

Protocollo di comunicazione

Sequenza di operazioni per il pairing:

1. Il server viene avviato e si mette in attesa di una connessione del client
2. Il client viene avviato e manda la richiesta di connessione con il server dato un indirizzo IP e una porta
3. Comincia la comunicazione tra client e server
 - 3.1. **ATTENZIONE:** Se il server non riesce a mandare un messaggio per intero, lo manderà un'altra volta. Quindi i messaggi che arrivano al client potrebbero essere ripetuti.
 - 3.1.1. Esempio: Il server manda 99 byte su 100. Il client riceve 99 byte che riesce comunque a leggere. Il server rimanda lo stesso messaggio finché non riuscirà a mandare 100 byte.
4. La connessione viene interrotta quando il client decide di chiudere la connessione (grazie ad un apposito messaggio) oppure in caso di errori
5. A seguito di un errore il server si resetta e accetta un'altra connessione in entrata

Messaggi (può subire variazioni):

- Client manda un comando da eseguire su una finestra - Server risponde con un messaggio che indica se il comando è stato eseguito
 - Sintassi client
 - command <window handle> <process ID> <numero di tasti> <tasto 1> <tasto 2> <...>
ATTENZIONE: ricorda di inserire uno spazio dopo l'ultimo comando!
Lista dei tasti supportati:
 - [https://msdn.microsoft.com/it-it/library/windows/desktop/dd375731\(v=vs.85\).aspx](https://msdn.microsoft.com/it-it/library/windows/desktop/dd375731(v=vs.85).aspx)
 - Sintassi server
 - command ok/error
- Server manda il numero di nuove finestre
 - Sintassi server
 - newwindows <numero di finestre>
- A seguito del precedente messaggio il server manda le nuove finestre (una per volta):
 - Sintassi server
 - <nome finestra>'\n'
<window handle>'\n' (da non visualizzare)
<pid processo>'\n'
<dimensione icona (int) espressa in byte (char)>'\n'
<icona> (facoltativo)
 - <dimensione icona> può assumere i seguenti valori:
 - > 0 : l'icona esiste e ha la dimensione specificata
 - == 0 : l'icona non esiste

- == -1 : l'icona esiste ma c'è stato un errore durante il salvataggio. Questi errori si verificano molto spesso per le applicazioni che sono presenti (forse) in una parte del menù start di windows. Quindi non sono direttamente aperti dall'utente, infatti non sono presenti nella barra in basso.
 - == -2 : l'icona è troppo grande e non ci entra nel buffer (non dovrebbe verificarsi mai, ma...)
 - == -3 : errore di lettura nel file dell'icona (sono stati letti 0 byte dal file)
 - == -4 : errore generico
 - ATTNEZIONE: non considerare l'icona come una stringa. I delimitatori non funzioneranno perché l'icona è composta da byte codificati secondo il formato .ico.
- Il server manda la lista delle finestre chiuse (in un unico messaggio)
 - Sintassi server
 - closed <numero finestre chiuse>\n'
 - <window handle> <process ID 1>\n'
 - <window handle> <process ID 2>\n'
 - ...
- Il server manda il nome della finestra in focus (se il focus cambia)
 - Sintassi server
 - focus <window handle> <process ID>
 - OPPURE (se non è presente una finestra in focus)
 - focus nullwindow
 - OPPURE (se è il sistema operativo a tenere il focus)
 - focus windowsoperatingsystem
- Client manda il messaggio di terminazione - Server risponde con un ack:
 - Sintassi client
 - stop
 - Sintassi server
 - stop

Prendi in considerazione che:

- Il server può essere raggiungibile o non raggiungibile. Infatti l'utente-server può decidere se avviare o disabilitare la connessione. Nel caso il server non sia raggiungibile la richiesta di connessione del client non avrà esito positivo.
- Il client deve sempre tenere una connessione TCP aperta quando si connette con un server.
- Il client dice al server cosa vuole ricevere.
- La connessione con il server viene interrotta quando il client chiude la connessione TCP (quindi non c'è bisogno di un messaggio esplicito per la chiusura della connessione).
- Il client deve connettersi con più server (in generale).

Tutorial

[https://msdn.microsoft.com/it-it/library/windows/desktop/ms738545\(v=vs.85\).aspx](https://msdn.microsoft.com/it-it/library/windows/desktop/ms738545(v=vs.85).aspx)

Per far funzionare il sistema seguendo questo tutorial, il client deve scrivere in continuazione sul socket del server perché, se il server non riesce a leggere niente sul socket, la comunicazione viene interrotta.

Posso introdurre nel server una sleep che rallenta un po' le letture (però bisogna sempre prendere in considerazione lato client la possibilità che il server si disconnetta e quindi bisogna riconnetterlo automaticamente).

Oppure si potrebbe introdurre un messaggio di disconnessione mandato dal client al server.