

Geschönte Quellen mit a2ps

ALEXANDER MAYER, TECHN. UNIVERSITÄT MÜNCHEN

Um sich bei der Entwicklung großer Software-Projekte zurechtzufinden, nutzen viele Programmierer trotz moderner Entwicklungsumgebungen vielfach die altbewährte Methode: das Ausdrucken des Quelltexts. Unix wäre nicht Unix, gäbe es für diese Aufgabe nicht ein kleines spezialisiertes Tool, das Quelltexte übersichtlich mit intelligentem Syntax-Highlighting ausdrückt. Dieser Allesdrucker heißt a2ps und ist bereits seit Jahren fester Bestandteil jeder Distribution.

Das GNU-Tool a2ps[1] nennt sich selbst »Any To Postscript Filter«, macht also im Idealfall aus jeder Datei eine Postscript- und somit druckfähige Datei. Anfangs war der Name allerdings etwas bescheidener gewählt: »ASCII To Postscript«. Im Laufe der Zeit entwickelte sich jedoch a2ps zu einem vollkonfigurierbaren Filter mit eigener Programmiersprache. Da aber für alle wichtigen Programmiersprachen bereits Filterdefinitionen, die sogenannten »Stylesheets«, im Paket enthalten sind, muß man sich im Regelfall damit nicht auseinandersetzen. Im einfachsten Fall reicht es, die gewünschte Datei als Argument a2ps zu übergeben:

```
$ a2ps main.c
```

Standardmäßig wird die Datei zweispaltig formatiert und ohne Syntax-Highlighting direkt an den Standarddrucker (*lp*) gesandt. Möchte man die Formatierung durch a2ps vor dem Drucken betrachten, kann man mit der Option *-o* eine Datei angeben:

```
$ a2ps main.c -o main.ps
```

Die Ausgabe erfolgt in diesem Fall in die angegebene Datei und nicht auf den Drucker. Solche Postscript-Dateien lassen sich am einfachsten mit Ghostview betrachten. Anschlie-

ßend kann man diese Dateien mit *lpr* ausdrucken:

```
$ gv main.ps
$ lpr main.ps
```

Natürlich druckt a2ps auch auf jeden beliebigen anderen Drucker, wenn man den Namen der Warteschlange mit der Option *--printer* angibt:

```
$ a2ps --printer=laserjet main.c
```

Da die Standardformatierung nicht immer passend sein wird, ermöglicht a2ps viele verschiedenen Anpassungen des Druckbilds.

Zeilenumbrüche

Besonders bei Quelltexten ist es wichtig, daß so wenig Zeilenumbrüche wie möglich auftreten, sie würden nur die Lesbarkeit beeinträchtigen. Wie viele Quelltextzeilen nicht mehr in eine Druckzeile passen, gibt a2ps als Statusinformation aus:

```
$ a2ps graph.c -o x.ps
[graph.c (C): 12 pages on 6 sheets]
[Total: 12 pages on 6 sheets] saved into
the file `x.ps'
[106 lines wrapped]
```

In der ersten Zeile meldet a2ps in diesem Beispiel, daß die Formatierung mit dem Stylesheet für die Pro-

grammiersprache C vorgenommen wurde. Außerdem erfahren wir, daß der Quelltext zwölf Seiten benötigt, aber auf sechs Papierseiten gedruckt werden kann. In der letzten Zeile folgt die Information, wie viele Quelltextzeilen nicht mehr auf eine Druckseite paßen. Im Beispiel waren es 106 Zeilen – ein zu hoher Wert. Um die Anzahl der Zeilenumbrüche auf ein erträgliches Maß zu reduzieren, sollte man die Anzahl der Zeichen pro Zeile erhöhen. Der Standardwert von 80 Zeichen ist für dieses Beispiel offenbar zu niedrig.

