



MAGAZIN FÜR PROFESSIONELLE INFORMATIONSTECHNIK

Anschluß für die Firma:

Internet-Provider

Vergleichstest:

3D-Grafik-Workstations

Systeme von Digital, HP, IBM, Intergraph, SGI und Sun

Know-how

Gigabit-Ethernet versus ATM
Red-Hat-Linux verwalten

Photoshop-Alternative

GIMP V 0.99.x

Java

Java Native Interface
IBM-Projekt San Francisco

Multimedia

Apples Quicktime 3.0

Softwareentwicklung

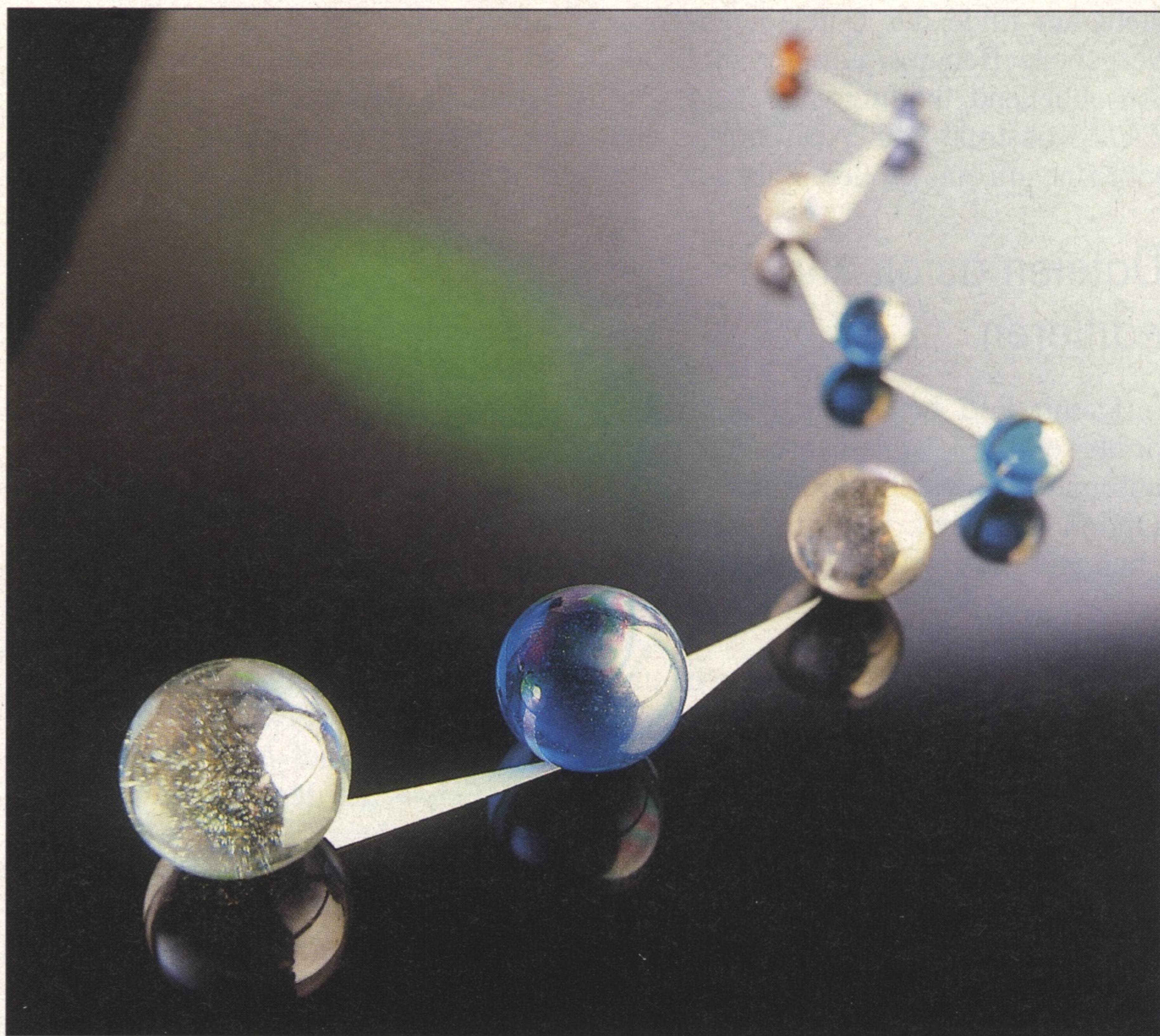
Portierung von Unix
nach NT



Marktübersicht:

