

Prova Pratica

Progettazione del Software

10/06/2019 - tempo a disposizione 2h

Si vuole realizzare un'applicazione client-server che permetta di accedere in remoto a due files contenenti simboli esadecimali: un file conterrà le lettere da A ad F, ed un altro cifre numeriche da 1 a 9. Un server multithreading accetta connessioni da più client. I client una volta connessi possono avviare una sessione di download dei simboli esadecimali contenuti nei files. Una volta avviata la sessione, il server invia ininterrottamente 16 simboli alla frequenza di 1 simbolo/secondo. I simboli ricevuti dal server sono visualizzati dal client nella Text Field con etichetta "hexadecimal". Durante la trasmissione dei simboli il client può decidere da quale dei due insiemi di simboli ricevere il successivo, o di interrompere la trasmissione. L'avvio di una nuova trasmissione, deve comportare la sovrascrittura del contenuto delle Text Field con le nuove informazioni. 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.

Nome Cognome 1234567

IP Address: 127.0.0.1 Port: 4400 [Connect] [Disconnect]

☐ A - F ☐ 0 - 9 ☒ hexadecimal DBA61

decimal 899681

binary 11011011101001100001

[Start] [Stop] [Converti]

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

L'interfaccia grafica del client dovrà essere composta come da figura. In particolare, il frame deve avere come titolo il *nome*, il *cognome* e la *matricola* dello studente. I 2 campi testuali dovranno indicare l'indirizzo IP e la porta a cui connettersi (IP e porta per il testing verranno comunicati in aula). I pulsanti *Connect* e *Disconnect* permettono di gestire la connessione col server. Una volta avvenuta la connessione, deve essere possibile o avviare il download (pressione del tasto *Start*) o disconnettersi dal server. A download avviato deve essere possibile interromperlo con la pressione del tasto *Stop* e deve essere possibile premere i tasti *A-F* e *0-9*. Terminata la trasmissione deve essere possibile effettuare la conversione del numero esadecimale sia in decimale che in binario, attraverso la pressione del pulsante *Converti*. Le Text Field non devono essere editabili.

Gestire correttamente la possibilità di premere i pulsanti. In particolare:

- All'avvio solamente il pulsante *Connect* è abilitato. Non deve essere possibile avviare 2 connessioni dallo stesso client spingendo ripetutamente il bottone;
- *Start* e *Disconnect* possono essere premuti solo se la connessione col server è già avviata;

- Dopo la pressione di *Start* gli unici pulsanti che possono essere premuti dovranno essere *Stop*, e *A-F*, *0-9* per la scelta dell'insieme di simboli da leggere;
- Alla pressione di *Stop* deve essere possibile premere il pulsante *Converti* per la conversione dal formato esadecimale al formato decimale/binario. Inoltre, dovranno tornare attivi i pulsanti *Start* e *Disconnect*, per effettuare un nuovo download o disconnettersi dal server.

Si implementi il seguente protocollo:

- Alla pressione del pulsante *Connect*, il client invierà una richiesta di connessione al server utilizzando Indirizzo IP e Porta indicati negli apposti campi.
- Alla pressione del tasto *Start* il client dovrà inviare al server la stringa "*start*", abilitare il pulsante *Stop*, disabilitare *Start* e *Disconnect* ed iniziare la ricezione dei simboli esadecimali di risposta inviati dal server. Inizialmente il server sceglierà il file contenente i simboli *A-F*. A download avviato i pulsanti *A-F* e *0-9* dovranno essere abilitati. Alla pressione del primo il client manderà la stringa "*lettere*", alla pressione del secondo la stringa "*cifre*". Il server manderà continuamente simboli in ordine sparso dell'insieme che si è specificato, fino alla ricezione di un altro simbolo.
- Il server inizierà ad inviare un simbolo al secondo fino ad un massimo di 16; L'insieme delle lettere è {"A","B","C","D","E","F"}, mentre l'insieme delle cifre è {"1","2","3","4","5","6","7","8","9"}. Alla ricezione di ogni simbolo esadecimale, il client dovrà aggiornare opportunamente la Text Field con etichetta "hexadecimal".
- Alla pressione del tasto *Stop*, il client deve inviare il comando "*stop*". Il server risponderà inviando "+" a fine trasmissione. Alla ricezione della stringa "+" (che non deve essere visualizzata), il client deve smettere di ricevere stringhe, disabilitare i pulsanti *Stop*, *A-F* e *0-9*, e riabilitare *Start* e *Disconnect*. Inoltre, dovrà essere abilitato il pulsante *Converti*. Alla pressione del pulsante *Converti* il client dovrà convertire il numero esadecimale ricevuto dal server sia in decimale che in binario, e riempire correttamente le Text Field con etichette "decimal" e "binary". Nel caso in cui si avvii una nuova comunicazione dopo l'interruzione (si preme di nuovo *Start*), preliminarmente il contenuto di ogni Text Field deve essere cancellato ed il pulsante *Converti* disabilitato.

HINT: Per convertire un numero esadecimale a decimale si consiglia l'uso del metodo `parseUnsignedLong(String hexadecimal, int hexadecimal_base)` della classe `Long`. Invece, per convertire un numero decimale a binario si consiglia l'uso del metodo `toBinaryString(long decimal)` che prenderà in input il numero decimale precedentemente convertito.

- Se non viene premuto il pulsante *Stop*, il server una volta inviati i 16 simboli invierà il carattere "+" per indicare al client la fine della trasmissione.
- Alla pressione del pulsante *Disconnect*, deve essere inviata la stringa "*disconnect*", e devono essere chiusi i canali di comunicazione generati in fase di connessione. Deve inoltre essere abilitato il solo pulsante *Connect* e deve essere cancellato il contenuto delle Text Field. Inoltre, deve restare possibile instaurare una nuova connessione senza che sia necessario il riavvio del client.