

# Laboratorio di Reti, Corsi A e B

## Distributed Hangman

### Progetto di Fine Corso – A.A. 2014/15

## 1 Descrizione del problema

Il progetto richiede di realizzare Distributed Hangman, una versione distribuita e multigiocatore di Hangman, il "gioco dell'impiccato" (wikipedia). Rispetto alla versione classica a due giocatori, il gioco viene modificato in modo che ogni partita possa essere giocata da un **Master** e da un insieme di **Guesser** che tentano di indovinare la parola una lettera per volta. I giocatori di Distributed Hangman sono geograficamente distribuiti e si collegano alla rete per partecipare al gioco. La grafica non è essenziale, e può essere visualizzato un contatore in luogo del tradizionale stickman.

Distributed Hangman deve essere sviluppato tenendo in considerazione le seguenti specifiche:

- Ogni giocatore si **registra** ad un servizio di gestione delle partite, specificando il proprio **nickname** ed una password;
- Prima di giocare un giocatore effettua il **login** fornendo il proprio nickname e la propria password;
- il Master può **aprire** una partita. La partita viene identificata dal suo nickname: quindi il Master può aprire solo una partita per volta. Il Master indica il numero  $k$  di Guesser che vuole incontrare;
- la **partita** inizia quando viene raggiunto un numero di Guesser uguale a  $k$ ;
- durante la partita qualunque Guesser (senza turni di gioco) può proporre una lettera;
- la partita termina quando
  - il numero di tentativi è esaurito, oppure
  - vengono indovinate tutte le lettere, oppure
  - scade un timeout lungo

Inoltre il Master può **annullare** la partita in qualsiasi momento, sia prima che dopo il suo inizio;

- un giocatore può **accettare** di partecipare ad una partita aperta precedentemente, diventando così Guesser;

- tutti i giocatori che non sono Master o Guesser ottengono un elenco delle partite che sono state aperte
  - all’atto del login e
  - ogni volta che viene aperta una nuova partita da un Master o un Guesser accetta una partita;
- ogni giocatore partecipa come Master o Guesser ad al più una partita per volta.

## 2 Implementazione del sistema

Si deve realizzare un’applicazione **Registry** che gestisce l’organizzazione delle partite di Distributed Hangman e un’applicazione **User Agent** che viene utilizzata dai giocatori.

### 2.1 Specifica della Registry

La Registry:

- gestisce la registrazione, il login ed il logout dei giocatori;
- gestisce le richieste di apertura e di chiusura di una partita. Il numero massimo delle partite aperte è limitato a 10: le richieste eccedenti devono essere rifiutate;
- deve inviare all’User Agent che non partecipa ad una partita come Guesser o Master un elenco delle partite aperte. L’invio avviene al momento del login, ed ogni volta che varia l’insieme delle partite aperte in attesa di giocatori, o l’insieme dei Guesser che hanno accettato di parteciparvi.
- gestisce l’accettazione alla partecipazione da parte degli User Agent: il server deve fornire agli User Agent gli indirizzi necessari alla comunicazione durante la partita;
- tutte le informazioni di configurazione del server (il numero massimo di partite, range di indirizzi dei gruppi ecc.) sono registrate in un file di configurazione in JSON.

### 2.2 Specifica dell’User Agent

Lo User Agent:

- gestisce l’interfaccia del giocatore con il Registry in fase di registrazione, login e logout, di apertura e chiusura della partita;
- se il giocatore non è Master o Guesser, lo User Agent deve visualizzare le notifiche ricevute dal Registry riguardo alle partite aperte;
- se il giocatore è Master di una partita, lo User Agent deve accettare dal giocatore la eventuale richiesta di chiusura.

- se il giocatore è Guesser lo User Agent deve accettare la proposta di una lettera, ed in risposta deve visualizzare lo stato del gioco modificato. Il giocatore propone una lettera ed attende sempre l'esito della sua proposta prima della proposta successiva.
- tutte le informazioni di configurazione dell'User Agent (l'indirizzo del server, ecc.) sono registrate in un file di configurazione in JSON.

### 2.3 Vincoli implementativi dello User Agent e del Registry

L'implementazione deve seguire queste linee guida:

- le richieste di **registrazione**, di **login**, di **logout** devono essere gestite come chiamate RMI;
- la **notifica** delle partite disponibili deve avvenire tramite RMI-callback;
- la richiesta di **apertura** di una partita viene realizzata con l'apertura di una connessione TCP tra lo User Agent del giocatore Master e la Registry: la connessione rimane attiva fino a che a) la partita inizia, oppure b) scade il time out lungo, oppure c) il Master abbandona la partita;
- la richiesta di **accettazione** di una partita viene realizzata con l'apertura di una connessione TCP tra lo User Agent del giocatore Guesser e la Registry: la connessione rimane attiva fino a che a) la partita inizia oppure b) scade il time out lungo oppure c) il master abbandona la partita;
- la richiesta di **abbandono** da parte del Master viene gestita sempre con l'annullamento della partita;
- la richiesta di **abbandono** da parte del Guesser determina il decremento del numero di giocatori, se avviene in fase di apertura, e la continuazione del gioco se avviene durante la fase di gioco;
- a tutti i giocatori di una partita la Registry consegna una **chiave di cifratura** per riconoscersi (simmetrica). Si può usare la libreria *jaspyt*;
- la comunicazione tra i Guesser partecipanti ad una partita ed il Registry avviene mediante **multicast**;
- all'inizio della **partita** (cioè, al raggiungimento dei  $k$  Guesser) la Registry:
  - comunica al Master ed ai Guesser indirizzo e porta del gruppo **multicast** e la **chiave** di cifratura;
  - infine chiude tutte le connessioni relative al Master e ai Guesser della partita;
- lo stato della partita è gestito dallo User Agent del Master;
- la proposta di una lettera da parte del Guesser viene implementata dallo User Agent con una transazione a due vie sul gruppo multicast. Il client del Guesser attende un ACK dal Master. In caso di timeout dell'ACK, la transazione viene ripetuta;

- dopo ogni proposta il Master invia in multicast a tutti i partecipanti l'ultima mossa effettuata e lo stato aggiornato (ha funzione di ACK per il Guesser che ha giocato).

### 3 Modalità di svolgimento del Progetto

Il progetto deve essere svolto singolarmente. Il materiale consegnato deve comprendere:

- il codice dell'applicazione e di eventuali programmi utilizzati per il test delle sue funzionalità;
- la relazione in formato pdf che deve contenere:
  - una descrizione generale dell'architettura del sistema. Motivare le scelte di progetto;
  - uno schema generale dei threads attivati da ogni componente e delle strutture dati utilizzate, con particolare riferimento al controllo della concorrenza;
  - una descrizione delle classi definite ed il codice eseguibile con indicazioni precise sulle modalità di esecuzione.

L'organizzazione e la chiarezza della relazione influiranno sul voto finale.

Relazione e codice sorgente devono essere consegnati sia in formato cartaceo, presso la portineria del Dipartimento, sia in formato elettronico, via e-mail al docente del corso. Gli eseguibili devono essere consegnati in formato elettronico.