

Java, reti e shell

Marco Danelutto
LPRb a.a. 2007-2008

Compilazione da riga di comando

- javac nomeFile.java
- estensione .java è necessaria
- attenzione al path e al package dei file
 - se file.java ha una package a.b
 - deve trovarsi in una a/b/file.java di una directory nel CLASSPATH

CLASSPATH

- cammino lungo il quale si cercano le classi (.class) per caricarle dinamicamente
- processo automatico interamente gestito dall'ambiente Java
- ma occorre predisporre il CLASSPATH nel modo opportuno (Eclipse lo fa automaticamente, sa shell lo dovete fare a mano)

Settaggi attivi

- per vedere quale classpath è attivo
 - echo \$CLASSPATH
 - env | grep CLASSPATH

Classpath da shell

- variabile di ambiente CLASSPATH
 - lista di directory e/o .jar separati da “:”
 - da (t)csh:
setenv CLASSPATH :.\$HOME/classes
 - da bash:
export CLASSPATH=.:\$HOME/classes
- sono comandi validi per la shell corrente

Classpath da shell (permanente)

- Comandi per il classpath nel file di inizializzazione eseguito ogni volta che si lancia una shell
 - \$HOME/.bashrc (per bash)
 - \$HOME/.cshrc (per la (t)csh)
 - normalmente a fine file

Classpath da shell (permanente) (2)

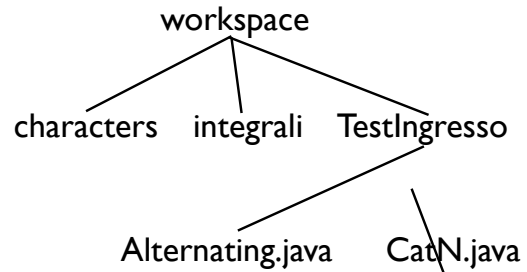
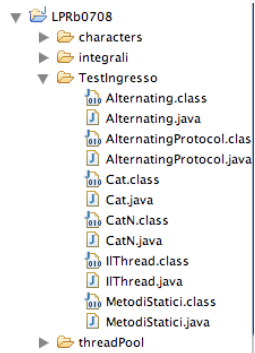
- per mantenere settaggi precedenti (se ce ne fossero)
 - export CLASSPATH=.:\$HOME/classes:
\$CLASSPATH
 - prima cerca nel nuovo path poi nel vecchio
 - export CLASSPATH=\$CLASSPATH:.:\$HOME/
classes
 - prima cerca nel vecchio path poi nel nuovo

Classpath da comando di compilazione

- javac -classpath directory nomeFile.java
- javac -cp directory nomeFile.java

- solo temporaneamente (per la durata della compilazione richiesta) aggiungi directory al classpath
- directory semplice o lista separata dai “:”

Package e CLASSPATH



.jar e Classpath

- pacchetto.jar
 - `jar cvf pacchetto.jar TestIngresso/`
 - `javac -cp pacchetto.jar ...`

Esecuzione da shell

- `java nomeClasse`
 - nomeClasse diverso da nomeFile.class
 - nome qualificato
 - `TestIngresso.Catn`
- classpath gestito come per la compilazione

Ancora var d'ambiente

- `JAVA_HOME`
 - utilizzata da molti tool che usano java
 - va puntata alla directory con gli eseguibili
 - tipicamente:
`JAVA_HOME=/usr/local/java/jdk1.5.0_06/bin`

Ancora ambiente (2)

- diverse versioni di Java disponibili su queste macchine
- fujih1:~> ls /usr/local/java/
docs j2sdk1.4.2_04 jdk1.5.0_06
jdk1.6.0_02 WTK2.2 WTK2.5.1
- potete “scegliere” quella che vi aggrada
 - settando opportunamente PATH
 - variabile d’ambiente
 - indica dove cercare i comandi shell

PATH

- stesse regole e convenzioni del CLASSPATH
- classpath derivato da path ...
- settaggi temporanei da riga di comando
- settaggi permanenti nei .bashrc/.cshrc

Comandi “remoti”

- ssh
 - esecuzione di una shell su un host remoto (senza param)
 - ssh nomeutente@nomehost
 - esecuzione di un comando su un host remoto (con parametri)
 - ssh utente@host cmd param ...

Comandi “remoti”

- scp
 - come cp ma sorgente o destinazione:
 - nomeHost:[path/]filename
 - scp pippo fujih2:
copia pippo da dir corrente nella \$HOME di fujh2
 - scp pippo fujih2:/tmp
lo copia nella /tmp di fujih2

Caratteristiche

- tutta la connessione avviene su SSL
 - vedi Reti (fwd)
 - cifrata => password “nascoste” sulla rete
- protocollo che prevede utilizzo di chiavi asimmetriche per stabilire una chiave di sessione

Autenticazione

- in condizioni normali
 - ssh host comando
 - chiede la passwd della macchina remota
- si può evitare
 - in modo da snellire la procedura
 - per i casi in cui il comando remoto sia in uno script

Procedura elimina richiesta passwd

- nella directory \$HOME/.ssh
- dovrete avere una coppia di chiavi
 - chiave privata: id_rsa
 - chiave pubblica: id_rsa.pub
- se non ci sono createle
 - ssh-keygen
 - passphrase vuota

Procedura (2)

- nella \$HOME/.ssh editare il file
 - authorized_keys
 - aggiungere il contenuto di id_rsa.pub come ultima riga del file
- diritti giusti (altrimenti non funziona):
.`ssh/ drwx-----`
`authorized_keys,id_rsa.pub -rw-r--r--`
`id_rsa -rw-----`