Guião N.º 2

Sockets TCP (em Java)
Protocolos de comunicação ao nível da aplicação

Sistemas Distribuídos

Ricardo Costa
rcosta@estg.ipp.pt

ESTGF | POLITÉCNICO DO PORTO

Outubro 2017

1 Estabelecer um protocolo de comunicação

É comum que na comunicação entre cliente/servidor esteja definido um protocolo que permita entender uma sequência de mensagens. Nesta aula pretende-se implementar um cliente e um servidor que se entendem segundo um protocolo de comunicação previamente acordado.

2 O protocolo Knock Knock

Quando o cliente se liga ao servidor e este aceita a respectiva ligação, deve iniciar-se uma troca de mensagens que poderá ter a seguinte sequência:

Server: "Knock knock!"
Client: "Who's there?"
Server: "Dexter."
Client: "Dexter who?"
Server: "Dexter halls with boughs of holly."Client: "Groan."

Esta sequência de mensagens obedece a um protocolo previamente acordado e a que chamaremos de "Knock Knock Protocol".

A classe KnockKnockProtocol (Listing **??**) implementa o protocolo usado na comunicação entre cliente e servidor e assegura a coerência da sequÊncia de mensagens. Por exemplo, se após o servidor ter enviado a mensagem "Knock knock!"o cliente responder com "Dexter who?"em vez de "Who's there"este não terá qualquer efeito, sendo obrigado a responder "Who's there"para conseguir passar ao passo seguinte na sequÊncia de mensagens.

De seguida, o servidor responde com a primeira de várias piadas que conhece:

Server: Turnip

O cliente deverá então responder:

Client: Turnip who?"

E o servidor retornará:

Server: Turnip the heat, it's cold in here! Want another? (y/n)

Se quiser continuar a comunicação com o servidor deverá responder y, caso contrário deverá responder n. No caso de responder n o servidor enviará a mensagem "Bye"e ambos cliente e servidor terminarão.

Se em qualquer altura introduzir uma mensagem que não está de acordo com o protocolo, o servidor responderá com uma mensagem:

Server: You're supposed to say "Who's there?"!

Após o erro o servidor reiniciará a comunicação:

Server: Try again. Knock! Knock!

3 Exercícios

3.1 Protocolo HTTP

Um exemplo conhecido de um protocolo de comunicação ao nível da aplicação é o HTTP.

Comunique com um servidor HTTP utilizando o comando telnet, por exemplo:

telnet www2.estgf.ipp.pt 80

De seguida teste o protocolo escrevendo:

GET /index.html

O servidor deverá devolver a página principal presente no servidor Web. O browser que utiliza vulgarmente comunica segundo este protocolo interpretando e mostrando as páginas que lhe são devolvidas do servidor.

3.2 Protocolo Knock Knock

Implemente um cliente e um servidor que utilizem o protocolo Knock knock para comunicarem.

4 Recursos

4.1 class KnockKnockProtocol

```
1  import java.net.*;
2  import java.io.*;
3
4  public class KnockKnockProtocol {
5      private static final int WAITING = 0;
6      private static final int SENTKNOCKKNOCK = 1;
7      private static final int SENTCLUE = 2;
8      private static final int ANOTHER = 3;
9
10     private static final int NUMJOKES = 5;
11
12     private int state = WAITING;
13     private int currentJoke = 0;
```

```
14
15      private String[] clues = { "Turnip", "Little Old Lady", "
            Atch", "Who", "Who" };
16      private String[] answers = { "Turnip the heat, it's cold in
            here!",
17                                   "I didn't know you could yodel!
                                         ",
18                                   "Bless you!",
19                                   "Is there an owl in here?",
20                                   "Is there an echo in here?" };
21
22      public String processInput(String theInput) {
23          String theOutput = null;
24
25          if (state == WAITING) {
26              theOutput = "Knock! Knock!";
27              state = SENTKNOCKKNOCK;
28          } else if (state == SENTKNOCKKNOCK) {
29              if (theInput.equalsIgnoreCase("Who's there?")) {
30                  theOutput = clues[currentJoke];
31                  state = SENTCLUE;
32              } else {
33                  theOutput = "You're supposed to say \"Who's
                        there?\"! " +
34                  "Try again. Knock! Knock!";
35              }
36          } else if (state == SENTCLUE) {
37              if (theInput.equalsIgnoreCase(clues[currentJoke] + "
                    who?")) {
38                  theOutput = answers[currentJoke] + " Want
                        another? (y/n)";
39                  state = ANOTHER;
40              } else {
41                  theOutput = "You're supposed to say \"" +
42                  clues[currentJoke] +
43                  " who?\"" +
44                  "! Try again. Knock! Knock!";
45                  state = SENTKNOCKKNOCK;
46              }
47          } else if (state == ANOTHER) {
48              if (theInput.equalsIgnoreCase("y")) {
49                  theOutput = "Knock! Knock!";
50                  if (currentJoke == (NUMJOKES - 1))
51                      currentJoke = 0;
52                  else
53                      currentJoke++;
54                  state = SENTKNOCKKNOCK;
55              } else {
56                  theOutput = "Bye.";
```

```
57                      state = WAITING;
58                  }
59              }
60              return theOutput;
61          }
62  }
```

Listing 1: Exemplo de uma implementação da class KnockKnockProtocol