

Programmazione in ambienti distribuiti I

(01FQT, 01FEL, 06EKE, 02HDF)

AA 2009-2010, Esercitazione di laboratorio n. 3

Esercizio 3.1 (server TCP concorrente)

Sviluppare un server TCP (in ascolto sulla porta specificata come primo parametro sulla riga di comando) che accetti richieste di trasferimento file da client ed invii il file richiesto.

Il server deve implementare lo stesso protocollo usato nell'esercizio 2.4 ed essere di tipo concorrente, attivando un massimo di tre figli simultanei.

Usando il client sviluppato nell'esercizio 2.4 provare ad attivare quattro client simultaneamente e verificare che solo tre riescono a collegarsi.

Provare a terminare un client brutalmente (battendo ^C nella sua finestra) e verificare che il server (padre) sia ancora attivo ed in grado di rispondere ad altri client.

Esercizio 3.2 (client TCP con multiplexing)

Modificare il client dell'esercizio 2.4 per gestire un'interfaccia utente interattiva che preveda i seguenti comandi:

- G file (richiede di fare GET del file indicato)
- Q (richiede di chiudere il collegamento col server col comando QUIT dopo che un eventuale trasferimento in corso è terminato)
- A (richiede di terminare immediatamente il collegamento col server, anche interrompendo un eventuale collegamento in corso)

L'interfaccia utente deve essere sempre attiva e permettere quindi il "type-ahead", ossia fornire input anche se c'è un trasferimento in corso dal server.

Esercizio 3.3 (server versatile)

Sviluppare un server iterativo che risponda sia al protocollo TCP che UDP su tutte le porte il cui numero è passato sulla riga di comando.

Quando riceve una richiesta (sotto forma di datagramma o collegamento TCP) il server risponde col seguente messaggio di testo:

```
Ciao NOME ( INDC : PORTC ), ti sei collegato a INDS:PORTS con PROTO
```

Ove NOME è il nome DNS del client, INDC e PORTC l'indirizzo IP e la porta del client, INDS e PORTS l'indirizzo IP e la porta del server, PROTO il protocollo di trasporto usato.

Il messaggio è contenuto in un unico datagramma (nel caso UDP) mentre nel caso TCP il canale viene chiuso dal server subito dopo avere inviato il messaggio.

Se il nome DNS del client non è disponibile il server deve usare la stringa "<ignoto>".

Sviluppare un client che si colleghi al suddetto server con protocollo TCP o UDP a seconda che sia fornita l'opzione **-t** o **-u** come primo parametro sulla riga di comando (default: protocollo UDP). Il server a cui collegarsi è fornito come ultimo parametro sulla riga di comando nella forma numerica "indirizzo:porta" oppure nella forma logica "nome_DNS:nome_servizio".

Nota: se si desidera rendere visibili dei nomi e relativi indirizzi si possono inserire nel file **/etc/hosts**. Per rendere noto un servizio non standard si possono inserirne i dati nel file **/etc/services**. Alla fine dell'esercitazione ripristinare il contenuto iniziale di questi file.