

Progetti per il Corso di Reti e Servizi

Indice dei progetti:

- server chat di Roberto Foglietta
 - Ultima versione disponibile: `libsocket-052.tar.gz` – Server LibSocket r0.52
 - Il pacchetto contiene i sorgenti e anche l'eseguibile che puo' essere avviato con `libsocket/server` e per default si mette in ascolto sulla porta 1616
 - Il server e' interrogabile anche da telnet e ha un solo utente di default `login myadmin myadmin` con il quale si puo' esplorare tutti i comandi `help help`
 - Il server non salva nulla su disco nemmeno il database degli utenti quindi al riavvio riparte come nuovo senza operazioni aggiuntive
 - Il server puo' essere ricompilato con `cd libsocket; make distclean; make`
- client chat di Pietro Marrone
 - Ultima versione disponibile: `Chat_v2.2.tar.gz` – Client Chat Versione 2.2
 - Il pacchetto contiene i sorgenti e anche il bytecode che puo' essere avviato con

```
cd Chat/build; java -cp ./log4j.jar:./chat.jar it/alma/chat/client/ui/ClientSwing
```

- Il client richiede Java Runtime Edition (JRE) versione 1.5 o maggiore.

Server Chat

Si tratta di sviluppare un server chat che permetta:

- una comunicazione via ASCII-7bit compresi i comandi di connessione
 - socket TCP/IP in C/C++ (comprensione profonda della loro natura metafisica!!!)
 - gestione contemporanea del servizio su più connessioni
 - sistemi di routing basati sulla registrazione dei clients

Le seguenti sono gli user cases:

- la registrazione e servizio di più utenti contemporaneamente:
 - il client crea un nuovo utente o oppure
 - il client si autentica con username unico e password segreta
 - il client puo' visualizzare gli utenti online e mandargli messaggi
 - il client puo' sconnettersi

Eventuale riutilizzabilità del progetto:

- salvataggio su disco delle informazioni interne al server in particolare il db utenti (attualmente tutto e' solo in RAM)
- messa in sicurezza del server per il corso di Sicurezza in maniera da usarlo anche sulle porte inferiori alla 1024.
- p2p: l'ip/porta del client associato allo username richiesto dopo autorizzazione.
- chatroom: possibilita' di entrare o creare delle stanze tematiche.
- mail: conferma di registrazione utente via mail.

IN PROGRESS

MAJOR:

- 25-10-2006, DONE: il login da un altro socket di un utente gia' loggato deve chiudere la connessione piu' vecchia
- 25-10-2006, DONE: l'utente myadmin (ma probabilmente anche qualsiasi altro utente) collegato a socket zero non funziona (-1).
- 25-10-2006, DONE: il tempo fra la spedizione di un messaggio e la sua ricezione non dipende piu' dal numero di

connessioni attive.

- 28-11-2006, DONE: bugfix il menu contestuale non funzionava correttamente mostrando l'help base anche per utente e admin (semafori).

MEDIUM:

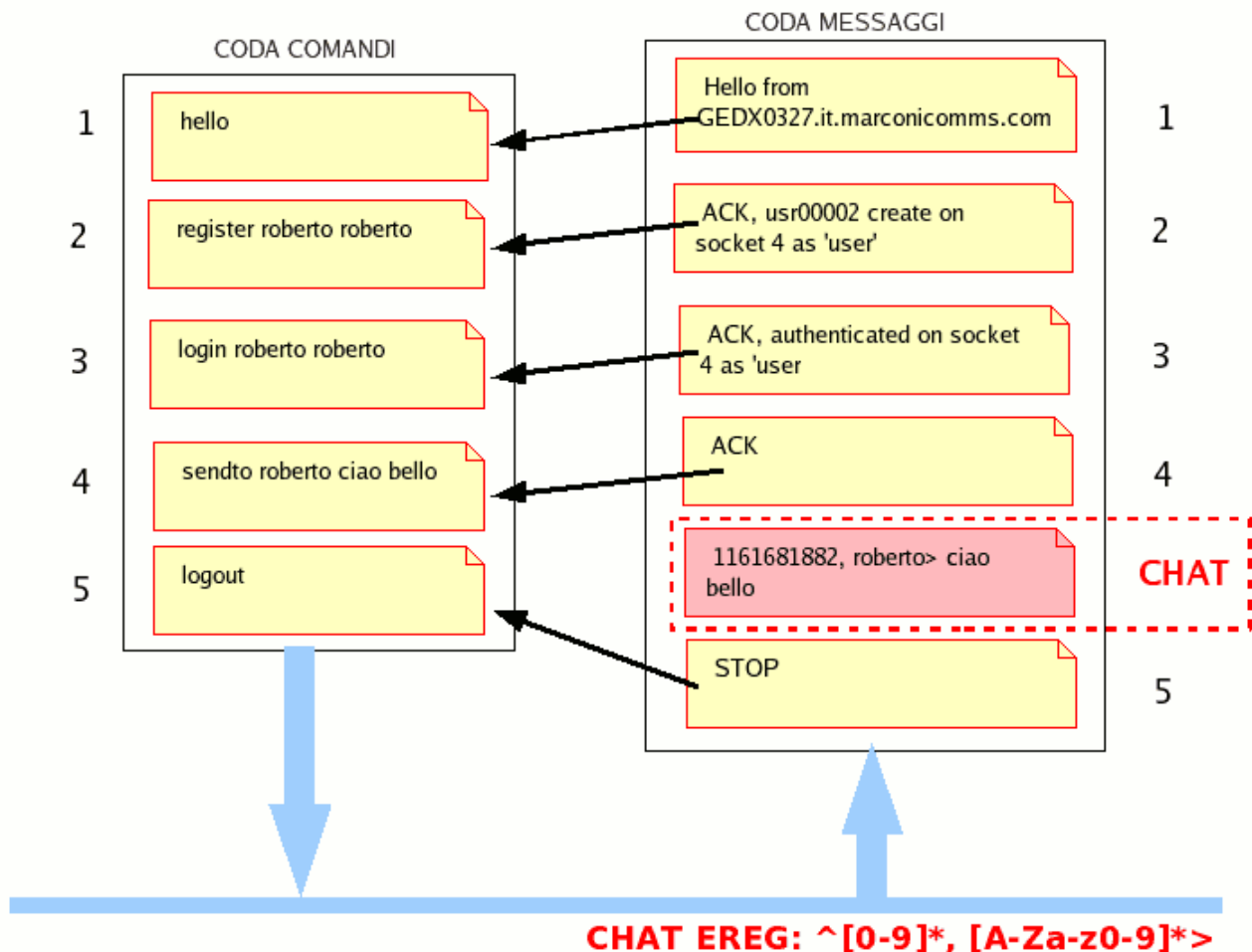
- 25-10-2006, DONE: la direttiva register permette di ricevere messaggi ma non permette di accedere allo user level.
- 25-10-2006, PENDING: imporre maxconnections a 1 e analizzare tutti i casi di quello che succede alla connessione appesa.
- 28-11-2006, DONE: Pietro ha riportato che il comando whoisol riporta ERRS se viene interrogato troppo di frequente (semafori).
- 29-11-2006, DONE: generare una cache per whoisol oppure creare una funzione ALERT che avverta spontaneamente i chat clients

MINOR:

- 24-10-2006, DONE: manca whoami, manca l'IP della contro parte.
- 25-10-2006, DONE: il comando LIST e' stato rimosso dalla view comandi degli utenti.
- 28-11-2006, DONE: tutti i comandi in level NONE e USER ritornano ACK quando riescono.
- 28-11-2006, DONE: l'utente che ripete il login sulla stessa connessione si vede sbattere fuori
- 28-11-2006, PENDING: diversificare LOGOUT dagli altri comandi di uscita

Schema della logica di comunicazione

- Schema della logica di comunicazione: si tratta di una logica accettabile solamente nell'ipotesi che nessuna comunicazione venga completamente perduta. In buona sostanza si delega agli strati sottostanti il problema di evidenziare e gestire la perdita/corruzione dei dati trasmessi.



