

6/98

LINUX
MAGAZIN

Das einzige deutschsprachige Linux-Magazin

LINUX

MAGAZIN

ISSN 1432 - 640 X

DM 9,80

ÖS 75,-

SFR 9,80

6/98

Linux News

**Neues aus dem
Internet**

Netzwerke

**Virtual Network
Computing**

WWW

**Die Website
META Language**

Grafik

Blender 3D

Ohren auf !

**Zeitgesteuerte
Audioaufzeichnung
unter Linux**



www.linux-magazin.de

Perl Snapshot

Aus der CGI-Trickkiste IV

von Michael Schilli

In dieser letzten Folge über CGI-Programmierung mit Perl wird erklärt wie man mit Non-Parsed-Header-Skripts ungepufferte Ausgaben aber auch Server-Pushs realisiert. Abschließend werden Client-Pulls, Umleitungen und Expire-Mechanismen besprochen.

Den Ausgaben normaler CGI-Skripts schickt der Web-Server immer einen Header-Leporello mit HTTP-Protokoll-Informationen voraus. Ein absolut minimales CGI-Skript wie

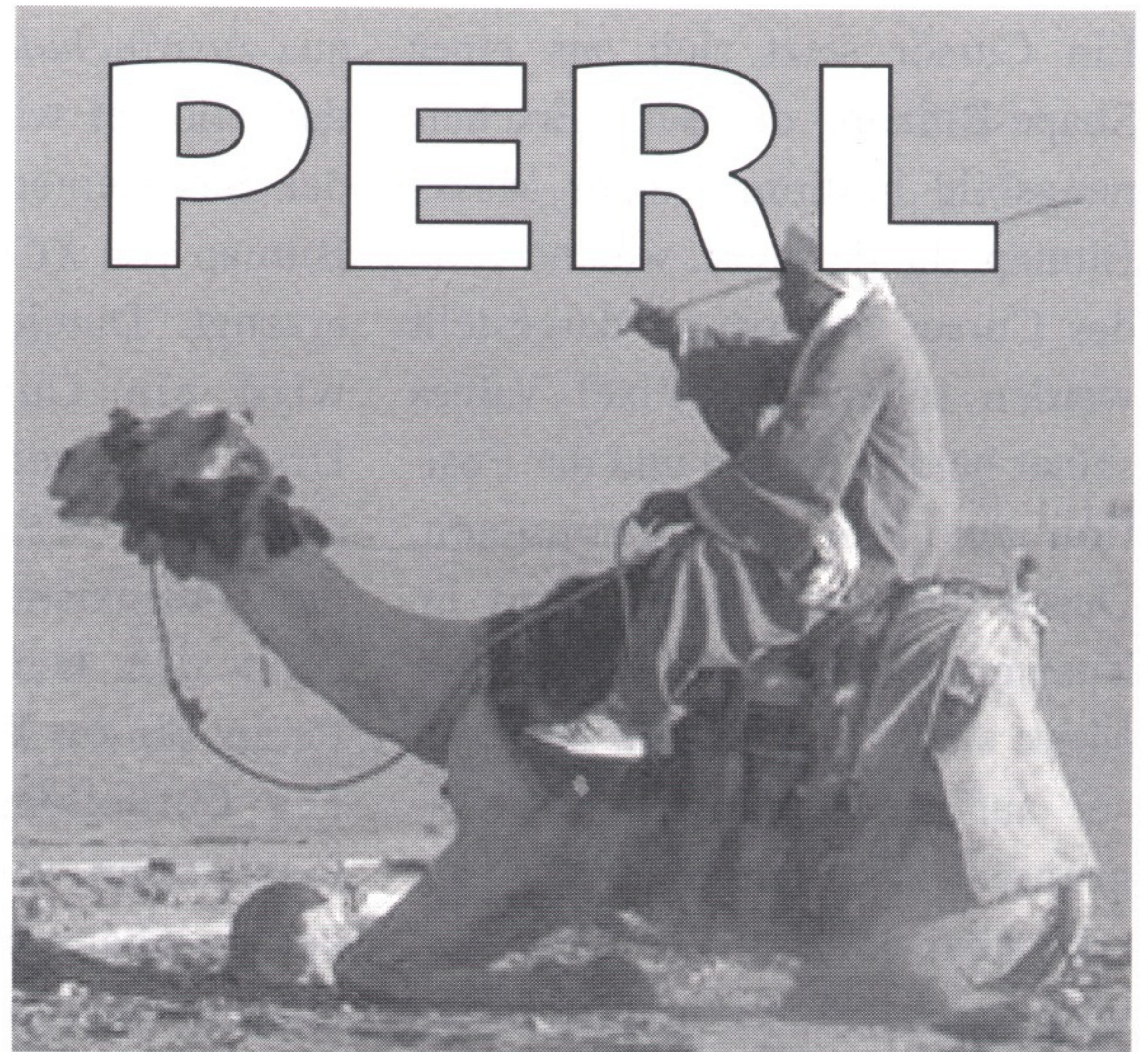
```
#!/usr/bin/perl -w
print „Content-Type: text/html\r\n\r\n“;
print „Hello, CGI!\n“;
```

