

Progetto FTP Mirror

Descrizione sintetica:

Realizzare una applicazione che un servizio di ftp mirroring. L'applicazione consta di due applicazioni, una server ed una client.

L'applicazione server implementa i comandi di list, get e put. Il primo trasferisce file dal server al client ed il terzo trasferisce file dal client al server. Il client invia legge comandi dall'utente, li invia al server ed esegue le azioni associate, i.e., visualizza la lista, memorizza il file per la get o invia file per la put. Il server è implementato da una applicazione multiprocesso che associa ad ogni nuovo client un nuovo processo.

NOTA: Per simulare l'effetto della latenza di rete, le applicazioni trasferiscono i file suddividendoli in pacchetti da 100 byte ed inviando pacchetti consecutivi ad intervalli di 1 secondo.

L'applicazione deve essere scritta utilizzando il linguaggio C. I processi comunicano attraverso socket TCP. L'elaborato deve essere corredato da una opportuna documentazione.

Descrizione dettagliata:

Processo di server:

Il processo server riceve su linea di comando il numero di porta su cui attivare il servizio ed il nome della "service directory" in cui opererà. Può ricevere i seguenti comandi:

- **List <par>**. Il comportamento è il seguente:
 - Se <par> corrisponde al nome di un file, il server invia al client la dimensione del file.
 - Se <par> corrisponde ad una directory, il server invia l'elenco dei file contenuti nella directory <par> e, per ognuno di essi, indica la dimensione.
 - Altrimenti il server invia un messaggio d'errore.
- **get <par>**. Il comportamento è il seguente:
 - Se <par> corrisponde al nome di un file, il server invia al client il contenuto del file.
 - Se <par> corrisponde ad una directory, il server invia al client tutti i file nella directory. Nota, l'invio NON è ricorsivo.
 - Altrimenti il server invia un messaggio d'errore
- **Put <par>**. Il comportamento è il seguente:
 - Se <par> corrisponde al nome di un file, il client invia al server il contenuto del file.
 - Se <par> corrisponde ad una directory, il client invia al server tutti i file nella directory. Nota, l'invio NON è ricorsivo.
 - Altrimenti il client visualizza un messaggio d'errore

Regole generali.

Il server ed il client vanno realizzati in linguaggio C su piattaforma UNIX/Linux. Le comunicazioni tra client e server si svolgono tramite socket TCP. Oltre alle system call UNIX, i programmi possono utilizzare solo la libreria standard del C. Sarà valutato negativamente l'uso di primitive non coperte dal corso (ad es., code di messaggi) al posto di quelle studiate.

Relazione

Il progetto va accompagnato da una relazione che contenga almeno le seguenti sezioni:

1. Una guida d'uso per il server e per il client, che illustri le modalità di compilazione e d'uso dei due programmi.
2. Una sezione che illustri il protocollo al livello di applicazione utilizzato nelle comunicazioni tra client e server (non il protocollo TCP/IP!).
3. Una sezione che descriva i dettagli implementativi giudicati più interessanti (con particolare riferimento alle system call oggetto del corso), eventualmente corredati dai corrispondenti frammenti di codice.
4. In appendice, la relazione deve riportare il codice sorgente integrale del progetto.

Consegna del progetto

Entro due giorni dalla data prescelta per lo scritto finale, vanno consegnati al docente il progetto e la relazione. Il progetto va inviato all'indirizzo clemente.galdi@unina.it in un archivio compresso in formato zip. Durante la correzione del progetto, il client ed il server verranno testati, eseguendoli su due o più macchine diverse.