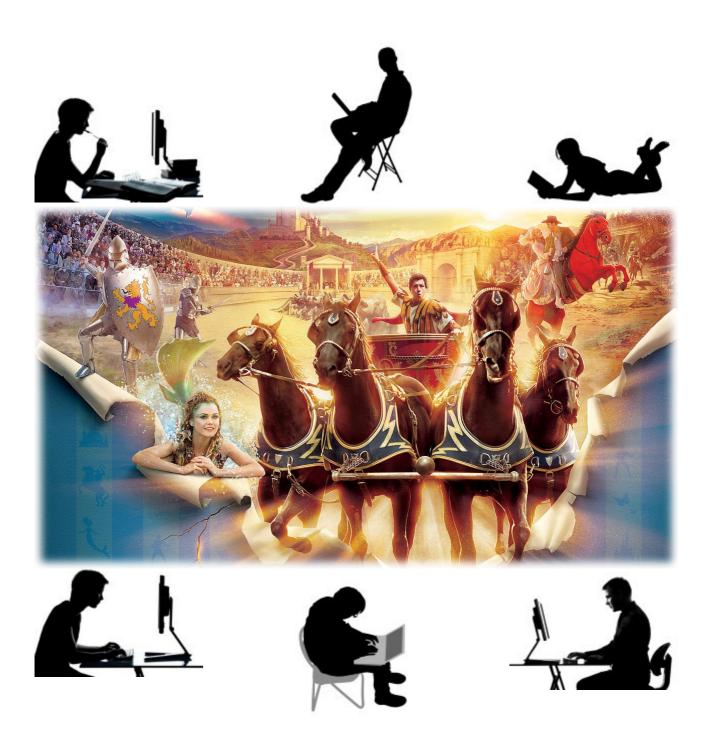
SHARED TALE



PERCHE LA FANTASIA......PUÒ ESSERE STIMOLATA......

Descrizione

"Shared tale" è uno strumento che permette di creare e condividere un racconto con gli utenti registrati. Tramite terminale, è possibile registrarsi al servizio ed accedere in maniera molto semplice ed intuitiva ai servizi offerti.

L'obiettivo di "Shared Tale" è la creazione di un racconto i cui autori sono (possono essere) TUTTI gli utenti della comunità.

Difatti ognuno potrà dare il suo contributo scrivendo il proprio capitolo e cosi far andare avanti la storia e, perché no, dare spunto ad altri utenti per il continuo.

"PERCHE LA FANTASIA......PUÒ ESSERE STIMOLATA....."

Ritengo che a volte la fantasia abbia bisogno di uno spunto per svilupparsi e prendere la giusta direzione, con "Shared Tale" questo è possibile.

Cosa aspettate? Ritenete di essere abbastanza bravi da tenere i lettori inchiodati allo schermo o semplicemente siete uno di questi? Iscrivetevi dunque, di sicuro vivrete un esperienza unica, e lo farete insieme ad altre persone. In seguito alla registrazione è possibile effettuare il login e scegliere il ruolo da interpretare :

- Lettore : ti permetterà di visionare i progetti già esistenti
- Scrittore : ti permetterà di contribuire a scrivere un progetto semplicemente scegliendolo fra quelli esistenti ed aggiungendo un capitolo.

Descrizione Tecnica

Il programma è scritto in C/Bash e testato con la versione 13 di Ubuntu (linux).

Un modello Client-Server con Client concorrenti, la concorrenzialità dei processi è garantita in due modi :

- Tra Client e Server in seguito all'apertura di un socket di comunicazione, quest'ultima avviene tramite "send" e "recv" con controllo di avvenuta ricezione
- Tra Clients con l'ausilio di semafori implementati in maniera che, se un utente è loggato e sta scrivendo, altri utenti non possono scrivere, devono attendere il completamento dell'operazione da parte del primo utente, possono però leggere.

Nella stessa directory del server sono presenti i files :

- "users.txt" il quale contiene tutti i nominativi degli utenti registrati e la loro password.
- "progetti.txt" che contiene la lista dei progetti disponibili, creati.
- "'nome progetto'.txt" sono i vari progetti esistenti.

Le funzionalità accedibili tramite il Client sono:

- Registrazione : username e password vengono spediti al server il quale li memorizza nel file users.txt.
- Login : username e password spediti al server che li confronta con gli utenti registrati nel file users.txt.
- Lettore : il server invia la lista dei progetti disponibili, leggendoli da progetti.txt al client che può scegliere quale leggere.
- Scrittore:
 - Nuovo progetto : il client invia il nome del nuovo progetto al server che lo crea e lo aggiunge in coda a progetti.txt.
 - Progetto esistente : il server invia la lista dei progetti al client che può scegliere quale modificare.