erzeugt zwar nur eine minimale Ausgabe, schickt aber an den Browser etwa mit

```
HTTP/1.0 200 OK
Connection: close
Date: Sat, 04 Apr 1998 05:51:35 GMT
Server: Apache/1.2.4 mod_perl/1.05
Content-Type: text/html
Client-Date: Sat, 04 Apr 1998 05:51:37 GMT
```

Hello, CGI!

eine Statusmeldung (*HTTP/1.0 200 OK*) und — zusätzlich zum explizit angegebenen *Content-type* — eine Reihe von informativen Headern, die die Art der Verbindung, den Servertyp und das Datum auf Client- und Serverseite festlegen. Möchte man auf diese Angaben Einfluß nehmen, muß das Skript im NPH- (Non-Parsed-Header)-Modus laufen, in welchem es explizit auf diese Leistung des Servers verzichtet und



alles selbst in die Hand nimmt — ohne Netz und doppelten Boden. Fängt der Name der Skript-Datei mit *nph-...* an, verzichtet der Server (z. B. der Apache) darauf, seine eigenen Header vor die eigentliche Ausgabe eines Skripts zu knallen oder die Ausgabe zu puffern. Vielmehr gelangen alle Daten, sobald das Skript sie ausgibt, ungefiltert zum Browser.

Hieraus ergeben sich interessante Anwendungen: Statt die Seite im Browser auf einen Schlag aufzubauen, kann man die Daten schrittweise hineinladen. Das dynamische Fortschreiben der Daten vermittelt dem Brower-Benutzer bei langlaufenden CGI-Skripts den Eindruck, daß es vorwärts geht und nicht etwa der Server hängt oder das Netz dicht ist.

■ Listing 1: nph-append.pl

```
#!/usr/bin/perl -w

use CGI qw/:standard/; # header()
exportieren

$| = 1; # Ausgabe
entpuffern

print header(-nph => 1); # NPH-Header
ausgeben

print (h1(„2“)); # 2!
sleep(1);

print (h1(„1“)); # 1!
sleep(1);

print (h1(„Boom!“)); # Bazong!
```

It's the fi-nal Countdown - *didodiedo!*

Als völlig sinnloses Beispiel sei in diesem Zusammenhang das Countdown-Skript *nph-append.pl* vorgestellt, das nichts anderes tut, als im Sekundentakt zuerst 2, dann 1, und dann *Boom!* anzuzeigen. Da weder der Server die Ausgabe puffert (nph-Skript) noch das Skript selbst (\$! enthält einen wahren Wert), füllt sich die im Browser angezeigte Seite schrittweise (siehe Listing 1).

nph-append.pl nutzt natürlich *Lincoln Steins* allgegenwärtiges CGI-Paket, das auch die Konstruktion von nph-Skripten unterstützt: Erhält die *header*-Funktion zusammen mit der *-nph*-Option einen wahren Wert überreicht, nimmt sie die Ausgabe der Status-Meldung zusammen mit allen wichtigen Headern in die Hand.

