Martin-Luther-Universität Halle-Wittenberg
Naturwissenschaftliche Fakultät III
Institut für Informatik

Prof. Dr. Stefan Braß, Dr. Henning Thielemann

1. Übung zur Vorlesung „Grundlagen des WWW“

Sommersemester 2009 — Ausgabe: 2009-04-16

Abgabe: 2009-04-23

Aufgabe 1.1: Day-Time-Client (2 Punkte)

Laden Sie das Programm unter der Adresse

`http://dbs.informatik.uni-halle.de/brass/www09/dt_client.c`

herunter und übersetzen Sie es mit dem Befehl

```
gcc -o dt_client dt_client.c
```

Das erzeugte Programm `dt_client` ist ein Client des „Day-Time-Servers“, es funktioniert aber auch mit dem Server aus Aufgabe 1.2. Führen Sie das Programm aus. Sie können Rechnernamen und Port-Nummer wie bei dem Programm `telnet` explizit angeben. Die Syntax ist: `dt_client host port`. Das Programm kann auch unter Windows übersetzt werden, wenn man zu Beginn das Präprozessorsymbol `OS_WINDOWS` definiert.
Was tut dieses Programm und was ist dessen Ausgabe (mit und ohne Parameter)?

Aufgabe 1.2: Hello-World-Server (2 Punkte)

Beschaffen Sie sich nun das Programm unter folgender Adresse:

`http://dbs.informatik.uni-halle.de/brass/www09/server1.c`

Finden Sie in dem Programm die eingebaute Port-Nummer des Servers. Übersetzen Sie es wieder und starten Sie es mit „`server1`“. Greifen Sie dann mit `telnet` oder `dt_client` auf den Server zu. Sie müssen dazu Rechnernamen und Portnummer wie im Beispiel in Aufgabe 1.1 in der Kommandozeile angeben.
Was gibt der Server aus? Geben Sie in der Lösung außerdem die verwendeten Port-Nummern von Client und Server an! (**Hinweis:** Sie können den Server danach mit Strg-C wieder beenden.)

Aufgabe 1.3: Echo-Server (6 Punkte)

Betrachten Sie das Programm unter

`http://dbs.informatik.uni-halle.de/brass/www09/server2.c`

Finden Sie auch hier heraus, was es tut.

a) In der vorliegenden Version wird die Anfrage vom Klienten erst ausgegeben, sobald sie komplett empfangen wurde (also nach der Leerzeile) oder der Puffer voll ist. Ändern Sie das Programm dahingehend, dass die einzelnen Zeichen unmittelbar nach dem Empfang ausgegeben werden. Lassen Sie sich nich davon verwirren, dass aufgrund der Pufferung der Zeichenströme die Zeichen tatsächlich nur zeilenweise verschickt und entsprechend vom Server ausgegeben werden.

b) Ändern Sie das Programm so ab, dass die Verbindung zum Klienten nach etwa 10 Sekunden abgebrochen wird. Sie können dies erreichen, indem sie die Verbindung mit `fcntl(socket, F_SETFL, O_NONBLOCK);` in den nicht-blockierenden Modus versetzen und mit `usleep` immer eine Zehntelsekunde warten, bevor Sie erneut Daten abfragen. Im nicht-blockierenden Modus gibt `read` den Wert -1 zurück und setzt `errno=EAGAIN`, falls noch keine neuen Daten angekommen sind. Mit `time(NULL)` können Sie die aktuelle Systemzeit in Sekunden ermitteln. Wir empfehlen, eine Funktion zu schreiben,

die wie `read` im blockierenden Modus auf ein eingehendes Zeichen wartet, aber nur bis maximal zehn Sekunden nach Entgegennahme der Anfrage.

Hinweis: Mit dem Shell-Kommando `man` oder dem `man:`-Präfix im Konqueror (unter Linux/KDE) bekommen Sie auch Erklärungen zu Systemfunktionen, beispielsweise welche Header-Dateien mit `#include` eingebunden werden müssen.