

10/98

LINUX
MAGAZIN

Das einzige deutschsprachige Linux-Magazin

LINUX

MAGAZIN

Serie für
Einsteiger!

ISSN 1432 - 640 X

DM 9,80

ÖS 75,-

SFR 9,80

10/98

Linux News

Neues aus dem
Internet

Sicherheit

TCP-Wrapper

tcpd

Desktop Publishing

Axene

Sound

MP3, CD-R und
CDDA

Ordnung im Versions-Chaos

Quelltext-Management mit CVS

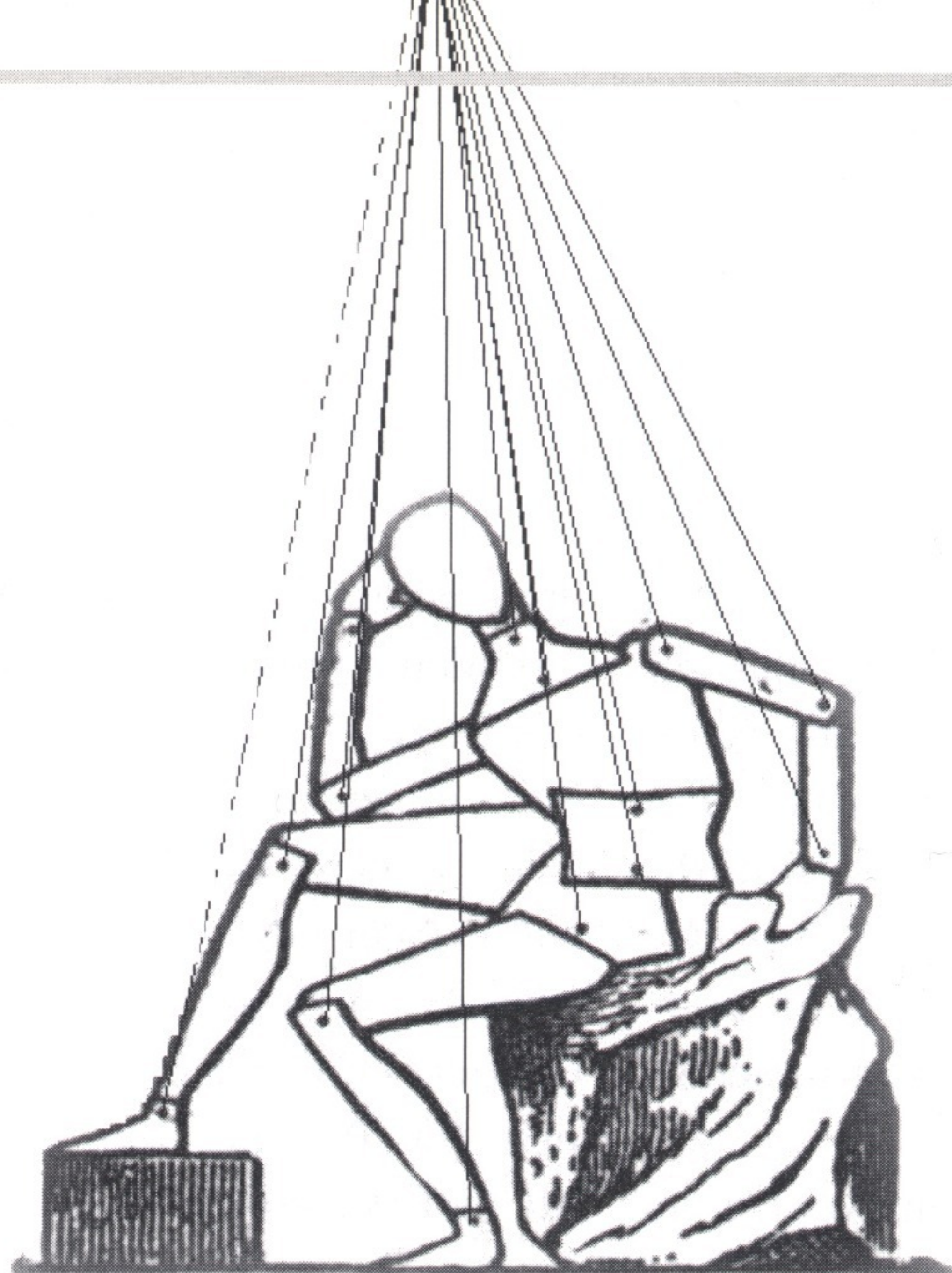


www.linux-magazin.de

Perl-Snapshot

Hurra, Expect ist da!

von Michael Schilli



Wem das klassische TCL-basierte Expect nicht behagt, findet vielleicht Gefallen am neuen Perl Expect-Modul. Damit lassen sich viele interaktive Aufgaben bequem automatisieren.

