Web-Seiten als Programmoberflächen - CGI-Scripte selbst erstellt

Der unter Linux übliche WEB-Server Apache bietet die Möglichkeit eigene CGI-Scripte zu erstellen. Wie das funktioniert, soll dieser Artikel zeigen.

Was ist ein CGI-Script?

Jeder, der mit seinen Schülern schon einmal WEB-Seiten erstellt hat, kennt das dahinter liegende Prinzip. Die Seiten werden in HTML erstellt und anschließend auf einen WEB-Server im Internet übertragen. Gibt nun ein Surfer die Adresse des Dokuments in seinen Browser in der Art *http://server/pfad/datei.html* ein, schickt der Browser eine Anforderung an den WEB-Server *server* für die Datei *datei.html* die sich im Pfad *pfad* befinden soll. Falls es die gewünschte Datei auf dem Server gibt, wird diese vom WEB-Server an den Browser übertragen.
Bei diesem Verfahren wird die HTML-Datei genau so an den Browser gesendet, wie sie zuvor vom Autor erstellt wurde. Der Inhalt der WEB-Seite ist statisch. Selbst nach mehreren hundert Aufrufen ändert sich an ihrem Inhalt nichts.
Im Gegensatz zu diesem Verfahren wird bei dynamischen WEB-Seiten der HTML-Code erst bei der Anfrage durch ein auf dem Server ablaufendes Programm generiert. Ein typisches Beispiel hierfür sind die Suchmaschinen im Internet. Der Suchende übermittelt durch das Absenden eines ausgefüllten Formulars seine Suchanfrage an die Suchmaschine. Dort erfolgt eine entsprechende Datenbankabfrage, deren Ergebnis in Form einer WEB-Seite wieder an den Suchenden zurückgesandt wird.

Das *Common Gateway Interface* kurz *CGI* stellt nun genau diese Schnittstelle zur Datenübertragung zwischen dem Internet-Browser und dem Server-Programm dar. Auf der Serverseite werden sehr häufig Perl-Scripte für die Erzeugung der dynamischen WEB-Seiten eingesetzt - daher rührt auch die eigentlich falsche Bezeichnung CGI-Script. Generell kann das Server-Programm in jeder beliebigen Programmiersprache erstellt werden.

Hello World

Wie beim Erlernen einer neuen Programmiersprache üblich wollen wir mit einem *Hello World*-Programm beginnen. Da der Text *Hello World* nun aber sehr statisch ist, werden wir zusätzlich die aktuelle Systemzeit des Servers, die wir durch einen Aufruf des Unix-Befehls */bin/date* ermitteln, mit ausgeben.

In der zentralen Konfigurationsdatei */etc/httpd/httpd.conf* des Apache WEB-Servers (S.u.S.E. 6.4) legen die beiden folgenden Einträge die Pfade fest, in denen der WEB-Server nach HTML-Dateien und nach CGI-Scripten sucht.

- `DocumentRoot „/usr/local/httpd/htdocs”`
- `ScriptAlias /cgi-bin/ “/usr/local/httpd/cgi-bin/”`

Entsprechend dieser Konfiguration erstellen wir mit einem Editor unserer Wahl die Datei */usr/local/httpd/cgi-bin/hello.cgi*, mit folgendem Inhalt:

```
#!/usr/bin/perl
#
# /usr/local/httpd/cgi-bin/hello.cgi
#
# von H. Baeurle – letzte Aenderung: 02.11.2000

# Ueber den Content-Type wird dem Browser mitgeteilt um
# welche Art von Daten es sich im folgenden handelt,
# z.B. image/gif, image/jpg ...
print „Content-Type: text/html\n\n“;

# Die Ausgabe des UNIX-Befehls date wird in der Variablen
# serverzeit gespeichert.
# Achtung: Die richtige Hochkommas verwenden!
$serverzeit=‘/bin/date‘;

# Ab hier wird mit print-Befehlen die HTML-Ausgabe erzeugt.
print „<html><head>\n“;
print “<title>Hello world!</title>\n”;
```

```
print "</head><body>\n";
print "<h1>Hello world!</h1>\n";
print "Die Serverzeit ist: ",$serverzeit;
print „</body></html>\n";

# Programmende
```