Komplettlösungen für
High Availability





Hilfreiche Perl-Einzeiler für Unix und Windows

Kurz und gut

Michael Schilli

Statt unleserliche Wunderwerke zu erzeugen – wie im regelmäßig stattfindenden Obfuscated Perl Contest – kann man mit Perl-Einzeilern auch flott Alltagsprogrammierjobs erledigen. Es folgt eine Sammlung der wichtigsten, die man immer im Handgepäck haben sollte.

Perl-Skripts müssen nicht in einer Datei stehen. Der Interpreter verarbeitet auch mit der Option `-e` hereingereichte Strings. Um zum Beispiel 61 000 DM Jahresgehalt mal schnell durch 12 zu teilen, reicht die Eingabe

```
perl -e 'print 61000 / 12'
```

und Perl rechnet fließkommagenau 5083.3333333333 aus. Zu beachten ist bei solchen Aufrufen von der Kommandozeile, daß der auszuführende Code in Unix-Shells in einfachen Anführungszeichen steht (sonst expan-

diert die Shell Sonderzeichen wie * und \$), der Command-Interpreter unter Windows 95 und NT jedoch doppelte Anführungszeichen verlangt. Die vorangestellte Option `-l` dient hier nur dazu, hinter das Ergebnis ein Newline-Zeichen einzufügen.

grep, awk und
sed in einem

Perls Stärke ist zweifellos die Manipulation von Texten, und so steht mit `-n` eine Option zur Verfügung, mit der

MEMORY

für SUN,SGI,HP,IBM,DEC u. NT
Bei Memory-Upgrades verlassen wir uns auf führende Hersteller wie z.B. KINGSTON, DATARAM u. CAMINTONN

DISK

4GB, 9GB, 18GB und 47GB
16-bit mit Ultra-SCSI und Ultra2-SCSI
SE-SCSI, Differential-SCSI oder LVD
Interne oder Externe Subsysteme
für SunOS, Solaris, HP-UX, AIX, IRIX,
Digital-UNIX und WindowsNT

BACKUP

DLT, DAT und AIT
Einzellaufwerke bis 70GB - 5 MB/s
Autoloader bis 4.2TB - 20 MB/s
Single-User, Server- oder
Netzwerk-Backuplösungen
für SunOS, Solaris, HP-UX, AIX, IRIX,
Digital-UNIX und WindowsNT

RAID

bis 820 GB KAPAZITÄT
Level 0, 1, 3 oder 5
Ultra-SCSI-based Hardware-Lösung
max. 128 MB CACHE
bis 32 MB/s Datentransferrate
für SunOS, Solaris, HP-UX, AIX, IRIX,
Digital-UNIX und WindowsNT

NETWORK

10 Mbit oder 100 Mbit
3COM, BAY NETWORK, ATI, DIGITAL...
Ethernet, Fast-Ethernet und FDDI
Transceiver, Repeater, HUBs, Switches
Netzwerkverkabelung

**JETZT KOSTENLOS
KATALOG ANFORDERN**

07121 / 483-100

info@advunibyte.de

oder im Internet lesen
<http://www.advunibyte.de>

**Advanced
UniByte**

Alle eingetragenen und sonstigen erwähnten Warenzeichen
sind das alleinige Eigentum ihrer jeweiligen Inhaber

sich auf der Kommandozeile angegebene (oder durch die Standardeingabe hereinkommende) Dateien bearbeiten lassen:

```
perl -n -e 'print $_ if /mschilli/' /etc/passwd
cat /etc/passwd | perl -n -e 'print $_ if /mschilli/'
```

