

Blatt 08

Besprechung: 28. Juni 2012

1. Finger (10P)

Implementiert das *Finger User Information Protocol* gemaess RFC 1288. (Der RFC ist online verfuegbar unter <http://tools.ietf.org/html/rfc1288>.)

Schreibt einen parallelisierten Server, der zur Beantwortung jeder Anfrage einen eigenen Kind-Prozess startet, auf Basis von Sockets.

```
fingerd [PORT]
```

Von den Anfragetypen muesst ihr nur die '{U}{C} query' (Sec. 2.5.2 im RFC) unterstuetzen. Das ist gleichfalls die minimale Anforderung des RFCs. Die Antwort eures Servers soll zumindest den Login-Namen, den realen Namen (GECOS Feld), das Home-Verzeichnis und die Login-Shell beinhalten. Diese Informationen bekommt ihr mit der Funktion *getpwnam()*. Sind die Dateien '.plan', '.project', oder '.pubkey' im Home-Verzeichnis des angefragten Benutzers vorhanden, soll ihr Inhalt ausgegeben werden. Interpretiert angefragte Namen als vollstaendige Login-Namen, das macht Anfragen stets eindeutig.

Euer Programm soll eine valide Implementierung des Finger-Protokolls sein.

Orientiert euch deshalb grundsatzlich am RFC. Achtet darauf, dass eure Server-Implementierung sicher ist, falls der Morris-Wurm vorbei kommt. ;-)

Als Client zum Testen koennt ihr anfangs telnet(1) verwenden. Beispiel:

```
$ telnet marmaro.de 79
Trying 46.163.115.27...
Connected to marmaro.de.
Escape character is '^]'.
meillo
[... informative output ...]
Connection closed by foreign host.
$
```

(Eingaben sind fett gedruckt.)

Schreibt euren Finger-Client so, dass er als

```
finger USER@HOST [PORT]
```

aufgerufen werden kann. Er soll dann zu HOST an PORT eine Verbindung oeffnen und Informationen zu USER anfragen.

Denkt daran, dass ihr auf den Solaris-Systemen die Programme mit '-lsocket' linken muesst. Siehe 'man -s3socket bind'.

Zum Einreichen der Loesung:

```
submit ss2 8 [team] [notes] fingerd.c finger.c
```

Fragen zur Selbstkontrolle

- Nenne Unterschiede zwischen Pipes und Sockets.
- Welche Arten von Sockets gibt es?
- Was ist ein Datagram?
- Mit welchen Funktionen greift man auf Sockets zu?
- Welche Adressfamilien gibt es und wann und wie waehlt man sie?
- Welche Informationen gehoeren zu einer Socket-Adresse?
- Wie war das noch mal mit der Byte-Order?
- Warum sind Sockets so kompliziert zu verwenden? Ginge das nicht auch einfacher?