ASCREEN

Version 3.2

Handbuch zum Programm

Anselm Lingnau

November 1992

Vorwort

Dies ist die Anleitung für ASCREEN, einen Alternativen SCREEN-Previewer für TEX auf dem Atari. Hier wird alles erklärt, was Sie wissen müssen, um ASCREEN zu installieren und anzuwenden.

ASCREEN ist ein schnelles und leistungsfähiges Programm, das die DVI-Dateien von TEX auf einem an einen Atari ST, Mega-ST, STE, Mega-STE oder TT angeschlossenen Monitor darstellt. ASCREEN läßt sich in beliebigen TEX-Implementierungen einsetzen und greift dabei auf die jeweils schon vorhandenen PK-Zeichensätze zu, denn die Auflösung der Zeichensätze und ihre Ordnerstruktur ist beliebig wählbar. Selbstverständlich versteht ASCREEN sich blendend mit den gängigen TEX-Shells wie der „TEXshell“ von Heidrich, Kießling und Maluschka oder *CTEX* von Christoph Strunk. ASCREEN schreibt auch Listen fehlender Zeichensätze jeweils im richtigen Format für die TEX-Versionen von Stefan Lindner oder Christoph Strunk. Übrigens ist auch Multitasking, etwa durch PAM's MultiGEM oder Ataris künftiges MultiTOS, für ASCREEN kein Problem, es ist sogar möglich, nach einem TEX-Lauf ASCREEN die neue DVI-Datei automatisch lesen und anzeigen zu lassen. ASCREEN unterstützt *Fontlibraries*, bei denen Gruppen von Zeichensätzen in einer einzigen Datei gesammelt werden, was Plattenplatz spart und bei langsamen Festplatten einen Geschwindigkeitsvorteil bringt. Die Fontlibraries von ASCREEN sind kompatibel zu denen von emTEX für IBM-artige Computer.

ASCREEN ist in GEM eingebunden und von der Bildschirmhardware unabhängig — es funktioniert in Schwarzweiß und Farbe in diversen Auflösungen auf den meisten Rechnern und Monitoren. Ferner ist ASCREEN „international“; landesspezifische Versionen können durch einfaches Ändern der Resourcedatei erstellt werden.

ASCREEN ist ein schnelles und komfortables Programm: es arbeitet auf Wunsch voraus, während Sie eine Seite Text lesen, so daß die nächste Seite ohne Verzögerung zur Verfügung steht. Sie können so viele Fenster öffnen, wie GEM Ihnen erlaubt, und verschiedene Textseiten gleichzeitig betrachten. Oder verschiedene Teile derselben Seite. Wenn Sie eine Übersicht über eine Seite bekommen möchten, können Sie sie auch verkleinern und so ganz auf den Bildschirm holen. Das funktioniert sogar für Doppelseiten. Wollen Sie Ihren Bildschirm ganz für die Textdarstellung nutzen und auf Fenster weitgehend verzichten, so ist auch das möglich — ASCREEN kann eine Textseite in einem Fenster ohne Kontrollelemente zeigen und stellt Ihnen so bis auf die Menüzeile den ganzen Bildschirm zur Verfügung. Weitere

„normale“ Fenster sind natürlich auch hier möglich.

Schließlich unterstützt ASCREEN auch HyperTEX, ein Verfahren zur Erstellung von Dokumenten mit „Querverbindungen“. Sie können mit HyperTEX Teile eines Dokuments so mit anderen verbinden, daß Sie durch einfaches Anklicken mit der Maus etwa zur Definition eines Begriffs oder zu weiteren Verweisen kommen können.

Und noch eine letzte gute Nachricht: Sie dürfen ASCREEN benutzen, ohne dafür etwas bezahlen zu müssen, und Sie dürfen das Programm auch allen Ihren Verwandten, Bekannten, Freunden und Feinden schenken — unter zwei kleinen Bedingungen: Sie dürfen dabei kein Geld verdienen, und Sie dürfen auch nicht so tun, als hätten Sie ASCREEN selbst geschrieben. (Ehre, wem Ehre gebührt.)

* * *

Diese Anleitung beschäftigt sich weder mit der Herstellung von TEX genehmen Manuskripten an sich (darüber gibt es dickere und bessere Bücher), noch mit der Bearbeitung solcher Manuskripte durch TEX, noch mit der Installation eines TEX-Systems. Ich setze voraus, daß Sie zumindest so viel von TEX verstehen, daß ich Ihnen nicht erklären muß, wofür ein Previewer überhaupt gut ist, und daß Sie über ein installiertes TEX-System verfügen.

Manche Absätze in der Dokumentation sind etwas kleiner gedruckt, und links von ihnen stehen die Symbole **Aha** oder **!!!**. **Aha** bezeichnet Absätze, in denen interessante, aber nicht so wichtige Sachen erklärt werden. **!!!** weist auf wichtige Stellen hin.

* * *

An dieser Stelle möchte ich verschiedenen Personen meinen Dank aussprechen: zuerst natürlich Donald E. Knuth, der TEX erfunden und verschenkt hat, und Leuten wie Stefan Lindner, Lutz Birkhahn oder Christoph Strunk, die sich abmühen, um gute und billige Implementierungen von TEX für den Atari zur Verfügung zu stellen. Roy Middleton stellte mir Zeit und einen Rechner zur Verfügung, als ich die allererste Proto-Version von ASCREEN schreiben wollte. Neben meinen Beta-Testern für ASCREEN 2 (und 3), Peter Dömel, Cornelius Krasel und Volker Kurz, müssen einige Benutzer von ASCREEN 2 erwähnt werden, deren Kommentare und Kritik maßgeblich zur Verbesserung jener Version beitrugen und auch in diese Neuimplementierung eingingen, zum Beispiel (in alphabetischer Reihenfolge) Thomas Braun, Peter Eckel, Gregor Fritz, Ralf Michael Handl, Johannes Hill, Markus Michalek, Frank Mittelbach und Jean-Pierre Seifert. Michael Kaib machte es möglich, daß ich ASCREEN 3 auf einem Atari TT ausprobieren konnte, und steuerte Kommentare über die Verwendung von ASCREEN mit *PAM's MultiGEM* bei. Lutz Birkhahn verdanke ich den Code für die IMG-Einbindung, die mir selbst zu umständlich war, und Eberhard Mattes (der Guru von der PC-Seite) erklärte mir sein FLIB-Format. Wichtige Beiträge zu ASCREEN 3 lieferten neben einigen schon genannten Personen unter anderem (ebenfalls in alphabetischer Reihenfolge) Stefan Eissing, Dieter Fiebelkorn, Michael Hoppe, Stefan Haake, Karsten Isakovic, Julian Reschke, Herbert Sauro, Thomas Uhl, Robert Wilhelm und Martin Wunderli(ch?). Zahlreiche Leute schickten mir auch die erbetene Postkarte, ich habe schon eine ganz nette

Sammlung, die aber durchaus noch wachsen darf (wink, wink). Vielen Dank an alle!

Eppstein, im November 1992 — A. L.

"And what is the use of a book," thought Alice,
"without pictures or conversations?"
— LEWIS CARROLL, *Alice in Wonderland* (1865)

Inhaltsverzeichnis

Kapitel 1

Installation

1.1 Wichtige Vorbemerkung

Willkommen zu ASCREEN! Sie haben sich also ASCREEN und diese Anleitung besorgt und sind kaum noch zu bremsen, weil Sie endlich sehen wollen, ob sich die Mühe gelohnt hat. Gleich werden Sie es wissen. Nehmen Sie sich aber zuerst etwas Zeit und machen Sie eine Sicherheitskopie der ASCREEN-Diskette, nur für den Fall, daß die Kaffeetasse umkippt, der Hamster ausbricht oder die Kinder ausprobieren, was der Menüpunkt „Diskette formatieren. . . “ so alles anstellt. Wo Sie dabei sind, machen Sie doch gleich ein paar Extrakopien für Ihre Verwandten, Freundinnen und Freunde. Bis gleich also!

1.2 Installation — Eine halbe Anleitung

Jetzt sind wir soweit: ASCREEN kann installiert werden. Die Anleitung zur vorigen Version erklärte hier haarklein den Installationsvorgang für die TEX-Version von Stefan Lindner. Heute gibt es aber mindestens vier allgemein verbreitete Implementierungen von TEX für den Atari ST (dazu kommen wohl nochmal so viele, die gewissermaßen „regionale Bedeutung“ haben), und ASCREEN verträgt sich potentiell mit allen. Es wäre wohl hochgradig unfair, die Benutzerinnen und Benutzer einer einzigen Implementierung so zu bevorzugen, und darum gebe ich nur einige eher allgemeine Hinweise.

Die Installation besteht aus den folgenden Schritten:

1. Das ASCREEN-Programm (ASCREEN.PRG) und seine Resourcedatei (ASCREEN.RSC) müssen an einen geeigneten Platz auf der Festplatte kopiert werden. (Gibt es TEX-Systeme auf Diskettenbasis? Unwahrscheinlich.) Der erste Vorschlag für einen solchen Platz wäre „in die Nähe des standardmäßig mitgelieferten Previewers“. Dieser heißt zum Beispiel SCREEN.TTP (bei Stefan Lindner) oder DVI_ST.PRG (bei Christoph Strunk), aber auch

etwas wie `PREVIEW.PRG` sieht wahrscheinlich aus. Kopieren Sie `ASCREEN` einfach in dasselbe Verzeichnis.

2. `ASCREEN` muß an das verwendete TEX-System angepaßt werden. Sie können dazu entweder die mitgelieferte Datei `ASCREEN.SET` mit einem beliebigen TEXt-Editor bearbeiten (der, mit dem Sie Ihre Dokumente schreiben, ist bestens geeignet) oder `ASCREEN` vom Desktop aus starten (doppelt anklicken) und die Einstellungen über die Dialoge des Programms vornehmen. Insbesondere der Dialog Pfade... ist hierbei interessant. Der einzige etwas trickreiche Punkt betrifft die Einstellung der Schablone für Zeichensatzdateinamen; Kapitel 5 erklärt dies genauer.

3. Inhaber einer TEX-Shell, etwa der *TEXshell* von Heidrich, Kießling und Maluschka (beim Lindner-TEXdabei) oder Christoph Strunks *CTEX*, sollten `ASCREEN` nun von der Shell aus zugänglich machen, also als Previewer oder Druckertreiber installieren. Für die HKM-TEXshell ist eine Datei `ASCREEN3.INF` beigefügt, die zur Installation gebraucht wird; bei *CTEX* müssen Sie den Previewer mit dem passenden Menüeintrag „finden“.

   Wenn Sie `ASCREEN` von einer kommandozeilenorientierten Shell wie *Mupfel* oder *Guläm* aufrufen möchten, sollten Sie dafür sorgen, daß die Datei `ASCREEN.SET` auch dann gefunden wird, wenn das Verzeichnis mit den TEX-Programmen nicht das aktuelle Verzeichnis ist. Am einfachsten setzen Sie dazu mit einem Befehl wie

   ```
   setenv ASCREENSET d:\tex\ascreen.set
   ```

   die Umgebungsvariable `ASCREENSET` auf den vollen Namen der Voreinstellungsdatei.

!!! Benutzer von Christoph Strunks *CTEX* sollten eine Kleinigkeit beachten: man kann Standardargumente für Previewer und Druckertreiber angeben, die dann immer an das betreffende Programm übergeben werden. Die normale Voreinstellung für den Previewer ist etwas wie `-p=a4`. `ASCREEN` versteht diese Option aber recht gründlich miß (sie führt dazu, daß der Cachespeicher auf eine Seite begrenzt wird), so daß Sie sie am besten gleich entfernen.

Und braucht man keine Klempner mehr,
na, da werd' ich halt Installateur!
— REINHARD MEY, *Ich bin Klempner von Beruf* (1974)

Kapitel 2

ASCREEN starten

2.1 Das Programm aufrufen

Sie können ASCREEN ganz einfach aufrufen, indem Sie es auf dem Desktop zweimal anklicken. Nach einer kurzen Weile präsentiert sich Ihnen die Menüleiste von ASCREEN (der unmißverständlichen Titel links oben verrät es). Jetzt können Sie mit dem Menüpunkt Datei öffnen im Menü Datei eine DVI-Datei aussuchen, deren erste Seite dann wenig später in einem Fenster angezeigt wird.

Von einer kommandoorientierten Shell wie der *Mupfel* aus können Sie ASCREEN genauso über den Programmnamen, ascreen, starten. Da ASCREEN ein GEM-Programm ist, also die graphische Bedienoberfläche des Atari-Computers benutzt, kann es bei manchen Shells nötig sein, ein Kommando wie

```
gem ascreen
```

zu geben. Darauf weisen wir im folgenden nicht jedesmal hin.

Wenn Sie eine Shell benutzen, werden Sie üblicherweise den Namen der DVI-Datei, die ASCREEN zeigen soll, gleich beim Aufruf mit angeben.

```
ascreen foo
```

zeigt etwa die Datei foo.dvi an. Die DVI-Datei sucht ASCREEN in demjenigen Verzeichnis, das in der Pfade-Dialogbox bzw. in der Setupdatei als DVI-Pfad angegeben wurde. Sie können aber auch einen kompletten Dateinamen (mit Pfad) angeben, die Voreinstellung wird dann ignoriert. Insbesondere sollten Sie etwas sagen wie

```
ascreen .\foo
```

wenn Sie die Datei foo.dvi im aktuellen Verzeichnis meinen, Ihr DVI-Pfad aber auf etwas anderes zeigt.

Aha Wenn Sie einen modernen Desktopersatz wie GEMINI verwenden, können Sie ASCREEN auch als Icon auf dem Bildschirmhintergrund ablegen. Dann können Sie Dateien anschauen, indem Sie die betreffenden Icons auf das ASCREEN-Icon ziehen.

Aha Was GEMINI angeht, so habe ich von Leuten gehört, die eine Datei DVI.MUP als Applikation für alle Dateien anmelden, deren Namen mit .DVI aufhören. DVI.MUP sieht ungefähr so aus:

```
d:\tex\ascreen &d:\tex\ascreen.set $*
```

Dadurch wird ASCREEN automatisch gestartet, wenn Sie eine DVI-Datei doppelt anklicken. (Dieser Tip wurde von Cornelius Krasel beigesteuert.)

!!! Damit ASCREEN funktionieren kann, braucht es eine passende *Resourcedatei*. In dieser Datei, typischerweise ASCREEN.RSC genannt, sind die Elemente der graphischen Bedienoberfläche definiert. ASCREEN sucht diese Datei mit der üblichen GEM-Methode. Wenn Sie also die Resourcedateien aller Ihrer GEM-Programme in einem besonderen Verzeichnis sammeln, dann sollte das auch mit ASCREEN funktionieren. Normalerweise sind Sie aber auf der sicheren Seite, wenn die Resourcedatei im selben Verzeichnis steht wie ASCREEN.PRG.

Aha ASCREEN ist so geschrieben, daß sämtliche landessprachlichen Zeichenketten in der Resourcedatei stehen. Daher ist es einfach, ausländische Versionen von ASCREEN zu machen, denn man muß nur die Resourcedatei ändern. Kapitel 6 hat hierzu einiges zu sagen. In der Tat enthält die Distribution eine fertige Resourcedatei für eine englische Bedienoberfläche (hallo Herbert!) Um die englische Version zu benutzen, können Sie zum Beispiel das (deutsche) ASCREEN.RSC in ASCREEN.GER und das (englische) ASCREEN.ENG in ASCREEN.RSC umbenennen. Oder Sie tun den Dateinamen der englischen Resourcedatei in die Umgebungsvariable ASCREENRSC:

```
setenv ASCREENRSC d:\tex\ascreen.eng
```

ASCREEN sucht nämlich in Wirklichkeit nur nach einer Datei ASCREEN.RSC, wenn die Umgebungsvariable ASCREENRSC nicht definiert ist, sonst wird ihr Inhalt direkt als Dateiname benutzt. Die zweite Variante macht es übrigens möglich, beide Sprachen gleichzeitig auf demselben Rechner zu installieren und zu benutzen.

2.2 Die Setupdatei

Beim Programmstart liest ASCREEN nicht nur die Resourcedatei, sondern auch eine *Setupdatei*. Diese enthält Voreinstellungen für viele Aspekte der Funktion von ASCREEN, zum Beispiel

- Pfadangaben für DVI-Dateien, IMG-Graphiken, Protokoll- und Zeichensatzlistendateien
- Suchpfade und Namensschablonen für Zeichensätze und Zeichensatzbibliotheken
- Parameter für die Speicher- und Zeichensatzverwaltung
- Voreinstellungen für die Gestalt und Funktion der Bedienoberfläche

und einiges mehr. Alle einstellbaren Parameter und Optionen, die innerhalb des Programms über Dialoge zugänglich sind, können Sie in einer Setupdatei permanent ablegen.

Der Name der Setupdatei, die ASCREEN beim Programmstart liest, kann auf eine von drei Arten festgelegt werden:

1. Sie tun gar nichts Besonderes. Dann sucht ASCREEN im selben Verzeichnis, wo auch die Programmdatei ASCREEN.PRG steht, nach einer Datei namens ASCREEN.SET
2. Sie legen den vollständigen Pfadnamen Ihrer Setupdatei in der Umgebungsvariablen ASCREENSET ab, wie im Kapitel über die Installation beschrieben. Dieser Name hat Vorrang gegenüber dem fest eingebauten Namen ASCREEN.SET.
3. Sie geben den Dateinamen auf der Kommandozeile an, und zwar als erstes Argument unmittelbar hinter einem kaufmännischen Undzeichen (&). Etwa so:

   ```
   ascreen &special.set foo
   ```

   (Die Endung .SET wird von ASCREEN *nicht* selbständig hinzugefügt.) Wenn Sie gut mit TEX Bescheid wissen, kennen Sie diese Syntax schon von den Formatdateien, in denen Voreinstellungen für TEX abgelegt werden. Wie TEX erkennt ASCREEN dies nur, wenn die Angabe als allererstes Argument erscheint. Ein so angegebener Name hat Vorrang gegenüber dem Inhalt von ASCREENSET.