Beide Befehle sehen nach, ob in der Unix-Paßwortdatei irgendwo der String 'mschilli' vorkommt, und geben passende Zeilen aus. Intern wickelt der Perl-Interpreter bei gesetzter Option *-n* nämlich eine *while*-Schleife um den angegebenen Kommandostring, die entweder auf der Kommandozeile angegebene Dateien oder den Datenstrom der Standardeingabe zeilenweise abarbeitet:

```
while (<>) {
    print $_ if /mschilli/;
}
```

Da die Paßwortdatei bekanntlich Einträge vom Format

```
mschilli:8KYD6mggn4tsl:501:100:Michael \
Schilli:/home/mschilli:/bin/bash
```

enthält, wäre es angebracht, die durch Doppelpunkte getrennten Einzelfelder gesondert zu untersuchen. Bei gesetzter Option *-a* zerlegt Perl die Felder der hereinkommenden Zeilen in die Elemente eines Array mit dem Spezialnamen *@F*, wobei es den in der *-F*-Option gesetzten Feldtrenner benutzt, oder, falls diese fehlt, an Leerzeichen oder Tabulatoren trennt. Die Benutzerkürzel, also die ersten Felder der Paßwortdateieinträge spuckt somit die Konstruktion

```
perl -l -a -F: -n -e 'print $F[0]' /etc/passwd
```

aus. Daraus läßt sich eine hilfreiche Anwendung ableiten: Das dritte Feld des Paßworteintrags enthält die im System eindeutige Benutzernummer. Möchte der Administrator einen Account für einen neuen Benutzer einrichten, sucht er – falls er, wie alle 'echten' Sysadmins, ohne GUI-Schnickschnack arbeitet – nach der nächsten, noch unbenutzten Nummer. Da Perl wie der gute alte *awk* ein *END*-Konstrukt bietet, dessen Inhalt der Interpreter nach der impliziten *-n*-Schleife ausführt, findet folgendes Konstrukt die gesuchte Nummer:

```
perl -l -n -a -F: -e '$high = $F[2] if $F[2] > $high;
END {print $high+1}' /etc/passwd
```

Zeile für Zeile durchsucht dieses Skript die Paßwortdatei und setzt *\$high* gleich der aktuellen Benutzernummer im dritten Feld, falls diese größer als der bislang gespeicherte Wert ist. So

steht am Ende der Datei die höchste vergebene Benutzernummer in *\$high*. Anschließend kommt der Code im *END*-Konstrukt zur Ausführung und gibt eine um eins größere Zahl aus.

Dateien automatisch editieren

Damit Perl bei solchen impliziten Schleifen die jeweils bearbeitete Zeile automatisch ausgibt, muß statt *-n* die Option *-p* stehen. Dies hilft beim Suchen und Ersetzen von Textstücken:

```
perl -p -e 's/a/b/g' datei
```

ersetzt in *datei* alle 'a' durch 'b' und gibt den modifizierten Inhalt auf der Standardausgabe aus. Um den Inhalt der Datei selbst zu verändern, muß die *-i*-Option (für *in-place-edit*) herhalten: Der Aufruf

```
perl -p -i.bak -e 's/a/b/g' datei
```

sichert zunächst für alle Fälle *datei* nach *datei.bak* und führt anschließend die Zeichenersetzung in *datei* selbst durch. Das obige Kommando funktioniert auch für mehrere Dateien gleichzeitig:

```
perl -p -i.bak -e 's/\bprintf\b/myprintf/g' *.c
```

ersetzt unter Unix, wo die Shell **.c* zu einer Reihe von Dateinamen expandiert, in allen C-Dateien im gegenwärtigen Verzeichnis sämtliche Aufrufe von *printf* (Wortgrenzen links und rechts) durch eine neue Funktion *myprintf* und sichert gleichzeitig die alten Dateien nach **.c.bak*. Kommt ein Kurzsript nicht mit dem Funktions-

