Aufgabe 9:

20.01.2000

Remoteshell mit Signalbehandlung (remsh) (17 Punkte)

Programmieren Sie basierend auf der Musterlösung von Aufgabe 5 (~i4sp/pub/aufgabe5) und Aufgabe 6 (~i4sp/pub/aufgabe6) eine Remote-Shell, die Hintergrundprozesse und Signalbehandlung unterstützt. Die unten gestellten Fragen sind in der Dokumentation (**remsh.doc**) zu erläutern. Die ausführbaren Programme müssen sich durch ein Makefile erzeugen lassen.

a) Shell Client (5 Punkte)

Schreiben Sie ein Client-Programm **remsh**, welches statt telnet verwendet werden kann, um mit dem rshd zu kommunizieren. Das Programm soll einen Socket aufmachen, von Standard-Input lesen und auf den Socket schreiben, bzw. vom Socket lesen und auf Standard-Output schreiben. Der Client muß gleichzeitig Standard-Input und Socket auf Verfügbarkeit von Daten prüfen (**select(2)**). Schreiben Sie für das Kopieren der Daten von zwei Eingabe-Filedeskriptoren in zugeordnete Ausgabe-Filedeskriptoren eine eigene Funktion `void copyfd(int in0, int out0, int in1, int out1)`.

b) vi-Problem (1 Punkt)

Starten Sie über den Remote-Shell-Client **remsh** das Programm **vi** in der Remote-Shell. Dokumentieren Sie Ihre Beobachtungen.

c) Pseudoterminal (6 Punkte)

Ändern Sie den rshd von Aufgabe 6 und markieren Sie die Teile des Programms, die sie geändert haben, durch Kommentare. Als Shell soll statt der yash die jsh gestartet werden. Die Ein/Ausgabe der Shell soll jetzt an ein Pseudoterminal erfolgen. Öffnen Sie ein Pseudoterminal (**forkpty(3)**) und kopieren Sie unter Verwendung der copyfd-Funktion von Aufgabe (a) die Ausgabe des Sockets in die Eingabe des Master-Filedeskriptors und umgekehrt. Im bei forkpty entstehenden Kindprozess müssen Sie die jsh ausführen.
Testen und dokumentieren Sie das Verhalten des **vi** in Ihrer Shell.

d) Terminaleinstellungen ändern (5 Punkte)

Modifizieren Sie das **remsh**-Programm so, daß das kein lokale Echo der Eingaben erfolgt. Dazu müssen Sie den Terminaltreiber entsprechend konfigurieren (***termios(3)***).

Wenn die Standard-Eingabe ein Terminal ist, wird die Eingabe üblicherweise zeilenweise gepuffert. Der Shell-Client **remsh** soll den Terminaltreiber nun so umstellen, daß jedes auf der Tastatur eingetippte Zeichen sofort zum Shell-Daemon **rshd** geschickt wird. (***termios(3)***)

Bei einem Terminal wird durch die Eingabe von CTRL-C, CTRL-Z bzw. CTRL-\ ein Signal ausgelöst. Dieses Signal darf jetzt nicht mehr auf der Client-Seite erzeugt werden, sondern vom Daemon. Stellen Sie den Terminaltreiber so ein, daß **jedes** Zeichen (auch CTRL-C, CTRL-Z oder CTRL-\ !) an den Daemon übertragen wird — diese Zeichen dürfen also auf der Client-Seite keine Signale mehr auslösen!

Vor dem Beenden des Programms soll der ursprünglichen Zustand des Terminaltreibers wieder hergestellt werden.

Abgabe: bis spätestens Donnerstag, 10.02.2000, 14:00

Bei Fragen zur Aufgabenstellung bitte Mail an *golm@informatik.uni-erlangen.de*