kann natürlich auch verwendet werden – hier unten wird beschrieben, wie.

Für die Zukunft bleibt zu hoffen, dass sich Unicode auf allen Betriebssystemen durchsetzt, womit das Eurosymbol auf allen Systemen an gleicher Stelle wäre und hiermit auch dieses Problem für alle Zeiten gelöst wäre. OpenOffice.org macht mit XML und Unicode vor, wie die Zukunft aussehen kann.

Die offizielle Definition des ISO-8859-15-Zeichensatzes findet man übrigens unter [7].

4.2.1 Schriften mit Eurozeichen installieren

Zunächst müssen Schriften mit Eurozeichen auf dem System verfügbar sein. Für die Bildschirmanzeige sind die Microsoft-TrueType-Schriften ein guter Ansatzpunkt. Werden sie korrekt im Zeichensatz Latin 9 installiert wie oben beschrieben[10], dann steht auch das Eurozeichen an richtiger Stelle zur Verfügung.

Einige unter XFree86 4.x mitgelieferte Schriften stehen standardmäßig bereits im Latin-9-Zeichensatz zur Verfügung und enthalten auch das Eurozeichen.

4.3 Eingabe mit der Tastatur

Normalerweise sollte X11 bereits entsprechend konfiguriert sein, sodass Sie das Eurozeichen auf deutschen Tastaturen mit AltGr+e eingeben können. Wenn Sie keine deutsche Tastatur verwenden, so hilft möglicherweise die Compose-Taste (normalerweise die rechte Strg-Taste, manchmal auch in Verbindung mit Shift). Drücken Sie zunächst Compose, dann e und schließlich =.

Funktioniert beides nicht, so kann der folgenden Eintrag in der Datei ~/.Xmodmap oder /etc/X11/Xmodmap bewirken, dass das Zeichen über die Tastenkombination AltGr+e eingegeben werden kann:

```
keycode  26 = e E EuroSign
```

Funktioniert auch das nicht, probieren Sie es mit dieser Zeile:

```
keycode  26 = e E 0x00a4
```

Die Änderungen werden entweder nach einem Neustart des X-Window-Systems wirksam, oder nachdem Sie den Befehl `xmodmap /etc/X11/Xmodmap` bzw. `xmodmap ~/.Xmodmap` ausgeführt haben.

[10] Das Programm *ttmkfdir* erkennt selbstständig, ob eine Schrift auch in der Kodierung ISO-8859-15 vorliegt, und erstellt die fonts.dir-Datei entsprechend.

4.4 Das Eurosymbol in verschiedenen Anwendungen

Unabhängig vom Eurosymbol finden Sie in diesem Abschnitt Tipps zum Ändern der Schriftarteinstellung der betreffenden Anwendungen. Die Vorgehendweise eignet sich also auch wenn Ihnen die Standardschrift z. B. nicht gefällt oder zu klein ist.

4.4.1 X-Terminals

Die Schrift von *XTerm* kann global in der Datei ~/.Xdefaults definiert werden. Möglich wäre beispielsweise folgende Einstellung:

```
XTerm*font: -misc-fixed-medium-r-normal--15-140-75-75-c-90-iso8859-15
```

Damit die Änderungen wirksam werden (natürlich nur bei neu gestarteten X-Terminals) ist entweder ein X-Neustart erforderlich oder man führt das Kommando `xrdb -merge ~/.Xdefaults` aus.

Zum Ausprobieren, d. h. um die Schrift in einem bereits gestarteten *XTerm* zu ändern ist folgendes Vorgehen möglich:

1. Zunächst muss im Programm *xfontsel* eine Schrift ausgewählt werden.
2. Dann muss diese Auswahl über den *select*-Button in die Zwischenablage kopiert werden.
3. Schließlich kann diese Schrift in einem *XTerm* verfügbar gemacht werden indem man im *VT-Fonts*-Menü, welches erscheint wenn man die STRG-Taste gedrückt hält und die rechte Maustaste betätigt, den Eintrag *Selection* auswählt.

4.4.2 OpenOffice.org

Da OpenOffice.org zum Anzeigen der Schriften – wenn möglich – nicht X11, sondern die Freetype-Bibliothek verwendet und die Schriftdatei ausliest, gibt es unabhängig von der X11-Installation keine Probleme mit dem Eurozeichen.

4.4.3 Gtk+-Programme

Die einzige Voraussetzung ist, dass der Font im Zeichensatz Latin 9 vorliegt.

Gtk+ 1.x

Der Standardfont für Beschriftungselemente kann z. B. mit dem Gnome-Kontrollzentrum ausgewählt werden. Alternativ trägt man in der Datei ~/.gtkrc den zu verwendeten Font gemäß diesem Beispiel ein:

```
style "user-font"
{
    font="-adobe-helvetica-medium-r-normal-*-*-120-*-*-p-*-iso8859-15
}
widget_class "*" style "user-font"
```

Gtk+ 2.x

Bei Gtk+ 2.x sind keine weiteren Maßnahmen erforderlich, um das Eurosymbol anzuzeigen. Um eine andere Schriftart oder einen anderen Schriftgrad zu verwenden, kann folgender Eintrag in die Datei ~/.gtkrc-2.0 vorgenommen werden:

```
style "user-font"
{
    font_name = "Helvetica 11"
}
widget_class "*" style "user-font"
```

4.4.4 QT-/KDE-Programmen

Es kommt v. a. auf die Einstellungen im KDE-Kontrollzentrum an. Normalerweise sollte es keine Probleme mit dem Eurozeichen geben.

