

Esercizio 1

- Scrivere una applicazione client/server in cui
 - il server visualizza:
 - L'IP da cui si e' connesso il client
 - I messaggi testuali inviati dal client
 - Il server termina quanto il client invia “exit”

Client e server comunicano attraverso socket TCP

Esercizio 2

- Scrivere una applicazione client/server in cui
 - il server accetta i seguenti comandi:
 - TIME: Il server invia al client l'ora corrente
 - DATE: Il server invia al client la data corrente
 - PORT: Il server
 - crea un nuovo socket, invia il numero di porta al client. Suiccessivamente al comando PORT, tutte le comunicazioni avvengono sul nuovo socket.
 - Il client riceve dall'utente il comando da inviare al server
- Nota: Se un process esegue la listen su un socket su cui NON ha eseguito la bind, il sistema operativo associa una porta libera casuale. Il processo puo' ottenere il numero di porta assegnato dal sistema utilizzando la funzione getsockname()

Esercizio 3

- Scrivere una applicazione client/server in cui
 - Il server puo' gestire concorrentemente piu' client.
 - Quando il client si collega al server, viene creato un processo figlio che gestisce l'intera sessione per il client.
 - Il server riceve dal client una stringa ed invia al client l'indirizzo IP associato alla stringa.
 - Il client invia al server le stringhe lette da standard input
 - termina quando l'utente digita la stringa exit.

Esercizio 4

- Scrivere una applicazione client/server in cui
 - Il server puo' gestire concorrentemente piu' client.
 - Il client
 - Legge da standard input una espressione aritmetica `<operando1> <operatore> <operando2>` e la invia al server
 - Il server esegue l'operazione ed invia il risultato al client
 - Il client visualizza il risultato.
 - termina quando l'utente digita la stringa exit.