Aha Die letzte Methode ist vor allem sinnvoll, wenn Sie eine Setupdatei haben, die Sie gewöhnlich benutzen (und deren Name zum Beispiel in ASCREENSET steht), aber gelegentlich eine andere einsetzen möchten. Zum Beispiel kann ASCREEN ein Dokument auch mit Ihren Druckerzeichensätzen anzeigen, wenn Sie mal etwas ganz genau überprüfen wollen. Die dafür nötigen Voreinstellungen tun Sie einfach in eine andere Setupdatei, deren Namen Sie bei Bedarf direkt im Aufruf nennen. Siehe hierzu auch Abschnitt 6.2.

Die Einstellungen in der Setupdatei haben Vorrang gegenüber den entsprechenden im Programm fest vorgegebenen Standardwerten. Was die Setupdatei konkret enthält und wie die diversen Voreinstellungen angegeben werden, erfahren Sie im Kapitel 5.

2.3 Kommandooptionen

Außer dem Namen der Setupdatei und dem der DVI-Datei können Sie im Aufruf von ASCREEN diverse *Kommandooptionen* angeben, die es Ihnen gestatten, die Funktion von ASCREEN zu beeinflussen. Kommandooptionen müssen zwischen dem Namen der Setupdatei (falls vorhanden) und dem Namen der DVI-Datei stehen. Sie bestehen immer aus einem Minuszeichen gefolgt von einem Buchstaben.

Es gibt zwei Arten von Kommandooptionen: solche, die als *Schalter* funktionieren und andere, die einen *Parameter* nach sich ziehen. Schalteroptionen können gesammelt hinter einem einzigen Minus angegeben werden:

```
ascreen -AIv
```

ist äquivalent zu

```
ascreen -A -I -v
```

Die letzte Option in einer solchen Gruppe darf sogar einen Parameter haben.

Aha Sollte der Name Ihrer DVI-Datei mit einem Minuszeichen beginnen (pfui, pfui), so können Sie mit einem doppelten Minus das Ende der Optionenliste bekanntgeben:

```
ascreen -- -punkt
```

zeigt die Datei `-punkt.dvi` an.

Die Kommandooptionen machen es möglich, alle in der Setupdatei enthaltenen Voreinstellungen auch auf der Kommandozeile zu treffen. Kommandooptionen haben dabei Vorrang gegenüber den Einstellungen in der Setupdatei. Insbesondere gilt für Schalteroptionen, daß sie den betreffenden Schalter immer *umlegen*. Ein Schalter, der in der Setupdatei eingeschaltet wird, wird durch die entsprechende Kommandooption also ausgeschaltet und umgekehrt. (Tritt die Kommandooption zweimal auf, so heben die beiden Male sich gegenseitig weg.) Bei Optionen mit Parameter gilt der letzte Wert, falls dieselbe Option auf der Kommandozeile mehrfach auftaucht. Damit können Sie TEX-Shells austricksen, die hartnäckig eigene Ideen darüber haben, welche Optionen ein Previewer erwartet. Auch die Kommandooptionen werden im Kapitel 5 aufgezählt.

!!! Unter normalen Umständen können Kommandozeilen auf dem Atari eine Länge von höchstens 125 Zeichen haben. Das ist nicht allzu viel, und darum wurde vor einiger Zeit von Atari ein Verfahren definiert, das diese Grenze umgeht. ASCREEN unterstützt selbstverständlich dieses sogenannte ARGV-Verfahren und kann deshalb in Zusammenarbeit mit geeigneten Shells, zum Beispiel *GEMINI/Mupfel*, Kommandozeilen von nahezu beliebiger Länge übernehmen.

2.4 Probleme

Vor allem Leute mit TEX-Shells unterschiedlicher Provenienz klagen ab und zu über Schwierigkeiten beim Aufrufen von ASCREEN. Das liegt in den allermeisten Fällen nicht an ASCREEN, sondern an unterschiedlich schlimmen „Hirnschäden“ der betreffenden Shells. (Das Problem mit `-p=a4` in der Strunkschen *CTEX*-Shell haben Sie ja schon im vorigen Kapitel kennengelernt.) Die HKM-TEXshell zum Beispiel besteht darauf, jedem Treiber die Option `-J` zu übergeben, ohne daß man das abschalten könnte. ASCREEN ignoriert diese Option daher ohne Fehlermeldung, um unnötige Verwirrung zu verhindern. Wenn Ihre TEX-Shell nicht so mitspielt, wie Sie das gerne hätten, dann sollten Sie zunächst mit einem Programm wie

```
#include <stdio.h>
int main(int argc, char **argv) {
    while (*argv)
        printf("%s\n", *argv++);
    return 0;
}
```

(oder etwas Äquivalentem in einer anderen Programmiersprache als C) anstelle des richtigen `ASCREEN` nachschauen, was genau `ASCREEN` von Ihrer Shell als Kommandozeile übergeben bekommt. Meist können Sie so das Problem eingrenzen und entweder selbst etwas dagegen tun oder den Autor von `ASCREEN` [mich] (oder den der Shell) mit einem detaillierten Fehlerbericht beglücken (siehe Anhang F).

— I can call spirits from the vasty deep.

— Why, so can I, or so can any man;
But will they come when you do call for them?

— WILLIAM SHAKESPEARE, *King Henry IV, Part 1* (1598)

Kapitel 3

Texte anschauen mit ASCREEN

3.1 Bedienungselemente

Wie die meisten GEM-Programme bietet ASCREEN in einer Menüleiste verschiedene Befehle zur Steuerung des Programms an. Die meisten dieser Befehle sind auch über die Tastatur zugänglich, die entsprechenden Tastenkombinationen stehen jeweils hinter dem Menüeintrag. Die verschiedenen Menübefehle werden im nächsten Kapitel genau erklärt. In diesem Kapitel wollen wir uns zunächst anschauen, wie man mit ASCREEN in einem Text herumspazieren kann, wie man Seiten zwecks besserer Übersicht verkleinern kann und welche Möglichkeiten es beim Anschauen von Texten noch gibt.

3.2 Herumfahren auf einer Seite

Beim Anschauen von Texten können Sie ASCREEN über die Maus oder über die Tastatur steuern. Über die Bedeutung der Rollbalken und -pfeile in einem GEM-Fenster muß hier nichts gesagt werden, denn diese benehmen sich ganz so, wie Sie das in einem GEM-Programm vermuten. Erwähnenswert ist lediglich, daß ein Druck auf einen Rollpfeil das Fenster relativ zum Inhalt gesehen um 8 Pixel in der jeweiligen Richtung verschiebt. (Das heißt, wenn Sie auf den „Pfeil nach unten“ drücken, rutscht der Fensterinhalt um 8 Pixel nach oben.)

Aha Beim „randlosen“ Fenster geht das natürlich nicht, denn es hat ja keinen Rand und demnach auch keine Steuerelemente. Dieses Fenster können Sie nur mit Tastaturbefehlen oder direkt mit der Maus rollen (siehe unten).

Statt der Rollpfeile können Sie auch die Tastatur verwenden. Hierbei gibt es mehrere Möglichkeiten:

- Die Pfeiltasten bewegen den Fensterinhalt ebenfalls in Schritten von 8 Pixeln. Ein Druck auf eine Pfeiltaste entspricht jeweils einem Anklicken des gleichbedeutenden Rollpfeils mit der Maus.

- Halten Sie die Shift-Taste gedrückt und betätigen dann Pfeiltasten, so bewegt sich der Fensterinhalt in „großen“ Schritten. Wie groß die Schritte sind, hängt von der Größe des Fensters ab; wir könnten dazu „seitenweises Rollen“ sagen, wenn der Begriff „Seite“ nicht schon anderweitig besetzt wäre. Es besteht eine gewisse Überlappung zwischen den jeweils gezeigten Fensterinhalten: Beim Rollen nach unten etwa erscheint ein ehemals am unteren Rand des Fensters sichtbarer 8 Pixel hoher Streifen am oberen Fensterrand. (Für die anderen Richtungen gilt sinngemäß dasselbe.) Die Shift-Pfeiltasten-Kombinationen entsprechen Mausklicks in die grauen Bereiche der Fenster-Rollbalken.
- Die Tasten 7, 8, 9 und) auf dem Ziffernblock entsprechen den Pfeiltasten. Die Tasten 1, 2, 3 und 5 auf dem Ziffernblock entsprechen den Pfeiltasten mit Shift. Dies ist praktisch, wenn Sie nicht immer mehrere Tasten gleichzeitig drücken möchten.
- Schließlich können Sie durch einen Druck auf die Taste Clr/Home die obere linke Ecke der Seite im Fenster sichtbar machen.

Sie können den Text auch direkt mit der Maus bewegen. Halten Sie dazu die Shift- oder Alternate-Taste fest, drücken Sie die linke Maustaste und halten Sie sie fest. Der Mauszeiger wird dann zu einer flachen Hand („Schiebe-Hand“), und Mausbewegungen verschieben den Fensterinhalt. Die Verschiebung ist dabei in Relation zur Mausbewegung größer, wenn Sie Shift festhalten, als wenn Sie Alternate festhalten. (*Merke:* Auch bei den Pfeiltasten führt Shift zu größeren Bewegungen.)

Aha Die Schieberichtung entspricht der Bewegungsrichtung bei den Rollbalken, das heißt, wenn Sie die Maus nach unten ziehen, rutscht der Text nach oben. Dies wirkt zuerst merkwürdig, aber ist konsistent zum Verhalten der Rollbalken — es wäre wohl verwirrend, den Fensterinhalt nach oben zu verschieben, wenn ein Rollbalken nach unten gezogen wird, aber den Fensterinhalt nach unten zu verschieben, wenn Sie ihn mit der Maus direkt nach unten ziehen.

Aha Es gibt einen von Atari nicht offiziell dokumentierten Trick, um in der Hauptschleife eines Programms beide Maustasten gleichzeitig abzufragen (normalerweise geht das nur für eine Taste). Dieser Trick funktioniert bei allen bekannten TOS-Versionen, und ASCREEN verwendet ihn auf Wunsch, um die eben erklärte Ziehfunktion auch über die rechte Maustaste auszulösen. Sie können diese Option im Dialog *GEM-Optionen* oder in der Setupdatei ausschalten, falls Sie allergisch gegen undokumentierte Funktionen sind oder eines Tages ein TOS erscheint, das dieses Vorgehen nicht mehr verträgt.

3.3 Blättern im Text

ASCREEN speichert bereits formatierte Seiten in einem *Cachespeicher*, dessen Größe im wesentlichen nur durch den in Ihrem Rechner verfügbaren Hauptspeicher begrenzt ist. (Sie können eine geringere obere Grenze festlegen, um in Multitaskingsystemen auch anderen Programmen etwas Luft zum Atmen zu lassen. Kapitel 5 verrät Ihnen, wie.) ASCREEN zieht allerdings nicht beim Start den ganzen freien Speicher an sich — das wäre ja wirklich nicht die feine englische Art —, sondern versucht für jede neu anzuzeigende Seite Speicher zu

reservieren, wenn sie angefordert wird. Wenn dies gelingt, ist alles in Ordnung; wenn der freie Speicher nicht genügt oder die von Ihnen gewählte Obergrenze erreicht ist, wird die am längsten nicht auf dem Bildschirm gezeigte Seite zugunsten der neuen Seite entfernt. Die Sprache der Informatik nennt das ein *least recently used-* oder *LRU-Schema*. Auf einem Atari mit 1 MB Hauptspeicher passen um die 7 normal große Seiten in den Cachespeicher; Rechner mit mehr Speicher können entsprechend mehr Seiten zur Verfügung halten. Insbesondere müssen Sie auf einem 4-MB-Rechner schon eine Diplomarbeit oder ein Buch schreiben, um den verfügbaren Cachespeicher zu überschreiten. (Natürlich kommt es auch ein bißchen darauf an, was Sie sonst noch im Speicher haben und ob Sie Multitasking verwenden.)

Blättern können Sie mit der Maus oder mit der Tastatur. Wenn Sie die linke Maustaste drücken und festhalten, während der Mauszeiger sich in einem ASCREEN-Fenster befindet, so erscheint ein kleines Klappmenü. Sie können dann — immer noch mit gedrückter Maustaste — einen der dort angebotenen Punkte wählen, indem Sie die Maus dorthinziehen, bis der Menüpunkt schwarz hinterlegt ist, und dann die Maustaste loslassen. Haben Sie es sich anders überlegt und wollen doch nichts aus dem Menü wählen, so ziehen Sie die Maus aus dem Menü heraus, bis kein Menüpunkt mehr schwarz hinterlegt ist, und lassen die Taste los.

Die ersten beiden Einträge im Klappmenü sind *Nächste Seite* und *Vorige Seite*. Die Bedeutung dieser Menüpunkte dürfte offensichtlich sein. Wenn Sie die erste Seite des Dokuments im Fenster haben und *Vorige Seite* wählen, so gelangen Sie auf die letzte Seite; analog können Sie von der letzten auf die erste Seite kommen. Äquivalent zu den Einträgen im Klappmenü sind die Tasten [+] und [−] auf dem Ziffernblock.

Aha Im Dialog *DVI-Optionen* können Sie *schlaues Umblättern* wählen (oder vielmehr abbestellen — es ist standardmäßig eingeschaltet). Schlaues Umblättern sorgt dafür, daß Sie, wenn im Fenster der untere Seitenrand gezeigt wird, beim Blättern auf die nächste Seite an deren oberen Rand gelangen und umgekehrt. Normalerweise bleibt die Position des Fensters auf der Seite beim Blättern erhalten.

Die [Enter]-Taste benimmt sich wie eine Mischung aus den Tasten [2] und [+] auf dem Ziffernblock: sie geht auf einer Seite Schritt für Schritt nach unten und blättert dann zum oberen Ende der nächsten Seite um. Die Taste [.] auf dem Ziffernblock funktioniert entsprechend für das Rückwärtsgehen und Zurückblättern.

Aha Im Dialog *DVI-Optionen* wird dieses Verhalten *Magisches Enter* genannt. Wenn Sie es ausschalten, sind [Enter] und [.] synonym zu [+] und [−] auf dem Ziffernblock, respektive.

3.4 Zu einer beliebigen Seite springen

Mit dem Klappmenüeintrag *Gehe zu...* oder einem Druck auf die Taste [0] auf dem Ziffernblock wird ein kleiner Dialog aufgeklappt, in den man eine Seitennummer eintragen kann. Diese Seitennummer wird dann direkt angesprungen. ASCREEN richtet sich dabei lediglich nach der Position der Seite in der DVI-Datei; die Seitenangaben aus der Sicht von

TEX werden geflissentlich mißachtet. Es zeigt sich nämlich, daß die TEX-Seitennummern nicht eindeutig sein müssen, wie alle bestätigen können, die mal ein Buch mit Vorwort, Inhaltsverzeichnis usw. in LATEX gesetzt haben.

Aha Wenn die entsprechende Option im *DVI-Optionen*-Dialog eingeschaltet ist, merkt `ASCREEN` sich die Position jeder Seite in der `DVI`-Datei, die er schon einmal formatiert hat. Sie können daher vergleichsweise schnell auf eine Seite zurückblättern, die Sie schon einmal in der Anzeige hatten, selbst wenn sie nicht mehr im Cachespeicher steht, weil `ASCREEN` nicht die `DVI`-Datei nach ihr durchsuchen muß, sondern nur in seiner Tabelle nachzuschlagen braucht. Der Speicherbedarf von 4 Bytes für jede Seite im Dokument ist gegenüber der so gewonnen Bequemlichkeit unerheblich.

3.5 Seiten verkleinern

Oft ist es angenehm, eine Seite in ihrer Gesamtheit beurteilen zu können. `ASCREEN` bietet darum die Möglichkeit, einzelne oder Doppelseiten jeweils um den Faktor 2 oder 3 linear zu verkleinern. Das heißt, die Fläche der Seite verringert sich um den Faktor 4 bzw. den Faktor 9. Lesen können Sie dann vermutlich nicht mehr allzuviel, aber dafür ist die Verteilung von Schwarz und Weiß auf der später gedruckten Seite besser auszumachen. Insbesondere die Möglichkeit zur Verkleinerung zweier nebeneinanderliegender Seiten ist für Autoren von Büchern oder längeren Aufsätzen praktisch, um die Wirkung des aufgeschlagenen Werks prüfen zu können.

Mit dem Klappmenü können Sie den Inhalt des aktuellen Fensters über die Funktion *Verkleinern* in einem neuen Fenster verkleinert zur Anzeige bringen. Die Verkleinerung erfolgt dabei mit dem gerade im Dialog *GEM-Optionen* gewählten Verkleinerungsfaktor. Die Funktion *Doppelseite* im Klappmenü verkleinert die Seite im aktuellen Fenster und die im Dokument unmittelbar folgende Seite um den gerade aktuellen Faktor und stellt die beiden in einem gemeinsamen Fenster dar. Bei der doppelseitigen Verkleinerung bezieht die Seitenangabe im Fenstertitel sich auf die links sichtbare Seite und wird von einem `D` eingeleitet.

