

Descrizione Sintetica

Realizzare un sistema client-server che consenta a più utenti di giocare ad un gioco di conquista di territori. Si utilizzi il linguaggio C su piattaforma UNIX. I processi dovranno comunicare tramite socket TCP. Corredare l'implementazione di adeguata documentazione.

Descrizione Dettagliata

Il server manterrà una rappresentazione dell'ambiente che rappresenta un insieme di territori da conquistare. L'ambiente sia rappresentato da una matrice in cui gli utenti si potranno spostare di un passo alla volta nelle quattro direzioni: S, N, E, O. Le caselle saranno libere o di proprietà di un utente. Ogni utente, una volta connesso al server, potrà partecipare alla conquista; il server comunicherà all'utente il punto di partenza con coordinata (x,y) e il numero di territori da conquistare (stabilito dal server ed uguale per tutti i giocatori). La locazione di partenza sarà anche la prima conquista dell'utente. Dopo ogni passo l'utente riceverà l'informazione sull'effetto proprio movimento: se lo spostamento porterà su di un locazione libera questa diventerà di proprietà dell'utente; se la locazione di arrivo appartiene già ad un altro utente sarà possibile conquistarla con un lancio di dadi simulato (2 estrazione di numeri random da 1 a 6, uno per l'attacco l'altro per la difesa, in caso di pareggio vince la difesa). Quando un utente avrà conquistato il numero stabilito di territori o alla scadenza di un limite di tempo fissato, il server notificherà agli utenti la fine della sessione e il vincitore per poi generare una nuova sessione. Se la sessione scade prima della conquista del numero predefinito di territori il vincitore sarà colui che ne ha conquistati di più.

Per accedere al servizio ogni utente dovrà prima registrarsi al sito indicando password e nickname.

Non c'è un limite a priori al numero di utenti che si possono collegare con il server. Il client consentirà all'utente di collegarsi ad un server di comunicazione, indicando tramite riga di comando il nome o l'indirizzo IP di tale server e la porta da utilizzare. Una volta collegato ad un server l'utente potrà: registrarsi come nuovo utente o accedere al servizio come utente registrato. Il servizio permetterà all'utente di: spostarsi di una posizione, vedere la lista degli utenti collegati, vedere i territori degli altri utenti in gioco, vedere il tempo mancante, disconnettersi.

Il server dovrà supportare tutte le funzionalità descritte nella sezione relativa al client. All'avvio del server, sarà possibile specificare tramite riga di comando la porta TCP sulla quale mettersi in ascolto. Il server sarà di tipo concorrente, ovvero in grado di servire più client simultaneamente. Durante il suo regolare funzionamento, il server effettuerà il logging delle attività principali in un file apposito. Ad esempio, memorizzando la data e l'ora di connessione dei client e il loro nome simbolico (se disponibile, altrimenti l'indirizzo IP) e la data e l'ora delle conquiste.

Regole generali

Per gli studenti vecchio ordinamento:

Il server ed il client saranno realizzati in linguaggio C su piattaforma UNIX/Linux. Le comunicazioni tra client e server si svolgono tramite socket TCP.

Per gli studenti del nuovo ordinamento:

Il server verrà sviluppato in linguaggio C su piattaforma UNIX/Linux. Il client andrà realizzato in linguaggio Java su piattaforma Android. Client e server devono comunicare tramite socket TCP o UDP. Per la realizzazione del client, in particolare per la comunicazione, è consentito esclusivamente l'utilizzo delle API standard (java.net.*).

Oltre alle system call UNIX, i programmi C possono utilizzare solo la libreria standard del C. E' sconsigliato l'uso di primitive non coperte dal corso (ad es., code di messaggi) al posto di quelle studiate.

Relazione

Il progetto va accompagnato da una relazione che contenga almeno le seguenti sezioni:

1. Una guida d'uso per il server e per il client, che illustri le modalità di compilazione e d'uso dei due programmi.
2. Una sezione che illustri il protocollo al livello di applicazione utilizzato nelle comunicazioni tra client e server (non il protocollo TCP/IP!).
3. Una sezione che descriva i dettagli implementativi giudicati più interessanti (con particolare riferimento alle system call oggetto del corso), eventualmente corredati dai corrispondenti frammenti di codice.
4. In appendice, la relazione deve riportare il codice sorgente integrale del progetto. Orientativamente, la relazione dovrebbe constare di almeno 10 pagine, esclusa l'appendice. Indicare sulla copertina della relazione i componenti del gruppo.

Consegna del progetto

Entro la data prescelta per lo scritto finale (con eccezione per il primo appello), vanno consegnati al docente il progetto e la relazione. Il progetto e la relazione vanno inviati all'indirizzo alberto.finzi@unina.it in un archivio compresso in formato zip o rar. La relazione va consegnata al docente anche in formato cartaceo al momento della discussione. Durante l'esame orale, il client ed il server verranno testati, eseguendoli su due o più macchine diverse.