Beim Erstellen der Datei müssen Sie natürlich schreibende Zugriffsrechte haben. Anschließend wird das Perl-Script mit:

```
chmod o+x hello.cgi
```

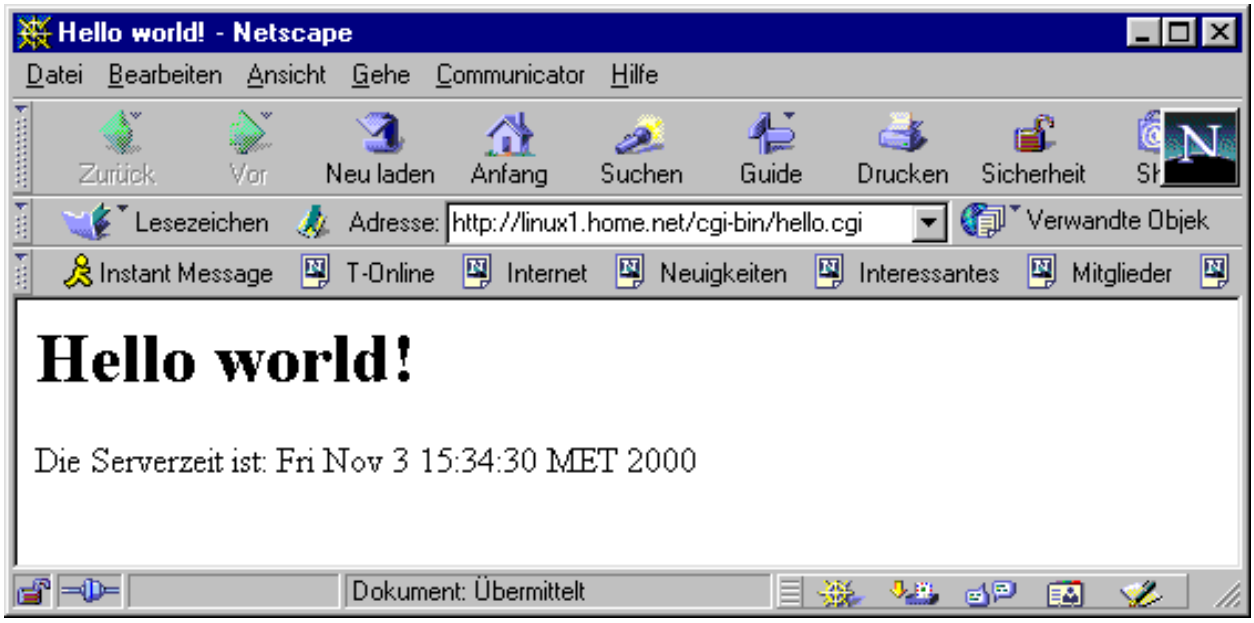
für alle (**o**ther + e**x**ecute) ausführbar gemacht. Ein *ls –l hello.cgi* sollte dies bei den Dateiattributen durch ein x an letzter Stelle anzeigen. Das Script kann jetzt direkt von der Kommandozeile aus über:

```
./hello.cgi
```

gestartet werden und führt zu der folgenden Ausgabe auf dem Bildschirm.

```
Content-Type: text/html

<html><head>
<title>Hello world!</title>
</head><body>
<h1>Hello world!</h1>
Die Serverzeit ist: Fri Nov 3 15:12:12 MET 2000
</body></html>
```

Erfolgt der Programmaufruf nicht lokal sondern durch eine Anforderung eines WEB-Browsers wird die Ausgabe des Programms einfach von der Standardausgabe an den Browser umgeleitet. Die Eingabe der Adresse *http://ihr-web-server/cgi-bin/hallo.cgi* führt dann zu folgendem Ergebnis:



Über *Ansicht*à *Seitenquelltext* kann der an den Browser übermittelte HTML-Code eingesehen werden.

