

Servidores http – web servers

Exemplo 1: Servidor de News

```
// TCPServerNews.java

// Aguarda pedido de conexão de algum cliente
// Envia uma msg ao cliente
// Volta para o início
// usa protocolo de aplicação HTTP

import java.io.*;
import java.net.*;

class TCPServerNews {
    public static void main(String argv[]) throws Exception
    {
        int i = -1;
        String LinhaCliente;
        String LinhaSaida = "Mensagem xxxxxxx";

        // cria socket de comunicação com os clientes na porta 80
        // este servidor será acessado por um browser IE, Mozilla, NN

        ServerSocket w = new ServerSocket(80);

        // espera msg de algum cliente e trata
        while(true) {

            i++;

            // espera conexão de algum cliente
            System.out.println("Esperando algum cliente se conectar\n");

            Socket c = w.accept();

            System.out.println("Um cliente se conectou\n");

            // cria streams de entrada e saída com o cliente que chegou

            BufferedReader in = new BufferedReader(new
InputStreamReader(c.getInputStream()));

            DataOutputStream out = new
DataOutputStream(c.getOutputStream());

            // lê uma linha do cliente
            LinhaCliente = in.readLine();

            //mostra a msg que chegou do cliente
            System.out.println("Chegou do cliente:" + LinhaCliente +
'\n');

            // escolhe msg a enviar ao cliente
```

```
        switch (i%5) {
            case 0: {
                LinhaSaida = "Mensagem zero  ";
                break;
            }
            case 1: {
                LinhaSaida = "Mensagem um    ";
                break;
            }
            case 2: {
                LinhaSaida = "Mensagem dois  ";
                break;
            }
            case 3: {
                LinhaSaida = "Mensagem tres  ";
                break;
            }
            case 4: {
                LinhaSaida = "Mensagem quatro";
                break;
            }
        }
        // envia as mensagens sob o protocolo HTTP
        // envia ao cliente a primeira msg do HTTP
        out.writeBytes("HTTP/1.1 200 Veja abaixo a mensagem\r\n");

        // envia ao cliente a header line com o tipo do arquivo
        out.writeBytes("Content-Type: text\r\n");

        // envia ao cliente a header line com o tamanho do arquivo
        out.writeBytes("Content-Length: " + 15 + "\r\n");

        // envia ao cliente uma linha em branco (lembre-se do HTTP)
        out.writeBytes("\r\n");

        // envia ao cliente a mensagem
        out.writeBytes(LinhaSaida + "\r\n");
    }
}
```

Exemplo 2: Servidor Web

```
// WebServer.java

// Servidor WEB simplificado - protocolo HTTP
// Recebe pedido de página .gif ou .jpg
// Le a página do disco e envia ao cliente
// Exemplo do livro de redes do Kurose&Ross

import java.io.*;
import java.net.*;
import java.util.*;
```

```
class TCPWebServer {
    public static void main(String argv[]) throws Exception {
        String requestMessageLine;
        String fileName;

        // cria socket de comunicação com os clientes na porta 80
        ServerSocket listenSocket = new ServerSocket(80);

        // espera conexão de algum cliente
        Socket connectionSocket = listenSocket.accept();

        // cria os dutos de comunicação de/para o cliente
        BufferedReader inFromClient =
            new BufferedReader(new InputStreamReader(
                connectionSocket.getInputStream()));
        DataOutputStream outToClient = new DataOutputStream(
            connectionSocket.getOutputStream());

        // le uma linha do cliente
        requestMessageLine = inFromClient.readLine();

        // separa os elementos da linha
        StringTokenizer tokenizedLine =
            new StringTokenizer(requestMessageLine);

        // verifica se o primeiro elemento é GET
        if (tokenizedLine.nextToken().equals("GET")) {
            fileName = tokenizedLine.nextToken();

            // se nome do file começar com / pula um caracter
            if (fileName.startsWith("/") == true )
                fileName = fileName.substring(1);

            // declara o File file e associa a fileName
            File file = new File(fileName);

            // pega o tamanho em bytes
            int numOfBytes = (int) file.length();

            // cria o duto inFile com o arquivo
            FileInputStream inFile = new FileInputStream (fileName);

            // declara vetor de bytes e le o arquivo neste vetor
            byte[] fileInBytes = new byte[4096];
            inFile.read(fileInBytes);

            // envia ao cliente a primeira msg do HTTP
            outToClient.writeBytes("HTTP/1.0 200 Document Follows\r\n");

            // envia ao cliente a header line com o tipo do arquivo
            if (fileName.endsWith(".jpg"))
                outToClient.writeBytes("Content-Type: image/jpeg\r\n");
            if (fileName.endsWith(".gif"))
                outToClient.writeBytes("Content-Type: image/gif\r\n");

            // envia ao cliente a header line com o tamanho do arquivo
            outToClient.writeBytes("Content-Length: " + numOfBytes + "\r\n");
        }
    }
}
```

```
        // envia ao cliente uma linha em branco (lembre-se do HTTP)
        outToClient.writeBytes("\r\n");

        // envia ao cliente o arquivo
        outToClient.write(fileInBytes, 0, numOfBytes);

        // fecha o cliente que se conectou
        connectionSocket.close();
    }
    else System.out.println("Bad Request Message");
}
}
```

1) HTML - exemplo de programas

Resposta ao web server acima