Stellen Sie unter *Kontrollzentrum, Erscheinungsbild, Schriften* alle Fonts auf den Zeichensatz *iso 8859-15* um, und wählen Sie außerdem unter *Persönliche Einstellungen, Land & Sprache* als Zeichensatz *iso8859-15* aus.

4.4.5 LaTeX-Dokumente

In diesem Dokument wurde das in der Schrift mitgelieferte Eurosymbol genutzt, das über die Anweisung `\usepackage[latin9]{inputenc}` im Dokumentkopf auch direkt über die Tastatur eingegeben werden kann (ansonsten müsste `\texteuro` verwendet werden). Es gibt aber noch viele Alternativen, die über das Thema dieses Dokuments hinausgehen würden. Ich verweise auf [5], Punkt 8.1.6 sowie v. a. auf [8].

4.4.6 Vim und GVim

Auf der Konsole muss dafür gesorgt werden, dass die korrekte Terminalschrift eingestellt ist, d. h., dass diese im Latin-9-Encoding vorliegt.

Folgender Eintrag in der Datei ~/.gvimrc schaltet die verwendete Schriftart und das Encoding im graphischen Vim um:

```
set guifont=-misc-fixed-medium-r-normal--15-140-75-75-c-90-iso8859-15
set encoding=iso-8859-15
```

4.4.7 Textdateien drucken

Zum Drucken von Textdateien verwendet der Druckerfilter in der Regel die Programme *enscript* oder *a2ps*. *a2ps* erkennt den richtigen Zeichensatz automatisch über die eingestellte Sprachumgebung. Funktioniert das nicht richtig, so kann in der Konfigurationsdatei /etc/a2ps.cfg der Zeichensatz explizit durch folgenden Eintrag auf Latin 9 umgestellt werden:

```
# Standardzeichensatz auf ISO-8859-15 wegen Eurosymbol
Options: --encoding=latin9
```

Verwenden Sie hingegen *enscript*, so sollte folgender Eintrag in der Datei /etc/enscript.cfg bewirken, dass das Encoding auf Latin 9 umgestellt wird:

```
# Standardzeichensatz auf ISO-8859-15 wegen Eurosymbol
DefaultEncoding: latin9
```

Ist bereits ein ähnlicher Eintrag vorhanden, so sollte er abgeändert werden. Alternativ kann der Zeichensatz beim Drucken einer einzelnen Datei verändert werden, indem die Option `--encoding=latin9` hinzugefügt wird. Diese Option wird sowohl von *a2ps* als auch von *enscript* verstanden. Allerdings kann dann nicht mit dem *lpr*-Kommando, sondern nur direkt mit *a2ps* bzw. *enscript* gedruckt werden. Allerdings bietet dies auch den Vorteil dass das Aussehen des Ausdrucks gezielter beeinflusst werden kann. Lesen Sie hierzu auch die Manpages *enscript(1)* bzw. *a2ps(1)*.

4.4.8 Das Eurozeichen in Mails mit Mutt

Mit der zunehmenden Verbreitung des Euro wird auch die Notwendigkeit entstehen, das Zeichen in E-Mails zu verwenden. Bei graphischen Mailclients ist das Ganze Toolkit-abhängig. Da ich selber Mutt verwende, beschreibe ich hier mal das Vorgehen in Mutt.

Zunächst muss sichergestellt werden, dass das verwendete Terminal den Latin-9-Zeichensatz unterstützt. Dazu einfach das Eurozeichen am Prompt eingeben und schauen, ob es angezeigt wird. Wenn nicht, können Sie in 4.4.1 nachlesen, welches Vorgehen erforderlich ist.

Weiterhin muss sichergestellt werden, dass die Umgebungsvariablen richtig gesetzt sind, siehe 4.1. Schließlich muss eine neuere Mutt-Version verwendet werden, entweder die Entwicklerversionen 1.3.x oder die neue stabile Version 1.4.x. Welche Version Sie einsetzen, erhalten Sie durch das Kommando `mutt -v`.

Als einzige Änderung an der Mutt-Konfiguration empfiehlt es sich, die Sendezeichensätze zu ändern. Normalerweise würde Mutt, wenn Sie das Eurosymbol in einer Mail verwenden, die gesamte Mail in UTF-8 kodieren. Da dies bei vielen Programmen zu Probleme führen kann, ist es sinnvoller, ISO-8859-15 zu verwenden. Fügen Sie folgenden Eintrag in Ihrer ~/.muttrc oder /etc/Muttrc hinzu:

```
# Sendezeichensätze
set send_charset="us-ascii:iso-8859-1:iso-8859-15:utf-8"
```

Damit steht der europäischen Kommunikation nichts mehr im Wege!

