

Progettazione del Software

Prova Pratica

16/07/2020 - tempo a disposizione 2h

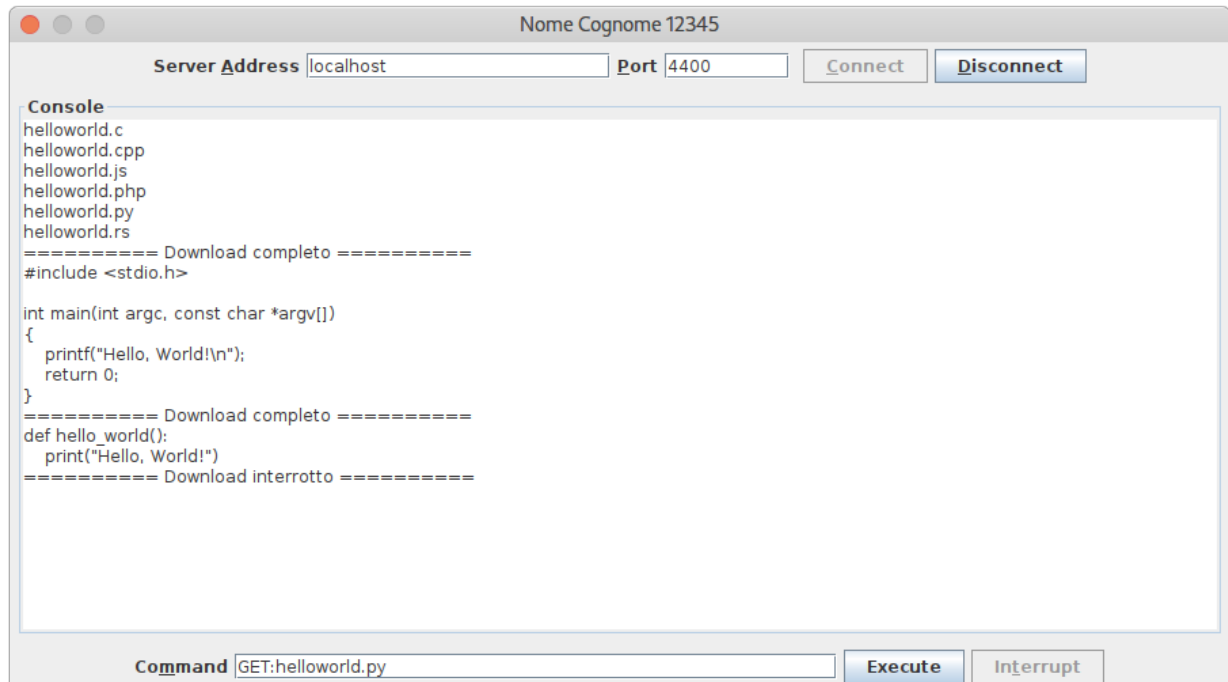
Si vuole realizzare un'applicazione *client-server* che per la condivisione ed il download di file testuali.

Il server ospita i file condivisi per i client, che, una volta connessi al server, possono eseguire tre comandi: 1) LIST: ottiene la lista dei file condivisi; 2) GET:*nome_file*: richiede il download del file dal nome *nome_file*. 3) INTERRUPT: richiede di interrompere l'eventuale download in corso. Il client permette, tramite l'interfaccia grafica, l'invio dei comandi che vengono inseriti in un campo testuale da parte dell'utente, la ricezione, e la visualizzazione delle risposte del server. Il server è *multithreading* ed accetta connessioni da parte di più client. Ciascun client è in grado di interagire con il server indipendentemente dagli altri client (le letture possono avvenire in parallelo).

La comunicazione è basata unicamente su scambio di stringhe. Tutte le stringhe sono inviate da client a server e viceversa utilizzando il carattere di fine linea come separatore.

Durante l'esame il server sarà raggiungibile al seguente indirizzo:

- Indirizzo IP: 80.211.232.219
- Porta: 4400



Si richiede la realizzazione del client, con interfaccia grafica e networking, in grado di comunicare con il server multithreading (fornito).

L'interfaccia grafica del client dovrà essere composta da un frame che abbia come titolo *nome cognome matricola* dello studente, da un pannello centrale contenente un'area di testo non editabile, e da tre campi testuali editabili per l'indirizzo, la porta del server, e l'inserimento dei comandi, e da quattro pulsanti che realizzino le seguenti funzioni: 1) *Connect*: permette di connettersi al server utilizzando indirizzo e porta specificati nei campi testuali; 2) *Disconnect*: permette di chiudere la connessione al server; 3) *Execute*: permette di eseguire il comando inserito nel campo di testo dall'utente; 4) *Interrupt*: permette di interrompere l'esecuzione del comando corrente;

L'area di testo centrale dovrà contenere il contenuto delle risposte ricevute dal server, e dovrà inoltre non essere editabile, e permettere lo *scrolling* del contenuto qualora quest'ultimo superi la dimensione visibile dell'area di testo (si faccia riferimento alla classe `JScrollPane` nelle API di Swing).