!!! Ist die zweite zu verkleinernde Seite gerade nicht im Cachespeicher vorhanden, wird sie zunächst aus der `DVI`-Datei geholt und formatiert. Für die doppelseitige Verkleinerung muß daher im allgemeinen im Cachespeicher Platz für eine weitere Seite sein.

Die Fenster mit verkleinerten Seiten werden von `ASCREEN` genau wie Fenster mit normal großen Seiten behandelt, was Rollen, Blättern usw. angeht. (Nein, Sie können ein verkleinertes Fenster *nicht* noch einmal verkleinern.) Sie können also eine normal große und eine verkleinerte Version Ihres Dokuments unabhängig voneinander gleichzeitig lesen.

Die Verkleinerung läßt sich auch über die Tastatur auslösen, und zwar über die Tasten `(` und `4` auf dem Ziffernblock. Dabei steht `(` für die Verkleinerung einer Seite und `4` für doppelseitige Verkleinerung. Die Eselsbrücke: `(` steht in der Nähe der Tasten für „kleine“ Schritte (`7` usw.) und verkleinert auch nicht so „viel“ wie `4`, die bei den Tasten für „große“ Schritte (`1` usw.) steht. Übrigens können Sie auch den Verkleinerungsfaktor über die Tastatur wählen: `/` und `*` auf dem Ziffernblock stellen den Verkleinerungsfaktor auf

den Wert 2 und 3, respektive.

Wenn Sie in einem Fenster blättern, das eine Doppelseite zeigt, so wird die zunächst rechts stehende Seite links dargestellt und rechts kommt die im Dokument folgende Seite ins Blickfeld. Oft ist es aber angenehm, wie in einem Buch zwei Seiten auf einmal „umblättern“ zu können. Hierfür gibt es den Eintrag *Rekto-Verso-Modus* im Klappmenü. (Die „rechten“ Seiten in einem Buch nennt man „rekto“ und die „linken“ auch „verso“. Der Rekto-Verso-Modus bezieht sich darauf, daß die Parität (gerade oder ungerade) der Seitennummern der jeweils links bzw. rechts stehenden Seiten gleich bleibt. Zu kompliziert? Probieren Sie's aus!) Im Rekto-Verso-Modus wird das `D` in der Titelzeile des Fensters durch ein `R` ersetzt, und alle Blätteroperationen blättern um zwei Seiten statt um eine, bis dieser Modus wieder aufgehoben wird.

Außerdem können Sie den Rekto-Verso-Modus auch durch einen Druck auf die Taste 6 auf dem Ziffernblock ein- und ausschalten. (Damit wären alle Tasten des Ziffernblocks belegt.)

Aha Sie haben die Wahl, ob Fenster mit Doppelseiten gleich beim Aufklappen in den Rekto-Verso-Modus versetzt werden. Dies wird im Kapitel 5 erklärt.

3.6 Klemmen und Lösen

Eine Erbschaft aus der Zeit, als `ASCREEN` noch keine Fenster benutzte, sind die Funktionen *Klemmen* und *Lösen* im Klappmenü. Mit ihnen ist es möglich, eine Seite unabhängig davon, ob sie in einem Fenster zu sehen ist, im Cachespeicher „festzuklemmen“ und so vor der Überlagerung durch andere Seiten zu bewahren. Dies ist in der neuen Version nicht mehr so wichtig, da in einem Fenster sichtbare Seiten implizit festgeklemmt sind und Sie ja mehrere Fenster haben können, aber ist trotzdem noch im Programm vorhanden. Wenn Sie nicht so viel Speicher in Ihrem Rechner haben und zum Beispiel den Index eines Buches redigieren wollen, kann es sinnvoll sein, die gerade bearbeitete Indexseite festzuklemmen, um so immer schnell zu ihr zurückkehren zu können. Das Festklemmen einer Seite kann man durch den Menüeintrag *Lösen* wieder aufheben, der bei einer geklemmten Seite anstelle von *Klemmen* im Menü erscheint.

3.7 Hilfe

Durch einen Druck auf die Taste Help können Sie eine Hilfstafel erscheinen lassen, die die wichtigsten Programmfunktionen erklärt.

Being well trained, [the bull] becomes naturally gentle.
Then, unfettered, he obeys his master.
— KAKUAN, *Ten Bulls* (c. 1150)

Kapitel 4

Die Menüleiste

4.1 Vorbemerkung

Wie jedes anständige GEM-Programm benutzt `ASCREEN` die Menüleiste am oberen Bildschirmrand, damit Sie Kommandos eingeben können. Für die meisten Menükommandos gibt es äquivalente Tastaturbefehle. Die erforderlichen Tastenkombinationen stehen jeweils rechts vom Menüeintrag. Das „Dach“ ^ bedeutet dabei *Control* — halten Sie die Taste `Control` fest, während Sie die darauffolgend genannte Taste drücken. Der merkwürdige schwarze Kasten steht für *Alternate* — hier müssen Sie analog die Taste `Alternate` gedrückt halten.

Aha Wenn die Tastaturbefehle Ihnen nicht zusagen — was unwahrscheinlich ist, denn sie halten sich an die üblichen Konventionen auf dem Atari ST —, so können Sie sie mit einem Resourceeditor modifizieren. Kapitel 6 gibt Tips dafür.

Manche Menüeinträge hören mit drei Pünktchen „...“ auf. Wie Sie vermutlich aus anderen Programmen wissen, heißt das, daß dieser Menüeintrag nicht sofort eine Funktion auslöst, sondern eine *Dialogbox* auf den Bildschirm holt, in der weitere Einstellungen gemacht werden können. `ASCREEN` unterstützt sogenannte fliegende Dialogboxen, die an ihrer oberen rechten Ecke ein auffälliges „Eselsohr“ haben. Sie können den Mauszeiger auf dieses Eselsohr bewegen und die Maustaste drücken und festhalten, um den Dialog an eine andere Stelle auf dem Bildschirm zu schleppen. Dies ist manchmal nützlich, wenn das Dialogformular etwas Wichtiges verdeckt.

Aha Andere Implementierungen fliegender Dialogboxen können das komplette Formular in Echtzeit bewegen. `ASCREEN` ist leider nicht so luxuriös; es wird nur ein Rahmen verschoben, dessen Größe der des Formulars entspricht.

Innerhalb der Dialogboxen gibt es zwei Arten von „Schaltern“: eckige Knöpfe zum Ankreuzen und runde zum Eindrücken. Die runden Knöpfe bieten Alternativen an, von denen jeweils nur eine aktiv sein kann. Wenn Sie einen runden Knopf wählen, wird der vorher

gewählte runde Knopf automatisch „entwählt“. Solche Knöpfe nennt man eingedenk der Stationstasten bei alten Autoradios auch „Radioknöpfe“.

Doch nun zu den Menüs, einem nach dem anderen:

4.2 Das ASCREEN-Menü

Ganz links in der Menüleiste steht das ASCREEN-Menü. Genaugenommen hat es gar nicht allzuviel mit ASCREEN zu tun, aber es ist üblich, daß das Programm, dem die gerade aktive Menüleiste gehört, sich im linkesten Titel identifiziert. In diesem Menü gibt es den ebenfalls üblichen Über...-Eintrag, der bei ASCREEN auch Informationen über das Kopieren enthält, und die Accessories, gegen deren Verwendung ASCREEN keine Einwände ins Feld führt. (Andere Programme sehen das mitunter anders.)

4.3 Das Datei-Menü

Dieses Menü stellt verschiedene Dateifunktionen zur Verfügung. Der unterste Eintrag, Programm beenden, tut genau das. Nach Verabredung ist dieser Eintrag immer der unterste im zweiten Menü.

Der Eintrag Öffnen... läßt eine Dateiauswahlbox erscheinen, in der Sie eine DVI-Datei zum Anzeigen auswählen können. Der zunächst angezeigte Pfad ist immer der der letzten gewählten Datei oder — wenn das Programm gerade gestartet wurde — die Voreinstellung aus der Setupdatei oder von der Kommandozeile.

Aha Im Gegensatz zu vielen anderen Programmen enthält ASCREEN keine eigene Dateiauswahlbox. Seit TOS 1.04, finde ich, ist die des Systems für alle praktischen Zwecke gut genug. Wenn die *Ihnen* zu primitiv ist, müssen Sie sich eben eine andere installieren.

Mit dem Eintrag Nochmal einlesen können Sie die gerade angezeigte DVI-Datei neu einlesen. „Was soll das?“ werden Sie sich jetzt fragen. Nun, im Zeitalter des Multitasking, dessen Morgenröte auf dem Atari ST inzwischen anbricht, ist es ohne weiteres denkbar, daß Sie TEX, ASCREEN und einen Editor gleichzeitig laufen lassen und ASCREEN immer nur die neueste Version desselben Textes anzeigen soll. Sie sparen sich mit der Wiederlesefunktion also den Aufwand, das Programm jedesmal zu beenden und neu zu starten. Es ist sogar möglich, ASCREEN automatisch mit dem Ende eines TEX-Laufs vom Vorliegen einer neuen Version der DVI-Datei zu unterrichten. Lesen Sie dazu im Anhang A über die Nachrichtenwarteschlange von GEM nach.

4.4 Das **Extra**-Menü

Im Extra-Menü sind einige weitere Befehle gesammelt. Die meisten davon beschäftigen sich mit Fenstern:

Fenster öffnen öffnet ein neues Fenster mit derjenigen Seite, die im bisher obersten Fenster zu sehen ist. Es ist gar kein Problem, dieselbe Seite gleichzeitig in mehreren Fenstern anzuschauen, da `ASCREEN` sich intern merkt, wie oft eine Seite gerade zu sehen ist und so beim Blättern nicht durcheinanderkommt.

Fenster auf Seite... ist Fenster öffnen sehr ähnlich. Allerdings wird nicht der Inhalt des oberen Fensters übernommen, sondern es erscheint dieselbe kleine Dialogbox wie bei der Gehe zu...-Funktion im Klappmenü bei der linken Maustaste. Hier können Sie eine Seitenzahl angeben, die `ASCREEN` dann in dem neuen Fenster anzeigt.

Nächstes Fenster dient dazu, die Fenster in zyklischer Reihenfolge durchzublättern. Es wird immer ein unteres Fenster nach oben geholt, bis jedes Fenster mal oben war. Mit Fenster schließen wird das oberste Fenster vom Bildschirm entfernt.

Die Fensterkommandos funktionieren auch für das randlose Fenster, das heißt, wenn Sie auf dem Bildschirm keinen Platz für Rollbalken und ähnliche Spielereien verschenken wollen und die erste Seite eines Dokuments, wie weiter unten beschrieben, in einem Fenster ohne Kontrollelemente auf den Bildschirm bringen, können Sie dieses Fenster dennoch mit Nächstes Fenster ganz nach vorne bringen (damit kann man schnell und bequem zwischen normaler und verkleinerter Darstellung hin und her schalten) oder mit Fenster schließen löschen.

Das Kommando Statistik schreibt eine Statistik des Speicherbedarfs und des Programmablaufs ins Nachrichtenfenster. Dazu gehören Daten wie die Größe des beanspruchten Speichers für formatierte Seiten, für Zeichensätze und für Glyphen (ausgepackte Zeichen) sowie die Anzahl der formatierten Seiten, der ausgepackten und gesetzten Glyphen und der gezogenen Linien. Es wird auch angegeben, ob der Platz zum Auspacken der Glyphen groß genug ist oder nicht.

Mit dem Kommando Hilfe können Sie eine Dialogbox auf den Bildschirm holen, die die wichtigsten Menü- und Tastaturkommandos von `ASCREEN` kurz aufzählt.

4.5 Das **Optionen**-Menü

Aus diesem Menü heraus können Sie verschiedene Dialogboxen aufrufen, mit denen Sie die Arbeitsweise von `ASCREEN` steuern können. Nahezu alle in der Setupdatei möglichen Einstellungen lassen sich auch in den Dialogen vornehmen und umgekehrt.

Eine genaue Diskussion der Optionen steht im Kapitel 5. Dort ist auch angegeben, in welcher Dialogbox jede Option eingestellt werden kann.

Interessant sind an dieser Stelle lediglich die beiden untersten Einträge: Lesen... erlaubt das Einlesen einer vorher gespeicherten Setupdatei, und Schreiben... legt die aktuellen Einstellungen in einer passenden Datei ab. Sie können also jederzeit einen Schnappschuß der gerade gültigen Einstellungen abspeichern und diese später mit Lesen... wieder wirksam machen.

The primary test of the user interface
is its success with users.
— APPLE COMPUTER INC., *Human Interface Guidelines* (1987)

Kapitel 5

Voreinstellungen und Optionen

5.1 Die Setupdatei

Es wurde ja schon ein paarmal angedeutet: Voreinstellungen für `ASCREEN` können in einer *Setupdatei* abgelegt und *en bloc* wieder eingelesen werden. Hier lernen Sie nun den Aufbau einer solchen Datei und die dort möglichen Einstellungen kennen. Im gleichen Aufwasch erkläre ich auch die Kommandooptionen, über die ich vor der Aufzählung aller Einstellungen noch kurz etwas sagen will.

Eine Setupdatei ist zeilenweise aufgebaut; bis auf Leerzeichen leere Zeilen werden überlesen und Zeilen, die mit einem Prozentzeichen `%` anfangen, als Kommentarzeilen angesehen und ignoriert (wie bei TEX). Andere Zeilen beginnen mit einer *Variablen*, gefolgt von Freiraum (Leerzeichen oder Tabulatorzeichen) und einem optionalen Gleichheitszeichen, auf die wiederum der *Wert* der Variablen folgt. Dieser erstreckt sich bis zum nächsten Leer- oder Tabulatorzeichen. Der Rest der Zeile wird ignoriert, kann also für Kommentare benutzt werden. Zeilen in der Setupdatei dürfen maximal 256 Zeichen lang sein. Beachten Sie aber, daß für einige Werte wie Pfadnamen in den programminternen Dialogen nur eine begrenzte Breite vorgesehen ist, die deutlich geringer ist als dieses Maximum. Sie können in der Setupdatei die volle Zeilenlänge ausnutzen, wenn Sie den betreffenden Dialog nicht aufrufen, denn das führt dazu, daß der Wert der entsprechenden Variablen auf die maximale im Dialog mögliche Länge gekürzt wird.

Die mitgelieferte Datei `ASCREEN.SET` enthält alle Variablen in alphabetischer Reihenfolge; dies ist aber keine Bedingung. Die Variablen in der Setupdatei können mehrmals und in beliebiger Folge auftreten; kommt dieselbe Variable mehrmals vor, so gilt der letzte Wert.

Aha Daß die mitgelieferte Datei `ASCREEN.SET` alle Variablen sortiert aufführt, liegt daran, daß sie mit der Funktion Sichern... im `Optionen`-Menü angelegt wurde. Diese Funktion arbeitet nun mal so.

5.2 Kommandooptionen

Die Kommandooptionen wurden ja schon im Kapitel 2 beschrieben. Hier muß über ihr prinzipielles Aussehen also nichts mehr gesagt werden. Jede Kommandooption entspricht einer Setupvariablen und kann diese überschreiben. Ich beschränke mich demnach darauf, die Kommandooptionen bei den dazugehörigen Setup-Variablen zu erwähnen und zum Schluß noch einmal alphabetisch aufzuzählen.

Aha Eben habe ich Sie belogen: Es gibt eine Kommandooption, die keiner Variablen im Setup entspricht und daher gesondert erklärt werden muß. Die Option `-s` hat ein Argument, das genau wie eine Zeile in der Setupdatei bearbeitet wird. Mit `-s` kann man also auch solche Variable setzen, die keine eigene Kommandooption haben (und von denen gibt es einige). Der Aufruf

```
ascreen -sfocus=Pointer foo
```

zum Beispiel sorgt dafür, daß Tastendrücke das Fenster beeinflussen, in dem der Mauszeiger steht, auch wenn dieses Fenster nicht das oberste ist.

Aha Wenn Sie das X11-System kennen, werden Sie sich an die Standard-X-Toolkit-Option `-xrm` erinnert fühlen, die die Option `-s` inspiriert hat.

5.3 Variable in der Setupdatei

Hier nun die versprochene Liste aller Variablen, die in der Setupdatei angegeben werden können. Ich führe sie in alphabetischer Reihenfolge auf und gebe, falls angebracht, Querverweise.