5 Anti-Aliasing von Bildschirmschriften

5.1 Allgemeines

Anti-Aliasing bezeichnet man ein Verfahren zur Kantenglättung von Kurven. Dies wird erreicht, indem die schwarzen „Treppen“ durch graue Felder aufgefüllt werden und so ein optisch geglätteter Eindruck entsteht.[11]

Erst mit XFree86 4.0.2 wurde ein Fontrenderer eingebaut, der Anti-Aliasing beherrscht. Dies geht aber nur mit skalierbaren Schriften im Type1- oder TrueType-Format. Erst muss sichergestellt werden, dass der X-Server mit Xft-Unterstützung konfiguriert wurde. Dies geschieht mit dem Befehl `xdpyinfo | grep RENDER`. Wird hier `RENDER` angezeigt, so ist diese Erweiterung mit einkompiliert worden. Außerdem gibt es Probleme mit manchen Grafiktreibern, beispielsweise mit den kommerziellen NVidia-Treibern. Die bei X selbst mitgelieferten sollten aber funktionieren.

5.2 Konfiguration

Die Konfigurationsdatei für Anti-Aliasing heißt XftConfig und befindet sich hier im Verzeichnis /usr/X11R6/lib/X11 (was aber nicht heißt, dass das bei Ihnen auch so sein muss; im Zweifel fragen Sie *locate*). Diese Datei ist eigentlich selbsterklärend. Primär müssen alle Pfade eingetragen werden, in denen sich Schriften befinden, die für Anti-Aliasing geeignet sind.

✎ Einige Dateien im Ghostscript-Pfad scheinen X zu stören. Deshalb am besten ein neues Verzeichnis anlegen und die entpsrechenden Schriften (Dateien mit den Endungen pfa, pfb und am besten noch afm) sowie die fonts.dir und fonts.scale verlinken. Störend wirken die gsf-Dateien.

Außerdem kann jeder Benutzer auf dem System in seinem Homeverzeichnis seine eigene Konfigurationsdatei mit dem Namen ~/.xftconfig anlegen, die dann entsprechend berücksichtigt wird. Dazu muss die systemweite Datei entsprechend vorbereitet sein, was aber bei aktuellen Linux-Distributionen der Fall sein sollte.

Insbesondere auf zwei Konfigurationsmöglichkeiten möchte ich hier eingehen. Zum einen ist es sinnvoll Anti-Aliasing erst bei einer bestimmten Schriftgröße einzuschalten, da bei kleinen Schriften oft eine Qualitätsverschlechterung statt -verbesserung eintritt. Durch den Eintrag

[11] Genauere Erklärungen mit Beispiel unter http://www.teamone.de/selfhtml/tif.htm auf den Seiten von SelfHTML.

```
match
    any size > 8
    any size < 15
edit
    antialias = false;
```

wird die Kantenglättung bei Schriftgrößen zwischen 8 und 15 Punkten deaktiviert. Außerdem gibt es eine Option, welche die Qualität des Subpixel-Rendering bei LCD-Bildschirmen verbessert:

```
match
edit
    rgba = bgr;
```

5.3 Unterstützung durch Anwendungsprogramme

Damit ist XFree86 entsprechend konfiguriert. Dies heißt aber noch lange nicht, dass die darauf aufbauenden Programme diese Erweiterung auch nutzen können. Momentan ist dies mit den QT- und Gtk-Bibliotheken möglich, und zwar ab QT 2.3.0 und Gtk 2.0. Bei QT muss die Unterstützung für Anti-Aliasing mit einkompiliert worden sein, die entsprechende Option für das `configure`- Skript lautet `-xft`. Damit Anti-Aliasing eingeschaltet ist, muss die Variable `QT_XFT` auf `Y` oder `1` gesetzt sein.

Auch das neue Gtk+ 2.0 unterstützt Anti-Aliasing. Um genau zu sein ist eignentlich die Hilfsbibliothek GDK für das Zeichnen von Zeichen zuständig und so ist die Anti-Aliasing-Unterstützung in GDK implementiert. Die Umgebungsvariable zur Steuerung des Verhaltens heißt `GDK_USE_XFT`.

OpenOffice.org 1.0.x verwendet zur Bildschirmanzeige der Schriften – mit Ausnahme von reinen Bildschirmschriften und den Menüschriften – übrigens nicht diese Erweiterung von X11, sondern die Freetype-Bibliotheken. Die Anti-Aliasing-Option lässt sich folgendermaßen ein- und ausschalten: *Extras, Optionen, OpenOffice.org, Ansicht, Bildschirmschriften glätten*.

✎ Möglicherweise funktioniert das Anti-Aliasing von StarOffice oder OpenOffice.org nur bei einer Farbtiefe von 24 Bit. Um die Farbtiefe zu ändern, muss die Datei /etc/X11/XF86Config angepasst werden oder die Konfiguration des X-Servers mit Hilfe der distributionseigenen Tools verändert werden. Bei SuSE Linux hilft beispielweise SaX2 weiter.

Ich habe diesen Hinweis von einem Leser bekommen, kann ihn zwar hier nicht nachvollziehen aber vielleicht hilft er Ihnen trotzdem.