A questo punto il client invia al server nome e testo nuovo capitolo, il server si preoccupa di aggiungerlo al progetto (nuovo o esistente).

Alcuni controlli inseriti:

- Registrazione : Username già esistente.
- Login: Username o pass errati.
- Lettore : Progetto non presente, nome errato.

 Database progetti vuoto, creane uno nuovo.
- Scrittore nuovo progetto : Il nome appartiene ad un progetto già esistente.
- Scrittore vecchio progetto: Database progetti vuoto, creane uno nuovo.
- Semaforo : Se il client crasha mentre cerca di modificare un progetto, il server se ne accorge ed effettua la dovuta operazione su semaforo.

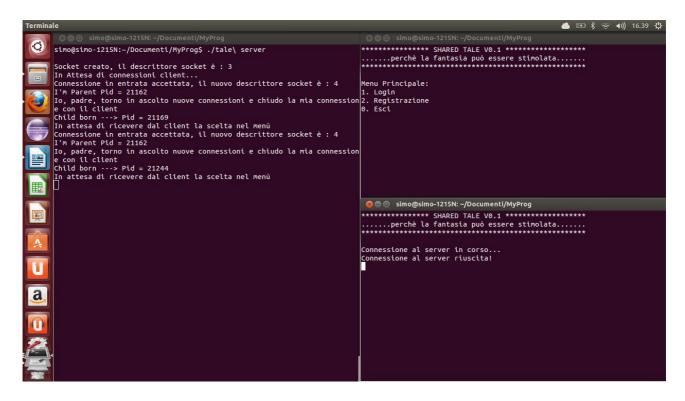
Se il semaforo per qualche ragione è già esistente all'avvio del server, lo distruggiamo e lo ricreiamo.

- In tutto il programma : controlli sull'avvenuta ricezione delle informazioni tra Client e Server.

P.S: nel codice, come commenti, sono presenti molte spiegazioni.

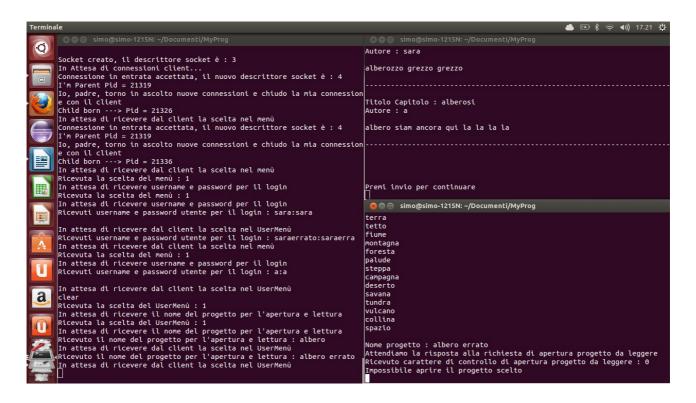
ESEMPI DI RUN

- Connessione al server, fork processo ogni client.

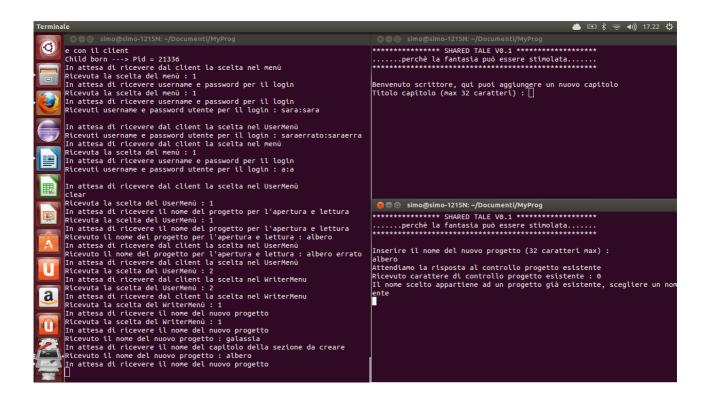


- Login effettuato ed errato

- Lettore progetto esistente e progetto non esistente



- Scrittore progetto, nome non esistente (quindi progetto creato) e nome già utilizzato



Utilizzo Semaforo, il primo Client riesce ad entrare, il secondo viene bloccato.

