

# Common Gateway Interface

**Dynamisch, Praktisch, Gut**

*Gerhard Poul*

Jeder, der heute mit dem Internet ernsthaft arbeiten will, kommt um das World Wide Web kaum herum. Egal ob man dabei einen Browser wie Lynx oder Mozilla einsetzt. Beide Browser können mit Formularen und ähnlichen interaktiven Inhalten umgehen.

Dynamisch erzeugte Web-Seiten scheinen heute schon fast wie ein Zwang im Internet zu sein. Jeder hat, auch wenn es noch so unnötig ist, auf seiner Homepage ein Guestbook und andere interaktive Elemente eingebaut.

Die sinnvollste Verwendung liegt, nach meiner bisherigen Erfahrung, aber im realisieren von Warenkorb-Systemen, dem ausfüllen von Formularen und dem integrieren von Datenbanken in die Web-Seiten.

Datenbanken sind bereits so verbreitet, dass sich der Aufwand zur Verwaltung der Homepage sehr gut durch ihre Verwendung minimieren lässt. Die Preisliste eines Computerhändlers muss man nicht mit table tags in HTML schreiben und sich dabei die Finger brechen. Es geht auch einfacher. Denn die Programme, die zwischen Datenbank und Web-Server stehen, erledigen die rasche Umsetzung der Inhalte nach HTML. Dadurch ist es auch möglich, eine sehr schnelle Änderung ohne Bearbeiten einer einzigen HTML Datei durchzuführen.

## Funktionsweise

Das Common Gateway Interface (CGI) ist grundsätzlich keine Programmiersprache sondern nur eine Schnittstellendefinition der Kommunikation zwischen Web-Server und dem Programm.

Programm ist hier deshalb so allgemein gehalten, da es sich sowohl um ein kompiliertes Binärprogramm, als auch um ein interpretiertes Skript handeln kann. Grundsätzlich sind viele Programmiersprachen dazu geeignet um CGI-Skripte zu schreiben. Hauptsächlich werden heute allerdings Perl und C/C++ verwendet.

An dieser Stelle beginnt wieder der ewige Krieg zwischen Perl und C Anhängern. Ich will hier deshalb nur die Vor/Nachteile bei der Verwendung einer interpretierten Sprache andeuten.

Interpretierte Sprachen haben hauptsächlich den Nachteil, dass sie eine viel geringere Performance aufweisen wie

kompilierte Programme. Dafür gewinnt man bei deren Verwendung ein paar Vorteile, wie etwa leichtere Wartbarkeit, da die Programme nicht nach jeder Änderung neu kompiliert werden müssen, sondern sofort in modifizierter Form zur Verfügung stehen. Dadurch entfallen auch die Probleme, die auftreten, wenn Web-Server und der Rechner des Entwicklers eine unterschiedliche Plattform verwenden.

## Ein Beispiel

Stellen wir uns nun einmal vor, ein Kunde der Firma xy würde sich die neueste Preisliste auf seinen Schirm holen. Was passiert dabei am Web-Server der Firma xy unter der Annahme, dass die Firma xy ein Datenbanksystem mit einem CGI-Skript verwendet?

Zuerst wird das CGI-Skript aufgerufen und dieses mit den Daten des Users versorgt. Es werden also Username, IP-Adresse, Ausgefüllte Formularfelder u.a. weitergegeben. Das Skript sucht sich nun die für sich interessanten Informationen heraus, erkennt in welcher Situation es aufgerufen wurde, holt die Daten aus der Datenbank, konvertiert diese Daten und gibt sie wieder an den Web-Server zurück. Dieser puffert die Daten und liefert sie nach dem Erfolgreichen beenden des Skripts an den Web-Browser des Benutzers zurück.

In diesem Szenario sind sehr komplexe Details enthalten die ich jetzt nach und nach erklären werde. Denn um die Abläufe zu verstehen muss man zuerst einige Grundsätzliche Tatsachen verstehen die beim erstellen einer Web-Site eine Rolle spielen.