Dokumente schrittweise überladen

Beim sogenannten Server-Push erwartet der Browser vom Server nicht nur ein Dokument, sondern mehrere aufeinanderfolgende, von denen er jeweils nur das neueste darstellt und bei ankommendem Nachschub das bisher dargestellte einfach überlädt. Die Verbindung des Clients zum Server bleibt dabei solange erhalten, bis auch das letzte Dokument eingetrudelt ist — unter Umständen eine Herausforderung für einen stark beschäftigten Web-Server, also Vorsicht! Den Startschuß hierzu liefert im Response-Header der Eintrag *multipart/x-mixed-replace* für den *Content-type*. Dieser spezifiziert gleichzeitig noch einen String, der eindeutig den Übergang zwischen zwei Dokumenten festlegt. Er muß mit zwei Gedankenstrichen (—) beginnen, und so eindeutig sein, daß die Zeichenfolge nirgends im nachfolgenden Dokument vorkommt. Der letzte Trennstring des Multi-Dokuments erhält zusätzlich noch „—“ angehängt. Listing 2 zeigt die Implementierung, ins *cgi-bin*-Verzeichnis und über einen Browser aufgerufen, liefert das Skript zunächst die Anzeige 2, die nach einer Sekunde von 1 überladen wird, worauf wiederum nach einer Sekunde schließlich *Boom!* folgt.

Das gleiche Problem löst auch ein CGI-Skript, das den Zählerwert eines hereingereichten Query-Parameters anzeigt, diesen herunterzählt und sich selbst nach einer Sekunde wieder mit dem neuen Wert des Query-Parameters aufruft.

Findet der Browser am Beginn eines HTML-Dokuments die Sequenz

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 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://bolug.uni-bonn.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.swb.de/personal/runner/dalug.html>
Linux User Group Dortmund (LUGRUDO) lugrudo@outerspace.de
Linux User Group Duisburg ???
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://www.uni-giessen.de/~g270>
Linux User Group Kassel (LUGK) jochen.hein@delphi.central.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.rz.uni-koeln.de/themen/linux/>
Leipziger Linux-Stammtisch <http://rzaix340.rz.uni-leipzig.de/~linux/>
Lüneburger Unix Stammtisch (LUSst) morningdew2@geocities.com
Linux-Usertreffen 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 Tübingen (LUGTü) <http://www.linux.de/lug/tuebingen/>
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/> ???

```
<META HTTP-EQUIV="Refresh"
  CONTENT="1;
  URL=http://host/cgi-
bin/clientpull.pl?count=2">
```

veranlaßt ihn das, nach der im *CONTENT*-Feld eingestellten Zeitspanne (eine Sekunde) den ebenfalls dort spezifizierten URL anzufordern. Im Beispiel bezieht

■ Listing 2: *nph-sp.pl*

```
#!/usr/bin/perl -w

use CGI qw/:standard/;

$| = 1;

$separator = „DerEindeutigeTrenner“;

print header(-nph => 1,
# Top-Header
      -type => „multipart/x-mixed-
replace;boundary=$separator“
      );

print „-$separator\n“;
# Erster Trenner

print header(-type => 'text/html'),
„\n“; # Teil-Dokument
print h1(2), „\n“;
print „-$separator\n“;
# Trenner

sleep(1);

print header(-type => 'text/html'),
„\n“; # Teil-Dokument
print h1(1), „\n“;
print „-$separator\n“;
# Trenner

sleep(1);

print header(-type => 'text/html'),
„\n“; # Teil-Dokument
print h1(„Boom!“), „\n“;
print „-$separator\n“;
# End-Header

Client-Pull
```

sich der URL auf das CGI-Skript, das die HTML-Seite erzeugt hat, und sich so nach der GET-Methode mit dem Wert 2 für den Parameter *count* aufruft. Aufgabe des CGI-Skripts ist es dann, eine neue Seite zu generieren, die den oben dargestellten HTML-Tag mit einem um eins heruntergesetzten Zählerwert repliziert — und schon fängt der Countdown an zu laufen.