Bei jeder Variablen ist auch der *Typ* angegeben, den ihr Wert haben muß. *Numerische* Werte können in dezimaler, oktaler oder sedezimaler Darstellung angegeben werden; die Syntax dabei entspricht der der Sprache C: der dezimale Wert `123` ist oktal `0173` und sedezimal `0x7b`. Der ebenfalls aufgeführte *Standardwert* tritt in Kraft, wenn die Setupdatei nicht existiert oder nicht gefunden werden kann. Schließlich steht einer der Buchstaben G, D, S, M, oder P für den Dialog im Optionen-Menü, in dem die Variable innerhalb von `ASCREEN` geändert werden kann, respektive „GEM-Optionen“, „DVI-Optionen“, „Seitenformat“, „Speicher“ und „Pfade“. Ein Strich (–) heißt, daß die Variable innerhalb von `ASCREEN` nicht zugänglich ist.

cacheahead (Numerisch, Standard 1, D, Option `-A`) Hat diese Variable einen von 0 verschiedenen Wert, so arbeitet `ASCREEN` eine Seite voraus: wenn eine neue Seite zur Anzeige kommt, formatiert `ASCREEN` gleich die nächste und tut sie in den Cachespeicher. Damit wird die „Lesezeit“ sinnvoll ausgenutzt, und die nächste Seite ist oft schon fertig, wenn sie gezeigt werden soll. Hat die Variable den Wert 0, so wird nicht vorausgearbeitet.

cachepages (Numerisch, Standard 5, M, Option `-p`) Die maximale Anzahl von Seiten, die `ASCREEN` im Cachespeicher hält (genug RAM-Speicher vorausgesetzt). Werte, die

kleiner als 1 sind, werden als 1 interpretiert.

colors (Numerisch, Standard 1, G, Option `-c`) Die Farben, die `ASCREEN` zur Anzeige des Dokuments benutzen soll, und zwar gemäß der Formel

$$256 \cdot \text{Hintergrund} + \text{Vordergrund}.$$

Der Standardwert 1 ergibt schwarze Schrift auf weißem Grund und muß normalerweise nicht geändert werden. Ansonsten ist diese Variable wohl die einzige, bei der die hexadezimale Eingabemöglichkeit sich so richtig lohnt.

dvibuflimit (Numerisch, Standard 20000, M) `DVI`-Dateien bis zu dieser Größe werden komplett ins RAM gelesen. Wenn Sie viel RAM-Speicher haben, können Sie durch eine großzügige Einstellung durchaus Zeit sparen. Siehe Kapitel 6 für eine wichtige Anwendung dieser Variablen.

dvibufsize (Numerisch, Standard 8192, M) `ASCREEN` puffert Zugriffe auf die `DVI`-Datei in Blöcken dieser Größe. Wenn Sie gerne an Knöpfen drehen, dann können Sie versuchen, hier einen für Ihren Rechner optimalen Wert zu finden (er dürfte von Ihrem Betriebssystem, Ihrer Platte und Ihrem Hostadapter abhängen). Ich jedenfalls bin mit dem Standardwert zufrieden und habe auch nicht viel damit herumprobiert.

dvipath (Verzeichnis, Standard `d:\texout`, P, Option `-D`) In diesem Verzeichnis sucht `ASCREEN` nach `DVI`-Dateien, deren Namen keine explizite Pfadangabe enthalten. Wenn Ihr `dvipath` nicht das aktuelle Verzeichnis bezeichnet, Sie aber eine Datei in demselben anzeigen wollen, so können Sie etwas wie `ascreen .\foo` sagen.

emergencyfont (Zeichensatzname, Standard `cmr10`, –) `ASCREEN` verwendet diesen Zeichensatz, wenn ein in der `DVI`-Datei angeforderter Zeichensatz nicht existiert. Bei vergrößerten Zeichensätzen wird zunächst versucht, den Zeichensatz in der Grundvergrößerung zu finden, und `ASCREEN` greift erst auf den `emergencyfont` zurück, wenn dieser auch nicht da ist.

emergencysize (Dimension, Standard 10 pt, –) Die Größe des `emergencyfont`, inklusive der Vergrößerung. Wäre der `emergencyfont cmr10 scaled1200`, so müßten Sie hier `12pt` angeben.

extension (Zeichenkette, Standard `scr`, –, Option `-e`) Die TEX-Implementierung von Stefan Lindner und Lutz Birkhahn verwendet für Zeichensätze Verzeichnisse, die die Auflösung des Geräts und eine Abkürzung für den Gerätetyp enthalten. `ASCREEN` muß diese Abkürzung wissen, um korrekte Zeichensatzlisten im Lindner-Stil schreiben zu können. Sie können mit dieser Variablen die Geräteabkürzung einstellen. Für den Strunk-Stil ist die Abkürzung unerheblich.

feeponerror (Numerisch, Standard 1, G) Wenn diese Variable einen von 0 verschiedenen Wert hat, so piept `ASCREEN`, wenn eine Fehlermeldung ausgegeben wird. Leider gibt es im „normalen“ GEM keine Funktion zum Piepen, so daß das BIOS benutzt werden muß; wenn das nicht klappen sollte (wer weiß?) oder das Piepen stört, kann man es

hiermit abschalten.

focus (Fokusbezeichnung, Standard `Window`, G) Normalerweise beziehen Tastaturkommandos zum Rollen oder Blättern sich auf das aktive (obenliegende) Fenster. Sie können hier wählen, ob die Tastaturkommandos statt dessen auf das Fenster wirken sollen, in dem der Mauszeiger gerade steht, was bequem sein kann, aber vom normalen Verhalten der AES abweicht. Die möglichen Werte sind `Window`, wenn die Tastaturkommandos für das obenliegende, und `Pointer`, wenn die Tastaturkommandos für das Fenster mit dem Mauszeiger gelten sollen.

fontname (Schablone, Standard `~\res%r.%e\mag____%5.3M\%f.pk`, P, Option `-z`) Jede TEX-Implementierung hat eigene Ansichten darüber, wie die Dateinamen für Zeichensätze aussehen sollen. Ein flexibler Previewer wie `ASCREEN` muß aber in der Lage sein, sich jeder Situation anzupassen. Dazu dient die Zeichensatzschablone, die so ähnlich funktioniert wie die C-Funktion `printf()`: besondere Zeichen in einer vorgegebenen Zeichenkette werden durch bestimmte Werte ersetzt, durch die sich der Dateiname ergibt. Allgemein gesagt bewirkt ein Prozentzeichen (`%`), daß etwas Besonderes in den Namen eingebaut wird; was genau, bestimmt das folgende Zeichen gemäß der Tabelle:

`%f`	Name des Zeichensatzes (ohne `.pk`)
`%m`	Vergrößerung in Promille als Ganzzahl
`%M`	Vergrößerung als Dezimalzahl
`%r`	(horizontale) Auflösung des Geräts in dpi
`%R`	fünffache Auflösung des Geräts in dpi
`%p`	(horizontale) Auflösung des Geräts multipliziert mit der Vergrößerung, als Ganzzahl
`%P`	Analog zu `%p` wie `%R` zu `%r`
`%e`	Der Wert der `extension`

Wie bei `printf()` kann man zwischen dem Prozent und dem Formatbuchstaben die Feldbreite und die Anzahl der Dezimalen angeben. So entspricht die Angabe `%5.3M` aus dem Standardwert der Vergrößerung des Zeichensatzes als Dezimalzahl, und zwar fünf Zeichen breit mit drei Nachkommastellen. Alle Werte werden über den `printf()`-Formatschlüssel für Ganzzahlen bearbeitet, mit Ausnahme von `%M` (Fließkommazahl) und `%f` und `%e` (Zeichenketten).

Aha Die Formatbuchstaben mit der Verfünffachung haben vornehmlich historische Bedeutung: in der Anfangszeit von TEX benutzte Donald E. Knuth diverse eigenartige Drucker von Xerox mit einer Auflösung von 200 dpi. Wenn man dies mit 5 multipliziert, kommt man gerade auf 1000, und Zeichensatzvergrößerungen werden bei TEX ja traditionell in Promille angegeben. Heute haben die wenigsten Drucker 200 dpi Auflösung, aber die Verfünffachung hält sich hartnäckig etwa bei der TEX-Implementierung von TEXsys (Schrod & Co., „TH-Darmstadt-TEX“). Brrr.

Eine Tilde (`~`) am Anfang des `fontname` wird durch die gerade durchsuchte Komponente des `fontpath` ersetzt.

fontpath (Suchpfad, Standard `d:\prtfonts`, P, Option `-F`) Eine durch Kommata getrennte Liste von Verzeichnisnamen, in denen nach Zeichensätzen gesucht wird. Normaler-

weise werden die genannten Verzeichnisse in Folge durchsucht, wobei an jedes der per `fontname` gebildete eigentliche Name des Zeichensatzes angehängt wird. Im `fontpath` können auch Zeichensatzbibliotheken mit der Dateiendung `.fli` benannt werden.

glyphmem (Numerisch, Standard 64000, M) Soviel Speicher (in Bytes) benutzt `ASCREEN` zum Auspacken von Zeichen, die in `PK`-Dateien ja gepackt vorliegen. Reicht der Speicher nicht, werden alle ausgepackten Zeichen weggeworfen und es wird mit leerem Speicher weitergemacht, was natürlich einen Effizienzverlust bedeutet. Es zeigt sich, daß der Standardwert normalerweise jenseits von Gut und Böse liegt, wie Sie mit der Statistikfunktion leicht prüfen können.

grafpath (Suchpfad, Standard `d:\graphics`, P, Option `-G`) Eine durch Kommata getrennte Liste von Verzeichnisnamen, in denen nach Graphiken im `IMG`-Format gesucht wird.

hresolution (Numerisch, Standard 101, P, `-r`) Die horizontale Auflösung der zu verwendenden Zeichensätze. Der Standardwert von 101 dpi ist der niedrigste Wert, für den Donald E. Knuth die Computer-Modern-Zeichensätze getestet hat. Die tatsächliche Auflösung eines Atari SM 124 liegt wohl mehr so bei 70 dpi; die meisten TEX-Implementierungen für den ST verwenden Werte um 100 dpi. Die vertikale Auflösung ist normalerweise gleich der horizontalen. Siehe auch `vresolution`. Die Kommandooption `-r` gestattet die Formen `-r`*h* oder `-r`*h*`x`*v*; im ersteren Fall wird die vertikale Auflösung gleich der horizontalen gesetzt.

hypertex (Numerisch, Standard 1, D) Hat diese Variable einen von 0 verschiedenen Wert, so interpretiert `ASCREEN` „HyperTEX-`\specials`“. Diese besonderen Kommandos können in `DVI`-Dateien eingebaut werden, um Querverweise zu ermöglichen, die durch Anklicken weiterverfolgt werden können. Anhang D erklärt dies genauer. Hat die Variable den Wert 0, so werden die `\specials` stillschweigend ignoriert.

leftmargin (Dimension, Standard 0 pt, S, Option `-H`) Die Breite des angezeigten linken Randes. Der Standardwert 0 pt bewirkt, daß der linke Rand der Seite aus der Sicht von TEX und der linke Rand des Fensters zusammenfallen, was auf einem kleinen Bildschirm Platz spart, aber mitunter Sachen abschneidet, die in den linken Rand ragen. Der konservative Wert ist 1 in, was der TEX-Vorgabe entspricht, aber oft Platz verschenkt und zu mehr horizontalen Rolloperationen zwingt. Siehe auch `topmargin`, `pageheight` und `pagewidth`.

logging (Numerisch, Standard 0, D, Option `-l`) Hat diese Variable einen von 0 verschiedenen Wert, so werden alle Ausgaben, die ins Protokollfenster geschrieben werden, auch in einer Datei vermerkt. Den Namen für diese Protokolldatei konstruiert `ASCREEN` aus dem Namen der `DVI`-Datei, indem die Endung `.dvi` durch `.prt` ersetzt wird; der Pfad wird anderweitig beibehalten.

magnification (Vergrößerung, Standard `magstep0`, –, Option `-M`) Mit dieser Variablen können Sie die Vergrößerungsangabe in der `DVI`-Datei überschreiben. Der Wert kann in Promille oder in der TEX-Form `magstep0`...`5` bzw. `magstephalf` angegeben werden.

maxfonts (Numerisch, Standard 32, M, Option `-Z`) Maximalzahl von Zeichensätzen in einem Dokument.

memcheck (Numerisch, Standard 0, G) Hat diese Variable einen von 0 verschiedenen Wert, so wird am rechten Ende der Menüzeile der jeweils gerade freie Systemspeicher angezeigt. Dieser wird mit der Funktion `Malloc(-1)` ermittelt und muß daher nicht stimmen (insbesondere ist der Wert die Größe des gerade größten freien Speicherblocks). Außerdem ist es nicht gerade höflich, Sachen rechts in die Menüzeile zu schreiben, wo doch die Uhr steht :-), und darum können Sie es hier abschalten.

missingfile (Dateiname, Standard `missing.fnt`, P) Der Name einer Datei, in der die Liste fehlender Zeichensätze abgelegt wird. Im Lindner-Stil wird dieser Name verwendet; im Strunk-Stil wird der Name aus dem Namen der `DVI`-Datei abgeleitet, indem die Endung `.dvi` durch `.fts` ersetzt wird.

missingstyle (Stilangabe, Standard `Lindner`, P) Diese Variable hat den Wert `Lindner` oder `Strunk`, je nachdem, ob Zeichensatzlisten im Format der TEX-Implementierungen von Stefan Lindner oder Christoph Strunk angelegt werden sollen (sorry Lutz).

movebyenter (Numerisch, Standard 1, D, Option `-m`) Wenn diese Variable einen von 0 verschiedenen Wert hat, so wird beim Druck auf die Taste `Enter` zunächst die gezeigte Seite nach unten gerollt, bis der untere Seitenrand erreicht ist. Danach wird zum oberen Rand der nächsten Seite geblättert. Die Ziffernblocktaste `.` wirkt analog umgekehrt. Hat die Variable den Wert 0, so wirken `Enter` und `.` wie die Tasten `+` und `-` auf dem Ziffernblock, respektive.

msgraise (Numerisch, Standard 1, G) Hat diese Variable einen von 0 verschiedenen Wert, so wird das Protokollfenster nach oben geholt, wenn etwas darin ausgegeben wurde. Hat die Variable den Wert 0, so bleibt es, wo es ist.

msgwindow (Geometrie, Standard `256x160+320+154`, –, Option `-w`) Größe und Position des Protokollfensters. Die ersten beiden Zahlen geben die Größe in Pixeln an, die letzten beiden die Position, wobei positive Werte den Abstand zwischen dem linken bzw. oberen Bildschirm- und Fensterrand, negative Werte dagegen den Abstand zwischen dem rechten bzw. unteren Bildschirm- und Fensterrand bezeichnen. Für diese Variable gibt es keine Einstellungsmöglichkeit in einem Dialogformular, denn Sie können ja das Fenster auf dem Bildschirm herumschieben.

nokeypress (Option `-J`) Diese Variable ist nur aus Kompatibilitätsgründen zu früheren Versionen von `ASCREEN` vorhanden und wird ignoriert.

pageheight (Dimension, Standard 0 sp, S, Option `-y`) Die maximale Höhe einer TEX-Seite. Der Standardwert 0 pt steht für „was auch immer in der `DVI`-Datei steht“. Der Wert, den `ASCREEN` verwendet, ist das Maximum des hier angegebenen Werts und des Werts in der `DVI`-Datei. Siehe auch `pagewidth`.

pagetable (Numerisch, Standard 1, D) Wenn diese Variable einen von 0 verschiedenen Wert hat, so legt `ASCREEN` eine Liste der bereits betrachteten Seiten und ihrer Positionen

in der DVI-Datei an. Dadurch ist es möglich, diese Seiten schnell wiederzufinden, was vor allem das Herumspringen in sehr großen Dokumenten beschleunigt. Hat diese Variable den Wert 0, so wird diese Liste nicht angelegt.

pagewidth (Dimension, Standard 0 pt, S, Option -x) Die maximale Breite einer TEX-Seite. Der Standardwert 0 pt steht für „was auch immer in der DVI-Datei steht“. Der Wert, den ASCREEN verwendet, ist das Maximum des hier angegebenen Werts und des Werts in der DVI-Datei. Siehe auch pageheight.

rightbutton (Numerisch, Standard 1, G) Wenn diese Variable einen von 0 verschiedenen Wert hat, benutzt ASCREEN einen nicht offiziell dokumentierten Trick, um beide Mausknöpfe gleichzeitig abzufragen und das Herumzerren des Fensterinhalts mit dem rechten Mausknopf zu erlauben. Hat diese Variable den Wert 0, so wird nur der linke Mausknopf abgefragt und das Herumzerren des Fensterinhalts über die Alternate-Taste ermöglicht.

rvmode (Numerisch, Standard 0, D) Wenn diese Variable einen von 0 verschiedenen Wert hat, werden verkleinerte Doppelseiten standardmäßig in den Rekto-Verso-Modus versetzt. Hat diese Variable den Wert 0, so geschieht das nicht.

savemem Diese Variable ist nur aus Kompatibilitätsgründen zu früheren Versionen von ASCREEN vorhanden und wird ignoriert.

showgraphics (Numerisch, Standard 1, D, Option -I) Wenn diese Variable den Wert 0 hat, so werden Graphikbefehle zur Anzeige von IMG-Dateien ignoriert. Hat die Variable einen von 0 verschiedenen Wert, so werden IMG-Dateien angezeigt.

shrinkfactor (Numerisch (2 oder 3), Standard 2, D, Option -f) Der Verkleinerungsfaktor. Die einzigen gültigen Werte sind 2 und 3.

smartpaging (Numerisch, Standard 1, D) Hat diese Variable einen von 0 verschiedenen Wert, so führt Vorwärtsblättern am unteren Seitenrand an den oberen Rand der folgenden Seite und analog Rückwärtsblättern am oberen Seitenrand an den unteren Rand der vorigen Seite Hat diese Variable den Wert 0, so bleibt die Position auf der Seite beim Blättern erhalten.

stats Diese Variable ist nur aus Kompatibilitätsgründen zu früheren Versionen von ASCREEN vorhanden und wird ignoriert.

tfmpath Diese Variable ist für künftige Erweiterungen vorgesehen und wird einstweilen ignoriert.

topmargin (Dimension, Standard 0 pt, S, Option -V) Die Breite des angezeigten oberen Randes. Der Standardwert 0 pt bewirkt, daß der obere Rand der Seite aus der Sicht von TEX und der obere Rand des Fensters zusammenfallen, was auf einem kleinen Bildschirm Platz spart, aber mitunter Sachen abschneidet, die in den oberen Rand ragen. Der konservative Wert ist 1 in, was der TEX-Vorgabe entspricht, aber oft Platz verschenkt Siehe auch leftmargin, pageheight und pagewidth.

usedesk (Numerisch, Standard 0, G, Option `-d`) Hat diese Variable einen von 0 verschiedenen Wert, so erscheint die erste Seite des Dokuments in einem Fenster ohne jegliche Kontrollelemente, um Platz auf dem Bildschirm zu sparen. Der Name der Variablen stammt aus einer Zeit, als `ASCREEN` zu diesem Zweck ein eigenes Desktop anmeldete; das ist heute verpönt, und das randlose Fenster tut seinen Job genausogut, wenn nicht gar besser (denn man kann es über andere Fenster hinweg nach oben holen). Hat diese Variable den Wert 0, so werden ausschließlich „normale“ Fenster benutzt.

verbose (Numerisch, Standard 0, D, Option `-v`) Hat diese Variable einen von 0 verschiedenen Wert, so führt `ASCREEN` ein ausführliches Protokoll über seine Tätigkeit, in dem zum Beispiel die geladenen Zeichensätze und die dazugehörigen Dateien auftauchen. Hat diese Variable den Wert 0, so geschieht das nicht.

vfpath Diese Variable ist für künftige Erweiterungen vorgesehen und wird einstweilen ignoriert.

vresolution (Numerisch, Standard 0, P, `-r`) Die vertikale Auflösung der zu verwendenden Zeichensätze. Normalerweise ist die vertikale Auflösung gleich der horizontalen, was Sie auch durch den Wert 0 zum Ausdruck bringen können. Einige Funktionen, etwa die Graphikbefehle, dürften nicht richtig funktionieren, wenn die Auflösungen nicht übereinstimmen. Daß man überhaupt verschiedene Werte angeben kann, liegt daran, daß der `DVI`-Interpreter in `ASCREEN` im Prinzip auch Drucker mit eigenartigen Auflösungen bedienen kann (selbst wenn das noch nicht implementiert wurde). Die Kommandooption `-r` gilt für beide Auflösungen, wobei deren Form `-r` ⟨h⟩`x`⟨v⟩ die horizontale und die vertikale Auflösung setzt.

