



LUMSA
UNIVERSITÀ

Reti di Calcolatori (A.A. 2020-2021)

Progetti RC

Prof. Valerio Rughetti

Istruzioni generali

I progetti proposti di seguito possono essere svolti dagli studenti del corso di Reti di Calcolatori dell'Anno Accademico 2020-2021, corso di laurea in Tecniche informatiche per la Gestione dei dati, secondo le modalità descritte nella pagina Web del corso:

<https://elearning.lumsa.it/course/view.php?id=165>

Il progetto è *facoltativo* e sostituisce l'esame "classico". Il progetto deve essere svolto in gruppi di 2 studenti. In caso di particolari esigenze il progetto può essere svolto singolarmente, ma è sconsigliato (carico di lavoro eccessivo), oppure da gruppi di 3 studenti (richiesto giustificativo). Sul sito web del corso verrà pubblicato un link a un modulo Google dove si potranno registrare i gruppi.

Quando e dove consegnare?

Il progetto deve essere consegnato *almeno 5 giorni* prima della data prevista per la verbalizzazione (ovvero la data in cui si sostiene l'esame). È quindi possibile presentare il progetto entro e non oltre i cinque giorni che precedono il secondo appello di settembre 2021 ovvero il **25 settembre 2021**. Il materiale va consegnato al docente sul sito del corso nell'apposita sezione.

Cosa consegnare?

Va consegnato il seguente materiale, sul quale devono essere chiaramente indicati il nome, cognome, matricola ed e-mail di tutti i componenti del gruppo:

- 1) Un file zip contenente il codice di tutta la sperimentazione, che deve essere chiaro e commentato. Per codice si intendono sia i sorgenti delle implementazioni, incluse le istruzioni per lanciare il programma per testare facilmente il corretto funzionamento dell'implementazione, sia l'elenco delle librerie usate. Non includere invece il codice delle librerie (solo i link alle librerie usate).
- 2) Una relazione (max 3/4 pagine) che contenga tutti i risultati della sperimentazione sia in forma tabulare che mediante grafici (se opportuni). Per ogni funzione implementata deve essere data una precisa descrizione dei parametri, del valore restituito e dell'operato della funzione stessa. Si richiede una versione elettronica della relazione (docx e/o pdf). La capacità di sintesi sarà premiata!
- 3) Un elenco delle esecuzioni eseguite (test esaurienti!).

Scopo del progetto

Il progetto richiede la creazione di un'applicazione che faccia uso delle tecnologie viste a lezione durante il corso. Il progetto può essere implementato in un linguaggio a scelta dello studente.

I progetti

In questo paragrafo vengono descritti alcuni progetti. Le implementazioni devono essere efficienti da un punto di vista asintotico e funzionanti. La descrizione dei progetti fornisce le specifiche minime per la realizzazione dei progetti stessi, gli studenti sono liberi di ampliare le funzionalità delle applicazioni stesse.

Esempi di progetti

Di seguito alcuni esempi di possibili progetti, vi ricordo che avete la possibilità di proporre quelli che volete.

Progetto numero 1. Implementazione di un server FTP.

L'obiettivo di tale progetto è l'implementazione di un client FTP e di un server FTP multithreaded in grado di implementare le seguenti funzionalità:

get: comando che permette di trasferire un file remoto nella directory locale al Client (comando RETR del protocollo);

put: comando che permette di trasferire un file locale nella directory locale al server (comando STOR del protocollo);

dir: comando che permette di ottenere la lista dei file nella directory remota (comando LIST del protocollo);

quit: per chiudere la connessione (comando QUIT del protocollo).

Si vogliono inoltre implementare i seguenti comandi del protocollo:

USER: per acquisire il nome dell'utente;

PORT: per comunicare la porta su cui trasferire i dati;

CWD: per cambiare la directory;

PWD: per conoscere la directory di lavoro;

DELE: per cancellare un file remoto.

Nel definire l'interazione tra il Client FTP ed il Server scritto dall'utente, si vogliono osservare le raccomandazioni rfc 959 che definiscono il protocollo FTP.

Progetto numero 2. Realizzazione di un Sistema SMTP.

L'obiettivo di tale progetto la realizzazione di un sistema Client/Server SMTP (Simple Mail Transfer Protocol) per l'invio della posta elettronica secondo lo standard RFC 821.

Il server SMTP dovrà svolgere le seguenti funzionalità:

- Per ogni utente registrato, mantenere una coda dei messaggi in uscita e una mailbox per i messaggi provenienti da altri utenti.
- Ricevere ed inviare i messaggi da un'applicazione per la posta elettronica di un utente locale registrato.
- Salvare i messaggi in arrivo e diretti ad un utente locale nella mailbox relativa. Quest' ultima dovrà essere realizzata come un file avente il formato richiesto da un'applicazione per la posta elettronica (ex: POP3 RFC 1939). A tale proposito può essere utile la consultazione della RFC 822.