In der Februar-Kolumne stellte ich *Net::Telnet* vor, jubelnd, daß endlich Ersatz für das praktische Tcl-*expect* (siehe dazu den Artikel in der Septemberausgabe 1998) zumindest im Telnet-Bereich getroffen sei. Wie wenig wußte ich um die Dinge, die da kommen würden, mir den Alltag zu erleichtern und das Leben zu versüßen! *Austin Schutz* stellte jüngst sein *Expect*-Modul vor - und nun kann jedes Perl-Skript alle Applikationen ansprechen, die auf die immer gleichen Ausgaben die immer gleichen Antworten erwarten: Ob man sich auf einem Cisco-Router einloggt, oder ein Paßwort-Programm aufruft, das sich gemeinerweise gegen Eingaben aus der Standardeingabe sträubt - die Send-Expect-Logik des Expect-Moduls erschlägt einfach alles.

Automatisches Einloggen

Das Skript aus Listing 1 (*telnet.pl*) loggt sich per *telnet* auf einem fremden Rechner ein und bietet anschließend interaktiven Zugriff auf die geöffnete Shell an. Da mir *Andreas König* einmal bescheinigt hat, daß *telnet* nur etwas für "Ewiggestrige" wäre (der Rest der Welt nimmt angeblich *ssh*), dient das Beispiel nur dazu, die prinzipielle Funktionsweise aufzuzeigen, schließlich beschränkt sich die Anwendung von *Expect* nicht nur auf *telnet*.

Jede beliebige Applikation läßt sich damit heinzelmännchengleich ansteuern.

Der Konstruktor *spawn* aus dem *Expect*-Paket nimmt den Pfad auf ein externes Programm sowie eine Reihe von (optionalen) Parametern dafür entgegen, startet es

und liefert eine Referenz auf ein *Expect*-Objekt zurück. In Listing *telnet.pl* startet das Programm */bin/telnet* mit dem anzusteuernenden Rechner *remote-host* als Parameter. Das Telnet-Programm wird erfahrungsgemäß etwas folgendes liefern:

```
Trying 205.44.189.17...
Connected to remothost.
Escape character is '^]'.
```

```
Linux 2.0.30 (ruebe.remote.de) (ttty3)
```

```
ruebe login:
```

und gleich anschließend auf die Eingabe des Login-Namens zu warten. Der Aufruf der *expect*-Methode in Zeile 15 verfolgt die Ausgabe aufmerksam und kehrt zurück, falls das eingestellte Suchmuster daherkommt. Taucht der eingestellte Ausdruck auch nach Ablauf der im Parameter *\$timeout* eingestellten Zeit nicht auf, bricht *expect* ab und liefert *undef* zurück. *telnet.pl* bricht in diesem Fall mit der *die*-Funktion aus Zeile 16 ab und meldet den aufgetretenen Fehler. Der Aufruf der *send_slow*-Methode in Zeile 17 antwortet dem Login-Programm, indem es ihm eine Sequenz von Zeichen schickt, die jenes genauso interpretiert, als würde ein Anwender den Namen eintippen und die Return-Taste drücken. Wichtig ist dabei, *send_slow* das *\n*-Zeichen mitzugeben, sonst tut sich nichts.

Langsam senden

Der erste Parameter von *send_slow* ist die Anzahl der Sekunden, die *send_slow* zwischen jedem gesendeten Zeichen warten soll - so kann man auch mit dem langsamsten Modem kommunizieren, ohne daß dieses den Faden verliert, für die Kommunikation mit dem Login-Programm ist jedoch keine Zeitverzögerung notwendig.

Wie man statt auf feste Strings auf Muster wartet, die regulären Ausdrücken genügen, zeigt in Zeile 20 die Wartefunktion auf die Paßworteingabe, die mit *[Pp]assword* entweder auf *Password* oder *password* wartet. Geht dem Wartemuster der Parameter *-re* voran, interpretiert *expect* den nachfolgenden String als regulären Ausdruck.

Enthält der Regex einen Match aufs Zeilenende, ist darauf zu achten, daß das angesprochene Terminal *\r* als Zeilenende-Sequenz sendet und so muß statt dem Perl-üblichen *\$* der Ausdruck *\r?\$* herhalten, der ein optionales *\r* vor dem eigentlich Umbruch erlaubt. Ein Zeilenanfang notiert wie gehabt mit *^*. Ausdrücke, die mehrere Zeilen abdecken, sind erlaubt.

Listing *telnet.pl* wartet in Zeile 25 lediglich auf das *\$-* Zeichen, den Prompt der Bourne- bzw. Bash-Shell, für C-Shells wäre dies etwa durch *%* zu ersetzen.

Die Krönung: Interaktivität

Der Knüller freilich ist die *interact*-Methode aus Zeile 28, die die Standardeingabe des Skripts mit dem Eingabekanal der geöffneten Telnet-Session kurzschließt und somit freie Interaktion erlaubt. Verläßt der Benutzer die Shell wieder, bekommt *interact()* dies mit, kehrt zurück, und *telnet.pl* beendet sich.

