

## Aufgabe 9: mt-httpd (17 Punkte, Abgabe Do, 29.01.2009 14:00)

Implementieren Sie einen mehrfädigen Webserver `mt-httpd` (Multi-Threaded HTTP Daemon) in einer Datei `mt-httpd.c`, der http-Anfragen auf einem anzugebenden Port entgegen nimmt und Dateien innerhalb eines festen Verzeichnisbaums ausliefert. Das Programm wird wie folgt aufgerufen:

```
mt-httpd www-path port #threads
```

Der erste Parameter gibt hierbei das Wurzelverzeichnis des auszuliefernden Verzeichnisbaumes an, der zweite Parameter die Portnummer, auf welcher der Server Anfragen entgegen nimmt, und der dritte Parameter die Anzahl der Threads, die zur Anfragebearbeitung erzeugt werden. Eine Anfrage ist max. 4096 Zeichen lang und kann wie folgendes Beispiel aussehen:

```
GET /docs/index.html
```

Der Pfadname enthält hierbei keine Leerzeichen. Weitere Wörter, die dem Pfadnamen evtl. folgen, **müssen** vom Server **ignoriert** werden. Die Zeile wird entweder mit ``\r\n`` oder mit ``\n`` terminiert. Der Server liefert dann die geforderte Datei an den Client aus und beendet nach Abschluss der Übertragung die Verbindung.

- Beginnen Sie zunächst mit einem einfädigen Webserver, in dem der Hauptthread die Anfrage sofort bearbeitet bevor eine neue Verbindung entgegengenommen wird (**accept(2)**). Testen Sie Ihren Webserver z.B. mit dem WWW-Pfad `/usr/share/doc/lame/html` und rufen Sie dann in einem Webbrowser (z.B. Firefox) die Webadresse `http://localhost:<PORT>/index.html` auf, worauf die Dokumentation des MP3-Encoders *lame* erscheinen sollte.
- Erweitern Sie den Server nun so, dass mehrere Arbeiter-Threads die Anfragebearbeitung übernehmen. Der Hauptthread soll nun nur noch die Verbindungen annehmen. Verwenden Sie zum Austausch gemeinsamer Daten zwischen den Arbeiter-Threads und dem Hauptthread einen Ringpuffer der Größe 128 (Modul **bbuffer**, Datei `bbuffer.c`), in welchen der Hauptthread die Dateideskriptoren der angenommenen Verbindungen einträgt. Die Arbeiterthreads entnehmen dann jeweils einen Dateideskriptor aus dem Ringpuffer und führen die Bearbeitung durch. Implementieren Sie zur Synchronisation der POSIX-Threads auf dem Ringpuffer Semaphoren im Modul **sem** (Datei `sem.c`) (**pthread\_mutex\_lock(3)**, **pthread\_cond\_wait(3)**). Die angegebene Zahl von Arbeiter-Threads soll einmal zu Beginn des Programmes gestartet werden und dann für die Laufdauer des Programmes bestehen bleiben.

### Hinweise:

- Eine ausführliche Beschreibung des http-Protokolls in der Version 0.9 finden Sie unter `http://www.w3.org/Protocols/HTTP/AsImplemented.html`
- Im Verzeichnis `/proj/i4sp/pub/aufgabe9` finden Sie Schnittstellenvorgaben für die Module **sem** und **bbuffer** sowie einige nützliche Hilfsroutinen im Modul **i4httools**. Sehen Sie sich die Vorgaben an, bevor Sie mit der Bearbeitung beginnen. Verwenden Sie insbesondere die Funktion **check-path()**, da es sonst leicht möglich ist, über Ihren Webserver an beliebige Dateien aus z.B. Ihrem Home-Verzeichnis zu gelangen. Sie können außerdem die bereitgestellten Funktionen zum Übertragen einer Fehlerseite bei einer nicht-existierenden Dateien bzw. einer fehlerhaften Anfrage verwenden.
- Im Vorgabenverzeichnis befinden sich auch das Semaphor- und Puffermodul sowie unsere ausführbare Referenzimplementierung unter dem Namen `mt-httpd`, welche Sie zum Testen und Vergleichen mit Ihrem Programm verwenden können.
- Die Schnittstellenvorgaben der Module **bbuffer** und **sem** finden Sie in einer Online-API-Beschreibung auf der Übungswebseite. Die Schnittstellen sind verbindlich einzuhalten. Die Headerdateien dieser beiden Module dürfen nicht verändert werden. Die Verwendung des Moduls **i4httools** ist optional und es kann auf eigene Bedürfnisse angepasst werden.