✎ Die Schriftenanzeige bei OpenOffice.org lässt sich – ganz unabhängig davon, ob Anti-Aliasing ein- oder ausgeschaltet ist – deutlich verbessern, wenn man statt der im Programm integrierten FreeType-Bibiliothek die systemeigene verwendet. Dazu muss vor dem Start von OpenOffice.org die Umgebungsvariable `LD_PRELOAD` auf den vollständigen Pfad der installierten FreeType-Bibiliothek gesetzt werden, z. B. `/usr/X11R6/lib/libfreetype.so.6`.

Am besten fügt man dies direkt in die Startdatei von OpenOffice.org ein, also beispielsweise /opt/OpenOffice.org1.0/program/soffice. Bei mir steht hier in Zeile 79 der folgende Befehl:

```
LD_PRELOAD=/usr/lib/libfreetype.so.6
export LD_PRELOAD
```

Allerdings hat dies zumindest bei mir den Nebeneffekt, dass bestimmte Schriftnahmen im Auswahlmenü nicht mehr angezeigt werden. Möglicherweise hängt dies aber auch mit der Version der genannten Bibliothek zusammen, die mit Ihrer Linux-Distribution geliefert wird.

6 Erzeugung von PDF-Dateien

PDF-Dateien eignen sich vorzüglich für den Datenaustausch. Die Vorteile liegen v. a. darin, dass

- das Dokument beim Empfänger exakt so ankommt, wie es sein soll, egal, ob er bestimmte Schriften installiert hat oder welchen Drucker er verwendet;
- der zur Anzeige nötige Acrobat Reader von Adobe kostenlos bezogen werden kann und heute auf fast jedem Computerystem schon installiert ist;
- das PDF-Format offengelegt wurde und deshalb eine große Zahl an Software verfügbar ist, um PDF-Dateien zu erstellen und anzuzeigen;
- das Format in keinster Weise an eine bestimmte Plattform gebunden ist und
- keine Gefahr besteht, dass der Rechner von Viren o. ä. infiziert wird, wenn die Datei geöffnet wird.

Gerade für Linuxer, die nicht mit Microsoft Word arbeiten und dies als „Standard“ ansehen, besteht daher ein großes Interesse an der Erzeugung solcher Dateien, um Daten beispielsweise per E-Mail auszutauschen.

Unter Linux hat man den Vorteil, dass ein Programm zur Erstellung von PDF-Dateien praktisch auf jedem System schon installiert ist: *Ghostscript*. Einen Vergleich der Qualitäten verschiedener Programme, die PDFs erzeugen können, findet man in [13]. Darin schnitt Ghostscript gar nicht so schlecht ab; zumindest aber bietet es das beste Preis-/Leistungsverhältnis!

Einige Hintergrundinformationen zum PDF-Format findet man im c’t-Artikel [14], die komplette Beschreibung des PDF-Formates von Adobe in [12].

6.1 Allgemeine Anleitung zur PDF-Erstellung

Wie oben schon beschrieben, druckt praktisch jedes Programm unter Linux zunächst Postscript. Dies ist der Ausgangspunkt zum Erstellen von PDF-Dateien. Liegt eine Postscript-Datei vor, beispielsweise mit dem Namen irgendwas.ps, so erhält man durch das Kommando

```
$ ps2pdf irgendwas.ps
```

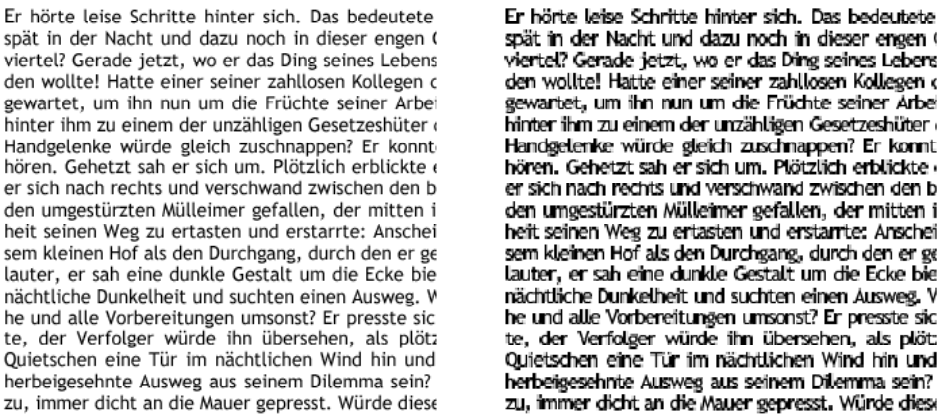

Abbildung 3: Unterschiedliche Einbettung von Schriften in PDF-Dateien

die Datei irgendwas.pdf im gleichen Verzeichnis. Letzlich ist das Programm *ps2pdf*, das im Lieferumfang enthalten ist, nur ein Skript, das Ghostscript mit einigen besonderen Optionen aufruft. Möchte man eine bestimmte PDF-Version erhalten, so gibt es die Skripte *ps2pdf12* für die Version 1.2, *ps2pdf13* für die Version 1.3 und *ps2pdf14* für die Version 1.4. Für die Frage, welche PDF-Version mit welcher Version des Acrobat Readers angezeigt werden kann, gilt folgende Regel: Man addiere die Zahlen vor und nach dem Punkt und erhalte so die Version des Acrobat Readers; beispielsweise benötigt man für PDF 1.4 mindestens den Acrobat Reader 5.