Freilich sollte man keine Paßwörter in öffentlich zugänglichen Skripten herumliegen lassen - *chmod 700 telnet.pl* stellt wenigstens sicher, daß niemand außer dem Eigentümer und dem Super-User den Inhalt lesen kann. Für mehr Sicherheit bot der Februar-Artikel eine Sandkasten-Shell an.

Die automatischen Kontaktaufnahme beschränkt sich

■ LISTING 1: *telnet.pl*

```
1 #!/usr/bin/perl -w
2
3 use Expect;
4
5 my $timeout = 10;          # Grundsätzlich 10 sec
6                             # auf Antwort warten
7 my $host      = „remotehost“;
8 my $host      = „localhost“;
9 my $passwd    = „super-geheim“;
10
11                             # Telnet starten
12 $telnet = Expect->spawn(„/bin/telnet“, $host);
13
14                             # Login
15 $r = $telnet->expect($timeout, 'login');
16 die „No 'login' prompt“ unless defined $r;
17 $telnet->send_slow(0, „$user\n“);
18
19                             # Paßwort
20 $r = $telnet->expect($timeout, -re => '[Pp]assword');
21 die „No 'password' prompt“ unless defined $r;
22 $telnet->send_slow(0, „$passwd\n“);
23
24                             # Unix-Prompt
25 $r = $telnet->expect($timeout, '\$');
26 die „No unix prompt (\$)“ unless defined $r;
27
28 $telnet->interact();        # In interaktiven
29                             # Modus schalten
```

ALPHA - Power für LINUX

Mainboard DIGITAL 164LX, 2MB Cache
Mainboard SAMSUNG 164UX-2/4, 2MB/4MB Cache
- Onboard Ultra-Wide-SCSI: SYMBIOS LOGIC 53C875
- Onboard 10/100 Mbs Ethernet: DIGITAL 21143 Controller
CPU ALPHA 21164, 533 MHz / 600 MHz

Individuell angepaßte Komplettsysteme nach Wunsch.
Installation und Konfiguration von LINUX Redhat 5.1 und
Windows NT.

Garantieverlängerung bis 48 Monate möglich.
(Behörden und öffentliche Einrichtungen bis 60 Monate.)

central service
Kapuzinerstr. 43
80469 München
Tel.: 089/202 395 09
Fax: 089/202 395 18

Enterprise Solution Server

ab 1.980,- DM
zzgl. MwSt.

Vertrieb
Bestellwesen
Auftragsabwicklung
Finanzbuchhaltung
Materialwirtschaft
Kostenrechnung

JAVA EURO
JAHR 2000 LINUX
Entwicklungsumgebung
WEB-Browser
Reportgenerator
SQL-Datenbank
Offene Systeme
GLOBAL COMMUNICATIONS
SYSTEMS
19.-23. OKTOBER
1999
Tel.: 0911/6003361
Fax: 0911/6003363
email: info@e2r-ixware.de
http: //www.e2r-ixware.de

TEAMWORK DEVELOPMENT

WIPEOUT

INTEGRATED SOFTWARE DEVELOPMENT

Java · C++ · C · Eiffel · Fortran

Linux · FreeBSD · Solaris · HP-UX

Version Control · Debugger
Symbol Retriever · Class Browser
Make Tool · Text Editor
Integrated Help System

WipeOut Standard : free
WipeOut Pro : DM 249 (Single User)

www.softwarebuero.de

softwarebuero m&b · Weststrasse 9 · 04425 Taucha · Germany
PREISÄNDERUNGEN VORBEHALTEN.
ALLE RECHTE AN FIRMENZEICHEN UND PRODUKTNAMEN LIEGEN BEI IHREN JEWEILIGEN EIGENTÜMERN.

wie gesagt nicht auf *telnet*-Programm - auch *ftp* bietet sich an (auch wenn *Net::FTP* schöner ist) und auch alle möglichen Prompt-gesteuerten Applikationen. (siehe Listing 1)

htpasswd austricksen

Eine weitere Anwendung zeigt Listing 2 (*htp.pl*): Um Programme im Batch-Betrieb anzusteuern, die eine Paßworteingabe verlangen, reicht ein Here-Dokument nicht aus:

```
htpasswd htpasswd.dat user <<EOT
pass
pass
EOT
```

ruft zwar das der Apache-Distribution beiliegende User-Paßwort-Programm auf, dieses nimmt aber die zwei per Standardeingabe hereingereichten Paßwörter gemeinerweise nicht entgegen, sondern wartet beharrlich mit

```
Adding user user
New password:
```

