
Aufgabe 2: rlsd

Schreiben Sie ein Programm **rlsd** (remote ls daemon), mit dem es möglich ist über das Internet ein Verzeichnis von einem entfernten Rechner aufzulisten.

a) Hello-World-Server

Im ersten Schritt soll ein Programm entstehen, welches einen Socket (**socket(2)**) erzeugt und von diesem Verbindungen (**listen(2)**, **accept(2)**) entgegen nimmt. Zum Entgegennehmen von Verbindungen ist es sinnvoll, den Socket zuvor an einen festen Port zu binden (**bind(2)**). Die Portnummer soll dem Programm als erstes Argument übergeben werden, zum Testen können Sie einen beliebigen Wert zwischen 1024 und 65535 wählen. Anschließend soll über die Verbindung die Zeichenkette "Hello World!" ausgegeben werden. Nachdem sie Ihren **rlsd**-Server gestartet haben können sie ihn mit dem Programm (**telnet(1)**) testen.

b) Makefile

Erstellen sie zum Übersetzen des Programmes ein minimales Makefile.

c) Verzeichnis Server

Im nächsten Schritt soll nach dem Entgegennehmen der Verbindung anstelle von "Hello World!" das Verzeichnis, in dem das Programm gestartet wurde gelesen und ausgegeben werden (**opendir(3)**, **readdir(3)**, **closedir(3)**).

d) Dateiattribute

Die Ausgabe soll nun um einige Dateiattribute erweitert werden. Um diese Informationen zu bekommen, muss zu erst die Inode-Information von jedem Verzeichniseintrag gelesen werden (**stat(2)**). Geben sie zu jedem Eintrag den Dateityp, den Zeitpunkt der letzten Änderung, die Größe, die Benutzerkennung (UID) und die Gruppenkennung (GID) aus (**getpwuid(3C)**, **getgrgid(3C)**).

e) Symbolische Links

Im letzten Schritt sollen symbolische Links nicht mehr verfolgt werden, sondern als solche wie bei "ls -l" angezeigt werden (**lstat(2)**, **readlink(2)**).

Hinweise:

- Informationen zu den verwendeten Kommandos bzw. Systemaufrufen finden sie im Unix-Manual, abrufbar mit dem Kommando **man**, oder in den Folien des Sommersemesters 2006: http://www4.informatik.uni-erlangen.de/Lehre/SS06/V_SOS1/Uebung/
- Achten Sie darauf das einige der angegebenen Bibliotheksfunktionen die internen Datenstrukturen bei wiederholten Aufrufen wiederverwenden.
- Beachten Sie bitte auch das **readlink(2)** keinen '\0'-terminierte-Zeichenkette in den Rückgabepuffer schreibt.
- Symbolische Links und Sockets sind nicht Bestandteil des POSIX.1-Standard. Um die Funktionen benutzen zu können, müssen Sie Ihr Programm mit **-D_BSD_SOURCE** übersetzen.