Die Option *--chars-per-line* gibt an, wie viele Zeichen eine Zeile fassen soll. Ohne Hilfe müßte man nun raten, wie lang die längste Zeile ist. Um sich diese Mühe zu ersparen, befindet sich auf der Heft-CD ein kleines Shellskript namens *longestline*. Es gibt die Zeilenlängen aufsteigend sortiert aus, die Länge der längsten Zeile ist somit die letzte ausgegebene Zahl:

```
$ longestline graph.c
...
130
133
133
133
141
143
162
```

Vielleicht fragen Sie sich, warum das Skript nicht einfach nur die maxi-

male Zeilenlänge ausgibt. Das obige Beispiel liefert bereits eine nahe-liegende Lösung für diese Frage: Die längste Zeile faßt 162 Zeichen, die zweitlängste jedoch bereits nur noch 143 Zeichen. Eine Zeilenlänge von 162 Zeichen würde die Schriftgröße beim Druck auf die Hälfte des normalen Werts reduzieren, was die Lesbarkeit deutlich verringert. Nur für eine einzige Zeile scheint das ein zu hoher Preis zu sein, deshalb wird man in der Praxis ein paar Zeilenumbrüche in Kauf nehmen, in diesem Beispiel würde sich ein Wert von 130 Zeichen anbieten.

Tabulatorlänge

Mehr Platz in einer Zeile kann man auch noch durch eine geringere Tabulatorenlänge erreichen. Standardmäßig macht a2ps aus einem Tabulatorzeichen acht Leerzeichen. Reduziert man das auf zwei Leerzeichen, paßt wesentlich mehr auf eine Drucksei-

te. Das Skript *longestline* berücksichtigt Tabulatoren und rechnet ebenfalls standardmäßig mit acht Leerzeichen, kann aber mit der Option *--tabsize* jede beliebige Tabulatorenlänge verwenden. Ruft man *longestline* mit dem Parameter *--tabsize=2* auf, zeigt sich, daß mit dieser Konfiguration bereits eine Zeilenlänge von 98 Zeichen reicht:

```
$ longestline --tabsize=2 graph.c | \
    tail -2
98
98
```

Nun muß man nur noch a2ps die gewünschte Tabulatorenlänge mit der Option *--tabsize* übergeben:

```
$ a2ps --tabsize=2 graph.c -o x.ps
[graph.c (C): 8 pages on 4 sheets]
[Total: 8 pages on 4 sheets] saved into
the file `x.ps'
[2 lines wrapped]
```

Wie die Ausgabe von a2ps zeigt, ist die Berechnung der maximalen Zei-

lenlänge nicht ganz exakt, da die Formatierung nicht unbedingt immer für jedes Zeichen die selbe Breite vorsieht. Aber sie ist sehr nahe am idealen Wert.

Falls ein Quelltext selbst mit den gerade besprochenen Methoden nicht handlich gedruckt werden kann, sollte man auf das zweispaltige Drucken verzichten. Standardmäßig druckt a2ps mit zwei Spalten und einer Zeile im Querformat, die Anzahl der Spalten und Zeilen, aber auch die Druckrichtung, können jedoch beliebig gewählt werden. Möchte man lediglich eine Seite auf einer Papierseite im Querformat, muß man a2ps folgendermaßen aufrufen:

```
$ a2ps --columns=1 --rows=1 --landscape
```

Falls Sie einen Drucker mit Duplex-Vorrichtung besitzen oder manuell gerne beide Seiten eines Papierbogens nutzen, werden Sie häufiger auch in Buchform ausdrucken, das heißt, die Reihenfolge der Seiten wird so gewählt, daß man die Papierbögen nur noch in der Mitte zusammenfalten muß, um ein handliches Heft zu erhalten. Auch für diesen Modus bietet a2ps Unterstützung an. Folgender Aufruf reicht, um zweiseitig in Buchform zu drucken:

```
$ a2ps --book graph.c -d
```

Allerdings funktioniert diese Funktion, die direkte Anweisungen in die Postscript-Ausgabe einfügt, in einigen Tests nicht immer zuverlässig mit jedem Drucker. Zu empfehlen ist vielmehr der Einsatz der bewährten Unix-Tools im Paket PSUtils [2], das in jeder gängigen Distribution enthalten ist. Hiermit klappt der zweiseitige Druck nicht mehr durch einen einzigen Aufruf, aber dafür zuverlässiger. Als erstes muß man a2ps eine Postscript-Datei im Hochformat mit einer Spalte und einer Zeile erstellen lassen:

```
$ a2ps --highlight-level=heavy \
    --portrait --columns=1 --rows=1 \
    graph.c -o graph.ps
```

Im zweiten Schritt übergibt man die



Bild 1: a2ps nutzt die Möglichkeiten von TeX optimal aus.

