Università degli Studi di Bologna
Scuola di Ingegneria

Corso di
Reti di Calcolatori T

Esercitazione 5 (proposta)
Focalizzazione conoscenze acquisite

Luca Foschini

Anno accademico 2013/2014

Specifica trasferimento struttura dati

Organizzare e sviluppare **il trasferimento delle due strutture dati** seguenti in linguaggio C.

```
typedef struct
{/*decidere ordine
dei campi */
   char* stringa;
   int intero;
   char carattere;
} struttura;
```

```
typedef struct
{
   char stringa[20];
   int intero;
   char carattere;
}
struttura;
```

Si presti particolare attenzione alla modalità di trasferimento usata per il campo ***stringa*** (utilizzo di un *separatore* oppure *invio lunghezza del dato trasmesso*) e **si giustifichino di conseguenza le scelte progettuali fatte**

Trasferimento stringhe

Organizzare e sviluppare **il trasferimento di stringhe da un pari ad un altro** in linguaggio C.

```
Scrittura di
stringhe di
lunghezza massima
20 byte
char * stringa;
Invio tipico con
una sequenza di
meno di 10 byte
```

```
Ricezione di stringhe
usando stringa[21]
per contenerle

Lettura e scrittura
a stringhe
```

Si presti attenzione alla lettura che potrebbe leggere più dati inviati in un colpo solo e potrebbe perdere contenuto

Se si stampa come stringa un campo che ne contiene una dopo l'altra, si stampa solo la prima ('/0')

Letture bufferizzate e coordinate

Si imposti una funzionalità per il trasferimento di dati binari di dimensione variabile: ad esempio frame di una sequenza stream di dati. Ad esempio, con:

- **uso di un buffer** (di grosse dimensioni) per il trasferimento del singolo frame;
- **controllo sul numero di byte effettivamente letti** a fronte di scritture molteplici da parte del pari
- **controllo del tempo di trasferimento** del file lato client e server

Controllo del tempo di trasferimento

Per valutare il tempo di trasferimento (**Ttrasf**) bisogna prendere il tempo di sistema **prima** (ad es. **T1**) e **dopo** (ad es. **T2**) le parti di codice che realizzano il trasferimento file. Il tempo di trasferimento viene valutato approssimativamente come:

Ttrasf = T2 – T1

Come ottenere il tempo di sistema?

Varie possibilità (con risoluzione diversa) a seconda del sistema operativo e del linguaggio utilizzato.

Ad esempio, in Java, la funzione statica:

```
public static long System.currentTimeMillis();
```

restituisce “the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC.”

Letture bufferizzate… con valutazione tempo trasferimento

In particolare, si realizzino i programmi client e server con le seguenti interfacce:

client serverHost serverPort bufferDim

server localPort bufferDim

Per ottenere il tempo di sistema in C si può utilizzare la funzione **gettimeofday** (definita in sys/time.h), che sfrutta le strutture:

```
// timeval salva il tempo, sempre dalle 00:00, dell'1/1/1970
// ha due campi, uno per i secondi (tv_sec) e uno per i microsecondi (tv_usec)
  struct timeval tv;
  struct timezone tz;
  gettimeofday(&tv, &tz);
tv.tv_sec; tv.tv_usec; // secondi e microsecondi
tv.tv_usec/1000;  // millisecondi
```

Si veda anche il **man di sistema**

Letture bufferizzate… con valutazione tempo trasferimento

I programmi **client** e **server** devono stampare a video il tempo totale di trasferimento del file

Si effettuino un po' di prove, **cambiando le dimensioni dei buffer** dai due lati

Quali considerazioni si possono fare?
Quale **dipendenza dalle dimensioni dei buffer** dai due lati?

Si provi a fare lo stesso nel mondo Java

Rivisitazione Esercitazione 3

Si provi a implementare una nuova realizzazione dell'Esercitazione 3, parte stream, che utilizzi **una sola socket** per la gestione di tutta la sessione con un client; in particolare:

- **Lato client per ogni invio bisogna indicare la lunghezza del file al server** (prima di spedirne il contenuto)
- **Lato server è necessario leggere il file a blocchi e usare un buffer di appoggio**, gestendo poi il trasferimento dei singoli blocchi (lunghezza e contenuto) dal server al client.

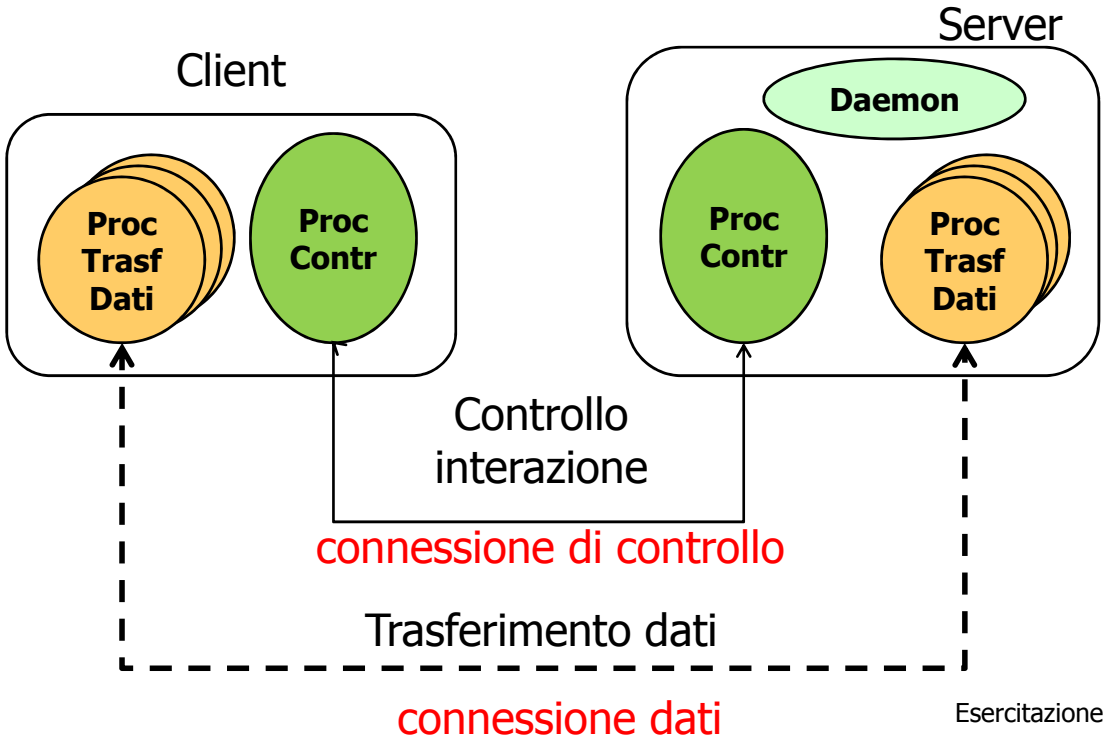
Proposta di estensione: Realizzazione servizio mget FTP