6.2 Einbettung von Schriften

Ein Problem, das häufig auftritt, ist die schlechte Anzeigequalität von Schriften im Acrobat Reader. Der Unterschied ist in Abbildung 3 deutlich zu sehen.

Es ist möglich, Schriften sowohl im TrueType-, im Type1- als auch im Type3-Format einzubetten. Letzteres wird hauptsächlich dazu verwendet, Bitmapschriften einzubinden und genau hier liegt das Problem: Bitmapschriften sind für eine bestimmte Auflösung gerastert, beim Druck ist i. d. R. alles in Ordnung. Anders am Bildschirm: Hier kann man den Text kaum entziffern.

Das Problem kann zum einen darin liegen, dass man eine veraltete Version von Ghostscript verwendet, zum anderen daran, dass schon das Anwendungsprogramm die Schrift falsch eingebettet hat. Letzteres kommt v. a. bei LaTeX vor.

Zunächst sollte daher sichergestellt werden, dass auf dem System die neueste Version von Ghostscript installiert ist: Derzeit ist GNU Ghostscript[12] 7.05 die neueste Version. Die Version 5.x ist zur PDF-Erstellung mehr oder weniger unbrauchbar, 6.x funktioniert normalerweise problemlos.

[12] Ghostscript gibt es in zwei Lizenzen: GNU General Public License (GPL) und Aladdin Free Public License (AFPL). Die GNU-Version ist jeweils älter. Die AFPL erlaubt zwar die unentgeltliche Verwendung, nicht jedoch das Verbreiten auf Datenträgern, die verkauft werden. Daher darf Linuxdistributionen nur die letzte GNU-Version beiliegen.

Da Ghostscript nun auch das Autoconf-System zum Kompilieren des Quellcodes verwendet, funktioniert eine manuelle Installation aus den Quellen recht unproblematisch, d. h. mit `./configure && make && su -c "make install"`.

✎ Wenn Sie spezielle Druckertreiber verwenden, die standardmäßig nicht im Lieferumfang von Ghostscript enthalten sind, müssen Sie vor der Installation dafür sorgen, dass Ghostscript entsprechend gepatcht wird. Ansonsten kann es sein, dass Sie nach dem Update nicht mehr ordnungsgemäß drucken können.

Wie die Schriften eingebettet werden bzw. ob sie überhaupt eingebettet werden, hängt vom jeweiligen Anwendungsprogramm ab. Normalerweise müssen alle Schriften, die Ghostscript nicht bekannt sind, eingebettet werden.

6.2.1 OpenOffice.org 1.0.x und StarOffice 6.0

Bei Type1-Schriften verhält sich OpenOffice.org standardmäßig richtig, d. h., die Schriften werden eingebettet und es gibt keine Probleme mit PDFs. Anders hingegen bei TrueType-Schriften: Mit dem Druckertreiber *Generic Printer* (SGENPRT) in der Druckerverwaltung *(spadmin)* werden TrueType-Schriften als Type3-Schriften gerastert eingebettet. Der Grund hierfür liegt darin, dass OpenOffice.org davon ausgeht, dass nicht alle Drucker mit Type42-/TrueType-Schriften zurechtkommen.

Deshalb besitzt OpenOffice.org den Druckertreiber *Acrobat Distiller* (ADISTILL): Hiermit werden TrueType-Fonts als Type42 eingebettet, die dann bei der PDF-Erstellung als TrueType eingebettet werden. Somit entstehen saubere PDFs.

✎ Ich habe die Erfahrung gemacht, dass Umlaute aus irgendwelchen Gründen mit Ghostscript 6.53 ohne Abstand gesetzt werden. Ein Update auf Ghostscript 7.05 hat das Problem behoben.

Da OpenOffice.org eine spezielle Möglichkeit besitzt, gehen Sie am besten folgendermaßen vor, um so einen Druckertreiber einzurichten:

1. Starten Sie die *OpenOffice.org-Druckerverwaltung*, indem Sie entweder den ensprechenden Menüentrag auswählen oder im OpenOffice.org-Installationsverzeichnis *spadmin* ausführen.
2. Wählen Sie *Neuer Drucker* . . .
3. Wählen Sie *einen PDF-Konverter anbinden* und klicken Sie auf *Weiter.*
4. Wählen Sie den *Adobe Distiller Treiber* und klicken Sie auf *Weiter.*
5. Übernehmen Sie die erste Zeile in der Auswahlliste, wählen Sie ein passendes *PDF-Zielverzeichnis* und klicken Sie auf *Weiter.* Falls Sie eine ältere PDF-Version erzeugen möchten, sollten Sie `-dCompatibilityLevel=1.2` (bzw. `1.3`) hinzufügen.[13]
6. Übernehmen oder ändern Sie den Namen und schließen Sie den Vorgang mit *Fertigstellen* ab.

[13] Zusätzlich hat dies den Vorteil, dass Standardschriften wie Times oder Helvetica nicht eingebettet werden und die erstellten PDF-Dateien somit kleiner werden. Diese Schriften werden nämlich sowieso mit dem Acrobat Reader mitgeliefert. Wenn man keine besonderen Zeichensätze verwendet, gibt es meines Erachtens keinen Grund, Helvetica oder Times einzubetten.

