
Programmieren für das Internet

Blatt 2

Abgabe: 12.11.2002

1. [5 Punkte] Programmieren Sie eine Funktion, die eine als String gegebene HTTP-Request-Line in die Bestandteile Methode, URI und Protokoll-Version zerlegt. Eine URI ist dabei immer in diesem vereinfachten Format gegeben:

```
RequestURI = "/" *path-segment
path-segment = *pchar * ( "/" path-segment )
```

Beachte, dass `pchar` hexadezimal kodierte Zeichen enthalten kann, die dekodiert werden müssen. Ein Beispiel: `datei%201` wird zu `datei_1` dekodiert.

2. [15 Punkte] Programmieren Sie einen einfachen Web-Server für HTTP 1.1, der die `GET`-Methode für die Request-URI aus Aufgabe 1 unterstützt. Der Server interpretiert die URI als Pfad in seinem Dateisystem und sendet den Inhalt der Datei als HTTP-Nachricht mit dem Content-Type „`text/plain`“ zurück. In der Anfrage enthaltene HTTP-Header liest der Server und gibt sie auf der Standardausgabe aus; die eingehenden Header werden also nicht weiterverarbeitet.

Der Server sucht die Datei unterhalb eines sogenannten *Document-Roots*: Der Pfad `/` in der Request-URI entspricht dem Pfad `/home/knauel` (bzw. deinem Home-Verzeichnis) im Dateisystem. Beispiel: Fordert der Client die URL `http://localhost/etc/moin.txt` an, so überträgt der Server den Inhalt der Datei `/home/knauel/etc/moin.txt`.

Als Statuscodes sollte der Server 200, 400, 403, 404, 501 und 505 unterstützen.

3. [4 Punkte] Erweitern Sie den Web-Server um die Request-Methode `HEAD`.
4. [6 Punkte] Das Verzeichnis `geheim` im Dateisystem des Web-Servers wurde hinterhältig in `topsecret` umbenannt! Damit die Dateien im Verzeichnis `geheim` auch weiterhin erreichbar sind, soll der Server eine entsprechende Weiterleitung bereitstellen.

Beispiel: Fordert ein Klient die Datei `/geheim/masterplan.txt` an, so antwortet der Server mit Statuscode 301 und leitet den Klienten zur neuen URI `/topsecret/masterplan.txt` weiter. Programmiere eine entsprechende Erweiterung für den Web-Server!

5. [8 Bonuspunkte] Der Unix-Befehl „`file -i`“ findet heraus, welchen MIME-Typ eine Datei hat¹. Benutze `file`, um den Header `Content-Type` in der Antwort des Web-Server richtig zu setzen. Setze außerdem den Header `Content-Length` entsprechend der Länge der übertragenen Datei!

Zur Vorlesung

URIs für Ressourcen mit hierarchischer Struktur:

```
absoluteURI = scheme ":" ( hier_part | opaque_part )
```

wobei gilt:

```
hier_part = ( net_path | abs_path ) [ "?" query ]
net_path  = "://" authority [ abs_path ]
abs_path  = "/" path_segments
path_segments = segment *( "/" segment )
segment   = *pchar *( ";" param )
param     = *pchar
pchar     = unreserved | escaped |
           ":" | "@" | "&" | "=" | "+" | "$" | ","
escaped   = "%" hex hex
unreserved = alphanum | mark
mark      = "-" | "?" | "." | "!" | "~" | "*" | "'" | "(" | ")"
```

HTTP-URLs:

```
http_URL = "http:" "://" host [ ":" port ] [ abs_path "?" query ]
host      = hostname | IPv4address
query     = see later
```

Aufbau der HTTP-Header-Felder:

```
message-header = field-name ":" [ field-value ]
field-name     = <printable ASCII characters>
field-value    = *( field-content | LWS )
field-content  = <any ASCII characters other than CR or LF>
LWS           = [CRLF] 1*(SP | HT)
```

¹Dies gilt zumindest für Linux- und FreeBSD-Systeme.