Das Skript muss vor der Bearbeitung von Daten zuerst seine Aufgabe erkennen. Denn meistens werden CGI-Skripte zur Datenbankabfrage rekursiv benutzt. Das Programm gibt in seiner Ausgabe, die es liefert, also wieder Links auf sich selbst aus, die eine andere Aktion auslösen. Ein gutes Beispiel hierfür ist etwa ein "Alle Anzeigen" oder "Weiter"-Button auf einer dynamischen Web-Seite. Dabei muss sich das Skript selbst mitteilen was getan werden muss. Denn das Programm läuft ja nur sehr kurz. Daher kann er sich seine Statusdaten nicht in einer Datei oder im RAM speichern.

Diese Laufzeitdaten werden dann meistens entweder als Cookies oder als ver-

stecktes Formular übergeben damit das Skript so schnell wie möglich fortsetzen kann. Es muss aber auch darauf geachtet werden, dass das Skript auch keine Fehler produzieren darf wenn die Parameter durch fremde Einwirkungen modifiziert werden, da dies ein potentielles Sicherheitsrisiko darstellt.

Die Daten aus der Datenbank werden meistens mit einer Datenbankabfragesprache, genannt SQL, durchgeführt. Dadurch ist es möglich aus einer Web-Seite mit einer externen Datenbank zu kommunizieren.

Um die Rückgabe der Daten an den Web-Server zu realisieren werden die Daten einfach mit HTML-Tags versehen und nach STDOUT geschrieben. Dadurch werden diese vom Web-Server gepuffert, mit den richtigen Headern versehen und dann auf ihre mehr oder weniger lange Reise zum Web-Browser des Users geschickt.



*Apache Web-Server (Lars Elibrecht), ISBN 3-8266-0438-5, ATS 504,-, International Thomson Publishing, 2. aktualisierte und erweiterte Auflage*

Das erste deutschsprachige Buch zum meistverbreiteten Web-Server. Der Autor, Web-Administrator bei der Unix-AG der Universität-Gesamthochschule Siegen, zeigt mit seiner fundierten und ausführlichen Darstellung, wie sich die enormen Möglichkeiten des Apache sinnvoll ausschöpfen lassen. Im ergänzten Anhang erfährt der Administrator sämtliche Neuerungen der aktuellen Apache 1.3 Distribution.

## Aus dem Inhalt

- Modulbeschreibungen, Übersetzung & Installation

# Ins Internet mit Wählaccount

Franz Fiala, August Hörandl

- Beschreibung aller Konfigurationsanweisungen
- Anwendungskonfigurationen, Beispielkonfigurationen
- Common Gateway Interface, Server-Side Includes
- Content Negotiation, Image-Maps
- Suchsysteme und Web-Robots
- Sicherheitsaspekte, Logfile-Analyse
- Secure Web-Server: Apache-SSL, SSLeay

## CD-ROM Inhalt

- Apache 1.2 & 1.3 Distributionen für Unix-Systeme, OS/2, AmigaOS & Windows 95/NT
- Zahlreiche Zusatzmodule
- Beispielkonfigurationen, Tools, RFCs, FAQs u.v.m.

Dieser Artikel zeigte natürlich nur einen kleinen Auszug der Fähigkeiten von CGIs. Wenn ich Ihr Interesse geweckt haben sollte können Sie weitere Informationen auf den folgenden Web Seiten erfahren.

## Infos

[1] The Common Gateway Interface, <http://hoohoo.ncsa.uiuc.edu/docs/cgi/>

[2] Apache HTTP Server Project, <http://www.apache.org/>

*Wer Verzug seiner Projekte meldet, sollte über die Pünktlichkeit seines Gehalts nachdenken.*

*C.-R. Schulze-Oberländer, Vorstand Finanzen Barmag*

**AG**

Während die ASN-Statistik viele Schulen im Internet wäht, gibt es immer wieder Anfragen, wie man an entlegenen Standorten, zu einer Stadtleitungsverbindung kommen kann, oder wie man zumindest mit einem Wählaccount einen temporären Internet-Betrieb in einer Rechnerlabor herstellen kann.

Windows-NT und LINUX werden dazu am häufigsten benutzt.