Option	Beschreibung
<code>--toc</code>	Druckt ein Inhaltsverzeichnis mit Seitenangaben.
<code>--line-numbers=n</code>	Druckt neben jeder n-ten Zeile eine Zeilennummer.
<code>--lines-per-page=n</code>	Druckt n Zeilen pro Seite.
<code>--no-header</code>	Druckt keinen Rahmen oder sonstige Randinformationen.
<code>--center-title</code>	Text in der Mitte oben (alle Informationstexte sind beliebig konfigurierbar, Escape-Sequenzen werden als Platzhalter verwendet, zum Beispiel steht »%p.« für die aktuellen Seitennummer).

Tabelle 1: Weitere nützliche Option von `a2ps`

Postscript-Datei an die PSUtils:

```
$ psbook graph.ps | psnup -2 > x.ps
```

`psbook` sortiert die Seiten richtig und `psnup -2` sorgt dafür, daß immer zwei Seiten auf eine Papierseite gedruckt werden. Anschließend kann die Datei `x.ps` entweder direkt an den Drucker mit `lpr` geschickt oder mit einem Betrachter wie `Ghostview` begutachtet werden.

Syntax-Highlighting

Das Einfärben von Schlüsselwörtern einer Programmiersprache ist beim Lesen von Quelltexten meist eine große Hilfe und macht das Druckergebnis ansprechender. Diese Syntax-Highlighting genannte Funktion ist mittlerweile Standard selbst bei kleinen Editor-Programmen, allerdings meist nur auf die Bildschirmdarstellung beschränkt. Nur wenige Programme können Syntax-Highlighting auch auf das Papier bringen.

`a2ps` beherrscht Syntax-Highlighting für die meisten gängigen Program-

miersprachen und kann sogar noch mehr als das Einfärben von Schlüsselwörtern. Da `a2ps` TeX zur Erstellung der Postscript-Datei verwendet, stehen alle technischen Möglichkeiten dieses professionellen Satzprogramms zur Verfügung. Schaltet man den Modus für das Syntax-Highlighting auf die Stufe »heavy«, werden viele Operatoren durch die üblichen typographischen Zeichen

ersetzt, aus »&&« wird beispielsweise »^« (siehe Bild 1):

```
$ a2ps --highlight-level=heavy graph.c
```

Falls Sie ein konservativeres Layout bevorzugen, können Sie auch den Modus *normal* verwenden. Er färbt lediglich die Schlüsselwörter ein und ersetzt keine Zeichen.

Wie so viele Programme in der Computerwelt verwendet `a2ps` standardmäßig das amerikanische Papierformat »Letter«. Falls die Distribution nicht bereits vorgesorgt hat, muß man mit der Option `--medium` die gewünschte Papiergröße angeben:

```
$ a2ps --medium=a4 graph.c
```

In `a2ps` stecken noch wesentlich mehr Fähigkeiten als hier angesprochen. Eine kleine Auswahl weiterer nützlicher Optionen zeigt Tabelle 1. Geben Sie einfach `a2ps` eine Chance. Es lohnt sich. Zumal die wenigsten GUI-Programme inklusive `KWrite` ein solch schönes Druckbild erzeugen. ♦

Anzeige



Literatur

- [1] `a2ps`-Homepage:
<http://www.infres.enst.fr/~demaille/a2ps/>.
- [2] PSUtils:
<http://www.tardis.ed.ac.uk/~ajcd/psutils/>.