umfang der Standard-Perl-Bibliothek aus, sondern benötigt Zusatzmodule, veranlaßt die Option *-M* den Interpreter, die angegebenen Module einzubinden. Somit steht Einzeilern die Welt offen: Das praktische Modul *LWP::Simple* aus der Programmsammlung *libwww* beispielweise gewährt einfachen Zugriff auf das World Wide Web:

```
perl -MLWP::Simple -e 'getprint('http://www.aol.com')'
```

holt den Inhalt der Web-Seite *http://www.aol.com* vom Netz und gibt ihn auf der Standardausgabe aus. Zu beachten ist jedoch, daß *LWP::Simple* nur einfache Web-Zugriffe beherrscht, Redirects sind nicht im Funktionsumfang enthalten. Verweist ein angegebener URL statt auf eine Datei auf ein Verzeichnis auf dem Zielrechner, vollführen moderne Browser einen Redirect, aus *http://www.aol.com/netfind* wird so flugs *http://www.aol.com/netfind/*, und nur der zweite URL ist für *LWP::Simple* geeignet.

Um den Inhalt einer Web-Seite nicht nur auszugeben, sondern gleich wegzuschreiben, hilft das Konstrukt

```
perl -MLWP::Simple -e \
'getstore('http://www.aol.com', 'aol.html')
```

Es speichert die geholte Information gleich in der Datei *aol.html* auf der lokalen Festplatte. Schon kein richtiger Einzeiler mehr, aber dennoch ganz praktisch ist

```
cat aol.html | \
perl -MHTML::TreeBuilder -e \
'print map { $_->[0]\n' } \
@{HTML::TreeBuilder->new->parse_file(\*STDIN)->\
extract_links}'
```

Benötigte Module

Das aktuelle Perl 5.004_04 für Unix-Plattformen enthält das verwendete Modul *File::Copy* bereits von Haus aus. *LWP::Simple* und *HTML::TreeBuilder* sind Bestandteil der Bibliothek *libwww* von Gisle Aas, die auf dem CPAN erhältlich ist, wo auch *MIME::Base64* zur Base64-Kodierung zu finden ist.

Die Perl-Distribution für Windows 95 und NT, die unter *ports/win95/Standard/x86/Perl5.00402-bindist04-bc.tar.gz* auf dem CPAN liegt, enthält alle verwendeten Module. Zu beachten ist, daß der unter Windows benutzte DOS-Kommando-Interpreter auf den doppelten Anführungszeichen (") zum Einschließen des Perl-Kommandostrings beharrt, eventuell im Perl-Code vorkommende Anführungszeichen sind mit ' zu maskieren. Auch fehlt die Expansion von Wildcard-Konstrukten wie **.c* auf Kommandointerpreterebene – diese Aufgabe ist an Perl zu delegieren, am besten mit Hilfe von *for*-Schleifen mit File-Globbing-Konstrukten.

Deutsche CPAN-Spiegel sind unter anderem:

<ftp://ftp.uni-hamburg.de/pub/soft/lang/perl/CPAN/>

<ftp://ftp.rz.ruhr-uni-bochum.de/pub/CPAN/>

<ftp://ftp.leo.org/pub/comp/programming/languages/perl/CPAN/>

Es extrahiert alle Hyperlinks aus dem per *cat* weitergereichten HTML-Dokument. Wie? Nun, das ist etwas komplizierter: Das erzeugte Objekt vom Typ *HTML::TreeBuilder* ruft seine *parse_file*-Methode mit einer Glob-Referenz des Standardeingabedeskriptors (**STDIN*) auf. Dadurch erstellt es ein *HTML::Parser*-Objekt, dessen Methode *extract_links* eine Referenz auf ein Array zurückgibt. Dieses besteht wiederum aus Array-Referenzen, die als erstes Element den Linknamen enthalten. Die Funktion *map* nimmt einen Codeblock und eine Liste entgegen, ruft für jedes Element der Liste den Codeblock auf (wobei der Wert des aktuell bearbeiteten Elements in der Spezialvariablen *\$_* liegt) und setzt dessen Rückgabewert in die Ergebnisliste ein. Nachdem das Skript also die von *extract_links* gelieferte Array-Referenz mit dem Konstrukt *@{...}* in ein Array verwandelt hat, steht im Codeblock mit *\$_* eine Referenz auf das Unter-Array zur Verfügung. *\$_->[0]* extrahiert dessen erstes Element.