Das waren also alle Setup-Variable. Hier noch die versprochene alphabetische Aufstellung der Kommandooptionen mit den dazugehörigen Variablen:

`-A`	`cacheahead`	`-m`	`movebyenter`
`-c`	`colors`	`-p`	`cachepages`
`-D`	`dvipath`	`-r`	`hresolution`
`-d`	`usedesk`		`vresolution`
`-e`	`extension`	`-s`	beliebig
`-F`	`fontpath`	`-V`	`topmargin`
`-f`	`shrinkfactor`	`-v`	`verbose`
`-G`	`grafpath`	`-w`	`msgwindow`
`-H`	`leftmargin`	`-x`	`pagewidth`
`-I`	`showgraphics`	`-y`	`pageheight`
`-J`	`nokeypress`	`-Z`	`maxfonts`
`-l`	`logging`	`-z`	`fontname`
`-M`	`magnification`		

Heuer's Law: Any feature that cannot be turned off is a bug.
— KARL W. Z. HEUER (The Walking Lint)

Kapitel 6

Tips und Tricks

6.1 Zeichensatzbibliotheken

Zeichensatzbibliotheken, neudeutsch auch *Fontlibraries*, sind eine hilfreiche Erfindung: die vielen kleinen `PK`-Dateien für Zeichensätze sind unübersichtlich und umständlich zu verwalten. Außerdem bewirken sie einen beträchtlichen Verschnitt auf der Platte, deren Platz ja in Blöcken von 1 KB oder mehr vergeben wird — ein Zeichensatz von 2100 Bytes braucht auf meiner Platte zum Beispiel 2 2048-Byte-Blöcke, wodurch ich gut 2000 Bytes verschenke. Eine Zeichensatzbibliothek faßt eine Gruppe von Zeichensätzen (`PK`-Dateien) zu einer großen Datei zusammen. Da die Zeichensätze in der Bibliothek dicht an dicht gepackt sind, tritt kein Verschnitt auf (außer dem am Dateiende, aber der ist dann nicht mehr schlimm). Je nach der Flottheit Ihrer Platte und der Version Ihres Betriebssystems kann es auch sein, daß Zeichensätze in einer Zeichensatzbibliothek durchaus schneller gefunden werden als solche in einzelnen Dateien, da nicht so viele Zugriffe auf Verzeichnisse nötig werden. `ASCREEN` liest das Inhaltsverzeichnis einer Zeichensatzbibliothek einmal und verwendet dann die Kopie im RAM.

`ASCREEN` unterstützt Zeichensatzbibliotheken in Eberhard Mattes' FLIB-Format, das für *emTEX* definiert wurde. *emTEX* ist die zur Zeit wohl leistungsfähigste TEX-Implementierung für PC-artige Computer unter DOS und OS/2 und stellt dort einen *de-facto*-Standard dar, an den `ASCREEN` sich damit anhängt.

Aha Auf dem Atari ST gibt es nur eine TEX-Implementierung, die Zeichensatzbibliotheken von Hause aus unterstützt, und das ist das kommerzielle TEX von TooLs. Die Verbreitung dieser Version ist allerdings gegenüber der der Implementierungen von Strunk und Lindner als vernachlässigbar anzusehen. `ASCREEN` ignoriert das TooLs-Format deshalb mit Absicht, auch um dem kommerziellen Vertrieb eines an sich freien Programms wie TEX nicht noch weiteren Vorschub zu leisten. Sorry.

Aha In der PC-Welt sind FLIB-Dateien für diverse Drucker im Umlauf. Alle Leute, die auf dem Atari TEX benutzen, sollten auf die Implementierer von Treibern Druck [sic!] ausüben, damit diese ihren Programmen das FLIB-Format beibringen. Der Aufwand dafür ist nicht groß, aber ein solcher Standard

könnte hier eine große Hilfe sein, nicht zuletzt um mit den vielen verschiedenen Verzeichnisschemata für Zeichensatznamen aufzuräumen.

Sie können `ASCREEN` dazu bringen, eine Bibliothek nach Zeichensätzen zu durchsuchen, indem Sie die FLIB-Datei im Suchpfad für Zeichensätze erwähnen. `ASCREEN` würde zum Beispiel mit der Zeile

```
fontpath d:\latex101.fli,d:\prtfonts
```

in der Setupdatei einen Zeichensatz zunächst in der Bibliothek `d:\latex101.fli` suchen. Ist der Zeichensatz dort nicht enthalten, sucht `ASCREEN` im Verzeichnis `d:\prtfonts` nach einer passenden einzelnen `PK`-Datei gemäß der Vorgabe für die Namensstruktur. (Für das Suchen in Zeichensatzbibliotheken ist die Namensstruktur unerheblich.) Selbstverständlich können Sie auch mehrere Zeichensatzbibliotheken angeben oder ein normales Verzeichnis an den Anfang des Suchpfads stellen.

Das FLIB-Format ist im Anhang C ausführlich dokumentiert. Leider gibt es auf dem Atari im Moment kein bequemes Programm zur Verwaltung von Zeichensatzbibliotheken. Vielleicht fühlt sich jemand inspiriert, ein solches Programm zu schreiben und (vorzugsweise) frei verfügbar zu machen. Einstweilen kann ich nur verschiedene vorgefertigte Zeichensatzbibliotheken und ein paar in Perl geschriebene Tools zum Einpacken und Auflisten von Zeichensatzbibliotheken anbieten, die als Basis für ein Verwaltungsprogramm dienen könnten. Wer Zugriff auf einen PC mit *emTEX* hat, kann das dortige Programm benutzen.

6.2 Spaß mit Druckern

Es ist leicht möglich, `ASCREEN` auch mit Druckerzeichensätzen zu benutzen. Vielleicht ist das für Sie interessant, wenn Sie mal etwas ganz genau nachschauen wollen. Am besten machen Sie sich eine Setupdatei, die für die Variablen `hresolution` und `vresolution` die Werte Ihres Druckers und für `fontpath`, `fontname` und `extension` die für Ihre TEX-Implementierung angebrachten Namen enthält. In meiner Datei `p6.set` stehen zum Beispiel die Zeilen

```
hresolution 360
vresolution 360
extension   neh
fontpath    d:\prtfonts
fontname    ~\res%r.%e\mag____%5.3M\%f.pk
```

(das sind die korrekten Werte für Lindner-TEX; der Wert für `fontpath` hängt natürlich von der Installation ab). Ich kann dann mit einem Kommando wie

```
ascreen &d:\tex\p6.set foo
```

die Datei `foo.dvi` in der Druckerauflösung anschauen. Der Anfangsteil des Kommandos ist natürlich ein hervorragender Kandidat für eine Aliasdefinition oder ein Shellskript.

Die dreieinhalbfache Auflösung des NEC P6 gegenüber dem Bildschirm führt selbstverständlich zu einer Vergrößerung des Speicherplatzes pro Seite auf mehr als das Zehnfache. Auf einem Atari mit 1 MB RAM dürfte dieser Trick also nicht funktionieren, zwei Megabyte sind für normal große Seiten wohl schon nötig. Jedenfalls passen entsprechend weniger Seiten in den Cachespeicher, und auch die Laufzeit steigt beträchtlich an. Trotzdem kann diese vergrößerte Darstellung ab und zu ganz sinnvoll sein.

6.3 Multitasking

Prinzipiell läuft `ASCREEN` unter Multitaskingerweiterungen wie PAM's MultiGEM, MiNT oder dem neuen MultiTOS. Hier noch ein paar Tips für optimale Ergebnisse.

Das Angenehme am Multitasking ist — zumindest auf TEX bezogen —, daß Sie den Editor nicht verlassen müssen, um TEX zu starten. Auch `ASCREEN` müssen Sie nicht beenden, wenn TEX eine neue Fassung der `DVI`-Datei bereitgestellt hat: Die Funktion Nochmal einlesen... macht dies unter normalen Umständen unnötig. Allerdings gibt es da ein kleines Problem: Aus Effizienzgründen besteht `ASCREEN` darauf, die `DVI`-Datei während des gesamten Programmlaufs offenzuhalten. Das führt dazu, daß TEX sie nicht mit der neuen Version überschreiben darf. Die einzige brauchbare Abhilfe setzt einen entsprechenden Speicherausbau voraus und besteht darin, das `dvibuflimit` so hoch zu setzen, daß die `DVI`-Datei komplett in den Puffer paßt. In diesem Fall wird sie nämlich nur beim Öffnen gelesen und gleich wieder geschlossen; TEX kann dann die nächste Version ohne Probleme drüberkritzeln. Wenn Sie nicht gerade *Krieg und Frieden* überarbeiten oder eine Diplom- oder Doktorarbeit abfassen, sollte ein Wert um 100000 für `dvibuflimit` für die meisten Zwecke ausreichen.

Sie können die GEM-Nachrichtenwarteschlange benutzen, um `ASCREEN` automatisch auf die Existenz einer neuen Version der `DVI`-Datei aufmerksam zu machen. Dazu brauchen Sie nur ein kleines GEM-Programm, das `ASCREEN` die entsprechende Nachricht schickt (siehe Anhang A für die Erklärung der Nachrichten). Anschließend müssen Sie nur dafür sorgen, daß dieses Programm nach jedem TEX-Lauf aufgerufen wird — benutzen Sie eine kommandoorientierte Shell, so ist das leicht mit einem Skript zu bewältigen. Leute mit einer richtigen TEX-Shell müssen dieser notfalls ein Programm unterschieben, das `TEX.TTP` oder so ähnlich heißt, das eigentliche TEX unter einem anderen Namen oder aus einem anderen Verzeichnis heraus aufruft und nach dessen Ende die Nachricht auf den Weg bringt. "The implementation is left as an exercise for the reader."

6.4 Ändern der Tastaturkürzel für Menüpunkte

Die meisten Funktionen, die über die Menüleiste am oberen Bildschirmrand zugänglich sind, können auch über Tastaturkürzel angesprochen werden. Kapitel 4 erklärt, wie diese Kürzel aussehen.

Ich habe mir Mühe gegeben, die Tastaturkürzel an die üblichen Gepflogenheiten auf dem Atari anzupassen: Fenster schließen ist `^U`, Nächstes Fenster ist `^W` und so weiter. Trotzdem kann es sein, daß Ihnen die eine oder andere Abkürzung nicht behagt. In diesem Fall können Sie die Kürzel mit einem *Resource Construction Set* editieren. Achten Sie aber darauf, daß Sie die Struktur der Menüs dabei nicht verändern, sonst dürfte die Bedienung von `ASCREEN` zu einem heiteren Ratespiel werden. Um Ihnen den Umgang mit der Resourcedatei zu erleichtern, habe ich eine `DEF`-Datei beigefügt, die die Namen der Objektbäume enthält. Diese `DEF`-Datei dürfte mit den meisten RCS-Programmen zu verwenden sein; ich selbst benutze *Interface* von der Firma *shift*, das ich bei dieser Gelegenheit wärmstens empfehlen möchte[1]. Für das Kuma-RCS muß man `ASCREEN.DEF` in `ASCREEN.RSD` umbenennen; das „neue“ RCS von Digital Research hat, glaube ich, ein Umwandlungsprogramm. Das Hauptmenü befindet sich, wer hätte das gedacht, im Baum mit dem Namen `MAINMENU`.

6.5 Andere Sprachen

`ASCREEN` ist sprachunabhängig geschrieben: alle Texte und Zeichenketten, die normalerweise deutsch sind und in ausländischen Versionen ausländisch sein sollten, stehen in der Resourcedatei. Darum ist es einfach, eine neue fremdsprachliche Benutzungsschnittstelle für `ASCREEN` zu machen: Das Programm selbst muß dafür nicht geändert werden, sondern nur die Resourcedatei. Ich würde mich freuen, wenn jemand zum Beispiel eine französische Version machte — alle ausländischen Fassungen, die ich zugeschickt bekomme, werden mit einem gebührenden Lob in die Distribution aufgenommen.

Alle sprachabhängigen Zeichenketten, die nicht Teil eines Menüs, eines Dialogs oder einer Alertbox sind, stehen im Baum `STRINGS` in der Resourcedatei. Dazu gehören alle Fehlermeldungen, der Titel für das Protokollfenster, die einzelnen Zeilen der Statistik und einiges mehr. Hacken Sie fröhlich drauflos, aber nicht ohne eine Kopie der Originaldatei an einem sicheren Ort untergebracht zu haben. Sie müssen lediglich darauf achten, die Struktur der Objektbäume nicht zu verändern, da `ASCREEN` sonst übel durcheinanderkommen kann. Vermeiden Sie insbesondere das Löschen von Objekten und das Sortieren von Dialogen. Manche Resource Construction Sets haben eine „Sicherheitsoption“, die strukturelle Änderungen an den Objektbäumen nicht zuläßt. Sollten Sie nicht zu den Glücklichen zählen, die ein solches Programm besitzen, kann die mitgelieferte C-Headerdatei einen Anhaltspunkt geben: Vergleichen Sie die Originaldatei mit der, die Ihr RCS bei der geänderten Resourcedatei produziert. Weichen die symbolischen Namen und die Objektindizes in den beiden Fassungen voneinander ab, dann sind Sie in Schwierigkeiten. Stimmen die beiden Dateien überein, dann können Sie einigermaßen sicher sein, daß die neue Resourcedatei funktioniert, ohne in `ASCREEN`s Innereien heftiges Sodbrennen zu verursachen. (Eine hundertprozentige Garantie ist das leider nicht.)

Und noch etwas: Natürlich ist es für Sie eine Ehrensache, daß Sie die Inhalte der drei Dialogboxen `INFOFM`, `COPYFM` und `CREDITFM` so exakt wie nur möglich übersetzen. Selbstverständ-

[1] *Dementi:* Ich habe keine Verbindung zur Firma *shift*, außer als zufriedener Benutzer.

lich dürfen Sie einen angemessenen Hinweis auf Ihre eigene Leistung nicht unterschlagen; eine Zeile wie

```
Traduction française par Monique Leroc
```

im `INFOFM` haben Sie sich sicherlich verdient.

Zuviel Honig essen ist nicht gut;
aber wer nach schweren Dingen forscht,
dem bringt's Ehre.

— DIE SPRÜCHE SALOMOS 25:27 (1. JT v. Chr.)

Anhang A

ASCREEN und GEM-Nachrichten

A.1 GEM-Nachrichten

Alle, die schon einmal ein GEM-Programm geschrieben haben, kennen *Nachrichten* und die Warteschlange: jedes GEM-Programm bekommt so zum Beispiel mitgeteilt, welche Menüpunkte gewählt wurden oder welche Fenster neu gezeichnet werden müssen. Meist werden Nachrichten von den AES erzeugt, dem Teil des Atari-Betriebssystems, das sich mit der graphischen Benutzungsoberfläche befaßt.

Auch `ASCREEN` verarbeitet natürlich diverse GEM-Nachrichten. Dazu gehören aber nicht nur die üblichen Nachrichten wie „Fenster neuzeichnen“ oder „Menüpunkt ausgewählt“, sondern auch einige besondere nur für `ASCREEN` definierte Nachrichten. Hiermit können andere Programme `ASCREEN` bitten, eine `DVI`-Datei einzulesen, die gerade gezeigte `DVI`-Datei nochmals zu lesen, ein neues Fenster zu öffnen und einiges mehr. In diesem Anhang erkläre ich Ihnen diese Nachrichten und zeige auch, wie Sie sie von einem Programm aus verschicken können.