Ein Formular

CGI-Scripte werden erst in Verbindung mit Formularen richtig interessant. Bei diesem Verfahren wird dem Surfer als Erstes eine normale HTML-Datei mit mehreren Formularfeldern zur Dateneingabe präsentiert. Sind alle Angaben gemacht, werden durch einen Klick auf den ebenfalls im Formular enthaltenen SUBMIT-Button alle Daten aus den Formularfeldern zum WEB-Server gesandt. Dort wird dann ein Programm z.B. ein CGI-Script gestartet, welches die übertragenen Daten verarbeitet und eine entsprechende Antwort generiert, die wiederum an den Surfer übertragen wird.

Typisches Beispiel für dieses Verfahren sind die im Internet zahlreich vorhandenen Suchmaschinen. Der Surfer gibt seine Suchanfrage in ein Eingabefeld auf der Startseite der Suchmaschine ein. Ein Klick auf *Suchen* überträgt die Suchanfrage an die Suchmaschine, die die Anfrage auswertet und als Ergebnis eine dynamisch generierte WEB-Seite zurückliefert.

Im folgenden Beispiel erfragt ein Formular die persönliche Anrede und den Namen des Surfers. Das dahinterliegende CGI-Script erzeugt aus diesen Angaben als Antwort eine HTML-Seite, die den Surfer persönlich begrüßt und ihm die Urzeit des Servers verrät.

Dazu erstellen wir zunächst die Datei */usr/local/httpd/htdocs/formular.html* mit folgendem Inhalt:

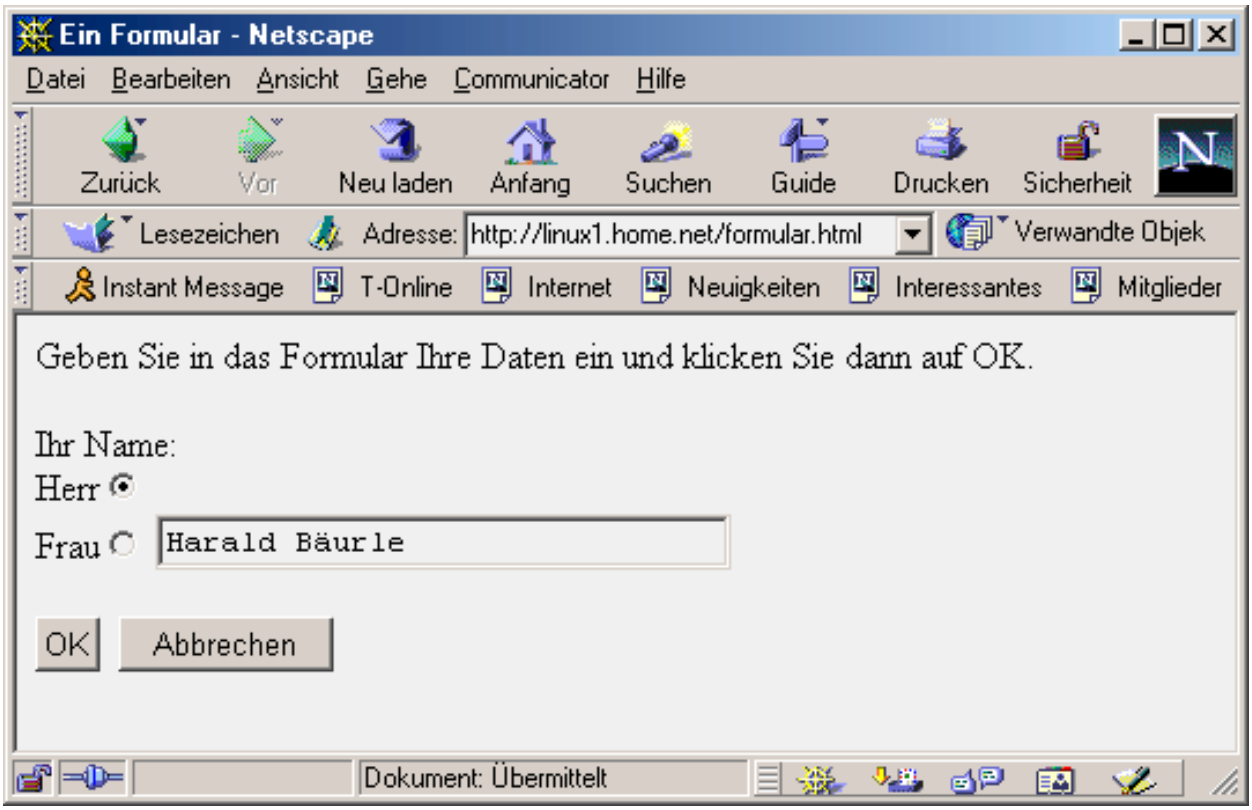
```
<html>
<head>
<title>Ein Formular</title>
</head>

<body>
Geben Sie in das Formular Ihre Daten ein und klicken Sie
dann auf OK.<p>
<form action="http://ihr-web-server/cgi-bin/form.cgi" method="get">
Ihr Name:<br>
Herr<input type="radio" name="anrede" value="Herr" checked><br>
Frau<input type="radio" name="anrede" value="Frau"<br>
<input type="text" size="30" name="name"><p>
<input type="submit" value="OK"> 
<input type="reset" value="Abbrechen">
</form>
</body>
</html>
```