Kopieren und Verschieben à la Unix

File::Copy schließlich erlaubt das Verschieben und Kopieren von Dateien im Unix-Look&Feel: Die Funktionen *move* und *copy* nehmen als zweiten Parameter auch ein Verzeichnis, in das sie die als ersten Parameter spezifizierte Datei verschieben beziehungsweise kopieren. Windows-Benutzer werden folgendes zu schätzen wissen:

```
perl -MFile::Copy -e 'for(<*.c>) {copy $_, '/backups'}'
```

kopiert alle *.c-Dateien im gegenwärtigen Verzeichnis ins Verzeichnis */backups*, falls dieses existiert. Die *for*-Schleife iteriert mit dem File-Globbing-Konstrukt über alle *.c-Dateien und setzt für den Schleifenrumpf *\$_* auf die aktuelle Datei. Da die Windows-Version des Perl-Interpreters dankenswerterweise auch Unix-Pfadnamen versteht, entspricht */backups* in Perl dem Windows-Pfad *\BACKUPS*, im Normalfall also *C:\BACKUPS*.

Alle Dateien im Verzeichnis *test*, die seit einer Woche nicht modifiziert wurden, verschiebt (diesmal wieder in der Schreibweise für Unix-Shells)

```
perl -MFile::Copy -e 'for(<test/*>) \
{move($_, 'test.old') if -M > 7}'
```

iX 7/1998

ins Verzeichnis *test.old*. Alles, was dort älter als drei Wochen ist, löscht

```
perl -e 'for(<test.old/*>) {unlink $_ if -M > 21}'
```

kommentarlos.

Uencode und Base 64 entziffern

Für schnelles Dekodieren von uencode- und Base64-kodierter Information stehen in Perl die *unpack*-Funktion sowie das Modul *MIME::Base64* zur Verfügung.

```
perl -e 'print unpack('u', join(' ', <>))' data.uu
```

entpackt die kodierte Information aus der Datei *data.uu* und leitet sie auf die Standardausgabe. Zu beachten ist jedoch, daß *unpack* nur die eigentlichen kodierten Daten verarbeitet, aus einer typischen UU-Datei, die zusätzlich noch den Namen der kodierten Datei und ein Ende-Tag im Format

```
begin 644 data
j2&D*:&D*:&D*
end
```

enthält, sind vor der Bearbeitung die erste sowie die letzte Zeile zu entfernen. Die besonders in EMail verwendete Base64-Kodierung knackt das Zusatzmodul *MIME::Base64*. Der Einzeiler

```
perl -MMIME::Base64 -e \
'print MIME::Base64::decode(join(' ', <>))' datafile
```

gibt die dekodierten Base64-Daten in *datafile* auf die Standardausgabe aus, wobei – analog zur UU-Kodierung – darauf zu achten ist, daß in *datafile* nur der Base64-Buchstabensalat liegt, MIME-Header und die einleitende Sequenz, die unter anderem den Dateinamen festlegt, müssen vorher manuell entfernt werden.

Genug der Beispiele. Wer noch tiefer in die Syntax von Perls Kommandozeilenoptionen einsteigen will, dem sei die Manual-Seite empfohlen, die auf das Kommando *perldoc perlrn* zum Vorschein kommt. (ck)

MICHAEL SCHILLI

arbeitet als Web-Engineer für America Online, Inc., San Mateo. Er ist Autor des im Juni bei Addison-Wesley erscheinenden Buches 'GoTo Perl 5'.

DEUTSCHE LINUX DISTRIBUTION

VERSION 5.4



"Linux ist langfristig der eigentliche Konkurrent von Windows NT"™
Dave Madden, Senior Product Manager Corel Corp.

Mit der neuen DLD 5.4 wird die nächste Runde eingeläutet. Konsequenterweise wurde die einfache Installation und Administration der DLD weiterentwickelt. Aktuelle freie und kommerzielle Software auf 4 CDs, sowie das komplett überarbeitete Handbuch machen die DLD 5.4 zur besten DLD die es je gab!

