

PostScript™-Einführung

Frank Richter

27.01.2003

Was ist PostScript?

- Einfache, interpretierte Programmiersprache, deren Hauptzweck die Beschreibung von Text, geometrischen Formen und Rasterbildern ist.
- reines ASCII, einfach und portabel
- geräteunabhängig, relativ high-level
- Grundlage für computer-unterstütztes Publishing und die gesamte Druckvorstufe. Zuerst in High-End-Geräten, jetzt auch in preiswerten Laserdruckern
- In den 80er Jahren von der Firma Adobe entwickelt. Adobe lizenziert PostScript™-Interpreter für Geräte- und Software-Hersteller

PostScript™ und Linux

- Linux hat die Verwendung von PostScript™ im Drucksystem von Unix übernommen
- Nahezu alle Anwendungen liefern für den Druck PostScript™.
Beispiele: L^AT_EX (über dvips), groff, OpenOffice, Browser, gimp, xv
- Direkte Ausgabe an PostScript™-fähige Geräte (über Spoolsystem)
- Indirekte Ausgabe an andere Geräte mittels `ghostscript`.
Sehr portable Software, auch für DOS, Windows, OS/2 etc.
`ghostscript` leistet auch Voransicht und Konvertierung in andere Formate (pdf, bmp, pnm, tiff, fax, usw.).

Wozu Kenntnisse über PostScript™?

- Druckausgabe für eigene Software. Manchmal kann es sinnvoll sein, ein Zwischenformat zu erzeugen (groff, T_EX). Meist wird man aber direkt PostScript™ ausgeben, da man so volle Kontrolle über die Ausgabe hat.
- Gerade mit Scriptsprachen oft schneller hingeschrieben, als Textverarbeitung zu automatisieren (Makros, StarBasic, VBA), besonders wenn es um genaue absolute Positionierung der Ausgabe geht (z.B. Eindruck in Formulare)
- Nachbearbeitung von PostScript™-Dateien
- Korrektur fehlerhafter PostScript™-Dateien

Dokumentation

- Adobe PostScript™Language Reference Manual (das sogenannte Red Book)
mitlerweile Version 3.
Detaillierte Beschreibung aller Features der Sprache, aller Operatoren usw.
<http://www.adobe.com/print/postscript/pdfs/PLRM.pdf>
- Adobe PostScript™Language Tutorial and Cookbook (das sogenannte Blue Book)
<http://www-cdf.fnal.gov/offline/PostScript/BLUEBOOK.PDF>
- Postscript Language Program Design (Green Book)
<http://www-cdf.fnal.gov/offline/PostScript/GREENBK.PDF>

Entwicklungsumgebung

- Editor, möglichst mit Syntax-Highlighting (vim!)
- PostScript™-Interpreter: `ghostscript`, aktuell Version 8.0, in vielen Distributionen noch 7.x (reicht völlig aus).
<http://www.ghostscript.com>
- Oberfläche für den Interpreter: `ghostview`, `kghostview`, `gv`
- PostScript™-Drucker oder entsprechende Filter (`CUPS`, `magicfilter`, `apsfilter`)

Funktionsweise der PostScript™-Sprache

- Werte werden auf dem Stack abgelegt, LIFO (last in first out)
- postfix-Notation
 - Der Operator nimmt die Werte vom Stack.
 - Danach sind diese erstmal weg! Die Sprache „verkonsumiert“ quasi ihre Daten
 - Ergebnisse werden normalerweise wieder auf den Stack gelegt.
- Einzelne Objekte werden durch Leerzeichen, Tab, CR oder NL getrennt.
- Kommentare beginnen mit % und gehen bis zum Zeilenende
- Mit dem Befehl `pstack` kann man den Stackinhalt inspizieren

Beispiel 1

Einfach interaktiv mit `gs` ausprobieren:

```
1 2      % legt 1 auf den Stack, dann eine 2
pstack
add      % nimmt zwei Operanden vom Stack,
         % addiert diese, und legt das Resultat wieder ab.
pstack
==       % entnimmt oberstes Stackelement und gibt es aus
quit     % beendet den Interpreter
```

Weitere Eigenschaften

- Notation des Stackinhaltes in der Literatur oft so:
links ist unten, rechts ist oben, \Rightarrow , Resultat ebenso.
Beispiel: 1 2 add \Rightarrow 3
- Datentypen: real, boolean, array, string, dictionary, mark
- Der Stack kann alle Datentypen aufnehmen, auch z.B. arrays und dictionaries
- Wegen postfix-Notation sind keine Klammern nötig:
3 4 add 1 3 add mul % entspricht $(3+4)*(1+3)$