Il Client SMTP dovrà recapitare ciascun messaggio di un utente all' SMTP server locale. Quest' ultimo lo inoltrerà all' SMTP server dell'utente destinatario del messaggio

Progetto numero 3. Gestione di una directory condivisa.

L'obiettivo del progetto è la realizzazione di un sistema Client/Server per la gestione di una directory condivisa. Quest' ultima è ospitata sul server e contiene un insieme di file ai quali i client possono accedere. Quando un nuovo client si collega al server, quest' ultimo gli invia la lista dei file presenti nella directory. Ciascun client può richiedere ogni file presente nella directory condivisa in modalità scrittura/lettura o solo lettura. Soltanto un client alla volta può accedere a un file in modalità scrittura/lettura, mentre più client possono accedere allo stesso file in modalità solo lettura. Una volta accettata una richiesta, il server invia copia del file richiesto al client. Quest' ultimo, una volta completata l'elaborazione

sulla copia del file richiesto, comunica al server il completamento dell'operazione e, qualora il file fosse stato aperto in modalità lettura/scrittura, invia al server una copia del file, per l'aggiornamento.

Più nel dettaglio, il server deve;

Attendere richieste di connessione dai client. Una connessione dura finché il client non ne richiede esplicitamente l'abbattimento. Durante ciascuna connessione un client può accedere a più file, anche contemporaneamente, purché distinti.

Una volta stabilita una nuova connessione verso un client, inviare a quest'ultimo la lista dei file contenuti nella directory condivisa.

Inviare la lista dei file presenti nella directory condivisa al client ogni volta che questo ne fa richiesta (questa funzionalità permette di comunicare al client cambiamenti nella directory condivisa, es. cancellazione/inserimento nuovi file).

Quando il client richiede un file <nomefile>, il server deve verificare che <nomefile> risieda nella directory condivisa. In caso contrario il server deve inviare un messaggio di errore al client. Qualora <nomefile> sia presente nella directory condivisa e il client lo richieda in lettura/scrittura, il server deve verificare che esso non sia in uso nella stessa modalità da parte di un altro client, altrimenti deve notificare al client l'impossibilità di accedere al file nella modalità richiesta. Infine, se la richiesta del client è accettata, il server deve inviare una copia del file richiesto al client.

Quando un client comunica al server il completamento di un'operazione di accesso, il server agisce in due modi distinti a seconda del criterio di accesso usato dal client:

Se il client ha effettuato l'accesso in modalità lettura/scrittura allora il server attende che il client gli invii copia (eventualmente modificata) di <nomefile>, poi sovrascrive la copia presente nella directory condivisa e infine invia una conferma al client.

Se l'accesso è avvenuto in modalità solo lettura il server dà conferma al client che l'operazione è conclusa. Si noti che in questo caso il client non invia una copia di <nomefile> al server.

Quando un client richiede la chiusura della connessione, il server esaudisce la richiesta senza ulteriori conferme.

Il client invece deve:

Una volta stabilita la connessione verso il server, attendere da quest'ultimo la lista dei file contenuti nella directory condivisa.

Ogni volta che l'utente richiede l'accesso a un file della directory condivisa, il client deve farne richiesta al server. A tale scopo:

Il client mette a disposizione dell'utente un'interfaccia che permette di scegliere un file tra quelli presenti nella directory condivisa, specificando una modalità di accesso.

Se una richiesta non va a buon fine (es. tentativo di accesso a file non esistente oppure accesso in modalità lettura/scrittura a un file già aperto da un altro client nella stessa modalità) il client stampa un messaggio di errore per l'utente.

Il client non deve permettere all'utente di richiedere file il cui nome non compare nella lista di cui è in possesso o due volte lo stesso file.

Quando un utente comunica al client che ha terminato la lettura e/o modifica di un file precedentemente scaricato dalla directory condivisa, il client deve completare l'operazione presso il server, secondo le modalità discusse sopra.

Quando l'utente richiede al client di chiudere la connessione presso il server, il client deve prima completare tutte le operazioni pendenti, poi chiudere la connessione. Per chiudere la connessione il client invia un messaggio al server e quindi abbate la connessione, senza aspettare conferma.

Si osservi che il client non deve mettere a disposizione dell'utente un editor di testo, ma soltanto un'interfaccia per visualizzare l'elenco dei file presenti nella directory condivisa e impartire i comandi previsti nelle specifiche.