Relazione sul Progetto di Reti di Calcolatori e Laboratorio

A.A. 2021/2022

Luca De Paulis

9 gennaio 2022

1 Introduzione e informazioni generali

Il progetto di Reti di Calcolatori e Laboratorio dell'Anno Accademico 2021/2022 consiste nella realizzazione di un Social Network chiamato **Winsome**. Il Social Network è basato su un'architettura di tipo *client-server*: le due parti comunicano attraverso la rete grazie ad un'API, che permette al client di interfacciarsi con l'applicativo del server.

Il progetto è stato realizzato in Java, versione 17 ed è stato testato in ambiente Linux.

2 Compilazione ed esecuzione

Il progetto è interamente compilabile ed eseguibile attraverso l'uso dei comandi `javac` e `java`; tuttavia, per semplificare il processo, è stato incluso un `Makefile` contenente alcune semplici regole.

`$ make` La regola di default compila il progetto: in particolare crea le directory necessarie al corretto funzionamento degli applicativi, compila il codice sorgente e ne crea due file `.jar`, che vengono salvati nella directory `bin/`.

`$ make run-default-server` Questa regola, se lanciata dopo la compilazione, esegue il server usando come parametri (ovvero come *path* del file di configurazione e della directory di logging, cf. **??**) i valori di default.

`$ make run-default-client` Analogamente, esegue il client usando i parametri di default (in questo caso solamente il *path* del file di configurazione, cf. **??**).

`$ make clean` Rimuove i file `.class`, i file `.jar` e i file di log del server contenuti nella directory `logs/`.

Per eseguire il client o il server passando dei parametri a riga di comando si possono usare gli script Bash `run-server.sh` e `run-client.sh`, entrambi contenuti nella directory `scripts/`. Per far ciò è necessario innanzitutto renderli eseguibili, tramite ad esempio `$ chmod a+x scripts/run-server.sh scripts/run-client.sh`, e poi eseguirli normalmente come `$ ./run-server.sh [...]`.

L'unica dipendenza del progetto è la libreria GSON (`gson-2.8.6.jar`), che si trova nella directory `lib/`.

3 Architettura generale del progetto

Il progetto è diviso in tre moduli principali:

- il **server**, implementato nel package `winsome.server` e avente come *entry point* la classe `winsome.server.WinsomeServerMain`: esso viene raccolto nel file `bin/winsome-server.jar`;
- il **client**, implementato nel package `winsome.client` e avente come *entry point* la classe `winsome.client.WinsomeClientMain`: viene compresso nel file `bin/winsome-client.jar`
- le **API**, implementate nel package `winsome.api`: insieme alle funzionalità di *utility* del package `winsome.utils` le API vengono compresse nel file `lib/winsome-api.jar`.

La scelta di separare le API dall'applicativo client ha lo scopo di separare il più possibile i vari moduli: così facendo si lascia aperta la possibilità di creare un client diverso che riusi le API fornite per interfacciarsi con il server. Inoltre sia il client che il server dipendono dalla libreria delle API per realizzare alcune funzionalità (ad esempio il parsing dei file di configurazione).

4 Il package `winsome.utils`

I tre moduli descritti nella Sezione 3 dipendono da alcune funzionalità messe a disposizione dal package di utilities:

- il file `winsome.utils.ConsoleColors` contiene i codici ASCII per usare i colori nei terminali: viene usato soprattutto dal client nella propria Command Line Interface;
- il package `winsome.utils.cryptography` contiene la classe `Hash`, che permette l'hashing di stringhe (utile per inviare e memorizzare le password), e l'eccezione `FailedHashException` per indicare il fallimento di un tentativo di hashing;
- il package `winsome.utils.configs` contiene le classi `AbstractConfig` e `ConfigEntry`: esse consentono di parsare file nel formato descritto nella Sezione 5.

5 I file di configurazione

Sia il client che il server necessitano di un file di configurazione per inizializzare i vari parametri. La sintassi dei due file è la stessa, e corrisponde alla sintassi base del linguaggio di markup YAML:

```
# comments and empty lines are allowed
key: value # comments after values are allowed
```

Due esempi di file di configurazione sono presenti nella directory `configs/` e di default, a meno di non specificare altrimenti negli argomenti a riga di comando del client/server, tali file sono usati come file di configurazione.

I campi dei file di configurazione devono essere tutti presenti una e una sola volta: in caso contrario il sistema segnala l'errore.

5.1 File di configurazione del server

Il file di configurazione del server ha i seguenti campi:

tcp-port Un intero non negativo che rappresenta la porta usata dal socket TCP.

udp-port Un intero non negativo che rappresenta la porta usata dal socket UDP.

multicast-addr L'indirizzo di multicast sul quale i client ricevono le notifiche del calcolo delle ricompense.

multicast-port La porta sulla quale i client ricevono le notifiche in multicast (è un intero non negativo).