Nun können Sie ganz einfach über den neu erstellten Druckertreiber PDF-Dateien erzeugen. Die fertigen PDFs liegen dann mit dem gleichen Dateinamen der Datei vor, die gedruckt wurde (plus die Endung .pdf), und zwar in dem oben ausgewählten Verzeichnis.

6.2.2 LaTeX-Dokumente

Standardmäßig verwendet LaTeX Schriften im Metafont-Format. Diese eignen sich zur PDF-Erstellung nur sehr eingeschränkt; sie werden immer als Type3-Bitmapschriften eingebettet und die resultierende Qualität ist eher mäßig.

Für die Computer-Modern-Schriften gibt es passende Type1-Äquivalente, sodass diese einfach eingebettet werden können. Sie müssen *dvips* nur die Optionen `-Pcmz` und `-Pamz` mitgeben, damit diese Schriften verwendet werden, bzw. gleich *pdflatex* verwenden. Außerdem muss sichergestellt werden, dass die Schriften installiert sind und dass die Distribution *TeTeX* verwendet wird. Bei anderen LaTeX-Distributionen heißt die Option evtl. anders. Probieren Sie auf jeden Fall auch `-Ppdf` aus oder lesen Sie die Anleitung Ihrer LaTeX-Distribution.

Die Verwendung der Computer-Modern-Schriften im deutschsprachigen Raum hat aber den Nachteil, dass bei der Verwendung von Sonderzeichen die Trennregeln nicht mehr richtig angewendet werden. Deshalb verwendet man normalerweise das T1-Encoding mit den EC-Schriften (European Computer Modern). Für diese gibt es neuerdings eine frei erhältliche Type1-Variante. Das Paket heißt *cm-super* und kann über das CTAN[14] heruntergeladen werden.

Alternativ kann auch das Paket ae.sty verwendet werden, dass trotz der Verwendung der Type1-Computer-Modern-Schriften die deutschen Trennregeln korrekt anwendet.

Eine ganz andere Möglichkeit wäre, auf die Computer-Modern-Schriften zu verzichten und stattdessen eine der 35 Standard-Postscript-Fonts zu verwenden. Die Dokumentation zu *psnfss*[15] enthält eine entsprechende Anleitung bereit, beispielsweise erreichen Sie eine übliche Kombination (was nicht unbegingt heißen soll, dass diese Kombination sehr geeignet ist) aus Helvetica, Times und Courier mit folgendem LaTeX-Code im Vorspann:

```
\usepackage{mathptmx}              % auch für den Mathematiksatz passend
\usepackage[scaled=.92]{helvet}    % etwas verkleinert, sonst zu groß
\usepackage{courier}
```

Dies ist die bevorzugte Variante, um platzsparende PDF-Dateien zu erstellen, da diese Schriften nicht in die PDF-Dateien eingebettet werden müssen, sondern im Standardlieferumfang des Acrobat Readers enthalten sind.

Welches die bevorzugte Option ist, kann nicht abschließend beantwortet werden. Es hängt von mehreren Faktoren ab, u. a. vom persönlichen Geschmack, vom verwendeten System und von dem jeweiligen Zweck.

[14] Das *Comprehensive TeX Archive Network* stellt ein Netzwerk von Servern dar, die LaTeX-Pakete zum Download anbieten. Der deutsche Server heißt ftp://ftp.dante.de.

[15] suchen Sie auf Ihrem System nach der Datei psnfss2e.pdf

7 Bezugsquellen von Schriften

Vor allem TrueType-Schriften findet man häufig kostenlos im Internet. Nur ist die Qualität meist enttäuschend. Hochwertige Schriften werden oft mit Softwareprogrammen mitgeliefert.

Die Microsoft-Grundschriften (Arial, Times New Roman, Verdana, Courier New, ...) können kostenlos im Internet unter folgender Adresse bezogen werden:

➠ http://corefonts.sourceforge.net

Hinweis: Die Dateien können normalerweise nur unter Windows ausgepackt werden. Ein geeignetes Programm für Linux wäre *cabextract*. Es kann hier heruntergeladen werden:

➠ http://www.kyz.uklinux.net/cabextract.php3

Viele Linux-Distributionen liefern darüber hinaus ein Skript namens *fetchmsttfonts* mit. Dadurch lässt sich das Herunterladen und Installieren dieser Schriften automatisieren. Allerdings dürften die meisten dieser Skripten nicht mehr funktionieren, da sie die Schriften direkt vom Microsoft-Server herunterladen wollen und das nicht mehr möglich ist. Microsoft hat im August 2002 die Schriften entfernt. Inwieweit sich schon angepasste Versionen dieser Skripten im Umlauf befinden weiß ich leider nicht.

Zusätzlich befinden sich z. B. auf den Corel-Draw-CDs eine Vielzahl von hervorragenden Schriften, die gut geeignet sind. Auch andere Grafikprogramme liefern solche mit. Fast immer als TrueType, oft noch zusätzlich als Type1.

