

Kommunikationsnetze – Übung 5

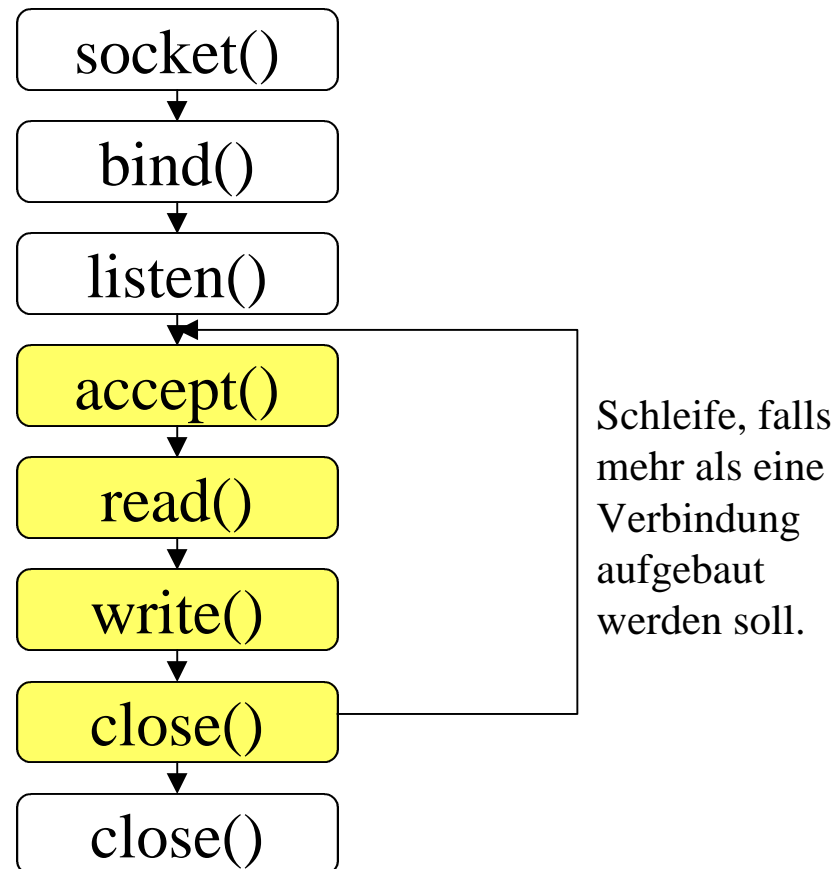
HTML-Datei an einen Browser senden

Lernziele:

- Daten von Browser empfangen `read()`
- Parameter der Funktion `read()`
- Daten zum Browser senden `write()`
- Parameter der Funktion `write()`
- Server für beliebig viele Verbindungen
- Web-Seite dynamisch erzeugen `sprintf()`
- Parameter der Funktion `sprintf()`

Prinzip

- 1) Socket erzeugen
- 2) Socket "binden"
- 3) Socket aktivieren
- 4) Verbindung annehmen
- 5) Daten empfangen (lesen)
- 6) Daten senden (schreiben)
- 7) Client-Socket schließen
- 8) Server-Socket schließen



Server- und Client-Socket

Problem: Server soll mehr als eine Verbindung aufbauen können

Lösung: Server erzeugt für jede angenommene Verbindung einen Socket

Ablauf:

- 1) Funktion `socket ()` erzeugt den ersten Socket: **server_socket**
- 2) Funktion `accept ()` wartet auf Verbindungsaufbau
und erzeugt danach einen zweiten Socket: **client_socket**
darin: empfangene Daten und IP des Senders (client)
- 3) Der **server_socket** ist bereit für den nächsten Verbindungsaufbau

Die Funktion `read()`

Syntax:

```
int read(int socketnummer, char *buffer, int laenge);
```

Übergabeparameter:

`socketnummer` = die Nummer des von `accept()` erzeugten Sockets

`*buffer` = ein Zeiger auf einen Datenpuffer

in diesen Puffer werden die gelesenen Zeichen kopiert

`laenge` = Größe des Datenpuffers in Bytes

Rückgabewert:

Anzahl der gelesenen Zeichen

Die Funktion `write()`

Syntax:

```
int write(int socketnummer, char *buffer, int laenge);
```

Übergabeparameter:

`socketnummer` = die Nummer des von `accept()` erzeugten Sockets

`*buffer` = ein Zeiger auf einen Datenpuffer

in diesem Puffer sind die zu sendenden Zeichen gespeichert

`laenge` = Größe des Datenpuffers in Bytes

Rückgabewert:

Anzahl der gesendeten Zeichen

Die Funktion `sprintf()`

Definiert in `string.h`

Schreibt eine Zeichenkette in einen Text-Puffer (ein Char-Array)

Beispiel:

```
char puffer[100];  
int zahl = 100;  
sprintf(puffer, "Zahl = %d", zahl);
```

Ergebnis: in der Zeichenkette `puffer` steht der folgende Text:

| | | | | | | | | | | | | | |
|---|---|---|---|--|---|--|---|---|---|----|--|--|--|
| Z | a | h | l | | = | | 1 | 0 | 0 | /0 | | | |
|---|---|---|---|--|---|--|---|---|---|----|--|--|--|

String-Ende-Zeichen

Musterlösung Aufgabe 2

```
listen(server_socket, 3);

while(1)
{
    client_socket = accept(server_socket, &clientinfo, &laenge);

    read(client_socket, empfangene_zeichen, sizeof(empfangene_zeichen));

    write(client_socket, web_seite, sizeof(web_seite));

    close(client_socket);
}

close(server_socket);
```

Musterlösung Aufgabe 3

```
#include <string.h>
...
int zaehler = 0;
...
listen(server_socket, 3);
while(1)
{
    client_socket = accept(server_socket, &clientinfo, &laenge);
    read(client_socket, empfangene_zeichen, sizeof(empfangene_zeichen));

    zaehler++;

    sprintf(web_seite, "HTTP/1.1 200 OK\r\n\r\n\
    <html><body><h1>Sie sind Besucher Nr. %d</h1></body></html>", \
    zaehler);

    write(client_socket, web_seite, strlen(web_seite));
    close(client_socket);
}
close(server_socket);
```