

Università degli Studi della Calabria
Corso di Laurea Specialistica in Ingegneria Informatica – A.A. 2003/2004
Corso di Griglie e Sistemi di Elaborazione Ubiqui

Esercitazione su Globus Toolkit 2:
Librerie Java – 1^a Parte

Paolo Trunfio
trunfio@deis.unical.it

Interrogazione di un server LDAP

```
import netscape.ldap.*;
```

```
import java.util.*;
```

```
public class LDAPSearch {
```

```
    public static void main (String args[]) {
```

```
        String server = "jupiter.deis.unical.it";
```

```
        int port = 2135;
```

```
        String basedn = "Mds-Vo-name=local, o=Grid";
```

```
        int scope = LDAPConnection.SCOPE_SUB;
```

```
        // oppure = LDAPConnection.SCOPE_ONE
```

```
        // oppure = LDAPConnection.SCOPE_BASE
```

```
        String filter = "objectclass=*";
```

```
        String attrNames[] = null;
```

```
        // oppure = {"attr1","attr2",...};
```

```
        boolean attrsOnly = false;
```

```
        // se = true non restituisce i valori degli attributi trovati
```

Interrogazione di un server LDAP (cont.)

```
try {
    LDAPConnection ld = new LDAPConnection();
    ld.connect(server, port);
    LDAPSearchResults res = ld.search(basedn, scope, filter, attrNames, attrsOnly);
    if (ld.isConnected())
        ld.disconnect();
    while (res.hasMoreElements()) {
        LDAPEntry findEntry = null;
        try {
            findEntry = res.next();
        } catch (LDAPReferralException e) {
            continue;
        } catch (LDAPException e) {
            System.err.println("Error: "+e.toString());
            continue;
        }
        String dn = findEntry.getDN();
        System.out.println ("dn: "+dn);
    }
}
```

Interrogazione di un server LDAP (cont.)

```
LDAPAttributeSet findAttrs = findEntry.getAttributeSet();
Enumeration enumAttrs = findAttrs.getAttributes();
while (enumAttrs.hasMoreElements()) {
    LDAPAttribute anAttr = (LDAPAttribute)enumAttrs.nextElement();
    String attrName = anAttr.getName();
    Enumeration enumVals = anAttr.getStringValues();
    while (enumVals.hasMoreElements()) {
        System.out.print (attrName+": ");
        String val = (String)enumVals.nextElement();
        System.out.println (val);
    }
}
System.out.println();
}
} catch( LDAPException e ) {
    System.err.println("Error: "+e.toString());
}
}
}
```

Trasferimento di un file con GridFTP

```
import org.globus.io.urlcopy.UrlCopy;
import org.globus.util.GlobusURL;

public class FileCopy {
    public static void main(String[] args) {
        String fromURL = "gsiftp://thebe.deis.unical.it//home/paolo/file1";
        String toURL = "gsiftp://jupiter.deis.unical.it//home/paolo/file1";
        boolean thirdPartyTransfer = true;
        try {
            GlobusURL from = new GlobusURL(fromURL);
            GlobusURL to = new GlobusURL(toURL);
            UrlCopy uc = new UrlCopy();
            uc.setSourceUrl(from);
            uc.setDestinationUrl(to);
            uc.setUseThirdPartyCopy(thirdPartyTransfer);
            uc.copy();
        } catch (Exception e) {
            System.err.println(e);
        }
    }
}
```

Sottomissione di un semplice job

```
import org.globus.gam.Gram;  
import org.globus.gam.GramJob;  
import org.globus.gam.GramJobListener;  
  
class SimpleGramJobListener implements GramJobListener {  
    public synchronized void statusChanged(GramJob job) {  
        String jobID = job.getIDAsString();  
        String status = job.getStatusAsString();  
        System.out.println("Job ID : "+jobID);  
        System.out.println("Status : "+status+"\n");  
        if (status.equals("DONE"))  
            System.exit(1);  
    }  
}
```

Sottomissione di un semplice job (cont.)

```
public class JobRun {  
    public static void main(String [] args) {  
        String contact = "jupiter.deis.unical.it";  
        String rsl = "&(executable=/bin/date)(stdout=/home/paolo/datefile)";  
        // esegue "/bin/date" sulla macchina remota e memorizza  
        // il risultato nel file remoto "/home/paolo/datefile"  
        GramJob job = new GramJob(rsl);  
        job.addListener(new SimpleGramJobListener());  
        try {  
            job.request(contact);  
            System.out.println("Job submitted: "+job.getIDAsString()+"\n");  
        } catch(Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```