



## 7363 - Web-basierte Anwendungen: **Übung 04, Meilenstein 2**

Umgang mit dynamischen Webseiten:  
CGI-Skripte, FCGI-Varianten  
Server Side Includes (SSI)



## Ziele



- **Teilziele**
  - Fertigstellung einer Entwicklungsumgebung für dynamische Web-Seiten.
  - Vertrautwerden mit dem CGI bzw. seiner Alternative
- **Übungen**
  - Performance-Vergleich: CGI vs. Alternativen (FCGI)
  - Umgang mit Formularen
  - Umgang mit SSI und Cookies
- **Übergeordnetes Ziel**
  - Einübung von Grundlagen-Fertigkeiten für den späteren Einsatz von Web Services
  - Vorbereitungen für die Projektarbeit



# Einbau des Zusatzmoduls zu FastCGI



## Zusatzmodul FastCGI



- Quelle
  - \$srcdir/mod\_fastcgi-2.4.2.tar.gz.
- Installation (z.Z. nur Fragmente)
  - Quelldatei an geeignetem Ort auspacken
  - README / Installationanleitung befolgen
    - Diese ist sehr einfach und schnell ausführbar
- Konfiguration (in Apache's http.conf)
  - "LoadModule"-Anweisung einbauen (noch Installationsteil)
  - Neues Apache-Verzeichnis \$ApacheHome/fcgi-bin anlegen (analog zum vorhandenen "cgi-bin")
  - "fcgi-bin" mit "fastcgi-script" assoziieren
  - <Location /fcgi> ...
  - AppClass /pfad/zu/fastcgi\_modul.fcgi
  - ... (hier zu ergänzen)



- Konfiguration, Teil 1
  - "LoadModule"-Anweisung in httpd.conf einbauen (noch Installationsteil)
  - Neues Apache-Verzeichnis \$ApacheHome/fcgi-bin anlegen
    - analog zum vorhandenen "cgi-bin"
  - Neue Verzeichnis per URL auffindbar machen
    - ScriptAlias-Eintrag aufnehmen, analog zu dem zu "cgi-bin"
  - FCGI *handler* für Einträge in diesem URL-Teil registrieren:
    - <Location /fcgi-bin>  
SetHandler fastcgi-script  
</Location>
    - Bem.: Seltsamerweise scheint dieser Eintrag nicht nötig zu sein...
  - "Directory"-Einträge für "fcgi-bin" anlegen
    - analog zu denen für "cgi-bin" (Siehe <Directory "...">...)



- Konfiguration, Teil 2

Für jede FastCGI-Anwendung wie folgt vorgehen:
- Anwendung ins "fcgi-bin"-Verzeichnis kopieren
- Anwendung in Apache registrieren:
  - Eintrag "AppClass /pfad/zu/fastcgi\_modul.fcgi" ergänzen



## Einbau weiterer Module



## Weitere Module



- Perl bzw. Ruby, CGI / FCGI
  - Ihr Apache-Server enthält nun die erforderlichen Module
- PHP-Anwender
  - Ergänzen Sie selbständig Ihre Apache-Installation um die für PHP notwendigen Module, testen Sie diese.
- Java-Anwender
  - Installieren Sie Apache Tomcat
  - Entscheiden Sie sich für die richtige Version
- Andere, sofern vorhanden
  - Ergänzen Sie Ihre Apache-Installation um die benötigten Module, bis dass Sie die von Ihnen gewählte Entwicklungsumgebung für dynamische Web-Seiten testbereit ist.



# Übungen zu CGI & Co.

Vorübung  
SSI und Cookies  
Formulare  
Belastungstest: Was bringt FastCGI?



## Vorübung



- Machen Sie sich vertraut mit den bereitgestellten einfachen CGI- und FastCGI-Skripten. Testen Sie diese!
- Wer PHP oder Java Servlets einsetzt, sollte sich statt mit FastCGI mit Beispielen dieser Machart vertraut machen.
- Anregung, freiwillig:
  - Schreiben Sie als CGI-Anwendung ein kleines C-Programm, das nur ein einfaches HTML-Dokument (oder auch nur ASCII-Text) ausgibt, und testen Sie seine Wirksamkeit durch Installation im "cgi-bin"-Verzeichnis. Achten Sie auf die extension ".cgi".