## Windows NT

### Voraussetzungen

- Windows-NT-Server 4.0
- Proxy-Server 2.0
- Remote Access Server
- Windows-NT mit Ethernet-Karte an LAN gekoppelt
- Proxy-Server generiert über LAT (Lokal Adress-Table) für jeden Arbeitsplatz eine nicht-geroutete Adresse, 10.0.0.1, 10.0.0.2...
- RAS wird konfiguriert, indem die IP-Adresse nicht fest, sondern auf die vom Provider zugewiesene IP-Adresse eingestellt wird
- Der Verbindungsaufbau erfolgt entweder händisch über "Dial-Up-Networking" (englische Version des DFÜ-Netzwerks) oder automatisch während der Selektion eines Web-Inhalts durch einen Benutzer.

Details zur Konfiguration finden Sie in PCNEWS-54, Seite 70 oder als Webversion unter <http://pcnews.at/thi/adm/conf/~nt.htm>. (Achtung: Dort wird im Punkt "DialUp" auf das Problem einer Standleitungsverbindung eingegangen. Eine Wählverbindung macht sogar weniger Probleme, weil in allen Menüs eine Wählverbindung angenommen wird.)

Eine Anlaufstelle für NT-Probleme ist der Webserver der HBLA Biedermannsdorf:

<http://www.edu-helpdesk.ac.at/KnowledgeBase/knowledge.htm>

Aktuelle Informationen von Microsoft:

<http://www.microsoft.com/switzerland/de/products/default.asp>

## LINUX

Das geht (natürlich) auch mit Linux. Der Vorteil gegenüber NT ist die viel geringere Hardwarevoraussetzung: ein alter 486er reicht - und selbst dann kann er als WWW- und Mail-Server für interne Zwecke bzw. zum Offline Lesen verwendet werden geringe (Online) Kosten.

### Weitere Hinweise siehe

- ADIM Band 81 ;-)
- PCNEWS 59, Seite 72
- <http://elina.htlwl.ac.at/~hoerandl/firewall.txt>

### Vorgehen

- Linux installieren z. B. Suse 6.0
- nicht vergessen: squid als Proxy, eventuell sendmail, apache, inn etc.
- eventuell bootp/dhcp server
- Modem einrichten, d. h. Verbindung zum Provider testen

Dial-On-Demand würde ich nicht verwenden - Probleme z. B. bei Nameserverzugriff d. h. es wird unnötigerweise oft die Verbindung aufgebaut (siehe diald howto) - besser ist ein einfaches Script mit entsprechenden Zugriffsrechten und einem "idle" Timeout, es gibt aber auch Lösungen, die die österreichische Tarifstruktur d. h. Tarifimpulszeiten kennen.

Die Rechner im Netz bekommen einige der freien Adressen (z. B: 10.x.x.x) entweder fix zugeteilt oder via bootp/dhcp.

Als Alternative zu squid ist wwoffle zu überlegen: damit kann man auch "offline" surfen, d.h., es werden dann die Seiten aus dem Cache gezeigt.

Mit fetchmail kann man Post aus externen Mailaccounts (z.B. bei schule.at) auf lokale Benutzer übertragen. Damit können die Lehrer jederzeit Mail lesen bzw. schicken, die Übertragung erfolgt automatisch beim nächsten Mal online.

### Siehe auch

<http://www.linux.at/>  
<http://www.luga.or.at/>  
<http://www.pcnews.at/poul/linux.html>

Eine ziemlich vollständige Anleitung gibt es unter

<http://www.dbg.rtt.bw.schule.de/lehrer/ritters/info/linux/linux.htm>

1 Es werden ca. 2000 Schulen als Internet-Schulen angeführt (siehe <http://www.bmuk.gv.at/>), dabei handelt es sich aber größtenteils um reine E-Mail-Accounts. Weiters gibt es ca. 800 Schulen mit einer Homepage. Aber auch diese Zahl sagt nicht viel, denn die Homepages werden in den meisten Fällen auf externen Servern gehostet. Ob eine Verbindung Schul-LAN <-> Internet besteht, kann aus den Zahlen nicht abgelesen werden.