A.2 Nachrichten für ASCREEN

Betrachten wir zuerst die Nachrichten, die für `ASCREEN` definiert sind. Neben den üblichen AES-Nachrichten, die Sie am besten in einem beliebigen GEM-Buch, etwa dem *Profibuch*, nachschlagen und die hier nicht aufgeführt werden, gibt es einige spezielle `ASCREEN`-Nachrichten.

Zunächst der allgemeine Aufbau einer GEM-Nachricht. Nachrichten müssen mindestens 8 Wörter, also 16 Bytes lang sein, sonst wird die Annahme bzw. Zustellung verweigert. Mit der Definition

```
WORD msg[8];
```

gilt die Zuordnung

`msg[0]`	Nachrichtentyp
`msg[1]`	Prozeßnummer des Absenders
`msg[2]`	„Überlänge“ der Nachricht

Der Rest der Nachricht kann ein beliebiges Format haben. Die „Überlänge“ ist die Anzahl von Bytes, die nicht mehr in die 16 Bytes der eigentlichen Nachricht gepaßt haben und daher mit einem expliziten Aufruf von `appl_read()` gelesen werden müssen.

Das erste Wort gibt den *Typ* der Nachricht an, zum Beispiel `MN_SELECTED` (0) für die AES-Nachricht „Menüpunkt gewählt“. Bei `ASCREEN`-Nachrichten hat das erste Wort immer den Wert `MSG_ASCREEN`, die Zahl 8196.

Aha Hierbei handelt es sich um eine willkürliche Definition, die Nummer 8196 ist hoffentlich noch frei...

`ASCREEN` unterscheidet die verschiedenen Arten von Nachrichten, bei denen `msg[0]` gleich `MSG_ASCREEN` ist, am Wert des vierten Worts (`msg[3]`). Hier gibt es die Werte

`MSG_READ`	`DVI`-Datei einlesen
`MSG_REREAD`	`DVI`-Datei wieder einlesen
`MSG_NEWPAGE`	Neue Seite formatieren
`MSG_NEWWIN`	Neues Fenster öffnen

Die numerischen Werte dieser Namen sollten Sie nicht allzusehr interessieren, es sei denn, Sie benutzen abartige Programmiersprachen wie Pascal oder Assembler. Im Verzeichnis `ALAUNCH` der `ASCREEN`-Distribution finden Sie eine Datei `ASCRMSGS.H`, die die entsprechenden Definitionen für C enthält.

Hier nun der Aufbau der einzelnen Nachrichten. Die ersten drei Wörter haben wir ja schon diskutiert; ich spare es mir, sie hier immer wieder mit anzugeben.

MSG_READ ⟨Dateiname⟩...
: Mit dieser Nachricht fordern Sie `ASCREEN` auf, eine neue `DVI`-Datei einzulesen. Die bisherige wird verworfen; ist der Name der bisherigen Datei gleich dem ⟨Dateiname⟩n, so entspricht die Nachricht der Nachricht `MSG_REREAD`. Diese Nachricht hat mit einiger Sicherheit Überlänge. Weiter unten können Sie sehen, wie Sie sie korrekt verschicken: der Dateiname folgt unmittelbar auf die ersten vier Wörter der Nachricht, und alles wird mit einem einzigen `appl_write()` auf die Reise geschickt.

MSG_REREAD (keine weiteren Parameter)
: Wenn `ASCREEN` diese Nachricht empfängt, liest er die gerade im Speicher befindliche `DVI`-Datei nochmals ein. Dies entspricht dem gleichnamigen Menüpunkt.

MSG_NEWPAGE ⟨Seitennummer⟩
: Diese Nachricht bittet `ASCREEN`, die Seite mit der Nummer, die `msg[4]` angibt, zu formatieren und in den Cachespeicher zu tun. Der praktische Nährwert dieser Nachricht außerhalb von `ASCREEN` ist zweifelhaft. `ASCREEN` verwendet die Nachricht jedoch intern, um das Vorausarbeiten zu koordinieren.

MSG_NEWWIN ⟨Art⟩ ⟨Seite⟩ ⟨x⟩ ⟨y⟩ ⟨w⟩ ⟨h⟩
Hiermit können Sie ASCREEN veranlassen, ein neues Fenster zu öffnen. Die ⟨Art⟩ des Fensters ist 1 für ein normales Fenster, 2 für ein um den Faktor 2 verkleinertes Fenster, 4 für ein um den Faktor 3 verkleinertes Fenster, 8 für ein um den Faktor 2 verkleinertes und 16 für ein um den Faktor 3 verkleinertes doppelseitiges Fenster. Das Fenster enthält die Seite mit der Nummer ⟨Seite⟩, und als Größe wünschen Sie sich die Breite ⟨w⟩ und Höhe ⟨h⟩ an der Position (⟨x⟩, ⟨y⟩). Diese Nachricht hat eine Überlänge von 2 Bytes.

Interessant für die äußerliche Anwendung sind vor allem die beiden erstgenannten Nachrichten.

A.3 Das Programm ALaunch

Die ASCREEN-Distribution enthält ein Programm zur Demonstration der Nachrichtenschnittstelle. ALaunch ist für Sie vielleicht interessant, wenn Sie einen GEM-Multitaskingzusatz wie *MultiGEM* benutzen — es scheint, daß es überhaupt nur dann für Sie sinnvoll ist! ALaunch wird mit dem Namen einer DVI-Datei als Parameter aufgerufen. Es prüft, ob gerade ein ASCREEN-Prozeß läuft. Falls nein, wird ein neuer ASCREEN mit der betreffenden Datei gestartet. Falls doch, so erhält der laufende ASCREEN eine MSG_READ-Nachricht mit dem angegebenen Dateinamen. Sie können also ALaunch als Applikation für Dateien vom Typ DVI anmelden (oder, ein hinreichend intelligentes Desktop wie *Gemini* oder das neue von Atari vorausgesetzt, ein ALaunch-Icon auf dem Desktophintergrund ablegen) und durch einfaches Anklicken (oder Abschleppen) von DVI-Dateien ASCREEN starten. ALaunch sorgt dann dafür, daß Ihr Speicher nicht durch mehrfache ASCREENs aufgefressen wird.

!!! Philosophische Personen können sich die Frage stellen, ob sie immer nur einen ASCREEN haben wollen. Wer lieber mehrere DVI-Dateien gleichzeitig auf dem Schirm hat, sollte ALaunch eben nur als Beispiel ansehen und nicht wirklich installieren.

Der Quellcode für ALaunch steht im entsprechenden Verzeichnis der ASCREEN-Distribution zur Inspektion bereit. Ich möchte hier nicht den ganzen Code wiederholen, sondern beschränke mich auf die wesentlichen Teile.

Die Anwesenheit von ASCREEN können Sie mit der AES-Funktion appl_find() wie im folgenden Ausschnitt überprüfen:

```
if ((id = appl_find("ASCREEN ")) < 0)
    /* Fehler */
else if (id <= app_count) /* geladenen ASCREEN weiterbenutzen */
    send_message(app_id, id, argv[1]);
else /* neuen ASCREEN starten */
    launch_ascreen(argv[1]);
```

app_count ist dabei eine Abkürzung für das zweite Wort des global[]-Felds von GEM, das die Maximalzahl der gleichzeitig möglichen Prozesse angibt. Bei einem GEM ohne Multi-

taskingerweiterung ist dies immer 1 (`ALaunch` tut dann nichts Sinnvolles); bei *MultiGEM* kann der Wert eingestellt werden. Beachten Sie auch, daß das Argument von `appl_find()` genau acht Zeichen lang sein muß.

Und `MSG_READ` verschicken Sie an `ASCREEN` ungefähr so:

```
void
send_message (const int from, const int to, const char *file)
{
    int  size;
    char msg[8 + MAX_CMDLINE]; /* MAX_CMDLINE ist z. B. 128 */
#define PUT_WORD(a, i, v) a[i] = (v) >> 8, a[i+1] = (v) & 0xff

    strcpy(msg + 8, file);
    size = 8 + (int)strlen(msg + 8) + 1;
    if (size < 16) size = 16;

    PUT_WORD(msg, 0, MSG_ASCREEN);
    PUT_WORD(msg, 2, from);
    PUT_WORD(msg, 4, size - 16);  /* die "Uberl"ange */
    PUT_WORD(msg, 6, MSG_READ);
    appl_write(to, size, msg);
}
```

`ALaunch` ist an einer Stelle von *MultiGEM* abhängig: es verwendet den Aufruf `Mfork()`, um einen neuen `ASCREEN` als unabhängigen Prozeß zu starten. Leider weiß ich nicht, inwieweit `Mfork()` etwa von *MultiTOS* oder *Mag!X* unterstützt wird und ob `ALaunch` daher in diese Umgebungen portabel ist. Erfahrungsberichte würden mich natürlich brennend interessieren. Damit Sie `Mfork()` mit Turbo- oder Pure-C und seiner Nichtstandard-Parameterübergabe in Registern benutzen können, brauchen Sie ein kleines Binding in Assembler, siehe `MFORK.S`.

A.4 Die Zukunft

Stefan Lindner, Lutz Birkhahn und Robert Kießling bereiten ein TEX-System vor, das auf der Steuersprache `tcl` von John Ousterhout beruhen wird. Dies wird vermutlich diverse Interaktionen zwischen TEX, TEXshell und Programmen wie Previewer und Druckertreiber zulassen. Wenn darüber mehr bekannt ist, wird auch `ASCREEN` natürlich entsprechend geändert werden. Wir sind alle sehr gespannt.

Die Botschaft hör' ich wohl,
allein mir fehlt der Glaube;
Das Wunder ist des Glaubens liebstes Kind.
— JOHANN WOLFGANG VON GOETHE, *Faust* (1808)

Anhang B

ASCREEN und Graphik

B.1 Überblick

Viele Dokumente gewinnen durch Hinzufügen von Graphik. Dazu gehören nicht nur Schemazeichnungen mathematischer oder physikalischer Sachverhalte, sondern auch zum Beispiel chemische Strukturformeln oder Fotos, die mit einem Scanner eingelesen wurden.

Das Einbinden von Graphiken in Texte ist ein traditioneller Schwachpunkt des sonst so leistungsfähigen TEX-Systems. In zehn Jahren ist es der Benutzergemeinde nicht gelungen, ein halbwegs brauchbares Verfahren zum portablen Verwenden von Graphik zu finden, geschweige denn zu standardisieren und überall verfügbar zu machen. Die einzige Methode, die wirklich sowohl weit verbreitet als auch für naive Benutzer einigermaßen zugänglich ist, ist die `picture`-Graphik von LATEX, deren Verwendung aber einen gewissen Masochismus voraussetzt. Inhaber moderner WYSIWYG-Textsysteme wälzen sich regelmäßig vor Lachen auf dem Boden, wenn der gestreßte LATEX-Bildchenmaler wieder zu Lineal und Millimeterpapier greift. Verständlicherweise haben sich daher verschiedene Schulen alternativer Graphikeinbindung gebildet: wohlhabende Personen benutzen PostScript mit einem entsprechenden Drucker, die Treiber für billigere Drucker unterstützen oft besondere (untereinander inkompatible) Graphikbefehle, und wieder andere Benutzer wandeln ihre Graphiken mit speziellen Programmen in TEX-Zeichensätze um, wozu wenigstens keine geräteabhängigen Treiber nötig sind (man braucht aber Treiber, die robust genug sind, um eventuell ziemlich große Zeichen zu verkraften). Der Turmbau zu Babel dürfte gegenüber dieser Vielfalt von „Sprachen“ geradezu ein Muster an Koordination und Verständigung gewesen sein.

Abhilfe ist leider zunächst nicht abzusehen. Der beflissene Implementierer eines Druckertreibers oder Previewers hat darum die Qual der Wahl (bzw. Wahl der Qual, wenn es ans Implementieren geht).

Auf dem Atari ST ist das Standardformat für Rasterbilder das IMG- oder *Image*-Format, das einfarbige oder bunte Bilder beliebiger Größe und Auflösung komprimiert speichern kann. Das IMG-Format hat seine Mängel, vor allem wohl die Tatsache, daß Farbinforma-

tionen nicht gespeichert werden (für einen Hack, der dies beheben kann, siehe Geiß[2], *Vom Anfänger zum GEM-Profi*, Heidelberg: Hüthig 1990). Nichtsdestotrotz sollte jedes Graphikprogramm, das sein Geld wert ist, das IMG-Format bearbeiten können.

Aha Nicht daß das immer so wäre. Ein bekannter Shareware-Iconeditor etwa gibt seinen IMG-Dateien offenbar völlig unmotiviert die Versionsnummer 2 statt 1, was konservative und vorsichtige Programme wie `ASCREEN` an den eigenen Fähigkeiten zweifeln läßt — unnötigerweise, wie sich zeigt. Bei den IMG-Dateien anderer Programme stimmt die Versionsnummer, aber dafür sind zum Beispiel die Abmessungen der Graphik falsch usw. usf. Seufz.

Sowohl die Treiber der TEX-Implementierung von Stefan Lindner als auch die der Implementierung von Christoph Strunk können IMG-Dateien einlesen, skalieren und ausgeben. Die Syntax hierfür ist selbstverständlich bei beiden Familien verschieden, wobei Christoph Strunk neuerdings auch die Lindner-Befehle unterstützt. `ASCREEN` versteht auch beides.

Die Strunkschen Treiber kennen außerdem noch einen Satz von `\special`-Befehlen für verschiedene Graphikprimitive. Dazu gehören Linien mit unterschiedlichen Mustern, Kreise und Ellipsen und heuer sogar Splines und andere nicht ganz simple Sachen. `ASCREEN` unterstützt die allermeisten Befehle dieser "CSG-Level-2-Graphik".

Was die Graphik durch `\special`-Befehle angeht, so ist der am besten etablierte Standard wohl der, den das UNIX-Programm *tpic* gesetzt hat. Viele Treiber auf unterschiedlichen Rechnern und Betriebssystemen verstehen heute diese *tpic*-`\special`s, nicht zuletzt auf dem Atari der PostScript-Treiber *dvips* und (wieder mal) die Treiber von Christoph Strunk. Und `ASCREEN`.

B.2 IMG-Graphik im Detail

Nach der etwas oberflächlichen Erklärung der IMG-Graphikeinbindung im vorigen Abschnitt hier eine etwas genauere Diskussion. Aber zunächst noch eine kritische Vorbemerkung:

IMG-Einbindung ist großer Mist. Daß so etwas nur auf Ataris funktioniert, ist notfalls zu verschmerzen — Nicht alle haben ja (etwa in der Uni) Zugriff auf eine UNIX-Workstation nebst schnellem Laserdrucker, um die daheim formatierten Dokumente zu drucken. Tatsache ist allerdings, daß die Auflösung bezahlbarer Drucker für eine saubere Halbtondarstellung effektiv nicht ausreicht, insbesondere wenn man an dem Bild noch herumzerren und -stauchen muß, um es ins richtige Format zu bringen: Unschöne Schmiereffekte oder Verzerrungen sind meist die Folge. Ferner kommen die dekorativen Graustufen-Füllmuster, die auf dem Bildschirm so nett aussehen, auf den meisten Druckern solide schwarz heraus (zumindest wenn diese Drucker das Bild mit Nadeln aufs Papier pieksen, soll's ja geben). Halbtongraphiken sind also meist Keine Gute Idee. Und für Strichzeichnungen sollte man, wenn irgend möglich, auf einen der `\special`-Sätze oder gar auf LATEX-`picture`s zurückgreifen, die wenigstens vektororientiert und daher auflösungsunabhängig sind. Es gibt heute schon viele Programme, für den Atari etwa *TEXdraw* oder *ZPCAD*, die das interaktive „Malen" von solchen Graphiken auch ohne Vorzeichnen auf Millimeterpapier unterstützen.

Wer nun partout die IMG-Einbindung verwenden möchte, hat bei ASCREEN die Auswahl zwischen dem Lindner-Stil und dem Strunk-Stil bei den \special-Befehlen:

L: \special{graphic␣img␣⟨Dateiname⟩}
S: \special{CS!g␣f␣⟨Dateiname⟩}

Beide Befehle bewirken im wesentlichen dasselbe: Die Datei ⟨Dateiname⟩ wird gelesen und auf der entsprechenden Seite dargestellt, wobei die linke untere Ecke der Graphik an der gerade aktuellen Position auf der Seite zu stehen kommt. (Der reelle Faktor f bei Christoph Strunk dient zur Skalierung der Graphik.) Die Position auf der Seite wird durch die Graphikeinbindung nicht verändert, so daß die Graphik aus der Sicht von TEX keinen Platz einnimmt. Dieses Problem lösen sowohl Stefan und Lutz als auch Christoph ziemlich auf dieselbe Weise; namentlich muß die TEX-Datei wissen, wie groß die Graphik sein wird, und entsprechend Platz reservieren. Die Anleitungen zu den jeweiligen TEX-Systemen enthalten hierzu und darüber, wie man diese Größen herausfinden oder gar ändern kann, ausführliche Informationen, für die hier leider kein Platz ist. Oft gibt es auch vorgefertigte Makros und Hilfsprogramme, die die Hauptarbeit des Einbindens übernehmen.