Zweitens besteht für den Server die Möglichkeit, in den Response-Header einen Refresh-Eintrag einzupacken. Er hat für die Nachladezeit *1s* und den URL des Skripts folgende Form:

```
Refresh: 1; URL=http://host/cgi-bin/client-
pull.pl?count=2
```

Das Skript in Listing 3 (*clientpull.pl*) implementiert das geforderte Verhalten. Solange der überlieferte Zählerwert noch größer als Null ist, packt es die Nachlade-Anweisung in den Response-Header, bevor es den aktuellen Zählerstand ausgibt. Die Environment-Variable *SCRIPT_NAME* enthält (gemäß Server-Standard) den URL-Pfad des ausgeführten Skripts. Erreicht der Zähler die Null, gibt *clientpull.pl* vor der letzten Ausgabe einen regulären Header aus und beendet so den Nachlade-Reigen.

Der Redirect-Mechanismus dient dazu, Benutzer beim Klick auf ein Dokument automatisch (und unsichtbar) auf einen anderen URL umzuleiten. Der Server spuckt hierzu lediglich den neuen URL mit- samt einiger Header-Informationen aus, und *zack!*

■ Listing 3: *clientpull.pl*

```
#!/usr/bin/perl -w

use CGI qw/:standard/;

$count = param('count'); #
CGI-Parameter abfragen
$count ||= 3; #
Parameter nicht gesetzt? Startwert.
$count--;

if($count) {
    print header(-Refresh => „1;
URL=$ENV{SCRIPT_NAME}?count=$count“);
    print h1($count);
} else {
    print header();
    print h1(„Boom!“);
}

Umleitung
```

■ Listing 4: *nph-redir.pl*

```
#!/usr/bin/perl -w

use CGI qw/:standard/;

print redirect(-nph => 1,
               -uri =>
               'http://other.host.com');
```

springt der Browser dorthin - ohne eine entsprechende Meldung anzuzeigen, nur die URL-Anzeige verändert sich. Verschiebt sich der Zugriffspfad einer Webseite oder zieht diese gar auf einen anderen Rechner um, lassen sich Benutzer, die noch den alten URL benutzen, zuverlässig umdirigieren.

Listing 4 (*nph-redir.pl*) zeigt, wie das mit *CGI.pm* funktioniert: Die Option *-redirect* der Header-Funktion bewirkt das gewünschte Verhalten.

Allerdings kostet diese Art der Umleitung relativ viel Strom: Der Perl-Interpreter muß hochfahren, das Skript parsen, *CGI.pm* laden — da können auf einem lahmen PC schon mal zwei Sekunden verstreichen. Thema einer der kommenden Folgen wird deswegen der Apache-Server mitsamt dem Modul *mod_perl* sein. Diese Kombination stellt eine Schnittstelle zur Verfügung, um derlei Tricks bei minimaler Serverbelastung zu realisieren.

Alterungskontrolle

Damit ein Browser z.B. ein stündlich modifiziertes Dokument nicht über Gebühr in seinem Cache hält und rechtzeitig die neueste Version holt, setzt man einfach serverseitig das Verfallsdatum des Dokuments auf die Erscheinungszeit der neuen Version.

Das CGI-Skript *expire.pl* in Listing 5 zeigt zu Testzwecken den Stundenanteil der aktuellen Uhrzeit an und teilt dem Client mit, daß das Dokument zum nächsten Stundenwechsel seine Gültigkeit verliert.

