Client SETI - Progetto 2025/2026

Implementazione completa del client per il protocollo SETI come descritto nel documento del progetto.

Struttura del Progetto

Il progetto è organizzato in moduli separati per una migliore manutenibilità:

- **client.h**: Header con dichiarazioni di funzioni e strutture
- **client_main.c**: Main del programma e loop interattivo
- **client_network.c**: Gestione socket TCP/UDP e comunicazione di rete
- **client_commands.c**: Implementazione dei comandi del protocollo
- **client_handlers.c**: Gestione delle risposte del server e thread UDP
- **client_utils.c**: Funzioni di utilità per codifica/decodifica
- **Makefile**: File per la compilazione

Compilazione

```bash
make
```

Per pulire i file oggetto:

```bash
make clean
```

Per ricompilare tutto:

```bash
make rebuild
```

Utilizzo

```bash
./client <id> <udp_port> <password> <server_ip:server_port>
```

Parametri:

- `id`: Identificativo del client (esattamente 8 caratteri alfanumerici)
- `udp_port`: Porta UDP per ricevere notifiche (< 9999)
- `password`: Password numerica (0-65535)
- `server_ip:server_port`: Indirizzo IP e porta del server

Esempio:

```bash
./client alice123 7878 1010 127.0.0.1:8888
```

Funzionalità Implementate

1. REGIS - Registrazione

Registra un nuovo client sul server con le credenziali fornite.

2. CONNE - Connessione

Connette un client già registrato usando ID e password.

3. LIST - Lista clienti

Richiede e visualizza l'elenco di tutti i clienti registrati sul server.

4. FRIE - Richiesta di amicizia

Invia una richiesta di amicizia a un altro client specificando il suo ID.

5. MESS - Invio messaggio

Invia un messaggio privato a un amico (massimo 200 caratteri).

6. FLOO - Flood

Invia un messaggio broadcast a tutti gli amici e agli amici degli amici (massimo 200 caratteri).

7. CONSU - Consultazione flussi

Consulta e rimuove i flussi in attesa:

- Richieste di amicizia (con possibilità di accettare/rifiutare)
- Messaggi privati ricevuti
- Messaggi di flood
- Risposte alle richieste di amicizia inviate

8. IQUIT - Disconnessione

Disconnette il client dal server in modo pulito.

Notifiche UDP

Il client riceve notifiche UDP asincrone dal server su un thread dedicato:

- **[0XX]**: Nuova richiesta di amicizia
- **[1XX]**: Richiesta di amicizia accettata
- **[2XX]**: Richiesta di amicizia rifiutata
- **[3XX]**: Nuovo messaggio privato
- **[4XX]**: Nuovo messaggio di flood

Dove XX è la codifica esadecimale little-endian del numero di flussi non consultati.

Dettagli Implementativi

Codifica dei Messaggi

- **Password**: 2 byte in little-endian
- **Porta**: 4 caratteri con zeri iniziali (es. "0052")
- **ID**: Esattamente 8 caratteri alfanumerici
- **Messaggi**: Massimo 200 caratteri, non possono contenere "+++"
- **Terminatore**: Tutti i messaggi TCP terminano con "+++"

Thread UDP

Il client crea un thread dedicato all'ascolto delle notifiche UDP che:

- Resta sempre attivo in background
- Decodifica le notifiche e le visualizza con informazioni chiare
- Non blocca l'interfaccia principale

Gestione Errori

Il client è robusto e gestisce:

- Errori di connessione
- Messaggi malformati

- Disconnessioni improvvise del server
- Input utente non valido

Modalità Verbose

Il client stampa automaticamente:

- Tutti i messaggi inviati con etichetta `[SEND]`
- Tutti i messaggi ricevuti con etichetta `[RECV]`
- Notifiche UDP dettagliate con tipo e numero di flussi
- Stati delle operazioni (connessione, registrazione, ecc.)

Note Importanti

1. **Formato ID**: Deve essere esattamente 8 caratteri alfanumerici
2. **Porte**: Devono essere < 9999 come specificato
3. **Password**: Numero intero tra 0 e 65535
4. **Messaggi**: Non possono contenere "+++" (terminatore del protocollo)
5. **Thread UDP**: Avviato automaticamente all'avvio del client

Esempio di Sessione

```
=== Client SETI ===
ID: alice123
Porta UDP: 7878
Password: 1010
Server: 127.0.0.1:8888
====================

========== MENU ==========
1. Registrarsi (REGIS)
2. Connettersi (CONNE)
...
Scelta: 1

[SEND] REGIS alice123 7878<password_bytes>+++
[RECV] WELCO+++
Registrazione accettata!
```

```
Scelta: 3
[SEND] LIST?+++
[RECV] RLIST 002+++
[RECV] LINUM alice123+++
[RECV] LINUM bob12345+++

=== Lista Clienti (2) ===
1. alice123
2. bob12345
======================
```

Test e Debug

Per testare il client:

1. Avviare il server
2. Avviare uno o più client con ID diversi
3. Testare registrazione, amicizie e messaggi
4. Verificare le notifiche UDP
5. Controllare i log verbose per debug

Conformità al Protocollo

Questo client implementa completamente il protocollo SETI 2025/2026 come specificato nel documento di progetto, rispettando:

- Tutti i formati dei messaggi
- Le codifiche little-endian
- I limiti di lunghezza
- La gestione dei flussi
- Le notifiche UDP