

Vorwort

*Programmieren ist wie Küssen:
Man kann darüber reden,
man kann es beschreiben,
aber man weiß erst, was es bedeutet,
wenn man es getan hat.*
Andree Beaulieu-Green

Perl ist bei manchen Programmierern schon nicht mehr so hip – Sprachen wie Ruby oder Python gewinnen neue Freunde. Auch aus den Schlagzeilen ist Perl ziemlich verschwunden, obwohl weiterhin viele Anwender Perl an allen Ecken und Enden einsetzen und nicht im Traum daran denken, etwas anderes zu verwenden.

Larry Wall erfand Perl seinerzeit als „glue language for Unix“ – eine interessante Mischung von Shell, sed, awk und C. Aus dieser Zeit stammt auch der Spitzname *Pathologically Eclectic Rubbish Lister*. Mit der Zeit wurde Perl auf nahezu alle Betriebssysteme portiert und zu einer universellen, unglaublich flexiblen (wenn auch nicht immer schönen) Programmiersprache entwickelt – oder wie Kristian Köhntopp treffend sagte: „Perl ist ein Schmerzmittel. Wenn Du bei der Speicherverwaltung oder der Stringverwaltung unter C Schmerzen hast, nimm Perl.“. Der Einsatzbereich von Perl ist nahezu unbegrenzt. Da gibt es den schnellen Einzeiler zur Systemadministration oder das „quick and dirty“ geschriebene Programm zum Füttern einer Datenbank mit Informationen aus einer eingescannten und mit OCR behandelten Tabelle, die nur noch als Ausdruck existierte. Andererseits existiert das gehobenerere Perl, das man verwendet, um wartbaren und lesbaren Code zu schreiben. So wären die Internet Movie Database, heise online, Slashdot, der BBC-Nachrichtenticker, Spam Assassin und viele andere lieb gewordene Anwendungen ohne Perl vielleicht nie in ihrer jetzigen Form entstanden. Dazu gibt es die Perl-Community, die es Anfängern und Einsteigern ermöglicht, Erfahrungen mit der Sprache zu sammeln, es werden massenhaft Workshops und Konferenzen veranstaltet, und vor allem existieren für nahezu jedes Problem schon die passenden Perl-Module – und wenn

es für das eigene Problem wirklich noch keines gibt, dann schreibt man selbst ein Modul und erwirbt damit vielleicht ein klein wenig Ruhm in der CPAN-Gemeinde.

Perl ist neben C/C++ diejenige Programmiersprache, zu der bereits sehr viel und sehr kompetente Literatur existiert. In den letzten Jahren ist es jedoch um die Perl-Literatur stiller geworden. Dabei sind ausgezeichnet geschriebene Bücher eine der Stärken der Perl-Kultur und der Perl-Szene. Man kann also eigentlich nichts Neues mehr erzählen, sondern nur das Bekannte neu kneten (ob es dann besser wird, ist eine weitere offene Frage). Aus diesem Grund versuche ich ein vom üblichen Standard abweichendes Konzept, das zumindest teilweise eine gewisse Alleinstellung in sich trägt. Zwar widmen sich der zweite und dritte Teil der Einführung in die Sprache, jedoch sind diese Teile nicht der Schwerpunkt des Buchs – was Sie schon daran erkennen, dass es mit allgemeinen Betrachtungen zur Programmierung beginnt, bevor es sich der Sprache Perl zuwendet.

Der Schwerpunkt liegt in der Programmierung und Anwendung von Algorithmen aus verschiedenen Gebieten, darunter numerische Mathematik (Gleichungssysteme, Regression, Interpolation), Berechnung von Datum und Uhrzeit (Datumsdifferenzen, Feiertage usw.), nichtnumerische Anwendungen (Suchen, Sortieren, insbesondere reguläre Ausdrücke (eine Domäne von Perl), Grafik, Benutzeroberflächen und Datenbankanbindung.

Man mag natürlich einwenden, dass für nahezu jedes Problem schon eine Lösung in Form eines Perl-Moduls beim CPAN existiert und es daher nicht nötig ist, diese Algorithmen zu behandeln. Das ist schon richtig, und an vielen Stellen wird auch auf die Module zurückgegriffen, jedoch ist es eines der Ziele dieses Buchs, Ihnen als Leser die Hintergründe verschiedener Algorithmen nahezubringen. Zum einen wird dadurch beispielsweise die Funktion des WWW-Protokolls HTTP durchschaubar – oder Sie werden in die Lage versetzt, vorhandene Module effizienter einzusetzen.