Highlights der neuen DLD 5.4

- automatisches Festplattenpartitionieren, Formatieren und Mounten von Partitionen
- automatische Konfiguration von PCI-Karten
- einfachste ISDN-Konfiguration mit Providerdatenbank
- problemloses Updaten mit ViPER, der Ihnen alle updatefähigen Pakete auf einen Blick zur Auswahl bietet
- automatische Konfiguration der meisten ISA-Soundkarten
- komfortable dialoggesteuerte Druckerkonfiguration
- menugeführte NFS-Konfiguration für Client und Server
- WEBLINK - Remote-Administration über Internet mit jedem Web-Browser - Betriebssystemunabhängig
- Optische Zustandsanzeige der gestarteten Prozesse
- automatisches und benutzergeführtes Starten der Prozesse
- komplett überarbeitetes Handbuch und zeitlich unbegrenzter, Installations-Support

4 CDs mit aktueller Software:

Netscape 4.05 / KDE Beta 4 / XFree86 V3.3.2 / Kernel 2.0.33 (2.1.98) / Solution-CD mit zahlreichen Demos und Programmen zum Freischalten, wie z.B. ADABAS D 10.0 Personal Edition, Yard, FlagShip, Wipeout, Smalltalk-X, Pro 5, Solid, IXess, Motif, Accelerated-X, u.v.m. / über 150 MB Games / das Beste vom Sunsire-Mirror (ca. 650 MB) / kompletter Sourcetreer aller nicht-kommerziellen DLD-Teile, und vieles mehr...

DEUTSCHE LINUX DISTRIBUTION:

- **DLD 5.4 CLASSIC F. STUDENTEN:** 49,-
4 CDs, inkl. OpenBackup, HotWire Fax, ca. 400 S. Dokumentation
- **DLD 5.4 CLASSIC:** 79,-
4 CDs, inkl. OpenBackup, HotWire Fax, ca. 400 S. Dokumentation
- **DLD 5.4 PRO F. STUDENTEN:** 149,-
wie Classic + Accelerated-X 4.1 OEM (inkl. Handbuch) und Disketten
- **DLD 5.4 PRO:** 199,-
wie Classic + Accelerated-X 4.1 OEM (inkl. Handbuch) und Disketten
- **DLD 5.4 PRO MOTIF:** 379,-
wie PRO + Motif 2.0 ELF Development-Kit
- **DLD 5.4 PRO CDE:** 499,-
wie PRO + Motif 1.2.5 Development-Kit + CDE 1.0.10 OEM
- **DLD 5.3 FÜR DEC ALPHA:** 79,-
1 CD, über 400 S. Dokumentation

UPDATES (NACHWEIS ERFORDERLICH):

- **DLD 5.4 PRO / PRO MOTIF / PRO CDE** 79,-
von jeweils entsprechender Vorgängerversion 5.3
- **DLD 5.4 PRO / PRO MOTIF / PRO CDE** 149,-
von jeweils entsprechender Version älter als 5.3

DLD ist eingetragenes Warenzeichen der delix GmbH.

Preise in DM

47057 Duisburg FiSh Systemtechnik, Bismarckstr. 142B,
Tel. 02 03 / 3 06 18 60, Fax 3 06 18 69
WWW: <http://www.fish.de>

70176 Stuttgart delix Computer GmbH, Schloßstr. 98,
Tel. 07 11 / 6 21 02 70, Fax 61 35 90
WWW: <http://www.delix.de>, eMail: delix@delix.de

Schweiz: Novir Datentechnik, Lagerstr. 11,
8910 Affoltern a.A., Tel. 01 / 7762550, Fax 01/7614171

Österreich: Frank CD-ROM, Simm. Hauptstr.-190/16/15,
1110 Wien, Tel. 02 22 / 7 68 36 26 Fax 7 68 60 06

Außerdem erhalten Sie die Produkte von delix auch im gut sortierten Buchhandel.

FORDERN SIE NOCH HEUTE UNSERE KOSTENLOSEN INFOS AN!