Wenn Sie zusätzlich Schriften kaufen wollen, bietet die Firma SoftMaker recht günstig qualitativ hochwertige Schriften an. Eine Schriftfamilie wird unter

➠ http://www.freefont.de

jeden Monat kostenlos zum Download gestellt – in beiden Formaten. Außerdem lassen sich kostengünstig Schriften-CDs erwerben.

Auch die Firma URW bietet Schriften an. Qualitativ hochwertig, für den Heimbedarf aber zu teuer. Dafür gibt es nette Spiele, bei denen man Schriften gewinnen kann (nur TrueType):

➠ http://www.urw.de

7.1 35 Standard-Postscript-Schriften

Jeder Postscript-Interpreter ab Level 2 muss folgende 35 Schriften eingebaut haben:

- Avant Garde (Normal, Fett, Kursiv, Fett-Kursiv)
- Bookman (Normal, Fett, Kursiv, Fett-Kursiv)
- New Century Schoolbook (Normal, Fett, Kursiv, Fett-Kursiv)
- Zapf Dingbats (Normal)
- Helvetica (Normal, Fett, Kursiv, Fett-Kursiv)

- Helvetica Narrow (Normal, Fett, Kursiv, Fett-Kursiv)
- Times (Normal, Fett, Kursiv, Fett-Kursiv)
- Courier (Normal, Fett, Kursiv, Fett-Kursiv)
- Palatino (Normal, Fett, Kursiv, Fett-Kursiv)
- Symbol (Normal)
- Zapf Chancery (Kursiv)

Da jeder Schriftschnitt eine eigene Schrift ist, macht sind es insgesamt 35 verschiedene Schriften, obwohl es sich nur um elf Schriftfamilien handelt (oder zehn, wenn man *Symbol* als *Times* einstuft).

Hochwertige „Kopien" dieser Schriften (von der Firma URW zur Verfügung gestellt) finden sich im Lieferumfang von Ghostscript bzw. können hier heruntergeladen werden:

➠ ftp://mirror.cs.wisc.edu/pub/mirrors/ghost/gnu/fonts/gnu-gs-fonts-std-6.0.tar.gz

Allerdings handelt es sich nicht um die Originalschriften sondern um Klone, deshalb tragen sie auch andere Namen. Die entsprechende Aliase sind für Ghostscript in der Datei Fontmap.GS bereits definiert. Um auch in anderen Programmen – also unter X11 – die Schriften unter dem „richtigen" Namen zur Verfügung zu haben, liefern verschiedene Linuxdistributionen angepasste fonts.dir-Dateien mit, teilweise auch fonts.alias. Under Debian müssen Sie dazu das Paket *gsfonts-x11* installieren.

Wenn dies bei Ihnen nicht der Fall ist und Sie daher keine skalierbaren Varianten dieser Schriften haben, bietet sich folgende fonts.dir-Datei an:

➠ http://www.bwalle.de/docs/gs-std-fonts.dir

Der Hersteller ist dabei auf *urw* gesetzt; damit wird eine Verwechslung mit den bei XFree86 mitgelieferten Pixelfonts vermieden. Einfach die fonts.dir sichern und durch die obige austauchen; Link oder Kopie auf fonts.scale nicht vergessen.

Natürlich muss das Verzeichnis auch als `FontPath` eingetragen sein.

Eine andere Möglichkeit zumindest an einige Standard-Postscript-Schriften (Times, Helvetica, Courier, Symbol, Zapf Dingbats) zu kommen sind die von Adobe im *Acrobat Reader 3.x* mitgelieferten Schriften. Diese veraltete Version befindet sich immer noch zum Download unter

➠ ftp://ftp.adobe.com/pub/adobe/acrobatreader/unix/3.x

Es werden nur die eigentlichen Schriftdateien mitgeliefert, die Metriken kann man sich separat herunterladen:

➠ ftp://ftp.adobe.com/pub/adobe/type/win/all/afmfiles/base35

Dann benennt man die Dateien entsprechend um, schreibt eine fonts.dir und fonts.scale und bindet den Pfad in die X11-Installation ein. Das Vorgehen kann allerdings zum Nachteil haben dass statt der mitgelieferten Pixelfonts in einigen Fällen (wenn die Größe nicht genau stimmt) die skalierbaren Schriften genommen werden, was sich in einer schlechteren Darstellungsqualität äußert (es sei denn, man bevorzugt Anti-Aliasing, dann sind diese Schriften wieder besser als Pixelfonts). Zudem verfügen diese Schriften über kein Eurosymbol.

Eine andere Möglichkeit ist es deshalb, die Schriften nur in OpenOffice.org mit Hilfe des Druckerinstallationsprogrammes einzubinden, siehe Abschnitt 3.3 auf Seite 18.

❢ Nach meiner Information ist es gemäß den Linzenbestimmungen des Adobe Acrobat Reader nicht erlaubt, die mitgelieferten Schriften für einen anderen Zweck einzusetzen. Ich beschreibe hier also nur die theoretische Vorgehensweise, die tatsächliche Durckführung erfolgt auf „eigene Gefahr".

