

Laboratorio di sistemi operativi

A.A. 2010/2011

Gruppo 2

Gennaro Oliva

22

Esercizi sui socket

I lucidi di seguito riportati sono distribuiti nei termini della licenza Creative Commons “Attribuzione/Condividi allo stesso modo 2.5” il cui testo integrale è consultabile all'indirizzo:
<http://creativecommons.org/licenses/by-sa/2.5/it/legalcode>

Esercizio 35

- Scrivere una applicazione client/server in cui:
 - Il client richiede come parametri sulla riga di comando, l'indirizzo IP del server e la porta su cui contattarlo, quindi una volta stabilita la connessione, esegue un ciclo infinito in cui legge stringhe dallo standard input e le invia al server
 - Il server è iterativo ed accetta come unico parametro il numero di porta su cui accettare connessioni e visualizza le stringhe inviate dai client che si connettono su quella porta
 - Il client termina l'esecuzione quando l'utente inserisce la stringa exit, mentre client e server terminano entrambi l'esecuzione quando l'utente inserisce la stringa close

accept

- L'**accettazione** di una connessione da parte di un server viene effettuata utilizzando:
- `int accept(int fd, struct sockaddr *client, socklen_t *dim);`
- **fd** è il socket in ascolto, **client** è il puntatore ad una struct `sockaddr` **allocata** dal server che viene assegnata dall'`accept` che vi memorizza le **informazioni** sul **client** (indirizzo IP, numero di porta da cui il client si connette)
- L'argomento **dim** è un parametro di input/output (infatti è passato come puntatore): in input contiene la dimensione della struttura indirizzata da **client** in output contiene l'effettiva dimensione dell'indirizzo della controparte
- È possibile **ignorare** queste informazioni passando **NULL**

Esercizio 36

- Trasformare il server dell'esercizio precedente in concorrente
- Modificare il client in modo che accetti nomi simbolici
- Modificare il server in modo che stampi l'indirizzo IP e la porta del client da cui provengono le stringhe

Esercizio 37

- Scrivere una applicazione client/server che implementi una shell remota:
 - Il client richiede come parametri sulla riga di comando, l'indirizzo IP del server e la porta su cui contattarlo, stabilisce una connessione ed effettua una fase di autenticazione in cui risponde alla richiesta del server di inserire una password
 - Ad autenticazione avvenuta con successo il client esegue un ciclo infinito in cui legge stringhe dallo standard input e le invia al server i comandi da eseguire
 - Il server è concorrente ed accetta come unico parametro il numero di porta su cui accettare connessioni ponendosi in ascolto sull'indirizzo di loopback 127.0.0.1 (non utilizzare INADDR_ANY)
 - Il server richiede ad ogni client una password verifica che corrisponda a quella contenuta in un file predefinito. Successivamente il server riceve ed esegue i comandi inviati dal client inviandogli l'output prodotto
 - Il client termina l'esecuzione quando l'utente inserisce la stringa exit, mentre client e server terminano entrambi l'esecuzione quando l'utente inserisce la stringa close