auf eine Eingabe, die tatsächlich vom Terminal (*tty*) kommt. Mit dem *Expect*-Modul hingegen, das mit Pseudo-*ttys* herumhantiert, ist das alles kein Problem: Listing *htp.pl* öffnet in Zeile 16 einen Kanal zum *htpasswd*-Programm, und wartet in Zeile 20 auf einen von zwei Strings: „*Changing*“ oder „*New password:*“. Erhält die *expect*-Methode nämlich mehr als einen String als zu erwartetes Muster mitgeliefert, kehrt *expect* schon dann zurück, falls nur einer der Strings in der Ausgabe des zu überwachenden Programms erscheint. Der Rückgabewert der Methode (in skalarem Kontext) zeigt dann an, welches der angegebenen Muster gefunden wurde: *1* bedeutet, daß das erste Suchmuster zutraf, *2* das zweite usw. *htpasswd* zeigt für den Fall, daß der angegebene User in der Paßwortdatei schon existiert, „*Changing password for user X*“ an, bevor der Paßwort-Prompt kommt. Während *htpasswd* in einem solchen Fall einfach eine Paßwortänderung eines bestehenden Benutzers erwartet, hat sich *htp.pl* in den Kopf gesetzt, das Skript abubrechen, falls der Benutzer schon existiert. Hierzu fängt es die Meldung ab und terminiert mit einer *die*-Meldung. Ist der Rückgabewert der *expect*-Methode *undef*, wurde bis zum Timeout keines der Muster gefunden, ist er *1*, war's die *Changing*-Meldung, ist er *2*, wartet *htpasswd* bereits auf das Paßwort

■ Listing 2: htp.pl

```
1 #!/usr/bin/perl -w
2
3 use Expect;
4
5 $username      = „user“;    # Neuer user
6 $passwd        = „pass“;    # ... und sein passwd
7
8 $passwdfile    = „htpasswd.dat“;
9 $htpasswd_prog = „./htpasswd“;
10
11 my $timeout = 10;           # Grundsätzlich 10 sec
12                               # auf Antwort warten
13
14                               # htpasswd-Programm
15                               # starten
16 $robot = Expect->spawn($htpasswd_prog,
17                       $passwdfile, $username);
18
19                               # Schon vorhanden?
20 $pattern = $robot->expect($timeout,
21                          „Changing“, „New password:“);
22
23 if(!defined $pattern) {
24     die „New password: not found“;
25 } elsif($pattern == 1) {
26     die „User already there“;
27 }
28                               # Erstes Paßwort
29 $robot->send_slow(0, „$passwd\n“);
30
31                               # Paßwort-Bestätigung
32 $robot->expect($timeout,
33               „Re-type new password:“) ||
34     die „Pattern Re-type new password: not found“;
35 $robot->send_slow(0, „$passwd\n“);
36
37                               # Bis zum Programmende warten
38 $robot->expect(undef);
```

```
# Schon vorhanden?
$pattern = $robot->expect($timeout, „New password:“);

die „New password: not found“ unless defined $pattern;

die „User already there“ if
$robot->exp_before() =~ /Changing password/;
```

eines neuen Benutzers. Als Alternative ginge auch Die Methoden *exp_before*, *exp_match* und *exp_after* auf ein *Expect*-Objekt zeigen nach einem Treffer jeweils an, was vor dem Match kam, welche Zeichenkette als Muster erkannt wurde, und was danach noch folgte. Findet *Expect* also die Zeichenkette „*New Password:*“, sieht es nach, ob vor dem Match im *Expect*-Akkumulator *Changing password* steht - falls ja, gab es den Benutzer schon und das Skript bricht ab. Zeile 38 wartet noch, bis sich das *htpasswd*-Programm verabschiedet, fehlt diese Zeile, würgt das sofort terminierende *htp.pl* das laufende *htpasswd*-Programm rücksichtslos ab und das Ergebnis ist eventuell ein nicht oder nur unvollständig geschriebener Record in der Paßwortdatei. (siehe Listing 2)

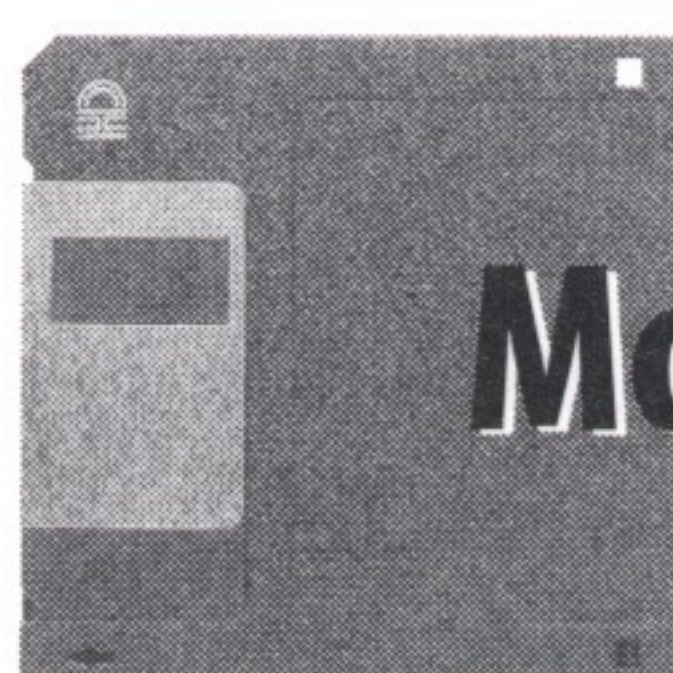
Debugging

Funktioniert etwas nicht, und es wäre hilfreich, wenn *Expect* erzählen würde, welche Muster es gefunden