Hierzu rechnet *expire.pl* die verbleibenden Minuten bis zum Anfang einer neuen Stunde aus und schickt mittels der *-expires*-Option der *header*-Funktion einen Expires-Header, der besagt, daß der Browser das Dokument nach X Minuten nicht mehr aus dem Cache hervorschummeln darf sondern tatsächlich vom Server holen muß. Ist die aktuelle Uhrzeit auf dem Server etwa 07:18:05, fehlen bis zum Stundenwechsel noch etwa 42 Minuten und mit *-expires => „+42m“* (die Option versteht die gleiche Syntax wie das letzten Monat vorgestellte *Cookie-expire*) sehen die zwei

ausschlaggebenden Header folgendermaßen aus:

Date: Sat, 18 Apr 1998 07:18:05 GMT

Expires: Sat, 18 Apr 1998 08:00:05 GMT

Damit ist zwar keineswegs garantiert, daß der Browser das Dokument in seinem Cache hält (manche Browser cachen keine CGI-Skripts oder haben in den Options-Menüs die Cache-Strategie deaktiviert), doch falls er den Mechanismus unterstützt, führt er bei weiteren Aufrufen der URL *http://host/cgi-bin/expire.pl* zwischen 07:18:05 und 08:00:05 keinen Netzzugriff mehr aus, sondern zeigt die olle Cache-Kopie. Danach oder beim Druck auf den Reload-Button (bei manchen Browsern CTRL-Reload) holt er sich die neueste Version vom Server.

■ Listing 5: *expire.pl*

```
#!/usr/bin/perl -w

use CGI qw/:standard/;

my ($sec,$min,$hour,$mday,$mon,
    $year,$yday,$yday,$isdst) =
    localtime(time());

$minutes_to_expiration = 60 - $min;

print header(-expires =>
             „+${minutes_to_expiration}m“);

print h1(„Die volle Stunde ist:
         $hour“);
```

So, damit ist erstmal Schluß mit dem CGI-Schmarr'n! Das nächste Mal gibt's einen Clipping-Agenten, der Nachrichten aus allerlei Zeitungen zusammensucht! Da werd's spitz'n! Bleibt's g'sund miteinander'!

DER AUTOR

Michael Schilli arbeitet als Web-Engineer für AOL Productions, San Mateo. Er ist Autor von „Effektives Programmieren mit Perl 5“ und unter *mschilli1@aol.com* oder *http://mission.base.com/mschilli* zu erreichen.

Autorenliste

Falko Bräutigam
 Marcus Brinkmann
 Marco Budde
 Kalle Dalheimer
 Bernhard Kuhn
 Florian Kühnert
 Patrick Murmann
 Günther Röhrich
 Michael Schilli
 Ansgar Schmidt
 Tom Schwaller
 Carsten Wartmann

Service für DeLUG-Mitglieder



Aktuelle Software

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 27.04.1998 18:11:52
 sha1=5a539e177b13a09008e8bd0934ce2c574c270a7d
<https://www.iks-jena.de/mitarb/lutz/logfile/>

Inserentenverzeichnis

Bdata	Seite 11
BEE	Seite 79
Concept Asa	Seite 2
Delix Computer	Seite 69
ixsoft	Seite 37
LinuxLand	Seite 11, 28/29
Llynch Meta Medien	Seite 80
Netzmarkt	Seite 43
Obelisk Computerbücher	Seite 59
O'Reilly	Seite 67
Sphinx	Seite 11
Thomson	Seite 33
Wagner	Seite 59

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 19
 81929 München

Tel.: 089/99315-310

Fax: 089/99315-199

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

Anzeigen: Agentur Neumann

Kalvarienbergstraße 31

Postfach 1352

87503 Immenstadt

Tel: 08323 / 98100

Fax: 08323 / 963226

Email: ThomasNeumann@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