

Aufgabe 6:

Remoteshell (rshd) (18 Punkte)

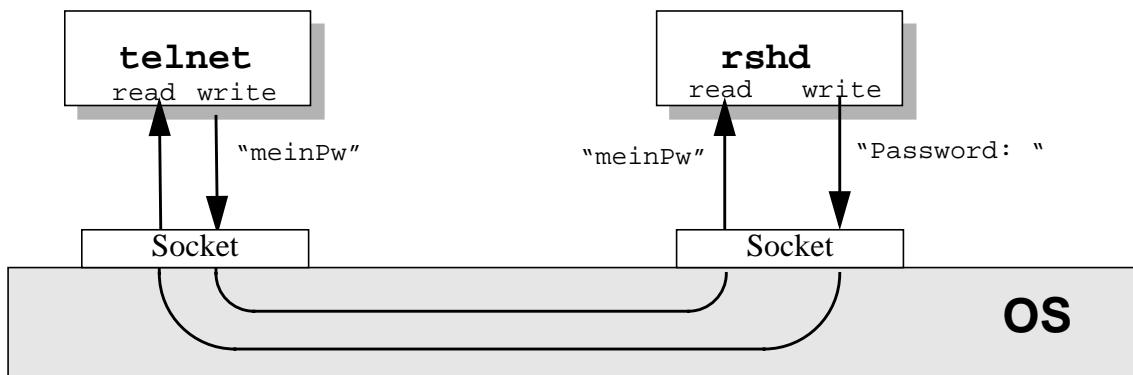
Programmieren Sie basierend auf der Musterlösung von Aufgabe 4 (~i4sp/pub/aufgabe4) eine Shell, die ihre Kommandos nicht mehr von der Tastatur erhält, sondern über einen Socket.

Schreiben Sie dazu einen Daemon, welcher von außen z.B. mittels **telnet** (1) kontaktiert werden kann: Dieser Daemon soll für jede Verbindung einen neuen Prozeß starten, in welchem das unveränderte yash-Programm von Aufgabe 4 ausgeführt wird. Gehen Sie folgendermaßen vor:

a) Socket-Verbindung (7 Punkte)

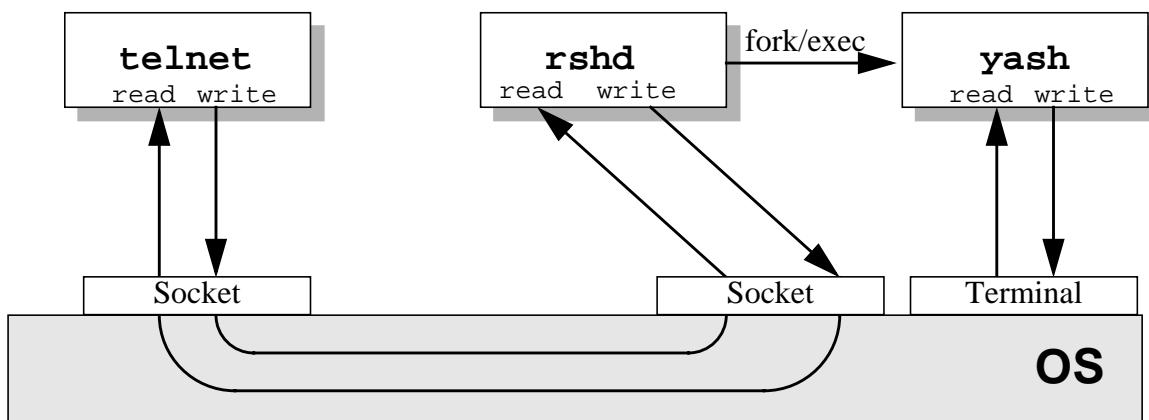
Beim Start des Daemons soll dieser nach einem "Paßwort" fragen. Dieses Passwort, soll dann bei jedem Verbindungsaufbau abgefragt werden. Das Passwort soll maximal 8 Zeichen lang sein.