Die Festlegung des Formulars erfolgt innerhalb der beiden Tags *<form></form>*. Dem öffnenden <form>-Tag wird über den Parameter *action="http://ihr-web-server/cgi-bin/form.cgi"* mitgeteilt an welches Programm die Eingaben aus dem Formular übertragen werden sollen. Mit dem Parameter *method* wird festgelegt, wie die Daten zum Server übertragen werden sollen. Zur Auswahl stehen *get* und *post*. Für das erste Beispiel verwenden wir die etwas weniger übliche Methode *get*, da sie die Formulareingaben für uns gut sichtbar an den CGI-Aufruf anhängt.
Das zugehörige CGI-Script /usr/local/httpd/cgi-bin/form.cgi sieht folgendermaßen aus:

```
#!/usr/bin/perl
#
# /usr/local/httpd/cgi-bin/form.cgi
#
# von H. Baeurle - letzte Aenderung: 02.11.2000

# Daten je nach gewaehlter Uebertragungsmethode in $eingabe
speichern
# Uebertragungsmethode GET
if($ENV{'REQUEST_METHOD'} eq 'GET')
{
  $eingabe = $ENV{'QUERY_STRING'};
}
# Uebertragungsmethode POST
else
{
  read(STDIN, $eingabe, $ENV{'CONTENT_LENGTH'});
}

# Die einzelnen Wertepaare trennen - Trennzeichen ist &
@paare = split(/&/, $eingabe);

# Jedes Wertepaar wird jetzt der Reihe nach abgearbeitet
foreach $paar (@paare)
{
  # Wertepaare selbst in Schluessel und Wert trennen -
Trennzeichen ist =
  ($schluessel, $wert) = split(/=/, $paar);

  # + Zeichen wieder in Leerzeichen umwandeln
  $wert =~ tr/+ / / ;

  # Hex-Codierung wieder in ASCII-Zeichen
  $wert =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack(„C“, hex($1))/eg;

  # Hash mit Daten erstellen
  $FORM{$schluessel} = $wert;

}

# Mime-Type
print „Content-Type: text/html\n\n“;

# Server-Zeit ermitteln
$serverzeit='/bin/date';

# Ab hier wird mit print-Befehlen die HTML-Ausgabe erzeugt.
print „<html><head>\n“;
```