7.2 Bildschirmschrift Terminus

Durch Zufall habe ich im Internet eine sehr gut lesbare Nichtproportionalschrift namens *Terminus* entdeckt, deren Installation ich nachfolgend beschreiben möchte. Diese Schrift eignet sich hervorragend als Standardschrift für ein *XTerm* oder auch für den Lieblingseditor.

Als erstes muss man sich die Schrift besorgen. Falls Sie Debian verwenden brauchen Sie nur das Paket *xfonts-terminus* zu installieren und sind schon fertig. Auch für SuSE gibt es ein (inoffizielles) Paket:

➠ ftp://ftp.suse.com/pub/people/mfabian/8.2-noarch/terminus-font-4.03-0.noarch.rpm[16]

Alles weitere erledigt Debian für Sie automatisch. Ansonsten müssen Sie sich die Schriften von Hand herunterladen, die Adresse der Homepage lautet:

➠ http://www.is-vn.bg/hamster/jimmy-en.html

Nun ist folgendermaßen vorzugehen:

1. Zunächst packt man das Paket aus, am einfachsten mit dem Befehl `tar xvfz ter-x400.tgz`. Dadurch wird ein neues Verzeichnis namens ter-x400 erstellt.
2. Wechseln Sie nun mit `cd ter-x400` in dieses Verzeichnis.
3. Nun muss ein Verzeichnis erstellt werden, in das die Schriften dann endgültig installiert werden sollen, beispielsweise /usr/X11R6/lib/X11/fonts/terminus. Dieses Verzeichnis muss natürlich dann in die Konfigurationsdatei von X11 eingebunden werden, siehe Abschnitt 3.2. Die Schriften können auch in ein bereits vorhandenes Verzeichnis hinzugefügt werden, davon rate ich aber eher ab, weil es beispielsweise zu Konflikten mit dem distributionseigenen Paketmanagementsystem führen *kann*.
4. Führen Sie `./configure -x11dir=...` aus. Natürlich müssen Sie anstelle der drei Punkte Ihr oben erstelltes Verzeichnis angeben.
5. Mit dem Befehl `make install`, der mit Administratorrechten ausgeführt werden sollte, wird die eigentliche Installation durchgeführt. Es werden die erforderlichen Schriftdateien in das vorher angegebene Verzeichnis kopiert und die Indexdateien werden erstellt.
6. Nun sind die Schriften installiert. Sie müssen `xset +fp ...` bzw. `xset fp rehash` ausführen; genauere Informationen dazu in Abschnitt 3.2.

Ich hoffe, dass diese wirklich gut lesbaren Schriften bald in jeder Linuxdistribution standardmäßig installiert sind oder als Paket vorliegen.

[16] Die Quellpakete finden Sie unter ftp://ftp.suse.com/pub/people/mfabian/8.2-src/terminus-font-4.03-0.src.rpm.

Literatur

[1] Font-HOWTO in Englisch von Donovan Rebbechi entweder auf der Festplatte Ihres Linux-Systems (normalerweise unter /usr/share/doc/howto/), oder auch im Internet unter http://www.linuxdoc.org/HOWTO/Font-HOWTO.html

[2] "OpenType specification", Microsoft Corporation: http://www.microsoft.com/typography/OTSPEC/default.htm

[3] "Introduction to OpenType fonts" (englisch) – Informationen über OpenType von Adobe: http://www.adobe.com/type/opentype/main.html

[4] "Adobe Font Formats, File Types and Q&A" http://partners.adobe.com/asn/developer/type/ftypes.html

[5] Fragen und Antworten (FAQ) über das Textsatzsystem TEX und DANTE, Deutschsprachige Anwendervereinigung TEX e. V.: http://www.dante.de/faq/de-tex-faq/

[6] "The ISO 8859 Alphabet Soup" (englisch) – umfassende Information über Zeichensätze: http://czyborra.com/charsets/iso8859.html

[7] "Final proof for Latin alphabet No. 9" (englisch) – offizielle Definition des Latin-9-Zeichensatzes: http://www.evertype.com/standards/iso8859/8859-15-en.pdf

[8] „Wie bekomme ich mit LATEX ein Euro-Symbol €?" http://home.vr-web.de/was/x/latexeuro.pdf

[9] Schriften und Drucker mit WordPerfect 8 für Linux (auf englisch): http://www.rodsbooks.com/wpfonts/

[10] OpenOffice.org-Installationshandbuch: ftp://sunsite.informatik.rwth-aachen.de/pub/packages/OpenOffice/contrib/deutsch/1.0.0/installations_handbuch.pdf (2,4 MByte)

[11] "The scourge of Arial" (auf englisch): http://www.ms-studio.com/articles.html

[12] Adobe Systems Incorporated, PDF Reference, third edition, Version 1.4; Addison-Wesley 2001, ISBN 0-201-75839-3; http://partners.adobe.com/asn/developer/acrosdk/docs.html

[13] Andrea Trinkwalder, „PDF selbst gebacken", Zwölf Tools, die Dateien ins Portable Document Format konvertieren; c't 7/2002, S. 152 ff

[14] Thomas Merz, „Ein Rezept für alle", Was im Portable Document Format steckt; c't 7/2002, S. 162 ff

[15] Mike Fabian, „CJK Support in SuSE Linux": http://www.suse.de/~mfabian/suse-cjk/fonts.html