Ein Problem beim Einbinden von IMG-Dateien sind die Pfadnamen. Insbesondere die TEX-Implementierung von Christoph Strunk legt die Verwendung absoluter Pfadnamen für die IMG-Dateien nahe. Dies ist aber nicht empfehlenswert, denn dadurch werden die DVI-Dateien systemabhängig — sie enthalten ja den Namen des IMG-Verzeichnisses auf dem System, wo der TEX-Lauf stattgefunden hat. Solche DVI-Dateien kann man nicht mehr ohne weiteres weitergeben, nicht mal, wenn man auf dem Atari bleibt und alle IMG-Dateien dazutut. Viel besser ist es, bei der Graphikeinbindung nur den eigentlichen Namen der IMG-Datei ohne den Pfadanteil anzugeben und den Treiber den Suchpfad beisteuern zu lassen. Sowohl die Treiber von Christoph Strunk als auch die von Stefan Lindner unterstützen eine Option, mit der man ein Verzeichnis für die IMG-Dateien festlegen kann. Auch ASCREEN erlaubt im Dialog „Pfade“, einen „IMG-Pfad“ einzugeben. Dies kann sogar eine durch Kommata getrennte Folge mehrerer Verzeichnisnamen sein, die dann von links nach rechts durchsucht wird.

Aha Der „populärste“ Fehler in der vorigen ASCREEN-Version bestand darin, daß Schrägstriche in absoluten Pfadnamen bei IMG-Dateien nicht in die Atari-üblichen Rückwärtsschrägstriche umgesetzt wurden. Zumindest gingen darüber die meisten Beschwerden ein — sicher zu Recht. Inzwischen wird diese Transformation in allen Dateinamen vorgenommen.

B.3 CSG-Graphik

Von Christoph Strunk stammt die Definition einer Reihe von Graphikprimitiven, die über den \special-Befehl implementiert werden. Der TEX-Befehl \special macht es möglich, Zeichenketten ohne Bearbeitung durch TEX in die DVI-Datei zu schreiben, wo der Gerätetreiber sie vorfinden und auswerten kann. Dies ist zwar eine praktische Möglichkeit, um Sachen zu tun, die in TEX selbst schwierig sind (etwa Graphik...), aber \special-Befehle

sind mit Vorsicht zu genießen: wenn Sie einen solchen Befehl geben, verlassen Sie sich darauf, daß der Gerätetreiber ihn auch verstehen kann. Und das ist beileibe nicht garantiert.

Von Christoph Strunks „CSG-Graphik-`\specials`“ gibt es zwei Geschmacksrichtungen: die etwas ältere Stufe 1 und die aktuelle Stufe 2. Die Stufe 2 ist zum größten Teil eine Obermenge von Stufe 1. `ASCREEN` unterstützt die allermeisten Befehle der Stufe 2, lediglich zwei Kommandos wurden nicht implementiert (siehe Tabelle). Hier ist eine Liste der CSG-Befehle der Stufe 2. Für eine genaue Diskussion verweise ich aus Platzgründen auf die Dokumentation zu Christoph Strunks TEX-Implementierung. Alle Befehle beginnen mit der Zeichenfolge „`CS!`“. Der Punkt (x, y) ist die aktuelle Position auf der Druckseite, alle Maße sind in Einheiten der mit `CS!u` festgelegten „Einheitslänge“. Die Koordinatenachsen sind wie in der Mathematik üblich angeordnet.

CS!a $dx\ dy\ \phi\ \alpha\ \beta$ Ellipsenbogen um (x, y) mit den Halbachsen dx und dy, gedreht um den Winkel ϕ zur x-Achse, beginnend bei α Grad (relativ zum Ende der Halbachse dx) und endend bei β Grad.

CS!b2 $dx_0\ dy_0\ dx_1\ dy_1$ Quadratische Bézierkurve mit den Kontrollpunkten (x, y), $(x + dx_0, y + dy_0)$ und $(x + dx_1, y + dy_1)$.

CS!b3 $dx_0\ dy_0\ dx_1\ dy_1\ dx_2\ dy_2$ Kubische Bézierkurve mit den Kontrollpunkten (x, y), $(x + dx_0, y + dy_0)$, $(x + dx_1, y + dy_1)$ und $(x + dx_2, y + dy_2)$.

CS!c r Kreis mit dem Radius r um den Punkt (x, y).

CS!c0 d Kreis mit dem Durchmesser d um den Punkt (x, y).

CS!d r Ausgefüllter Kreis mit dem Radius r um den Punkt (x, y).

CS!d0 d Ausgefüllter Kreis mit dem Durchmesser d um den Punkt (x, y).

CS!e $dx\ dy$ Ausgefüllte Ellipse um (x, y) mit den Halbachsen dx und dy.

CS!g $f\ n$ Bindet die IMG-Graphik n um den (reellen) Faktor f vergrößert ein. Die linke untere Ecke der Graphik liegt bei (x, y).

CS!G $f\ n$ Entspricht in `ASCREEN` dem vorigen Befehl. In Christoph Strunks Treibern wird für Rückwärtskompatibilität ein anderer Algorithmus verwendet.

CS!i n Liest die Datei mit dem Namen n ein. n muß weitere CSG-Befehle enthalten; das genaue Format kann hier nicht erläutert werden (siehe Dokumentation zum StrunkTEX). In der CSG-Stufe 1 sind in einer solchen Datei weitere `i`-Befehle zulässig. Diese Möglichkeit wurde in der Stufe 2 entfernt; `ASCREEN` unterstützt dies jedoch weiter.

CS!l $dx\ dy$ Zieht eine Linie von (x, y) nach $(x + dx, y + dy)$. Dabei wird der mit `CS!t` eingestellte Linientyp verwendet.

CS!m n Bindet das GEM-Metafile n ein. Dieses Kommando wird von `ASCREEN` *nicht* unterstützt, da es sich (ohne ein hypermodernes GDOS) nicht sauber realisieren läßt.

CS!r Setzt die Einheitslänge auf `1pt` und den Linientyp auf „durchgezogene Linie“ zurück.

CS!t *t* Wahl eines Linientyps. Die Typen sind: 0 – durchgezogene Linie, 1 – lang gestrichelt, 2 – gepunktet, 3 – Strich-Punkt, 4 – kurz gestrichelt, 5 – Strich-Punkt-Punkt, 6 – weit gepunktet

CS!u *l* Setzt die Einheitslänge auf *l*. *l* kann dabei eine beliebige TEX-Dimension sein.

CS!w *w* Setzt die Linienbreite auf *w*. Dieser Befehl wird einstweilen ignoriert.

Ursprünglich sollten die CSG-Befehle eine effiziente Implementierung der auch in LATEX-`picture`-Umgebungen erlaubten Elemente ermöglichen. Wie man sieht, sind sie über dieses Ziel inzwischen hinausgeschossen. Programme wie *TEXdraw* oder *ZPCAD* können CSG-Befehle erzeugen.

B.4 tpic-Befehle

Konzeptuell den CSG-Befehlen ähnlich sind die *tpic-Befehle*. Auch sie definieren `\special`-Kommandos für Graphikoperationen. Dieser Satz von Graphikbefehlen geht auf das Programm *tpic* zurück, eine Anpassung des bekannten Unix-Programms *pic* an TEX. Für das Programm *tpic* selbst braucht man eine Unix-Quellcodelizenz von AT&T, die wohl die wenigsten Ataribenutzer haben dürften; die *tpic*-`\special`s sind aber inzwischen sehr weit verbreitet und werden von bekannten Programmen wie *Gnuplot*, *Xfig* (unter Unix) und *TEXdraw* unterstützt. Die Stiloption `eepic.sty` für LATEX benutzt sie, um in `picture`-Umgebungen Linien beliebiger Steigung zuzulassen. Auch auf der Treiberseite sind *tpic*-`\special`s weithin bekannt: außer von Christoph Strunks neuen Treibern werden sie von Tom Rokickis *DVIPS* (der auch auf dem Atari ST läuft) oder von *xdvi* (unter Unix) verstanden. Im übrigen enthält das *groff*-Paket, der *troff*-Clone der *Free Software Foundation*, ein zu *pic* und *tpic* kompatibles Programm, *gpic*. Auch von *groff* gibt es inzwischen eine Anpassung für den Atari ST.

Hier wieder eine Tabelle mit den *tpic*-`special`s. Alle Maße sind hier zur Abwechslung in Milli-Zoll. Die Koordinatenangaben beziehen sich auf die obere linke Ecke eines gedachten Rechtecks, das die ganze Graphik umschließt. Die y-Achse verläuft also verkehrtherum. Winkel werden im Gegenuhrzeigersinn gemessen (also von $+x$ nach $+y$) und werden immer im Bogenmaß angegeben.

pn *s* Setzt die Stiftbreite auf *s* Millizoll. Dieser Befehl wird einstweilen ignoriert.

pa *x y* Fügt den Punkt (x, y) einem intern gespeicherten „Pfad“ hinzu.

fp Gibt den vorher definierten Pfad aus, d. h. die Punkte auf dem Pfad werden mit einer Linie der aktuellen Stiftbreite verbunden. Danach werden die Punkte gelöscht. Wenn Ausmalen gewünscht wurde und der Pfad geschlossen ist, dann sollte er ausgemalt werden.

ip Entspricht `fp`, aber der Pfad selbst wird nicht gezeichnet. Ausmalen findet statt, falls gewünscht.

da f Entspricht `fp`, aber der Pfad wird als gestrichelte Linie gezeichnet. Ein Strich ist f Zoll lang (f ist reell).

dt f Entspricht `da`, aber der Pfad wird als gepunktete Linie gezeichnet. Die Strecke zwischen den Punkten ist jeweils f Zoll.

sp d Entspricht `fp`, aber zwischen den Punkten wird eine Splinekurve gezeichnet. Diese Kurve geht nur durch den ersten und den letzten Punkt, aber ihr Aussehen hängt von den relativen Positionen der anderen Punkte ab. Das reelle Argument d ist fakultativ; falls $d = 0$ oder gar nicht da ist, wird die Kurve durchgezogen; falls $d > 0$, wird sie wie bei `da` d gestrichelt und, falls $d < 0$, wie bei `dt` $-d$ gepunktet.

ar $x\ y\ dx\ dy\ \alpha\ \beta$ Zeichnet einen Bogen mit dem Mittelpunkt bei (x, y), beginnend beim Winkel α und endend bei β. Handelt es sich um einen Vollkreis oder eine Ellipse, dann bestimmen dx und dy die Halbachsen in x- bzw. y-Richtung. Ansonsten ist $dx = dy$, und es wird ein Kreisbogen gezeichnet.

ia $x\ y\ dx\ dy\ \alpha\ \beta$ Dies verhält sich zu `ar` wie `ip` zu `fp`.

sh s Malt das nächste aus drei oder mehr `pa`- und einem `fp`- oder `ip`-Kommando oder aus einem `ar`- oder `ia`-Kommando bestehende Objekt aus. Der Parameter s ist eine Gleitkommazahl zwischen 0 und 1; 0 steht für weiß (inklusive Übermalen des Untergrunds) und 1 für schwarz; 0.5 ist grau. Wenn s nicht angegeben wurde, wird 0.5 benutzt. Dieser Befehl ist laut dem *tpic*-Standard optional; `ASCREEN` unterstützt darum nur die Parameter 0 und 1.

wh Entspricht `sh 0`.

bk Entspricht `sh 1`.

tx Veraltetes und nicht unterstütztes Kommando.

The nice thing about standards is
that you have so many to choose from.
Furthermore, if you do not like any of them,
you can just wait for next year's model.

— ANDREW S. TANENBAUM, *Computer Networks* (1989)

Anhang C

Das FLIB-Format

`ASCREEN` kann Zeichensatzbibliotheken im FLIB-Format von Eberhard Mattes lesen. Leider gibt es für den Atari ST im Moment noch kein komfortables Programm zum Erstellen und Verwalten solcher Bibliotheken. Darum soll hier zumindest das Format dokumentiert werden, vielleicht schreibt ja mal jemand ein schönes Programm dafür.

C.1 Der Aufbau einer FLIB-Datei

FLIB-Dateien bestehen aus einem Kopf, gefolgt von einem Kommentar und einem Inhaltsverzeichnis. Das Inhaltsverzeichnis enthält Informationen über die verschiedenen Vergrößerungsstufen von Zeichensätzen in der Bibliothek und pro Vergrößerungsstufe dann über die Zeichensätze selber. Nach dem Inhaltsverzeichnis kommen die eigentlichen Daten:

⟨Kopf⟩
⟨Kommentar⟩
⟨Größen⟩ ...
  ⟨Zeichensätze⟩ ...
⟨Daten⟩ ...

Hier ist das Format des Dateikopfes. Die Zahlen links sind der Abstand vom Dateianfang in Bytes, die Indizes geben die Länge des jeweiligen Wertes in Bytes an.

0	`FLIB`	magisches Langwort
4	2 0	Versionsnummer (2 Bytes)
6	l_2	Länge des Verzeichnis in Bytes
8	s_2	Anzahl der Größen
10	f_2	Anzahl der Zeichensätze
12	c_2	Länge des Kommentars (0: kein Kommentar)

Freundlicherweise sind die Datenwörter alle im *big-endian* oder auch Motorola-Format: das höherwertige Byte steht an erster Stelle. Hier der Aufbau eines Größeneintrags mit den entsprechenden Offsets:

0	m_2	Länge des Größeneintrags (ohne Kopf)
2	t_2	Anzahl der Zeichensätze in dieser Größe
4	dpi_4	Auflösung (Festpunkt, 16.16 Bits)
8		Zeichensätze

Eberhard Mattes besteht tatsächlich auf einer Angabe der Auflösung bis hin zu 1/65536 dpi! Hierzu gleich mehr. Der Eintrag für einen Zeichensatz hat das Format

0	fl_4	Länge des Zeichensatzes
4	a_4	relative Byteadresse des Zeichensatzes
8	n_1	Länge des Zeichensatznamens
9+	Name	

C.2 FLIB-Tips

Ein Problem mit den `FLIB`-Dateien ist die eben schon angesprochene Tatsache, daß die Größenklassen von Zeichensätzen in der Datei mit einer Genauigkeit von 1/65536 dpi angegeben werden muß. Im wirklichen Leben ist es kaum möglich, im Treiber so genau zu rechnen. Ein Zeichensatz mit 101 dpi bei der Vergrößerung `magstephalf`, also 1095/1000, hätte zum Beispiel eine Auflösung von 110.595 dpi, also 110 38993.92/65536 dpi. Nicht ganz exakt darstellbar, wie Sie sehen. `ASCREEN` hilft sich in dieser Lage, indem die „Gleichheit“ zweier Vergrößerungsstufen nicht ganz so pingelig betrachtet wird. Beim Vergleich zweier Auflösungen in einer `FLIB`-Datei toleriert `ASCREEN` Abweichungen von bis zu 32 Einheiten oder 1/2048 dpi, was das Problem vermeidet, aber nicht zu sichtbaren Abweichungen führen sollte.

OK? C'est facile quand vous avez le hang.
— MILES KINGTON, *Let's Parler Franglais One More Temps* (1982)

Anhang D

HyperTEX

D.1 Was ist HyperTEX?

Seit einiger Zeit gibt es eine neue Art, Dokumente für den Gebrauch mit Computern zu strukturieren: *Hypertext* nennt man einen Text, bei dem gewisse Stichwörter mit anderen Stellen des Textes „verbunden“ sind, so daß man bequem an diese Stellen gelangen kann. Referenzen wie „siehe oben“ und „vergleiche Tabelle 5“ lassen sich so leicht verfolgen. Hypertext macht es aber auch möglich, das Dokument in einer fast beliebigen Reihenfolge durchzugehen — Bücher dagegen unterstützen im wesentlichen nur das Lesen von vorne nach hinten, alles andere ist mühselig. Ein anderes oft gehörtes Stichwort ist *Hypermedia*; das Dokument enthält dann auch Bilder, Töne und andere „Effekte“, die sich leicht abrufen lassen.

Wäre es nicht schön, Hypertext mit TEX zu erstellen? Onlinedokumentation zum Beispiel wäre damit viel bequemer zu benutzen und würde außerdem nett aussehen. Tatsächlich gibt es verschiedene Ansätze für Hypertext mit TEX. Einer davon ist HyperTEX, das von `ASCREEN` unterstützt wird. HyperTEX besteht aus einer Stiloption für LATEX und einem Perl-Programm, das gewisse Hilfsarbeiten erledigt. Die prinzipielle Vorgehensweise ist wie folgt:

1. Sie schreiben Ihren Text wie üblich in LATEX und zeichnen die Querverbindungen mit besonderen Kommandos aus.
2. Beim LATEX-Lauf wird außer der `aux`-Datei auch eine `hyp`-Datei angelegt, die Informationen über die Querverbindungen enthält.
3. Die `hyp`-Datei wird mit dem Perl-Programm `hyp.pl` zu einer `hl`-Datei gemacht. Diese enthält die Querverbindungen in sortierter und für `ASCREEN` verdaulicher Form.
4. Sie schauen sich die `DVI`-Datei mit `ASCREEN` an. Sie können sie auch ausdrucken; gute Treiber müßten die Querverbindungen ignorieren.

HyperTEX unterscheidet zwischen der *Definition* eines Begriffs und einem *Verweis* auf den Begriff. Angenommen, Sie schreiben ein Dokument über elementare Geometrie. Die Definition des „Satzes von Pythagoras“ wäre dann die Stelle, wo der Satz eingeführt wird. An verschiedenen Stellen im Dokument, wo der Satz benutzt wird, können Sie dann auf die Definition verweisen. So ist es leicht, den Satz bei Bedarf „nachzuschlagen“.

Begriffe, für die Querverbindungen existieren, werden von `ASCREEN` auf dem Bildschirm eingerahmt. Drücken Sie die linke Maustaste, während der Zeiger sich innerhalb eines solchen Rahmens befindet, so erhalten Sie nicht das übliche Popupmenü, sondern ein weiteres mit den folgenden Einträgen:

Definition Wenn Sie diesen Menüpunkt wählen, bringt `ASCREEN` Sie in die Nähe der Definition des Begriffs. `ASCREEN` versucht, die Definition so gut wie möglich in die Mitte des Fensters zu bringen.