- Per supplire ad eventuali carenze degli strati sottostanti di assicurare che ogni comando o messaggio e relative risposte vengano effettivamente recapitate si e' attivata, su consiglio di Pietro, la gestione di numeri di progressione al massimo di 6 cifre decimali con la seguente sintassi `123456|<comando>` a cui seguirebbe `123456|>risposta>` dove, ovviamente, i due numeri coincidono.
- Le risposte ai comandi sono di quattro tipi ma il client chat utilizza solo i primi due:
 - singola linea `ACK, <risposta>\n\n`
 - multi linea `ACK, <intestazione>:\n<valore 1>\n...\n<valore N>\n\n`
 - valori o errore: per quei comandi di amministrazione tipo `GET`, `GETVALS` e `LIST`
 - codice di ritorno: `ACK`, `NAK`, `FAIL`, `ERRS`, `BUSY`, e `STOP`
- `ACK`, admitted command list for nobody level:
 - `hello hello: get hello from remote host`
 - `ML help help [cmd]: get the list of available commands or help about a command`
 - `bye bye: close the connection with remote host`
 - `exit exit: close the connection with remote host`
 - `quit quit: close the connection with remote host`
 - `register register nickname password [e-mail]: create a new user as specified`
 - `login login username password: to become the specified user`
- `ACK`, admitted command list for user level:
 - `logout logout: to drop the user identity (and close connection)`
 - `ML whois whois: visualizza una lista degli utenti registrati`
 - `whoami whoami: visualizza lo username, l'hostname del server e l'IP del client`

- `sendto sendto username "message ..."`
- ACK, admitted command list for admin level:
 - `get get [classpath] keyword|*`: report the keyword(s) contents
 - `getvals getvals [classpath] keyword`: report only the keywords contents
 - `set set [classpath] keyword value`: set the value in the keyword
 - `setup setup class|keyword [classpath] object [value]`: create the object with that name
 - `list list [classpath]`: lists the contenets of the class
- Come si vede sopra ci sono 3 livelli di accesso possibile al server: *nobody*, *user* e *admin*.
 - per entrare come amministratore occorre fare il login con `login myadmin myadmin`
 - per entrare come utente occorre fare il login con username e password.
 - per entrare come sconosciuto basta connettersi al server e per creare un account basta fare `register username password`

– Main.RobertoAFoglietta - 02 Nov 2006

Client Chat

Si tratta di sviluppare client Java che permetta:

- comprensione profonda delle classi socket in Java
- capacità p2p per trasferimenti dati/file che non devono instasare la banda del server
- tale capacità deve essere autorizzata dal client ricevente/destinatario

Le seguenti sono gli use cases del client:

- Connessione:
 - il client apre un canale di comunicazione (Socket) verso il server indicato alla porta indicata
- Registrazione:
 - il client crea una nuova utenza sul server fornendo username e password
- Login:
 - Il client effettua il login presso il server fornendo username e password (autenticazione)
- Lista Utenti Connessi:
 - Il client richiede la lista completa degli utenti connessi sul server
- Invio Messaggio ad Utente:
 - Il client invia un messaggio ad un altro utente connesso fornendo lo username del destinatario ed il messaggio
- Logout:
 - Il client richiede al server di effettuare il logout
- Disconnessione:
 - Il client chiude il canale di comunicazione (Socket) verso il server

RELEASE NOTE 2.2:

- Eliminato bug segnalato dal Roberto: il client ora gestisce la caduta del server.
- Aggiunto il supporto per la nuova sintassi del comando WHOISOL

RELEASE NOTE 2.0:

- Gestisce la eventualità che venga perso un pacchetto. La feature è stata gestita nella classe !CommandTokenizer, essa è un wrapper dei Command. I Command non sono stati modificati minimamente, oltre all'aggiunta della classe su indicata sono state necessarie solo poche modifiche alla inner class !CommandList.
- E stata inoltre irrobustita la gestione delle eccezioni in vari punti del codice.

RELEASE NOTE 1.1.2:

- Risolto il Bug rilevato da Foglietta quando non c'era nessun User on Line la Chat si Bloccava definitivamente

RELEASE NOTE 1.1:

- Password mascherata (***)
- Etichetta pulsante modificata da ai a Esegui
- Inserita una linea divisoria fra il pannello degli utenti on line e pannello messaggi
- Aggiunta etichetta su pannello "Utenti on Line"
- Aggiunto orario ad ogni messaggio visualizzato
- Migliorata la usability:
 - Se il focus si trova su Server, Porta o pulsante Connetti con invio ci si connette;
 - Dopo essersi connessi il focus va al testo da inserire per iniziare la chat
 - Con il focus sul testo da scrivere, basta premere invio per inviare il testo.

REQUIREMENTS:

- Richiede la versione JRE 1.5

LAUNCHER:

- cd build
- java -cp ./log4j.jar:./chat.jar it/alma/chat/client/ui/ClientSwing

Esempio di conversazione:



– Main.PietroMarrone - 10 Jun 2006

0506/progettireti.txt · Ultima modifica: 2007/06/07 11:27 da foglietta