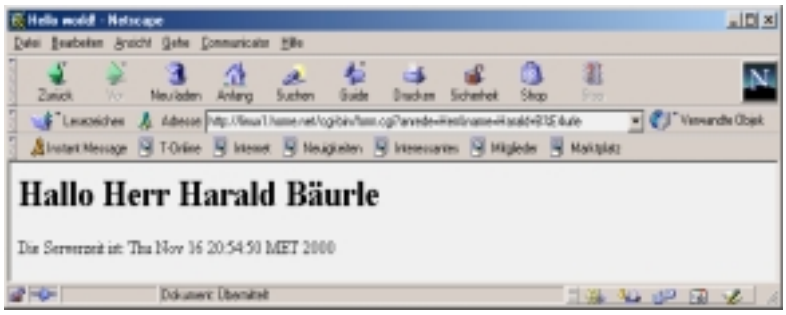
```
print "<title>Hello world!</title>\n";
print "</head><body>\n";
print "<h1>Hallo ";
print "$FORM{anrede} $FORM{name}";
print „</h1>\n“;
print „Die Serverzeit ist: „,$serverzeit;
print „</body></html>\n“;

# Programmende
```

Bevor wir uns den Programmcode von *form.cgi* etwas genauer ansehen testen wir zunächst einmal seine Funktion.



Der Klick auf den SUBMIT-Button *OK* führt zum Aufruf des CGI-Scriptes. Wie in der Befehlszeile des Browsers zu erkennen ist, werden die Formulareingaben bei der Methode *get* einfach an den CGI-Aufruf mit einem *?-Zeichen* angehängt. Dabei werden der Feldname und die im ausgefüllten Formular diesem Feld zugewiesene Eingabe als ein durch ein *=-Zeichen* getrenntes Wertepaar übermittelt. Die einzelnen Wertepaare werden wiederum durch ein *&-Zeichen* getrennt.

Zusätzlich werden für die Übertragung alle Leerzeichen durch *+-Zeichen* sowie alle Sonderzeichen durch Ihren Hexcode ersetzt. Im Beispiel wird aus *ä* ein *%E4*.



Die für die Übertragung vorgenommenen Änderungen müssen vom CGI-Script für die Auswertung rückgängig gemacht werden. Dazu wird in einem ersten Schritt entsprechend der gewählten Übertragungsmethode der im Environment abgelegte Eingabestring der Variablen *$eingabe* zugeordnet. Anschließend werden die durch das *&-Zeichen* getrennten Wertepaare im Array *@paare* abgelegt. In der folgenden *for each - Schleife* wird jedes der durch das *=-Zeichen* getrennten Wertepaare in Schlüssel und zugehörigen Wert aufgeteilt. Die im Anschluss folgenden zwei Befehle stellen dann in den gespeicherten Werten alle Leer- und Sonderzeichen wieder her. Jedes so bereinigte Wertepaar wird am Ende der Schleife im Hasch *$FORM* abgelegt. Ein Hasch ist nichts anderes als ein Array, bei dem der Zugriff über selbst festzulegende Schlüssel erfolgt. Nach der Schleife wird aus den aufbereiteten Daten die eigentliche HTML-Seite wie im ersten Beispiel mit print-Kommandos hergestellt.

Im Unterschied zur Sendemethode *get* werden bei *post* die Daten nicht an den CGI-Aufruf angehängt, sondern direkt gesendet. Außerdem gibt es keine Beschränkung bezüglich der zu versendenden Informationen. Ändern Sie die Methode in der Datei *formular.html* und probieren Sie es aus.

Noch ein Tipp: Das Konvertieren der Ein- und Ausgabe kann durch Einbinden des Modules *CGI* mit *use CGI;* wesentlich vereinfacht werden. Testen Sie selbst.

```
#!/usr/bin/perl
#
# /usr/local/httpd/cgi-bin/formcgi.cgi
#
# von H. Baeurle - letzte Aenderung: 02.11.2000

use CGI;

# Formular auswerten
$form = new CGI;
$anrede = $form->param('anrede');
$name = $form->param('name');

# Server-Zeit ermitteln
$serverzeit='/bin/date';

# Ausgabe erzeugen
print „Content-type: text/html\n\n“;
print <<„END_OF_TEXT“;
<html><head>
<title>Hello world!</title>
</head><body>
<h1>Hallo $anrede $name</h1>
Die Serverzeit ist: $serverzeit
</body></html>

END_OF_TEXT

# Programmende
```

Harald Bäurle •

E-Mail: HaraldBaeurle@t-online.de