

IL PROGETTO – LE FEATURES

Il progetto scelto è il gioco multiplayer.

Le features implementate per il server sono:

- ✓ Controllo dei parametri da riga di comando;
- ✓ Verifica della eventuale presenza di altri server;
- ✓ Output dei parametri da riga di comando, entrata, uscita dei giocatori, tentativi di risposta, classifica aggiornata e ordinata, vincitore;
- ✓ Gestione corretta della connessione e disconnessione di client;
- ✓ Gestione corretta in caso di chiusura del terminale;

Le features implementate per il client sono:

- ✓ Verifica presenza del server;
- ✓ Visualizzazione informazioni su connessione e disconnessione di altri client;
- ✓ Visualizza la classifica ordinata alla conclusione del gioco;
- ✓ Prevista la modalità di test con invio automatico delle risposte;
- ✓ Gestione corretta in caso di chiusura del terminale o del server;

FILES

GAME.H

Il file contiene la lista delle librerie da includere, dichiarazioni delle funzioni principali che poi vengono usati dal server o client e le strutture comuni.

MAIN.C

Il file contiene la funzione main dove vengono passati i dati principali come i punti che deve raggiungere un client per vincere, il numero massimo dei giocatori e ovviamente il tipo di esecuzione: server o client. Se i dati soddisfano le regole del gioco si prosegue con l'esecuzione, altrimenti viene mostrato il messaggio d'errore.

SERVER.C

Il file, come dice il suo nome stesso, contiene le funzioni che deve eseguire un server e le strutture alle quali si deve appoggiare: salva nella lista i clienti connessi, gestisce i punti dei giocatori, gestisce gli eventi come connessione degli utenti e risposte alle domande.

CLIENT.C

Il file contiene le funzioni principali del client: esecuzione del client e lo scambio dei messaggi con il server.

COMMON.C

Il file contiene le funzioni di scrittura e lettura sicure che poi vengono usate da client e server.

PIDFILE.C

Il file che ci permette di localizzare i PID, controllare la loro esecuzione o rimuoverli. Il file proviene da un altro progetto, rsyslog, quindi non è stato prodotto dagli autori del progetto di SO.

SPECIFICHE DELLA IMPLEMENTAZIONE

CONTROLLO DEL FILE PID

All'avvio del server il programma verifica l'esistenza di un file con nome ben conosciuto. Se il file non esiste, possiamo assumere che nessun server sia in esecuzione. Se questo file esiste, si prosegue con la lettura; ci aspettiamo di leggere un numero intero che rappresenta il PID del processo che ha registrato il file. Se effettivamente è in esecuzione un processo con questo PID, possiamo supporre che sia il server (non è possibile comunque esserne certi). Se il processo con il PID specificato non esiste, possiamo supporre che il server precedentemente in esecuzione sia stato terminato e quindi non abbia avuto la possibilità di cancellare il PID file al termine dell'esecuzione. Il controllo del PID file da parte del client è identico se non per il fatto che non è necessario registrare il proprio PID.

SCAMBIO MESSAGGI TRAMITE FIFO

Se il server è avviato correttamente viene aperta una FIFO in lettura dove il server si aspetta di leggere le richieste di connessione, inizialmente, e in seguito risposte e messaggi di servizio. Quando un client viene accettato, viene stabilito un ulteriore canale di comunicazione dal server verso il client. Questo è necessario perché un messaggio dal client al server deve essere consumato una singola volta, mentre un messaggio dal server ai client potrebbe dover essere consumato più volte, una volta per ogni client.

Ogni messaggio ha una dimensione fissa indipendentemente dai campi effettivamente utilizzati. Il primo campo di ogni messaggio specifica per convenzione quali campi devono essere ispezionati e quali possono essere ignorati.

IMPLICAZIONI SULLA SICUREZZA

Poiché il sistema è realizzato usando FIFO, anziché socket, non è possibile garantire l'autenticità dei messaggi. È quindi possibile per un client malintenzionato inviare messaggi fasulli. Per questo motivo è necessario assumere che i processi siano legati da una natura collaborativa, anche per via del fatto che sono in esecuzione sulla stessa macchina e non all'interno di una rete.

CLASSIFICA GIOCATORI

Il server mantiene una classifica ordinata rispetto al punteggio dei giocatori. Per realizzarla, abbiamo deciso di utilizzare una *linkedlist*.