

Übungszettel 6

Aufgabe 1: Parallel-Server

a) Client

Schreibt ein Client-Programm, das eine Socket-Verbindung mit dem Server-Programm herstellt und anschließend alle Tastatur-Eingaben direkt an den Server weiterleitet. Bei Eingabe von ESC (ASCII-Code 27_{10} , bzw. $1B_{16}$) wird der Client beendet.

Das Programm bekommt als Parameter den Hostname des Rechners übergeben, auf dem der Server läuft, sowie den Port. Zu Beginn des Hauptprogramms muss überprüft werden, ob die richtige Anzahl von Parametern übergeben wurde und ob diese gültig sind.

Die IP-Adresse zum Hostname erhält man folgendermaßen:

```
struct hostent* hostInfo;  
hostInfo = gethostbyname(argv[1]);  
memcpy(&serverAddr.sin_addr, hostInfo->h_addr, hostInfo->h_length);
```

Anstatt der Bibliothek *netinet/in.h* aus dem Beispielprogramm muss dafür die Bibliothek *netdb.h* eingebunden werden.

b) Server

Schreibt ein Server-Programm, das an einem Port auf eine eingehende Verbindung eines Clients wartet. Sobald ein Client eine Verbindung hergestellt hat, soll sich der laufende Server-Prozess mit Hilfe von *fork()* in zwei Prozesse aufspalten.

Der Eltern-Prozess wartet erneut auf eingehende Verbindungen, der Kind-Prozess verarbeitet die bereits eingegangene. Alle eingehenden Daten werden solange in eine Datei geschrieben werden, bis die Verbindung wieder abgebaut wird. Dabei soll für jede Verbindung eine eigene Datei mit Namen *daten1.txt*, *daten2.txt*, usw. angelegt werden. Es sollen maximal vier Client-Verbindungen gleichzeitig bearbeitet werden.

Ausser in die Dateien soll der Server die Daten auch auf die Standardausgabe schreiben, mit dem Namen der jeweiligen Datei davor. Wird ein ESC empfangen, wird der Kindprozess zur Verarbeitung terminiert.

Das Programm bekommt als Parameter den Port übergeben.

Der Aufbau der Verbindung und die Aufteilung in Eltern- und Kindprozesse soll im Hauptprogramm erfolgen, die Verarbeitung der Daten (Erzeugen des Dateinamens, Öffnen der Datei, Schreiben in die Datei, usw.) in einer Funktion

```
void handleConnection(int childfd, int childnum);
```

Es können also zwei Leute für den Server vorrechnen.

Hinweise

Um die Programme auszuprobieren, ist es am besten, wenn man auf einem Linux-Rechner mehrere Terminals für den Server und die Clients startet. Der Server muss natürlich als erstes gestartet werden.

Die Ports mit den Nummern kleiner als 1024 sind vom System belegt und können nicht verwendet werden.

Den Hostname eines Rechners fragt man unter Linux mit dem Kommando *hostname* ab.