

Esercitazione su Globus Toolkit 2:
Librerie Java – 2^a Parte

Paolo Trunfio
trunfio@deis.unical.it

1

Cancellazione di un Job

```
private static boolean jobCancel(String jobID) {  
    GramJob job = new GramJob("");  
    try {  
        job.setID(jobID);  
        Gram.cancel(job);  
    } catch(MalformedURLException e) {  
        System.err.println("Invalid job handle");  
        return false;  
    } catch(Exception e) {  
        System.err.println("Job cancel failed: " + e.getMessage());  
        return false;  
    }  
    System.out.println("Job cancel successful");  
    return true;  
}
```

2

Test di autenticazione

```
private static boolean authenticationTest(String resourceManager) {  
    try {  
        Gram.ping(resourceManager);  
    } catch(Exception e) {  
        System.out.println("Authentication test failed: " + e.getMessage());  
        return false;  
    }  
    System.out.println("Authentication test successful");  
    return true;  
}
```

3

Redirezione dell'output su video

```
import org.globus.gam.*;  
import org.globus.io.gass.server.*;  
  
class SimpleGramJobListener implements GramJobListener {  
    public synchronized void statusChanged(GramJob job) {  
        String status = job.getStatusAsString();  
        if (status.equals("DONE"))  
            System.exit(1);  
    }  
}
```

4

Redirezione dell'output su video (cont.)

```
public class JobRun2 {
    public static void main(String [] args) {
        String contact = "jupiter.deis.unical.it";
        String rsl = "&(executable=/bin/date)";
        // esegue "/bin/date" sulla macchina remota
        try {
            GassServer gassServer = new GassServer();
            String gassUrl= gassServer.getURL();
            rsl += "(stdout="+gassUrl+"/dev/stdout)";
            rsl += "(stderr="+gassUrl+"/dev/stderr)";
            // standard output e standard error sono rediretti sullo schermo del client
            GramJob job = new GramJob(rsl);
            job.addListener(new SimpleGramJobListener());
            job.request(contact);
        } catch(Exception e) { e.printStackTrace(); }
    }
}
```

5

Redirezione dell'output su uno stream

```
import org.globus.gam.*;
import org.globus.io.gass.server.*;
import java.io.*;

class SimpleGramJobListener implements GramJobListener {
    public synchronized void statusChanged(GramJob job) {
        String status = job.getStatusAsString();
        if (status.equals("DONE"))
            System.exit(1);
    }
}

public class JobRun3 {
    public static void main(String [] args) {
        String contact = "jupiter.deis.unical.it";
        String rsl = "&(executable=/bin/ls)";
        // esegue "/bin/ls" sulla macchina remota
        try {
            PipedOutputStream outputStream = new PipedOutputStream();
            PipedOutputStream errorStream = new PipedOutputStream();
```

6

Redirezione dell'output su uno stream (cont.)

```
GassServer gassServer = new GassServer();
String gassUrl= gassServer.getURL();
String uniqueString = new java.rmi.server.UID().toString(); // una stringa unica
rsl += "(stdout="+gassUrl+"/dev/stdout"+uniqueString+)";
rsl += "(stderr="+gassUrl+"/dev/stderr"+uniqueString+)";
gassServer.registerJobOutputStream("out"+uniqueString, outputStream);
gassServer.registerJobOutputStream("err"+uniqueString, errorStream);
// standard output e standard error sono rediretti rispettivamente
// su outputStream ed errorStream
StreamManager outputManager = new StreamManager(outputStream);
StreamManager errorManager = new StreamManager(errorStream);
outputManager.start();
errorManager.start();
GramJob job = new GramJob(rsl);
job.addListener(new SimpleGramJobListener());
job.request(contact);
} catch(Exception e) { e.printStackTrace(); }
}
}
```

7

Redirezione dell'output su uno stream (cont.)

```
class StreamManager extends Thread{
private BufferedReader input;
public StreamManager (PipedOutputStream stream) {
try {
input = new BufferedReader (new InputStreamReader
(new PipedInputStream(stream)));
} catch (IOException ioe) { ioe.printStackTrace(); }
}
public void run() {
try {
String line;
while (true) {
while ((line = input.readLine()) != null)
System.out.println (line); // stampa semplicemente su video
Thread.sleep(1000);
}
} catch (IOException ioe) { ioe.printStackTrace(); }
catch (InterruptedException ie) { ie.printStackTrace(); }
}
}
```

8