Suggerimento: Si consiglia di impostare la dimensione dell'area di testo centrale in modo tale da contenere 80 colonne per 25 righe, e di impostare la dimensione del campo testo dei comandi a 20 caratteri.

Si implementi il seguente protocollo:

- All'avvio solamente il pulsante *Connect* deve essere abilitato.
- Alla pressione del pulsante *Connect*, il client invia una richiesta di connessione al server utilizzando *indirizzo e porta indicati negli appositi campi*.
- Una volta stabilita correttamente la connessione, il client deve abilitare i pulsanti *Execute* e *Disconnect*. Il client deve inoltre cancellare tutto il contenuto dell'area di testo centrale. Alla pressione del pulsante *Execute*, il client deve inviare il contenuto del campo comando al server, abilitare il pulsante *Interrupt*, e disabilitare i pulsanti *Execute* e *Disconnect*. Alla pressione del pulsante *Interrupt*, durante il download, il client dovrà inviare la stringa "INTERRUPT" al server, che provvederà ad interrompere il processamento. È ammissibile premere il pulsante *Execute* anche se il contenuto è vuoto, ed equivale ad inviare un comando non riconosciuto al server (che risponderà con un errore).
- Una volta ricevuto un comando, il server inizierà ad inviare, ad intervalli regolari, stringhe al client, che dipenderanno dal comando richiesto:
 - Alla ricezione del comando `LIST`, il server restituirà la lista dei nomi dei file condivisi che è possibile scaricare, inviandoli uno per riga, una riga alla volta.
 - Alla ricezione del comando `GET:nome_file`, il server verificherà l'esistenza del file, e, qualora esista, inizierà ad inviarne il contenuto al client, una riga alla volta. Se invece il file con il nome richiesto non esiste, il server invierà la stringa "File Not Found". In entrambi i casi, al termine dell'invio della risposta il server invia la stringa "END" al client a segnalare la completa esecuzione del comando, mentre qualora il client richieda l'interruzione dell'invio, tramite il comando `INTERRUPT`, il server risponderà con la stringa "INTERRUPTED", e rimarrà in ascolto di nuovi comandi.
 - Per qualsiasi altra stringa ricevuta, il server risponderà con il messaggio "ERROR", ad indicare che il comando richiesto è errato, e chiuderà immediatamente la connessione con il client.

Alla ricezione di ciascuna risposta del server, il client deve:

- Se si tratta della stringa "END", appendere alla fine dell'area di testo la stringa "Download completato", eventualmente rendendola più evidente aggiungendo dei separatori (come da figura). Il client dovrà inoltre disabilitare il pulsante *Interrupt*, e riabilitare i pulsanti *Execute* e *Disconnect*.
- Se si tratta della stringa "INTERRUPTED", appendere alla fine dell'area di testo la stringa "Download interrotto", eventualmente aggiungendo dei separatori come sopra.

Il client dovrà inoltre disabilitare il pulsante *Interrupt*, e riabilitare i pulsanti *Execute* e *Disconnect*.

- Se si tratta della stringa “ERROR”, appendere alla fine dell’area di testo la stringa “Comando errato”. Il client dovrà inoltre disabilitare il pulsante *Interrupt*, e riabilitare i pulsanti *Execute* e *Disconnect*.

N.B.: Poiché la connessione è stata chiusa dal server, per poter continuare ad inviare comandi sarà necessario disconnettere e riconnettere il client.

- Tutte le altre stringhe vanno aggiunte all’area di testo come ricevute, in quanto fanno parte del contenuto della risposta.

Suggerimento: Al fine di semplificare la fase di debug si suggerisce di far stampare a terminale tutte le stringhe ricevute dal server.

- Alla pressione del pulsante *Disconnect*, il client deve inviare la stringa “DISCONNECT”, chiudere tutti i canali di comunicazione generati in fase di connessione, e deve inoltre abilitare nuovamente il pulsante *Connect* in quanto deve essere possibile instaurare una nuova connessione senza che sia necessario il riavvio del client. Il contenuto dell’area di testo deve rimanere visibile, fino alla creazione di una nuova connessione o alla chiusura del client.