Autorenliste

Andreas Bohne
Marco Budde
Robert Gehring
Patricia Jung
Cornelia Karl
Matthias Kranz
Bernhard Kuhn
Andreas Lindenblatt
Elke Moritz
Günther Röhrich
Oliver Schade
Michael Schilli
Martin Schulze
Tom Schwaller
Chris Weber
Michael Weyrauch

Service für DeLUG-Mitglieder



Monatsdiskette

Krypto-Info:

DFN-PCA Schlüssel: 2048/35DBF565 1997/04/16 DFN-PCA,
CERTIFICATION ONLY KEY (Low-Level: 1997-1998) <not-for-mail>
Key fingerprint = 09 7C 09 19 D3 C3 86 DC 7A 30 15 11 12 95 8D E3

IN-Root-CA Schlüssel: 2048/19980101 SIGN EXPIRE:1999-12-31
Key fingerprint = B3 06 9A 8D 38 04 3C 75 41 32 EE DC 8B 7D 61 0D
<http://www.in-ca.individual.net/>
Hashwert des Ewigen Logfiles vom 28.08.1998 22:02:31
sha1=2b24118b8c4dc8502c660e66ef601abcf4a8f105
<https://www.iks-jena.de/mitarb/lutz/logfile/>

Inserentenverzeichnis

American Magatrends GmbH	Seite 15
bdata Systems GmbH	Seite 35
Bee Baastrup: EDV-Entwicklungs GmbH	Seite 2
Million Bytes (Borkner)	Seite 35
Central Service	Seite 75
Computerbücher Obelisk	Seite 35
concept asa	Seite 79
Delix Computer GmbH	Seite 27
Exept Software AG	Seite 13
IOS-Gesellschaft für inovative Softwareentwicklung mbH	Seite 13
ixsoft Softwareentwicklung und -vertrieb	Seite 65
Lachner Universitätsbuchhandlung	Seite 65
Llynch Meta Medien GmbH	Seite 80
s2r-ixware Gbr	Seite 75
science + computing GmbH	Seite 23
Softwarebüro m&b	Seite 75
Solution-The Computer People	Seite 25
Sphinx Computer Vertriebs GmbH	Seite 9
SW Datentechnik	Seite 57
International Thomson Publishing	Seite 11
Linuxland	Seite 17,19,53
Best of Linux-Magazin	Seite 21

IMPRESSUM

Informationen, Berichte und Neuigkeiten aus der Linux-Welt

Herausgeber: Rudolf Strobl

Redaktion:

Email: redaktion@linux-magazin.de

Post: **Linux - Magazin Verlag**
Stefan-George-Ring 23
81929 München

Tel.: 089/99315-310

Fax: 089/99315-329

WWW: <http://www.linux-magazin.de>

ISSN: 1432 - 640 X

Chefredakteur: Tom Schwaller,
tom.schwaller@linux-magazin.de (ts)

Coverdesign: Tilo Wondollek, TW-Design

Layout : M & S Art&Work

Mitarbeiter: siehe Autorenliste

Vertriebsmarketing: info@linux-magazin.de

Aboverwaltung: Monika Scheck

Tel: 089/99315-310

Email: info@linux-magazin.de

Anzeigen: Agentur Jaser

Plattenweg 4

86850 Fischach

Tel: 08204 90040

Fax: 08204 90050

Email: pjaser@t-online.de

Anzeigenpreise: Es gilt die Anzeigenpreisliste vom 01.12.1997

Das Linux - Magazin erscheint monatlich.

Einzelheft: **DM 9,80**

Preis des Jahresabonnements:

Incl. DeLUG-Mitgliedschaft **DM 180,-**

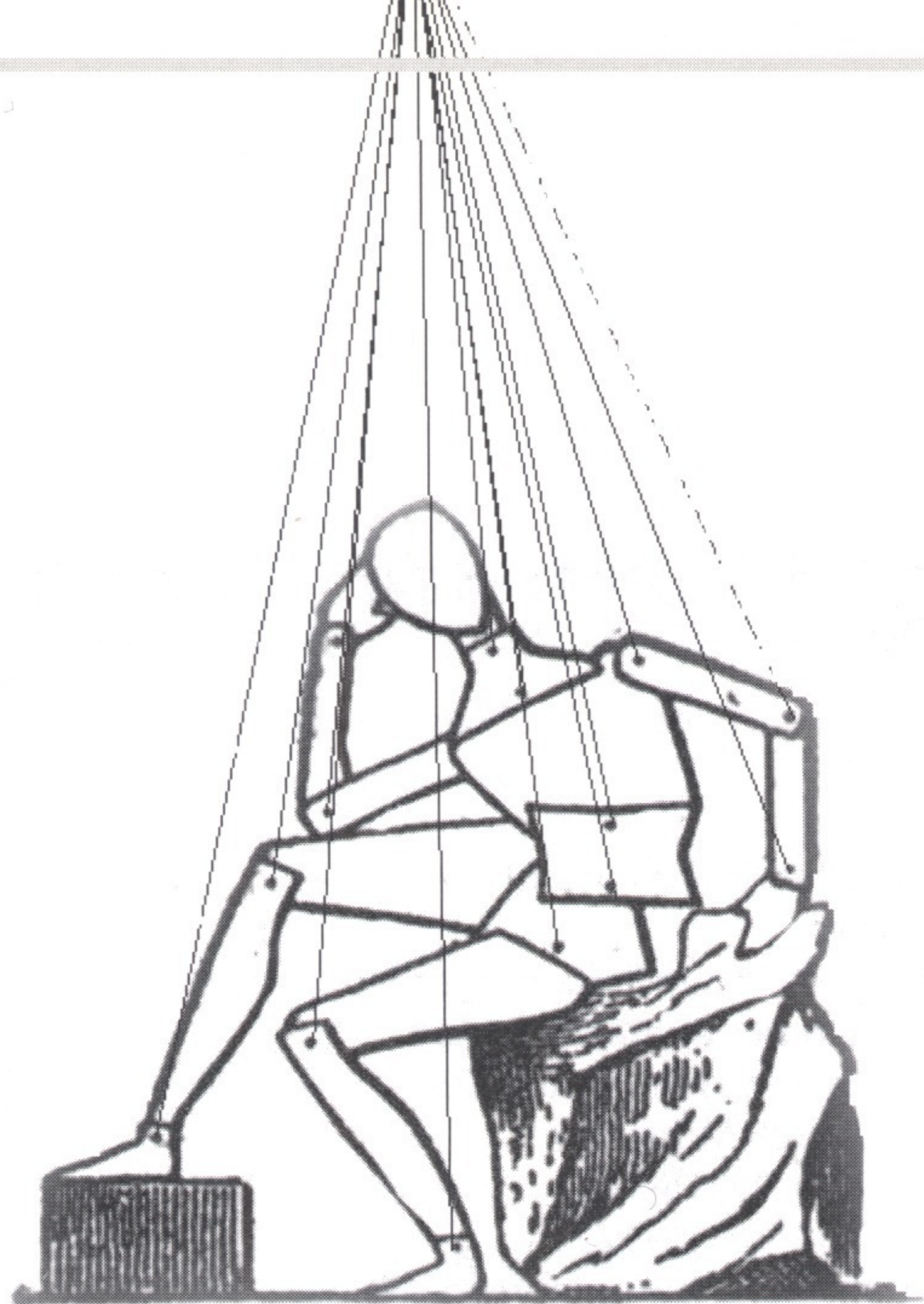
Ohne Mitgliedschaft **DM 103,-**

(Ausland: 115,-)

**Für Schüler und Studenten 20% Ermäßigung
(Vorlage Schüler- oder Studentenausweis)**

Der Begriff **Unix** wird in dieser Schreibweise als generelle Bezeichnung für die Unix-ähnlichen Betriebssysteme verschiedener Hersteller, zum Beispiel Eurix (Comfood), Ultrix (Digital Equipment), HP/UX (Hewlett-Packard) oder Sinix (Siemens) benutzt, nicht als die Bezeichnung für das Trademark von X/Open. Eine Haftung für die Richtigkeit der Veröffentlichung kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Honorierte Arbeiten gehen in das Verfügungsrecht des Verlags über. Nachdruck nur mit schriftlicher Genehmigung des Verlags. Mit Übergabe der Manuskripte und Bilder an die Redaktion erteilt der Verfasser dem Verlag die Exklusivrechte zur Veröffentlichung. Für unverlangt eingesandte Manuskripte kann keine Haftung übernommen werden. Sämtliche Veröffentlichungen im Linux-Magazin erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Copyright © 1994, 1995, 1996, 1997, 1998
Linux-Magazin Verlag



hat und was darauf als Antwort folgte, läßt sich ein *Expect*-Objekt einfach in den Debug-Modus schalten:

```
$exp->debug(1); # Debug an
$exp->debug(2); # Gesprächiger Debug
$exp->debug(0); # Debug wieder ausschalten
```

Dokumentation

Die vollständige Dokumentation zu *Expect* kommt — wie gehabt — nach der Installation des Moduls mit *perldoc Expect* zum Vorschein. Fröhliches Automatisieren!

Installation

Das *Expect*-Modul benötigt noch zwei TTY-Module, so daß insgesamt drei Distributionen her müssen (in dieser Reihenfolge):

```
AUSCHUTZ/IO-Stty-0.02.tar.gz
GBARR/IO-Tty-0.02.tar.gz
AUSCHUTZ/Expect.pm-1.07.tar.gz
```

Das *IO::Tty*-Modul führt bei der Installation durch ein längliches Konfigurationsskript, das an eine Perl-Installation erinnert - blindes Hämmern auf die Return-Taste übernimmt die Default-Werte, die unter Linux passen. Der Rest geht nahtlos.

