TFA 42
Sistemi e Reti di Calcolatori per la Didattica

19/04/2013
Laura Ricci

"COMPITO PER LE VACANZE"

- progettare altre esperienze di laboratorio che consentano la sperimentazione di qualche aspetto relativo ai protocolli di rete sviluppati nelle lezioni teoriche

- con riferimento a JAVA, alcuni esempi:
  - verifica che il protocollo UDP non implementa controllo di flusso
  - protocollo HTTP
  - gestione URL
  - ……???

- altri esempi anche in riferimento ad altri linguaggi/ambienti?

LABORATORIO: ACCEDERE A RISORSE IN RETE

- Prerequisiti:
  - Struttura di una URL
  - Identificare risorse in Internet

- Scopo dell'esercizio:
  - capire come è formata una URL
  - leggere la risorsa individuata da una URL
  - risorse considerate
    - pagine web
    - file

LABORATORIO: ACCEDERE A RISORSE IN RETE

- Utilizzare JAVA come ambiente di programmazione
- leggere la risorsa individuata da una URL
- se la risorsa è una pagina web, scaricare il contenuto della pagina
- analizzare l'HTML della pagina

```
public static void main(String args[])
    { try
        { new ProvaURL("http://www.unipi.it"); }
     catch(MalformedURLException ex)
                { ex.printStackTrace(); }
     catch(IOException ex)
                {  ex.printStackTrace(); }
     }
```

LABORATORIO: ACCEDERE A RISORSE IN RETE

```
import java.io.DataInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
public class ProvaURL
{
   public ProvaURL(String urlStr) throws MalformedURLException,
                                         IOException
    {
```

LABORATORIO: ACCEDERE A RISORSE IN RETE

```
URL url = null;
url = new URL(urlStr);
URLConnection urlConnection;
urlConnection = url.openConnection();
InputStream urlStream;
urlStream = url.openStream();
String type = null;
type = urlConnection.getContentType();
System.out.println(type);
if(type==null)
    return;
else if( type.compareTo("text/html; charset=utf-8") != 0 )
     return; }
```

LABORATORIO: ACCEDERE A RISORSE IN RETE

```
    StringBuffer content;
    content = new StringBuffer();
    DataInputStream html;
    html = new DataInputStream(urlStream);
    String line;
    while ((line = html.readLine()) != null)
          { content.append(new String(line));
            content.append('\n');
          }
    System.out.print(content);
}
```

LABORATORIO: ACCEDERE A RISORSE IN RETE

Da osservare:

- Quando un server invia un documento ad un client (Ad esempio ad un browser), invia anche un header HTTP, content-type
- Content type: tipo dei dati contenuti nel documento
- Documento di tipo testo (text/HTML)
- Parametro charset in the HTTP header specifica la codifica dei caratteri contenuti nel documento

LABORATORIO: ACCEDERE A RISORSE IN RETE

Cosaviene stampato.....

```
text/html; charset=utf-8
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it" lang="it">


<head>
    <base href="http://www.unipi.it/" /><!--[if lte IE 6]></base><!
[endif]-->
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta name="robots" content="index, follow" />
  <meta name="keywords" content="UniversitÃ  di Pisa, corsi di laurea,
ricerca scientifica, master, dottorati, scuole di specializzazione, servizi
agli studenti" />
  <meta name="og:url" content="http://www.unipi.it/" />
  <meta name="og:title" content="Home" /> ….....
```

LABORATORIO: DOWNLOAD DI UN FILE

```
import java.io.*;
import java.net.*;
public class DownLoadFromURL {
    public static void main(String[] args) {
        try {
            URL url = new URL("http://www.bluej.org/tutorial/tutorial-201.pdf");
            // The server thinks this request is from an Opera browser!
            String userAgent = "Opera/9.63 (Windows NT 5.1; U; en)Presto/2.1.1";
            System.out.println("Downloading ...");
            downloadFromUrl(url, "tutorial.pdf", userAgent);
            System.out.println("OK");
          } catch (Exception e) {  System.err.println(e); }
    }
```

LABORATORIO: DOWNLOAD DI UN FILE

```
public static void downloadFromUrl(URL url, String localFilename, String userAgent)
                                                                    throws IOException
{       InputStream is = null;
        FileOutputStream fos = null;
        try { URLConnection urlConn = url.openConnection();
              if (userAgent != null) {
                   urlConn.setRequestProperty("User-Agent", userAgent); }
                    is = urlConn.getInputStream();
                    fos = new FileOutputStream(localFilename);
                byte[] buffer = new byte[4096]; int len;
                while ((len = is.read(buffer)) > 0) {
                      fos.write(buffer, 0, len);  }
            } finally { try { if (is != null) { is.close(); }
            } finally { if (fos != null) { fos.close();
                } } }
}
```

