Betriebssysteme **WS 2007/2008**
Dr. Henrik Brosenne Georg-August Universität Göttingen
Institut für Informatik

Übung 4

Aufgabe 1

1. Klassifiziere die in der Vorlesung vorgestellten nachrichtenbasierten Prozesskommunikationsmechanismen *Pipes*, *Sockets vom verbindungslosen Datagramm-Typ* sowie *Sockets vom verbindungsorientierten Stream-Typ* dahingehend, ob bei ihnen send und receive jeweils blockierend oder nicht-blockierend sind!

2. Erkläre, wieso es zwei Arten von Pipes gibt. Erläutere dazu kurz, unter welchen Umständen sich *Unnamed Pipes* und unter welchen Umständen sich *Named Pipes* anbieten!

3. Wieso benötigen Sockets, die den verbindungslosen Datagramm-Typ verwenden, keinen `accept` und `connect`-Betriebssystemaufruf? Wieso ist das beim verbindungslosen Datagramm-Typ zu verwendende `sendto` nicht-blockierend, das beim verbindungsorientierten Stream-Typ zu verwendende `write` hingegen blockierend?

4. Gebe auf auf einer Kommandozeile `netstat -a` ein, um Informationen über aktive Sockets auf dem System anzuzeigen.[1] Was bedeutet in der Ausgabe für die Sockets der Internet-Domain jeweils die Paarangabe bei "Local Address" bzw. "Foreign Address"?

5. Schaue in der Datei `/etc/services`[2] den Namen desjenigen Dienstes nach, der mit der *well-known port number* 25 verbunden ist. Recherchiere, was dieser Dienst macht.

6. Falls `netstat -a` ergeben hat, dass auf Port 25 des lokalen Rechners ein Socket der Internet-Domain eingehende Verbindungen akzeptiert[3], stelle einer Verbindung mit dem Socket her, durch Eingabe von `telnet localhost 25`. Was passiert, wenn man `help` eingibt und die Eingabetaste drückt? Beende die Verbindung, durch `quit` gefolgt von der Eingabetaste.

[1] Geht auch unter MS Windows mittels `netstat /a`.
[2] Unter MS Windows XP in: `C:/WINDOWS/SYSTEM32/DRIVERS/ETC/SERVICES`
[3] Falls nicht, aber der Rechner ist mit dem Internet verbunden: führe das Kommando `telnet www.swe.informatik.uni-goettingen.de 80` aus und gebe `get /` gefolgt von der Eingabetaste ein. Was wird daraufhin angezeigt?

Aufgabe 2

1. Schreibe für eine Linux/Unix-Umgebung zwei *C*-Programme, die über Sockets der lokalen Unix-Domain vom verbindungsorientierten Stream-Typ miteinander kommunizieren.

   Der Client-Prozess soll einen genau 80 `char` langen null-terminierten String an den Server-Prozess schicken und sich beenden. Der zu verschickende String soll sich aus dem ersten Kommandozeilenparameter des Client-Programms ergeben.[4] Falls der dort angegebene String länger als 79 Zeichen ist, soll er abgeschnitten und mit `'\0'` terminiert werden. Falls er kürzer als 79 Zeichen ist, soll der Rest mit `'\0'` aufgefüllt werden.

   Der Server soll endlos Verbindungen annehmen und den übermittelten String sowie ein Newline auf der Standardausgabe ausgeben.

2. Starte den Server. Prüfe mittels `ls -l /tmp/` und `netstat -a` nach, ob der Server den lokalen Unix-Socket angelegt hat.

3. Übertrage mit dem Client ein paar Nachrichten.

4. Beende den Server per `kill` oder *CTRL-C*. Existiert danach die Socket-Datei immer noch im Verzeichnis `tmp`? Falls ja, lösche sie per `rm`.[5] Was passiert, falls der Server gestartet wird, wenn die Socket-Datei nicht gelöscht wurde und daher bereits existiert?

Hinweise.

- Orientierung gibt der in der Vorlesung vorgestellten Ablauf der Socket-Kommunikation. Schaue zusätzlich in den Man-Pages der einzelnen Betriebsystemaufrufe nach. (`man unix` liefert Informationen über Sockets der lokalen Unix-Domain.) Behandle auch die Rückgabewerte der Betriebssystemfunktionen, indem mittels `perror` eine Fehlermeldung zurückgeliefert und das Programm beendet wird, falls der Rückgabewert `-1` war. (`perror` hilft enorm beim Debugging!)

- Beim `socket`-Aufruf kann als Wert für den Parameter `protocol` die Zahl 0 angegeben werden.

- `bind` bekommt als zweiten Parameter die Speicheradresse einer `struct` vom Typ `sockaddr_un` übergeben.[6] Deren Strukturelement `sun_family` muss auf `AF_UNIX` gesetzt werden. In das Strukturelement `sun_path` muss ein null-terminierter Dateiname kopiert werden. Dieser dient als eindeutige "Adresse" dieses lokalen Sockets. Verwende als Dateinamen `/tmp/aufgabe8_username`, wobei `username` durch das persönliche

[4] Falls dieser Kommandozeilenparemeter Leerzeichen enthält, muss er auf der Kommandozeile in doppelte Anführungszeichen gesetzt werden.

[5] Wer will, kann auch im Server einen passenden *sighandler* installieren, der dies per `unlink` erledigt.

[6] Dieser Typ ist kompatibel zum in der Man-Page von `bind` angegebenen Typ `struct sockaddr` und kann gefahrlos darauf gecastet werden.

Login zu ersetzen ist. Als dritter Parameter von `bind` muss die Größe des übergebenen `sockaddr_un` Wertes übergeben werden. Diese kann mit dem vordefinierten Macro `SUN_LEN(...)` ermittelt werden, dem als Parameter ebenfalls die Speicheradresse der verwendeten Variable vom Typ `struct sockaddr_un` übergeben werden muss.

- Als `backlog`-Parameter[7] von `listen` wird traditionellerweise die Zahl 5 eingesetzt.
- `accept` liefert einen *descriptor* zurück. Dieser kann dann zur eigentlichen Kommunikationen mit dem Client benutzt werden, indem aus diesen mittels `read` gelesen bzw. in diesen mittels `write` geschrieben wird. Als zweiter Parameter von `accept` muss eine `struct` vom gleichen Typ wie bei `bind` übergeben werden. Diesmal wird diese Datenstruktur jedoch von `accept` mit Werten versehen. Als dritter Parameter muss die Adresse einer Speicherstelle vom Typ `socklen_t` übergeben werden, in der – wie bei `bind` – die Größe der im zweiten Parameter übergebenen Datenstruktur eingetragen werden muss.
- Da der Client sich per `connect` mit der vom Server bei `bind` angegebenen Socket-Adresse verbindet, muss das `connect` beim Client als zweiten und dritten Parameter die gleichen Werte wie beim `bind` des Servers übergeben bekommen.

[7] `connect`-Anfragen der Clients landen in einer Warteschlange des Server-Sockets. Durch `accept` wird jeweils eine Anfrage vom Server aus der Warteschlange entnommen. `listen` legt diese Warteschlange an und aktiviert sie, wobei `backlog` die Länge dieser Warteschlange angibt.