Si vuole sviluppare **un'applicazione C/S basata su socket con connessione** per il trasferimento di tutti i file di un direttorio remoto dal server al client (**multiple get**). Si dovranno realizzare i programmi **client** e **server**.

Si preveda **un'interazione sincrona (realizzata con una richiesta al processo di controllo lato server)** per trasferire il **nome del direttorio** da cui si devono prelevare i file da inviare; quindi, una seconda fase di **trasferimento dei file** dal server al client (realizzata da **processi figli** ulteriori con socket connesse).

Si vogliono realizzare due modalità di trasferimento, la prima con **client attivo** (**il client effettua la connect**), la seconda con **server attivo** (**il server effettua la connect**).

Proposta di estensione: Architettura



Trasferimento di file in direttori

Per ogni **Cliente**, il server prevede di creare un figlio, **processo di sessione** (chiamiamolo **processo di controllo del server**), a cui delegare tutta la interazione con quel cliente (**processo di controllo client**).

La **connessione di controllo** viene quindi usata tra il **processo di controllo server** e il **processo controllo cliente**. Questa deve servire per tutta la sessione di lavoro per preparare i trasferimenti dei file: ad esempio su questa connessione, diciamo **connessione di controllo**, passa il **nome del direttorio** da cui si devono trasferire i file; assumiamo il cliente ciclico che lavora fino alla fine del file di input.

Per ogni richiesta di direttorio del **cliente**, si vogliono **trasferire i file** attraverso **una nuova connessione dati** gestita da processi diversi (figli dei precedenti processi): il tutto va coordinato in modo corretto.

Trasferimento dei file

Per ogni **Direttorio**, il **processo di controllo del server** e il **processo di controllo client devono coordinarsi per il trasferimento dati creando una coppia di figli ad hoc.**

Si vogliono realizzare **due modalità** di trasferimento dei dati e una politica opportuna di gestione del trasferimento stesso.

- la prima con **client attivo** (**il client effettua la connect**),
- la seconda con **server attivo** (**il server effettua la connect**).

Nel primo caso, il cliente deve **richiedere la connessione dati** (attraverso un connect) e il server deve **disporsi ad accettarla** (su una porta specificata).

Nel secondo caso, il cliente deve **aspettare la richiesta di connessione** (attraverso una accept) e il server deve disporsi a **richiedere la connessione dati** (connect su una porta specificata dal client).

Trasferimento più direttori: Client Attivo

Per ogni richiesta di direttorio, il Cliente passa il **nome del direttorio** e aspetta la **porta di ascolto del server** (il numero di porta su cui connettersi) prima di attivare il **figlio per il trasferimento dati**.

Il **Client** richiede ciclicamente all'utente il **nome del direttorio** da cui trasferire i file e **comanda la operazione, ricevendo la porta di ascolto**, su cui dovrà attivare un **processo figlio** che **stabilisce una connessione** con il server remoto e riceve i file salvandoli nel direttorio locale in modo autonomo.

Il **Server** lavora come **processo di controllo (per quel client) che gestisce l'intera sessione**. Per ogni interazione, il processo di controllo, dopo **aver accettato la richiesta**, **crea una socket di ascolto** (su porta scelta) e su questa attiva un **processo figlio (processo trasferimento dati)**, dedicato a ricevere i file del direttorio; quindi il **processo di controllo server** restituisce il **numero di porta al processo di controllo del cliente corrispondente, in attesa per attuare la connessione**.

Trasferimento più direttori: Server Attivo

Per ogni richiesta di direttorio, il Cliente passa il **nome del direttorio** e la **porta di ascolto del processo trasferimento dati (figlio) in attesa** (il numero di porta su cui il processo di trasferimento dati server deve connettersi), dopo avere attivato il figlio con una certa porta per il trasferimento.

Il **Client** richiede ciclicamente all'utente **il nome del direttorio** da trasferire, **crea la socket di ascolto**, (delegandone la gestione ad un processo figlio), **poi effettua la chiamata per il trasferimento**.

Nel **Server** lavora il **processo di controllo (per quel client) che gestisce l'intera sessione**. Per ogni interazione, il processo di controllo, dopo aver **accettato una richiesta iniziale**, attiva un processo figlio a cui affida la **creazione della socket**, l'esecuzione della **connect** e il completamento del servizio richiesto.