LABORATORIO: PROTOCOLLO HTTP

- Scopo dell'esercizio: analisi del protocollo HTTP
- Comprensione dei metodi supportati dal protocollo HTTP
- Esempio: POST
  - Metodo utilizzato per inviare dati al server
- Esercizio
  - Inviare un POST ad un server
  - Analizzare la risposta del server

PROTOCOLLO HTTP: POST

```
import java.io.*; import java.net.*;
public class POST {
     public static void main(String args[]) throws Exception
     { String nome = "pippo";
      //L'URL a cui fare la POST
        URL url = new URL("http://www.google.it");
        HttpURLConnection connection = (HttpURLConnection) url.openConnection();
        connection.setDoOutput(true);
        connection.setRequestMethod("POST");
        connection.setRequestProperty("Content-Type","application/x-www-form-
                                                      urlencoded");
        connection.setRequestProperty("charset", "UTF-8");
        connection.setRequestProperty("Content-Length", Integer
                                              .toString(nome.getBytes().length));
        connection.setUseCaches(false);
```

PROTOCOLLO HTTP: POST

```
DataOutputStream wr=null;
wr = new DataOutputStream(connection.getOutputStream());
wr.writeBytes("nome=" + nome);
wr.flush();
wr.close();
int responseCode = connection.getResponseCode();
System.out.println(responseCode);
}}
```

```
Risposta dal server: 405
```

Server Web “in sola lettura”: gestisce pagine Web statiche, non prevede un input dall'utente, non consente il metodo POST.

Server Web non abilitato a ricevere dati dall'utente

PERDITA DI PACCHETTI UDP

- Prerequisiti:
  - Conoscenza protocolli UDP e TCP
  - Tecniche di controllo del flusso

- Obiettivo della lezione
  - Illustrare differenze tra TCP ed UDP
  - Scrivere un programma JAVA che dimostri che UDP non effettua controllod el flusso

PERDITA DI PACCHETTI: SERVER UDP

```
import java.net.*;import java.io.*;
public class server {
public static void main(String[] args) {
  int cd;
  try
  {   DatagramSocket ds =new DatagramSocket(10001);
      InetAddress ia = InetAddress.getByName("LocalHost");
      byte [] dataR=new byte[516];
      DatagramPacket dpR = new DatagramPacket(dataR,dataR.length);
      ds.receive(dpR);
    ByteArrayInputStream bis = new ByteArrayInputStream(dpR.getData());
    DataInputStream dis = new DataInputStream(bis);
    cd=(int)dis.readInt();
    System.out.println("client sent to me number "+cd);
```

PERDITA DI PACCHETTI: SERVER UDP

```
byte [] data=new byte[516];
ByteArrayOutputStream bos = new ByteArrayOutputStream();
while(cd!=0){
        DataOutputStream dos = new DataOutputStream(bos);
        try {
            dos.writeInt(cd);
            } catch (IOException e) { e.printStackTrace();}
          data = bos.toByteArray();
          DatagramPacket dp = new DatagramPacket(data,data.length,
          dpR.getAddress(),dpR.getPort());
          ds.send(dp);
          bos.reset();
           cd--; } }
       catch (Exception e) {e.printStackTrace();} }
```

PERDITA DI PACCHETTI: CLIENT UDP

```
import java.net.*;import java.io.*;import java.util.*;
public class client {
public static void main(String[] args) {
   int number, read_number=0;
   Scanner scanner = new Scanner(System.in);
   InetAddress ia;
   ByteArrayOutputStream bos = new ByteArrayOutputStream();
   DataOutputStream dos = new DataOutputStream(bos);
   System.out.println("select the nuber to send:");
   number=scanner.nextInt();
   try {
    dos.writeInt(number);
    dos.flush();
       } catch (IOException e) { e.printStackTrace(); }
```

PERDITA DI PACCHETTI: CLIENT UDP

```
byte [] data = bos.toByteArray();
try {
    DatagramSocket ds =new DatagramSocket( );
    ia = InetAddress.getByName("LocalHost");
    DatagramPacket dp = new DatagramPacket(data,data.length, ia,
                                                    10001);
    ds.send(dp);
    byte [] dataR=new byte[516];
    DatagramPacket dpR = new DatagramPacket(dataR,dataR.length);
    ByteArrayInputStream bis = new ByteArrayInputStream(dpR.getData());
    DataInputStream dis = new DataInputStream(bis);
```

PERDITA DI PACCHETTI: CLIENT UDP

```
while(number!=0){
    try {
        Thread.sleep(2000);
    } catch (InterruptedException e) { e.printStackTrace(); }
    ds.receive(dpR);
    read_number=(int)dis.readInt();
    System.out.println(""+read_number);
    number--;
   dis.reset();};
}catch (Exception e) {
e.printStackTrace();
}
}}
```