Nächster Verweis `ASCREEN` springt zum nächsten Verweis auf den Begriff, d. h. auf den in Richtung auf das Dokumentenende nächstliegenden.

Voriger Verweis `ASCREEN` springt zum vorigen Verweis auf den Begriff, d. h. auf den in Richtung Dokumentanfang nächstliegenden.

Zurück `ASCREEN` springt an die Stelle zurück, von der aus zuletzt eine Definition oder ein Verweis aufgesucht wurde. Mit dieser Funktion kann man eine Kette von Sprüngen „zurückspulen“, um an den Ausgangspunkt zurückzukehren und in der Lektüre des Dokuments fortzufahren. `ASCREEN` merkt sich die Sprünge für jedes Fenster einzeln.

!!! HyperTEX ist im Moment als „experimentell“ anzusehen. Insbesondere das Perl-Programm soll auf lange Sicht hin verschwinden und durch zusätzliche Intelligenz im Treiber ersetzt werden.

D.2 Die LaTeX-Seite von HyperTEX

Um HyperTEX in einem LaTeX-Dokument benutzen zu können, müssen Sie die Stiloption `hypertex` einbinden, indem Sie sie im optionalen Teil des `\documentstyle`-Befehls aufführen. Dadurch werden die folgenden Befehle definiert:

`\hyperdef{`⟨**Text**⟩`}` „Definiert“ den ⟨Text⟩.

`\hyperto{`⟨**Text**⟩`}` Richtet eine Verbindung auf ⟨Text⟩ ein.

Die Datei `HYPERTEX/HYPTEST.TEX` in der `ASCREEN`-Distribution enthält ein Beispieldokument.

D.3 Implementierung

Die `\hyper`...-Makros schreiben einen `\special`-Befehl in die `DVI`-Datei, der die Abmessungen des ⟨Text⟩es enthält. Diese Informationen werden von `ASCREEN` verwendet, um beim Anzeigen einen Rahmen um den ⟨Text⟩ zu zeichnen. Der ⟨Text⟩ kann keinen Zeilenumbruch enthalten.

Aha Eigentlich kann der ⟨Text⟩ aus einigermaßen beliebigem LaTeX-Code bestehen. Unglücklicherweise wird er aber in die `hyp`-Datei geschrieben und sollte daher nicht allzu lang sein.

Die `hyp`-Datei enthält nicht nur den Text des Querverweises, sondern auch die Seitennummer, auf der der Verweis aufgetreten ist. Das Perl-Programm `hyp.pl` sortiert alle Einträge in der `hyp`-Datei nach dem Text und sammelt die Seitennummern, um `ASCREEN` die Suche nach Verweisen und Definition zu erleichtern. `ASCREEN` selbst liest zu einer `DVI`-Datei auch die `hyp`-Datei, sofern vorhanden, und bietet die entsprechenden Querverweise zum Anklicken an. Die Rahmen um die Verweise werden von `ASCREEN` erzeugt, was keinen besonders großen Aufwand darstellt (Linien muß er sowieso zeichnen können). Man könnte sie auch von LaTeX erzeugen lassen, aber dann würden sie auch beim Ausdrucken erscheinen. So kann man ein HyperTeX-Dokument immerhin ausdrucken, ohne daß die dann nutzlosen Rahmen es entstellen; allerdings dürfte der Druckertreiber die HyperTeX-`\special`-Befehle als unbekannt monieren.

D.4 Die Zukunft

HyperTeX ist, wie gesagt, experimentell und wird sich sicherlich noch ändern, wenn ich die Zeit finde, das ganze umzuschmeißen und neu zu machen :-) Insbesondere denke ich daran, das Perl-Programm überflüssig zu machen. Ob HyperTeX jemals Verbreitung für Onlinedokumentation finden wird, halte ich für zweifelhaft, denn schließlich ist das zum Anzeigen erforderliche TeX-Paket nicht überall vorhanden, und es wäre wohl etwas mühsam, TeX zu installieren, nur um die Hilfe zu einem ganz anderen Programm lesen zu können. Allerdings habe ich die Idee, ein erweitertes `DVI`-Format zu implementieren, in dem die benötigten Zeichensätze für den Bildschirm gut versteckt enthalten sind. Eine solche „`DVJ`-Datei“ könnte durch einen Postprozessor aus einer `DVI`-Datei erzeugt und mit einer „Lightversion“ von `ASCREEN`, die nur die nötigsten Features enthält, angezeigt werden — auch wenn kein komplettes TeX installiert ist! Zeit müßte man haben! Ich wäre an Kommentaren interessiert, die mir einzuschätzen helfen, ob das eine Gute Idee ist und ob Leute ein solches Paket gerne benutzen würden. Ich möchte auch ganz generell um Meinungen zu HyperTeX bitten.

O, what a tangled web we weave
When first we practise to deceive!
— SIR WALTER SCOTT, *Marmion* 6:17 (1808)

Anhang E

FontList

E.1 Was ist FontList?

Das ASCREEN-Paket enthält auch noch eine weitere Dreingabe: Das Programm FontList gibt eine Liste der in einem Dokument benutzten Zeichensätze aus. Neben dem „Interesse-Wert“ einer solchen Liste hat das auch praktischen Nutzen: wenn Sie ein Dokument weitergeben wollen, können Sie mit der Liste sichergehen, daß keine exotischen Zeichensätze darin vorkommen, oder Sie können anhand der Liste die Zeichensätze zum Dokument mitliefern. Wie die Liste genau aussehen soll, können Sie selbst festlegen — FontList kennt von Haus aus den Lindner- und den Strunk-Stil für Zeichensatzlisten, den das respektive METAFONT verstehen kann (naja, die HKM-TEXshell bei Stefan Lindner, und das METAFONT ist ja auch von Lutz Birkhahn, und überhaupt...), aber Sie können auch über ein Format à la printf() eine nahezu beliebige Form der Ausgabe bestimmen.

E.2 FontList benutzen

FontList ist ein TOS-Programm, das Sie am besten von einer kommandoorientierten Shell wie der *Mupfel* aus aufrufen. Die grundlegende Syntax ist

fontlist [⟨Optionen⟩] ⟨DVI-Datei⟩

wobei die Liste auf der Standardausgabe landet. Diese kann wie üblich in eine Datei umgelenkt werden.

Die folgenden Optionen sind definiert:

-l Liste im Lindner-Stil (res101.scr cmr10 1.200)

-s Liste im Strunk-Stil (cmr10␣in␣magnification␣1.200)

-f ⟨Format⟩ Formatangabe für die Liste (siehe unten)

-e ⟨Endung⟩ Geräteendung für den Lindner-Stil (o. ä.)

-r ⟨Auflösung⟩ Auflösung für den Lindner-Stil (o. ä.)

-m ⟨Zahl⟩ Vergrößerung des ganzen Dokuments (in Promille)

-o ⟨Datei⟩ Name einer Ausgabedatei (anstatt *stdout*)

Im Format, das mit `-f` angegeben wird, können beliebige ASCII-Zeichen vorkommen. Einige Sequenzen haben jedoch eine besondere Bedeutung:

`%F`	Zeichensatzname
`%R`	Auflösung (siehe `-r`-Option)
`%M`	Vergrößerung (als Fließkommazahl)
`%m`	Vergrößerung (als Ganzzahl)
`%E`	Geräteendung (siehe `-e`-Option)

Das Format für den Lindner-Stil wäre beispielsweise

```
res\%R.\%E␣%M␣%F
```

das für den Strunk-Stil

```
%F␣in␣magnification␣%M
```

Geben Sie kein Format an, so ist die Vorgabe ein schlichtes

```
%F␣%M
```

Mehr muß über das Programm `FontList` an sich hier wohl nicht gesagt werden. Der Quellcode ist mitgeliefert, und `FontList` sollte auch auf anderen Computern ohne große Probleme laufen, wenn es dort einen einigermaßen zum ISO-Standard konformen C-Compiler gibt (ich habe es auf einem IBM RISC System/6000 getestet). Falls jemand weitere schöne Eigenschaften in das Programm einbaut, wäre ich selbstverständlich sehr an der neuen Version interessiert.

It is clear that Lists are very general structures;
indeed, it seems fair to state
that any structure whatsoever can be represented as a List
when appropriate conventions are made.
— DONALD E. KNUTH, *Fundamental Algorithms* (1973)

Anhang F

Zum Schluß

F.1 Geschichtliches

Jetzt kommen die Leute zu ihrem Recht, die immer alles genau wissen wollen: wo kommt `ASCREEN` her, und warum ist das Programm so, wie es ist? Wir müssen uns zurückbegeben in die finstere Vergangenheit:

Wir schreiben das Jahr 1988. Anselm leiht das *TEXbook* aus der Unibibliothek aus und ist alsbald rettungslos verloren. Nach einigen Experimenten mit einem selbstgebastelten Textformatierungsprogramm, das immerhin für Praktikumsmitschriften im Werte von einem Semester ausreicht (wenn auch gelegentlich mitternächtliche Debuggingsitzungen vor einem Abgabetermin nötig waren), kratzt er sein Erspartes zusammen, holt sich eine Festplatte für seinen Atari und bestellt sein erstes TEX, seinerzeit erhältlich bei der Firma TooLs in Bonn. Der mitgelieferte kombinierte Drucker- und Bildschirmtreiber ist allerdings nicht gerade das Gelbe vom Ei.

Später in demselben Jahr macht Anselm sich auf, um zwei Semester im Ausland zu studieren. Weit und breit ist kein Atari in Sicht, aber der Gedanke an den primitiven Druckertreiber daheim läßt A. keine Ruhe, insbesondere weil die andere Uni Sun-Rechner hat, wo es schon bequemere Previewer und Druckertreiber gibt. Eigentlich möchte er ja einen neuen, schnellen Treiber für seinen treuen NEC P6 haben. Zum Glück hat die andere Uni auch das Buch *TEX: The Program*, wo das `DVI`-Format dokumentiert ist, und A. gelingt es, den TUGboat-Artikel mit den Spezifikationen für `PK`-Zeichensätze aufzutreiben. In einigen durchgehackten Nächten (A. hat einen Schlüssel fürs Institut) entsteht auf einer Sun-3 der allererste Proto-`ASCREEN`, der allerdings mehr an eine Mischung aus *PKType* und *DVIType* erinnert, denn Anselm weiß nichts über SunView-Graphik. Wiederum später bekommt A. Gelegenheit, eine Atari-Transputer-Workstation zu benutzen (die alte Version mit dem extra Mega-ST), und er fügt dort die bisher fehlenden Graphikroutinen hinzu, hauptsächlich um mal zu sehen, ob die `DVI`-Bearbeitung einigermaßen stimmt. Die Graphik erscheint selbstverständlich nicht auf dem ST-, sondern auf dem ATW-Monitor, und die Graphikroutinen gehen an die

rohe Hardware. TEX gibt es auf der ATW natürlich keins, so daß ein paar DVI-Dateien und Zeichensätze zum Ausprobieren von einem Unix-Rechner über ein serielles Kabel heruntergeladen werden mußten. Das resultierende Programm ist in Geschwindigkeit und Komfort in etwa mit dem TooLs-Treiber zu vergleichen.

Wieder zuhause macht Anselm sich daran, den Mickymaus-Graphiktreiber für die ATW durch eine vernünftige Version für den Atari selbst zu ersetzen. Naja, was man damals so vernünftig nannte; heutzutage rümpfen wir über Line-A-Routinen und Programme, die vom 640 × 400-Bildschirm ausgehen, zu Recht die Nase. Irgendwann kam dann noch die Idee mit dem Cachespeicher dazu (die Anselm auch heute noch für seine ureigene Erfindung hält), und viele Patches und Versionen später ist ASCREEN 2 wirklich zu einem halbwegs brauchbaren Programm geworden. Zumindest diverse Kommilitonen und andere Leute an der Unität sind dieser Ansicht.

Aber die Zeit bleibt nicht stehen. Der Atari TT kommt auf den Markt, und Anselm bekommt Briefe und Besuch von Leuten, die sich über „unsaubere Programmierung“ und solche Sachen aufregen. Eigentlich wollte A. ja erst mal den P6-Treiber schreiben, aber schließlich sieht er ein, daß doch mal wieder etwas Arbeit in ASCREEN gesteckt werden muß. Mitte 1991 beginnt er damit, ASCREEN ein GEM-Kostüm zu verpassen und dabei diverse Sachen komplett umzustoßen, die er schon lange ändern wollte usw. Wieder wurden einige Leute mit zahllosen zumeist katastrophal fehlerhaften Versionen gequält, bis im Dezember 1991 ASCREEN 3.0 verteilt werden konnte. A. war entschlossen, nach dieser halbwegs stabil laufenden Ausführung erst mal eine Weile zu warten und Fehlerberichte, Vorschläge usw. zu sammeln und nach und nach zu implementieren. HyperTEX zum Beispiel ist Anselms „wir warten aufs Christkind“-Ersatz vom Heiligabend 1991. Entsprechend wurde erst im März 1992 ASCREEN 3.1 fertig, die erste Version, die auch einem breiteren Publikum angeboten wurde. Und Anselm ist zufrieden: die Kritiken sind überwiegend gut. Er hofft, daß ASCREEN 3.2 genausogut aufgenommen wird, denn, wie sein großes Idol Larry Wall sagt, *it pleases the Author of his Story.* Der P6-Treiber ist inzwischen übrigens abgesagt worden.

F.2 Warum ASCREEN kein Sharewareprogramm ist

Warum ist ASCREEN eigentlich kein Sharewareprogramm? Einige Leute haben mich das seit dem Erscheinen der letzten Version gefragt. Nun, TEX ist ja auch kein Sharewareprogramm (von einigen Implementierungen mal abgesehen). Fast alle Software im TEX-Umfeld kostet so gut wie nichts, und für viele andere Programme gilt dasselbe — zum Beispiel für die hervorragenden GNU-Werkzeuge. Ich benutze jede Menge freie Software, und ich glaube daran, daß man nicht nur nehmen darf, sondern, wenn möglich, auch geben muß. Außerdem habe ich die Erfahrung gemacht, daß ohnehin die wenigsten Leute eine Sharewaregebühr bezahlen. ASCREEN kostet also nichts bis auf Medien- und Übertragungskosten: „Live Free Or Die!“

Das heißt natürlich nicht, daß ich keine Spenden und milden Gaben annehmen würde. Wenn ASCREEN Ihnen so gut gefällt, daß Sie mir mal eine gemütliche Pizza gönnen oder meinen

nächsten Urlaub ein bißchen subventionieren wollen: meine Bankverbindung steht weiter unten bei der Kontaktadresse.

F.3 Rückkopplung und neue Versionen

Soweit ich dafür Zeit habe, werde ich `ASCREEN` erweitern und Fehler beheben. Ihre Kommentare, Vorschläge und Kritik sind mir dabei sehr wichtig, und ich möchte Sie darum gerne einladen, mir Ihre Meinung zu sagen. Die besten Chancen auf eine prompte Antwort haben Sie, wenn Sie mir E-Mail schicken; meine Adresse ist

`lingnau@math.uni-frankfurt.de`

Die Adresse für die gelbe Post, die ich natürlich auch beantworte, nur meist nicht so schnell:

Anselm Lingnau
Buchenweg 1
W-6239 Eppstein i. Ts.
Tel. (06198) 8555

Und schließlich, wie angekündigt, noch meine Kontonummer:

Kto.-Nr. 126 187
Genossenschaftsbank Main-Taunus eG., Kriftel
BLZ 500 694 88

Neue Versionen von `ASCREEN` werden in elektronischen Foren wie `TEX-D-L` und `de.comp.tex` angekündigt. Die jeweils neueste Version steht auf dem Rechner

`ftp.math.uni-frankfurt.de` [141.2.90.2]

für anonymes FTP zur Verfügung, dort gibt es auch „Zubehör“ wie Zeichensatzbibliotheken und andere Goodies für den Atari oder TEX. Ich habe leider keine direkte Möglichkeit, `ASCREEN` in Mailboxsystemen wie Maus oder Fido unterzubringen; hier muß die „natürliche Diffusion“ helfen.

Wer keinen Netzzugang hat, kann mir eine 3,5-Zoll-DD-Diskette nebst einem ausreichend frankierten Rückumschlag schicken und dabei die Versionsnummer eines etwa vorhandenen `ASCREEN` angeben. Ich schicke dann die aktuelle Version zurück oder hebe die Diskette bis zur nächsten Version auf, falls die Versionsnummer die der neuesten Version ist.

Unter der Adresse

`ascreen@math.uni-frankfurt.de`

gibt es eine Mailingliste für Ankündigungen, Diskussion usw. zum Thema `ASCREEN`. Schicken Sie Mail an

`ascreen-request@math.uni-frankfurt.de`

wenn Sie auf die Liste gesetzt werden möchten.

Das war's von mir aus. Nun viel Spaß mit `ASCREEN`, und ich freue mich auf Ihre Postkarte!

Why is programming fun?

. . . First is the sheer joy of making things.
As the child delights in his mud pie,
so the adult enjoys building things,
especially things of his own design.

. . . Second is the pleasure of making things that are useful to other people.
Deep within, we want others to use our work and to find it helpful.

. . . The magic of myth and legend has come true in our time.
One types the correct incantation on a keyboard,
and a display screen comes to life,
showing things that never were nor could be.

Programming then is fun
because it gratifies creative longings built deep within us
and delights sensibilities we have in common with all men.

— FREDERICK P. BROOKS, JR., *The Mythical Man-Month* (1975)