DER AUTOR

SMichael Schilli arbeitet als Web-Engineer für America Online Inc., San Mateo. Er ist Autor des 1998 bei Addison-Wesley erschienenen Buches „GoTo Perl 5“ und unter *michael@perlmeister.com* oder *http://perlmeister.com* zu erreichen.

LINUX USER GROUPS

Im folgenden die Liste der uns bekannten Linux User Groups im Deutschsprachigen Raum in Kurzfassung. Änderungen und Updates bitte der Redaktion (*redaktion@linux-magazin.de*) mitteilen (Name, Beschreibung, Treffpunkt, Adresse, Ansprechpartner, Homepage, Email, Tel, Fax, Mitgliederzahl,...).

Aachener Linux User Group (ALUG): <http://aachen.heimat.de/alug/>
 Linux User Group Allgäu (LUGAL): ???
 Linux User Group Augsburg (LUGA): <http://bsmux.rz.uni-augsburg.de/luga>
 Linux User Group Austria: <http://www.luga.or.at>
 Bamberger Linux/Unix-Stammtisch (BLUES): ???
 Berliner Linux User Group (BeLUG): <http://www.belug.org>
 Linux User Group Bochum (LugBo): <http://www.ping.de/sites/lecter/lug.html>
 Linux Anwendergruppe der Universität Bochum: <http://www.ruhr-uni-bochum.de/linux/>
 Linux User Group Bodensee (LLUGB): <http://llugb.amsee.de/>
 Bonner Linux User Group (BoLUG): <http://www.bonn.linux.de/>
 Linux User Group Braunschweig: <http://www.tu-bs.de/initiativen/LUG>
 Chemnitz Linux User Group (CLUG): <http://www.tu-chemnitz.de/linux/>
 Linux Users Group Darmstadt: <http://www.mathematik.tu-darmstadt.de/dalug/>
 Linux User Group Dortmund (LUGRUDDO): lugrudo@outerspace.de
 Linux Users Group Duisburg: <http://mti26.mti.uni-duisburg.de>
 Düsseldorf Linux User Group (DLUG): <http://www.hsp.de/~dlug/>
 Essener Linux-Stammtisch (ELiSta): <http://www.uni-essen.de/initiative/elista/>
 Freiburger Linux User Group (FLUG): <http://www.freiburg.linux.de>
 IN-Kompetent e.V. Fulda: <http://www.rhoen.de/>
 Linux User Group Giessen (LUGG): <http://lugg.tg.fh-giessen.de>
 Linux User Group Kassel (LUGK): jochen.hein@delphi.central.de
 Linux User Group Krefeld (LUG-KR): <http://www.OnLinux.de>
 Unix-Gruppe der Hamburger MH e.V.: <http://swt-www.informatik.uni-hamburg.de/~1willamo/hmh.html>
 Linux User Group Hannover (LUGH): <http://www.han.de/lugh/>
 Projektgruppe Linux der UNIX-AG Kaiserslautern: <http://www.unix-ag.uni-kl.de/~linux/>
 Karlsruher Linux User Group (KaLUG): <http://www.karlsruhe.linux.de>
 Linux-Workshop Köln (LiWoK): <http://www.uni-koeln.de/themen/linux/>
 Leipziger Linux-Stammtisch: <http://rzaix340.rz.uni-leipzig.de/~linux/>
 Lüneburger Unix Stammtisch (LUSt): morningdew2@geocities.com
 Linux-User treffen in Mörs : ???
 Münchner Linux Usergroup (MLUG): <http://www.muc.de/~cygnus/mlug/mlug.html>
 Linux Stammtisch Münster (MueSLi): <http://hottemax.uni-muenster.de/~grover/muesli.html>
 Linux User Group Nürnberg (LUGNü): <http://www.linux.noris.de/>
 Oldenburger Linux Stammtisch (OLiS): <http://www.infodrom.north.de/Linux/Stammtisch/>
 Linux User Group Osnabrück (LUGO): <http://www.lugo.de>
 Linux User Group Paderborn (LUG-PB): <http://www.bowle.de/~lug-pb/>
 Linux User Group Saar (LUGSaar): <http://www.lugs.linux.de/>
 UNIX-AG Siegen: <http://www.si.unix-ag.org>
 Linux User Group Stormarn: <http://www.geocities.com/SiliconValley/Bay/7117/>
 Linux User Group Switzerland: <http://www.lugs.ch>
 Linux User Group Traunstein (LugTra): <http://www.spotlight.de/lug/>
 Linux User Group Trier (TriLUG): <http://www.trilug.fh-trier.de/>
 Linux User Group Tübingen (LUGTü): <http://www.tuebingen.linux.de>
 Linux User Group TU Wien (LLL): lll@radawana.cg.tuwien.ac.at
 Linux User Group Wolfsburg (LUGWo): http://www.escape.de/user/laura/lug_wolfsburg.html
 Brainstorm Computer-Club Würzburg (BraCCWü): <http://www.wuerzburg.de/brainstorm/> ???

