

Kommunikationsnetze

Sommersemester 2003 - Übung 4


Thema: Server-Socket erzeugen und eine Verbindung annehmen

In dieser Übungsstunde sollen Sie einen Server-Socket erzeugen und für eine Verbindung vorbereiten. Mit dem Client-Programm der vorhergehenden Übung können Sie dann eine Verbindung auf einem frei programmierbaren Port aufbauen.

Beispielprogramm:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>

int main()
{
    int server_socket, client_socket;
    int laenge;
    struct sockaddr_in serverinfo, clientinfo;
    server_socket = socket(AF_INET, SOCK_STREAM, 0);
    serverinfo.sin_family = AF_INET;
    serverinfo.sin_addr.s_addr = inet_addr("192.168.103.xxx");
    serverinfo.sin_port = htons(5000);
    laenge = sizeof(serverinfo);
    bind(server_socket, &serverinfo, laenge);
    listen(server_socket, 3);
    printf("\n Der Server wartet...");
    client_socket = accept(server_socket, &clientinfo, &laenge);
    printf("Verbindung mit %s",inet_ntoa(clientinfo.sin_addr));
    close(client_socket);
    close(server_socket);
    printf("\n Server beendet");
}
```

hier bitte die
IP-Adresse
Ihres PC
eintragen !


Aufgabe 1: Übersetzen Sie das Beispielprogramm und erzeugen Sie eine ausführbare Datei mit dem Namen `server`. Starten Sie den Server so, dass Sie weiterhin Linux-Kommandos eingeben können. Dazu müssen Sie nach dem Dateinamen ein Leerzeichen und ein `&`-Zeichen eingeben:

```
server &
```

Lassen Sie sich die Prozessnummer Ihres Serverprogrammes ausgeben (`ps`)

Lassen Sie sich alle Netzwerkverbindungen ausgeben (`netstat -l`)

Aufgabe 2: Schreiben Sie das Client-Programm der letzten Übung so um, dass es ebenfalls den Port 5000 verwendet. Starten Sie das Client Programm. Versuchen Sie auch, eine Verbindung zu einem Server eines anderen Übungsteilnehmers herzustellen.

Aufgabe 3: Versuchen Sie, Ihr Server Programm auf Port 80 zu starten. Erklären Sie das Ergebnis.