Erzeugen Sie in Ihrem Daemon einen Streamsocket (**socket(2)**) und binden Sie ihn an alle Adressen des verwendeten Rechners (**bind(2)**). Die Portnummer soll Ihrer UID entsprechen (**id(1)**). Stellen Sie die Anzahl der wartenden Verbindungen auf 5 (**listen(2)**). Nehmen Sie dann Verbindungen über den Socket entgegen (**accept(2)**). Fragen Sie als erstes das beim Daemonstart gesetzte Passwort ab, d.h. schicken Sie der Client-Applikation (telnet) den String "Password:" und werten Sie die nächste Eingabe als Paßwort aus. Unterscheidet sich das Paßwort von dem, daß Sie beim Starten des Daemon angegeben haben, dann soll die Verbindung beendet werden: **Die Shell läuft unter ihrer UID und kann von JEDEM angesprochen werden!**



b) Fork der Shell-Prozesse (4 Punkte)

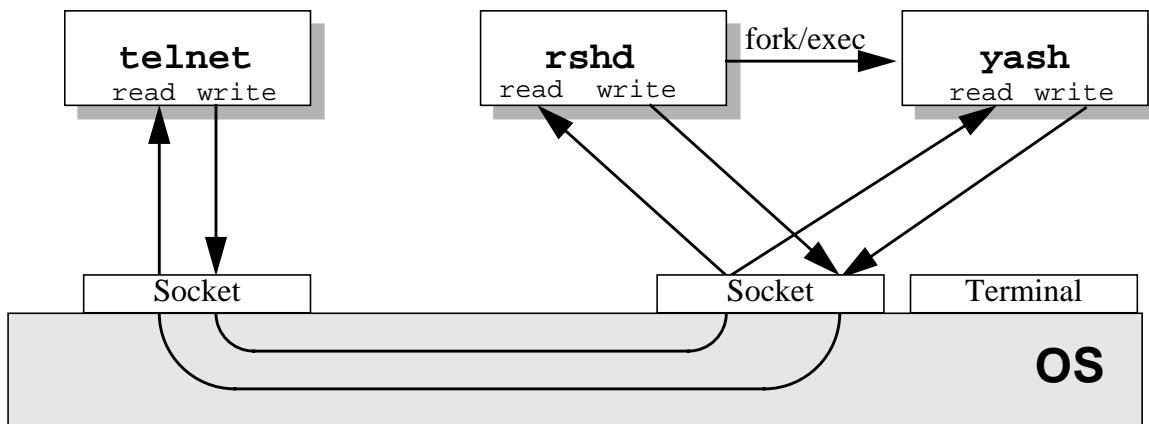
Erzeugen Sie für jede neue Socket-Verbindung einen neuen Prozeß, der diese Verbindung bearbeitet, während der Elternprozeß auf die nächste Socket-Verbindung wartet. Im Kindprozeß soll das yash-Programm von Aufgabe 4 ausgeführt werden. Die Ein- und Ausgaben der yash sollen noch nicht über den Socket gehen.



c) Ein-/Ausgabeumleitung

(3 Punkte)

Leiten Sie die Ein- und Ausgabe des neuen Prozesses (der Shell) auf den Socket um (**dup(2)**). Die Ein- und Ausgaben der vom yash-Prozeß gestarteten Kommandos werden damit ebenfalls über den Socket geschickt. Denken Sie bitte daran, Filedeskriptoren in Elternprozessen zu schliessen, sobald die Filedeskriptoren dort nicht länger benötigt werden.



d) Verwaltung der Shell-Prozesse

(4 Punkte)

Die Musterlösung der yash beendet sich bei Eingabe des Schlüsselwortes “exit”. Der Daemon soll das Beenden der von ihm gestarteten yash-Prozesse auf der Standardausgabe anzeigen.

Verwalten Sie die für die Verbindungsbearbeitung gestarteten yash-Prozesse. Geben Sie beim Starten eines solchen Prozesses auf der Standardausgabe des Servers die Meldung “Started shell with pid <pid>” aus, wobei <pid> für die Prozeß-ID des gestarteten Prozesses steht. Bei Beendigung dieser Prozesse soll der Exitstatus ausgegeben werden: “Exitstatus shell <pid>: <status>”

Verwenden Sie zur Verwaltung der Shell-Prozesse die joblist-Implementierung aus Aufgabe5.

Hinweis zur Lösung dieser Aufgabe:

- Sockets sind nicht Bestandteil des POSIX-Standards. Deshalb müssen Sie Ihr Programm mit dem Define `-D_XOPEN_SOURCE` übersetzen, um dem Compiler mitzuteilen, daß das Programm die im “Single Unix Standard” der “Open Group” enthaltene Funktionalität nutzt.
- Schreiben Sie ein Makefile, mit dem das gesamte Programm erstellt werden kann. Mit “make clean” soll das src-Verzeichnis von unnötigen Objectfiles gesäubert werden.

Dokumentieren Sie Ihre Vorgehensweise in **rshd.doc**!

Abgabe: bis spätestens Donnerstag, 13.01.2000, 14:00

Diese Aufgabe wird für Wirtschaftsinformatiker nach dem Credit Point System benotet.