Autorenliste

Andreas Bohne
Marco Budde
Robert Gehring
Patricia Jung
Cornelia Karl
Matthias Kranz
Bernhard Kuhn
Andreas Lindenblatt
Elke Moritz
Günther Röhrich
Oliver Schade
Michael Schilli
Martin Schulze
Tom Schwaller
Chris Weber
Michael Weyrauch

Service für DeLUG-Mitglieder



Monatsdiskette

Krypto-Info:

DFN-PCA Schlüssel: 2048/35DBF565 1997/04/16 DFN-PCA,
CERTIFICATION ONLY KEY (Low-Level: 1997-1998) <not-for-mail>
Key fingerprint = 09 7C 09 19 D3 C3 86 DC 7A 30 15 11 12 95 8D E3

IN-Root-CA Schlüssel: 2048/19980101 SIGN EXPIRE:1999-12-31
Key fingerprint = B3 06 9A 8D 38 04 3C 75 41 32 EE DC 8B 7D 61 0D
<http://www.in-ca.individual.net/>
Hashwert des Ewigen Logfiles vom 28.08.1998 22:02:31
sha1=2b24118b8c4dc8502c660e66ef601abcf4a8f105
<https://www.iks-jena.de/mitarb/lutz/logfile/>

Inserentenverzeichnis

American Magatrends GmbH	Seite 15
bdata Systems GmbH	Seite 35
Bee Bastrup: EDV-Entwicklungs GmbH	Seite 2
Million Bytes (Borkner)	Seite 35
Central Service	Seite 75
Computerbücher Obelisk	Seite 35
concept asa	Seite 79
Delix Computer GmbH	Seite 27
Exept Software AG	Seite 13
IOS-Gesellschaft für inovative	Seite 13
Softwareentwicklung mbH	
ixsoft Softwareentwicklung	Seite 65
und -vertrieb	
Lachner Universitätsbuchhandlung	Seite 65
Llynch Meta Medien GmbH	Seite 80
s2r-ixware Gbr	Seite 75
science + computing GmbH	Seite 23
Softwarebüro m&b	Seite 75
Solution-The Computer People	Seite 25
Sphinx Computer Vertriebs GmbH	Seite 9
SW Datentechnik	Seite 57
International Thomson Publishing	Seite 11
Linuxland	Seite 17,19,53
Best of Linux-Magazin	Seite 21

IMPRESSUM

Informationen, Berichte und Neuigkeiten aus der Linux-Welt

Herausgeber: Rudolf Strobl

Redaktion:

Email: redaktion@linux-magazin.de

Post: **Linux - Magazin Verlag**
Stefan-George-Ring 23
81929 München

Tel.: 089/99315-310

Fax: 089/99315-329

WWW: <http://www.linux-magazin.de>

ISSN: 1432 - 640 X

Chefredakteur: Tom Schwaller,
tom.schwaller@linux-magazin.de (ts)

Coverdesign: Tilo Wondollek, TW-Design

Layout : M & S Art&Work

Mitarbeiter: siehe Autorenliste

Vertriebsmarketing: info@linux-magazin.de

Aboverwaltung: Monika Scheck

Tel: 089/99315-310

Email: info@linux-magazin.de

Anzeigen: Agentur Jaser

Plattenweg 4

86850 Fischach

Tel: 08204 90040

Fax: 08204 90050

Email: pjaser@t-online.de

Anzeigenpreise: Es gilt die Anzeigenpreisliste vom 01.12.1997

Das Linux - Magazin erscheint monatlich.

Einzelheft: **DM 9,80**

Preis des Jahresabonnements:

Incl. DeLUG-Mitgliedschaft **DM 180,-**

Ohne Mitgliedschaft **DM 103,-**

(Ausland: 115,-)

**Für Schüler und Studenten 20% Ermäßigung
(Vorlage Schüler- oder Studentenausweis)**

Der Begriff **Unix** wird in dieser Schreibweise als generelle Bezeichnung für die Unix-ähnlichen Betriebssysteme verschiedener Hersteller, zum Beispiel Eurix (Comfood), Ultrix (Digital Equipment), HP/UX (Hewlett-Packard) oder Sinix (Siemens) benutzt, nicht als die Bezeichnung für das Trademark von X/Open. Eine Haftung für die Richtigkeit der Veröffentlichung kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Honorierte Arbeiten gehen in das Verfügungsrecht des Verlags über. Nachdruck nur mit schriftlicher Genehmigung des Verlags. Mit Übergabe der Manuskripte und Bilder an die Redaktion erteilt der Verfasser dem Verlag die Exklusivrechte zur Veröffentlichung. Für unverlangt eingesandte Manuskripte kann keine Haftung übernommen werden. Sämtliche Veröffentlichungen im Linux-Magazin erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Copyright © 1994, 1995, 1996, 1997, 1998
Linux-Magazin Verlag