- SSI
  - Erstellen Sie eine SHTML-Seite "besucherzaehler.shtml" mit einem SSI-Element, das einen Besucherzähler realisiert (vgl. Vorlesung).
  - Entwickeln Sie eine CGI-Anwendung zaehler.cgi dazu, die den Zähler liefert!
  - Hätte der Weg über FastCGI Vorteile?
- Cookies
  - Generieren Sie eine kleine HTML-Seite dynamisch per (F)CGI-Skript cookiezaehler.cgi (bzw. fcgi)
  - Realisieren Sie darauf einen Besuchszähler
    - Vergeben Sie Cookies an neue Besucher
    - Lesen Sie den Cookie von erneuten Besuchern (incl. eines Zählerwertes), geben Sie den Zählerwert aus, inkrementieren Sie den Zähler im Cookie.



- Einarbeitung
  - Den Umgang mit HTML-Formularen können Sie z.B. bei SelfHTML erlernen.
- Szenario:
  - Abgabe von Hausaufgaben.
- Anmeldung
  - Erstellen Sie ein (statisches) HTML-Formular "anmeldung.html"
    - Eingabefelder: Name, MatrNr, email, LV-Nr, Aufgabe-Nr (01 ... 15)
    - Radio Button-Gruppe: PL oder SL
    - "Abgabe"-Knopf soll folgende Anwendung aufrufen:
  - "anmelden.cgi" (bzw. .fcgi). Schreiben Sie diese Anwendung.
    - Das Formular soll ausgelesen werden. Speichern Sie die Angaben unter einer Session ID persistent!
    - Session Management per Cookie (SessionID darin speichern).
    - Fehlerfall: "Weiterleitung" an eine statische Seite "abgabefehler.html"
    - Normalfall: Weiterleitung an Formular "abgabe.html"



- Abgabe
  - Die Seite "abgabe.html" soll zur Auswahl einer lokalen Datei auffordern und deren Übertragung an "abgabe.cgi" (bzw. .fcgi) einleiten.
  - Die Anwendung "abgabe.cgi" soll
    - die Verwaltung der *session* fortsetzen (Cookie lesen, Eingabedaten wiederherstellen)
    - die eingehenden Daten annehmen und unter matr/aufgXX in einem geeigneten Verzeichnis speichern.
    - im Fehlerfall (z.B. "kein Cookie") an eine statische Fehlerseite leiten.



- Was bringt FastCGI?
  - Erstellen Sie eine (einfache) statische HTML-Seite "perftest.html"
  - Erstellen Sie ein CGI-Programm "perftest.cgi", das bei Aufruf die statische Seite liest und ausgibt. Wahlweise können Sie eine gleichartige Seite auch dynamisch generieren.
  - Erstellen Sie ein FastCGI-Programm "perftest.fcgi" mit gleichartigem Verhalten.
  - Erstellen Sie ein Testprogramm "perfmeter", das die o.g. drei alternativen Dokumentenquellen per HTTP n-mal abrufen und die dafür benötigte Zeit ermittelt.
  - Stellen Sie fest, wieviele Sekunden die drei Varianten für dieselbe Anzahl Abrufe benötigen. Wählen Sie n so, dass die schnellste Variante ca. 10 Sekunden benötigt.
  - Was hat FastCGI hier gebracht?



- Was
  - Bearbeiten Sie mind. 2 der drei Aufgaben
  - Dabei sollte auf jeden Fall die Formularaufgabe sein!
- Wann
  - Spätestens in 2 Wochen (erneut Do 12 Uhr)
  - Hintergrund: Nächste Woche gibt es weniger zu tun, daher gleichen wir hiermit über zwei Wochen aus.