Weiter geht es dann mit speziellen Algorithmen, darunter die Programmierung von Web-Anwendungen (CGI) und Netzwerkprogrammierung (Clients und Server), Programmierung von Front-Ends für beliebige Anwendungen (z. B. Suchmaschinen, Indexierung von Texten, Informationsdienste etc.). Behandelt werden die Codegenerierung oder die Anbindung von \LaTeX zum automatischen Erzeugen von PDF-Dokumenten. Dabei geht es auch immer wieder um die Anwendung bereits existierender Perl-Tools, darunter insbesondere Tools für Administratoren und nicht zuletzt die Ansteuerung von Hardware-Komponenten des PC.

Das Buch behandelt die aktuelle Version der Sprache, Perl 5. Dem aktuellen Perl macht man gerne zum Vorwurf, es beherrsche keine richtige Objektorientierung. Der Hash-Referenz-Pfeil-Stil gehört nicht unbedingt zu den Favoriten in Perl. Zur Beruhigung: Perl 6 wird richtiges OOP für echte Männer mit Klassen, Methoden und allem, was das Herz begehren können. Warum dann aber nicht gleich Perl 6 hier im Buch? Die Antwort ist ganz einfach: Seit der Ankündigung von Perl 6 im Jahr 2000 gibt es bis heute kein handliches Distributionspaket. Aber es wird daran gearbeitet – genauer gesagt, an Rakudo, dem Perl 6, das auf der Parrot Virtual Machine basiert. Da Perl 6 zu allererst eine Sprachreferenz ist, kann sich auch jeder sein eigenes Perl 6 implementieren, weshalb es bereits einige eigenständige Perl-6-Varianten gibt.

Es wird aber bis zum ersten „offiziellen“ Perl 6 sicher noch bis 2010 dauern. Die Entwickler der Perl-Version 6 haben Anfang April 2009 die Version 1.0 der virtuellen Maschine

Parrot veröffentlicht, auf der die Perl-6-Implementierung Rakudo aufsetzt. Weiterhin soll Parrot, das im Unterschied zu Perl 5 Compiler und Runtime voneinander trennt, auch andere dynamische Sprachen unterstützen (z. B. die Python-Implementierung Pynie und die Ruby-Umsetzung Cardinal). Eine Liste mit den Features, die in Perl 6 bereits funktionieren (und mit solchen, die nicht oder nur fehlerhaft klappen) findet sich bei der Perl Foundation (<http://www.perlfoundation.org/>). Wer Dokumentation zu der neuen Perl-Version braucht, dem hilft ein Blick in die Blog-Serie über Perl 6 von Moritz Lenz (<http://perl-6.de/>). So muss die Behandlung von Perl 6 auf die folgende Auflage des Buchs warten. Einen kleinen Ausblick erhalten Sie am Ende des ersten Teils auf Seite 154.

In dieses Buch sind Informationen aus Hunderten von Manualseiten sowie Info-, Readme-, FAQ- und HOWTO-Dokumenten eingeflossen, ebenso aus zahllosen Beiträgen diverser Perl-Newsgroups, aus Artikeln in den Zeitschriften c't, iX, Linux-Magazin, Linux Journal und aus Wikipedia. Es ist daher unmöglich, ein exaktes und vollständiges Quellenverzeichnis zusammenzustellen. Zur Vertiefung der einzelnen Themen, die in etlichen Kapiteln manchmal nur angerissen werden konnten, empfehle ich die im Literaturanhang aufgeführten Buchtitel.

Nicht zuletzt möchte ich mich bei allen bedanken, die mir direkt oder indirekt bei der Arbeit an diesem Buch geholfen haben, und auch bei jenen, die wegen der Arbeit am Manuskript vernachlässigt wurden. Mein besonderer Dank gilt Frau Metzger und Frau Weilhart vom Hanser-Verlag für die gute Zusammenarbeit, Jürgen Dubau für das Finden aller Tippfehler und seine Tipps, Larry Wall für Perl, Hermann Zapf für die Palatino, Donald E. Knuth für \TeX , Leslie Lamport für \LaTeX , Linus Torvalds für Linux und Carl Barks für den Schabrackenschriller.

München, Juni 2010
Jürgen Plate (plate@netzmafia.de)
<http://www.netzmafia.de/>