Progetto File Sharing P2P

Descrizione sintetica:

Realizzare una applicazione che implementi un servizio distribuito di *file sharing peer-to-peer*. All'atto dell'avvio, l'applicazione si registra presso un server di *rendez-vous* da cui riceve gli indirizzi di tutte le applicazioni connesse. Terminata la fase di registrazione, l'utente può effettuare la ricerca di un file indicandone il nome. L'applicazione contatta tutti i peer attivi ed invia la richiesta ricevuta dell'utente. Tra tutti i peer che rispondono positivamente alla richiesta di download, l'applicazione ne seleziona uno ed avvia il download del file richiesto dal peer selezionato, indicando agli altri peer presenti l'annullamento della richiesta. Se l'applicazione riceve una richiesta di download, ricerca localmente il file richiesto e, se presente, si propone per l'invio. Su richiesta dell'utente l'applicazione può terminare, nel qual caso invia al server un messaggio di terminazione ed attende la terminazione di eventuali download in corso.
Il server di rendez-vous notifica tutti i peer all'atto della modifica dello stato della rete, i.e., quando un peer termina o un nuovo peer si connette al servizio.

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 rendez-vous:
Il processo server riceve su linea di comando il numero di porta su cui attivare il servizio. Il server riceve dai peer le seguenti richieste:

- **Connessione**. Il peer invia l'indirizzo IP ed il numero di porta su cui e' in ascolto. All'atto della richiesta il server invia:
  - Al peer appena connesso la lista degli indirizzi IP e dei numeri di porta degli altri peer registrati
  - Agli altri peer già attivi l'indirizzo IP ed il numero di porta del nuovo peer.
- **Disconnessione**. Il peer comunica la prossima disattivazione del servizio. Il server invia a tutti i peer attivi l'indirizzo IP ed il numero di porta del peer che ha richiesto la disconnessione.

Ad ogni peer connesso, il server associa un thread che gestisce tutte le comunicazioni con il peer.

Processo peer:
Il processo peer riceve su riga di comando:

- L'indirizzo IP del server di rendez-vous
- Il numero di porta del server di rendez-vous
- Il numero di porta su cui mettersi in attesa di connessioni da altri peer
- Il nome della directory in cui sono memorizzati i file da condividere
- Il nome della directory in cui memorizzare i file scaricati

Il peer, all'atto dell'avvio, contatta il server di rendez-vous inviando il numero di porta su cui attende le connessioni da altri peer. Il server invia gli indirizzi di tutti i peer attivi. La comunicazione tra peer e server di rendez-vous è gestita da uno specifico thread.
Il peer crea un thread per ogni peer attivo nella rete e stabilisce una connessione con il peer.
Terminata la fase di inizializzazione, il peer legge da standard input i comandi dell'utente.

L’utente può inviare al peer uno dei seguenti comandi:

- connessioni: Visualizza lo stato di tutti i peer cui e’ connesso, i.e., Indirizzo IP, numero di porta e “stato del peer” (attivo/in terminazione)
- download: Visualizza le statistiche per ogni download in corso (Numero di byte scaricati, dimensione del file)
- stop: Indica al volontà di terminare il peer. In questo caso, lo stato del peer diventa “in terminazione”. Il peer informa il server di rendez-vous e non accetterà più richieste di download. Prima di terminare attende la terminazione di eventuali trasferimenti in corso.
- Nome di un file. In questo caso, il peer invia ad ogni peer cui è connesso il nome del file. Attende le risposte dei peer e selezione uno dei peer che hanno segnalato la presenza di un file con il nome indicato. Viene eseguito il download del file che viene memorizzato nella directory opportuna. Il download è eseguito a blocchi di 100 byte inviati ad intervalli di un secondo. Una volta avviato il download, l’utente può indicare un altro comando o nome di file SENZA ATTENDERE la terminazione del download.

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.