

Lezione 2: Comandi avanzati della Shell di Unix

Laboratorio di Elementi di Architettura e Sistemi Operativi

6 Marzo 2013

Parte 1: Metacaratteri, link e file speciali

I metacaratteri

- La shell riconosce alcuni caratteri speciali, chiamati *metacaratteri*, che possono comparire nei percorsi.
- Quando l'utente invia un comando, la shell lo scandisce alla ricerca di eventuali metacaratteri, che processa in modo speciale.
- Una volta processati tutti i metacaratteri, viene eseguito il comando.
- Questi metacaratteri (o *wildcard*) sono:
 - * una stringa qualsiasi di 0 o più caratteri
 - ? un singolo carattere

Esempi

- `cp * ../Backup` copia tutti i file contenuti nella directory corrente nella directory Backup ("sorella" della directory corrente)
- `cp /JAVA/Area*.java /JAVA_backup`
copia tutti i file il cui nome inizia con la stringa Area e termina con l'estensione .java nella directory JAVA_backup
- `ls *.txt`
elenca tutti i file con estensione .txt presenti nella directory corrente
- `ls /dev/tty?`
`/dev/ttya /dev/ttyb`

Tipi di file in Unix

- Un solo tipo di file fisico: *byte stream*
- Quattro tipi di file logici:

Directory Contiene nomi e indirizzi di altri file

Special file Oggetto per agire su un dispositivo di I/O

Link Collegamento ad un altro file

File ordinario Tutti gli altri file

File speciali

- Ogni dispositivo di I/O visto come un file
- I programmi non sanno se operano su file o device di I/O
- Lettura/scrittura su special file causano operazioni di I/O sul relativo device
- Indipendenza dai dispositivi reali !

I link

- **Hard link**
 - Un nome di file che punta a un i-node puntato anche da altri nomi di file
 - Non c'è differenza tra il nome originale e l'hard link: entrambi puntano allo stesso i-node !
- **Symbolic link**
 - Un file che contiene il nome di un altro file
- Particolarità
 - Non si può fare hard link di directory
 - Non si può fare hard link a file su partizioni diverse
 - Un file viene rimosso solo quando tutti i suoi hard link sono stati rimossi

Creazione e distruzione di link

- Hard link `ln fileesistente nomelink`
- Symbolic link `ln -s fileesistente nomelink2`
- Notare le dimensioni dei tre file (con `ls -al`)
- Per distruggere un link è sufficiente usare il comando `rm`
- Se il file originale viene rimosso, il link simbolico rimane pendente

Parte 2: Permessi e proprietari dei file

Ripartiamo da `ls`

Eseguendo il comando `ls -l /bin` si ottiene il seguente output:

```
...
lrwxrwxrwx 1 root root      4 Dec 5  2000 awk -> gawk
-rwxr-xr-x 1 root root  5780 Jul 13  2000 basename
-rwxr-xr-x 1 root root 512540 Aug 22  2000 bash
...
```

da sinistra a destra abbiamo:

1. tipo di file (– file normale, d directory, l link, b block device, c character device),
2. permessi,
3. numero di hard link al file,
4. proprietario del file,
5. gruppo del proprietario del file,
6. grandezza del file in byte,
7. data di ultima modifica,
8. nome del file.

I permessi dei file

- Unix è un sistema *multiutente*.
- L'amministratore del sistema (`root`) ha tutti i permessi
- Per gli altri utenti l'accesso ai file è regolato dai permessi:

```
-rw-r--r-- 1 root root 981 Sep 20 16:32 passwd
```
- Il blocco di caratteri `rw-r--r--` rappresenta i permessi di accesso al file:
 - I primi 3 (`rw-`) sono riferiti al proprietario.
 - Il secondo blocco (`r--`) è riferito al gruppo.
 - L'ultimo blocco (`r--`) a tutti gli altri utenti.
 - La prima posizione rappresenta il permesso di lettura (`r`),
 - la seconda il permesso di scrittura (`w`)
 - e la terza il permesso di esecuzione (`x`).
 - Un trattino (`-`) indica l'assenza del permesso corrispondente.

Cambio di proprietario

- cambia il gruppo del file

```
chgrp [-R] gruppo file
```
- cambia proprietario [e gruppo] del file

```
chown [-R] utente[:gruppo] file
```
- l'opzione `-R` indica di propagare il comando alle sottodirectory

Cambio dei permessi

- assegna i permessi al file

```
chmod [-R] permessi file
```
- permessi assoluti: un numero di quattro cifre corrispondenti alla *codifica binaria* dei permessi:

	proprietario	gruppo	altri
4 2 1	4 2 1	4 2 1	4 2 1
s S t	r w x	r w x	r w x

Il primo numero si può omettere

- permessi simbolici: una stringa di tre caratteri a scelta tra quelli tra parentesi quadre

```
[ugoa] [+ -=] [rwxst]
```

Simboli dei permessi

- u user
- g group
- o other
- a all (user, group, other)
- + assegna il permesso
- toglie il permesso

- = assegna i permessi specificati, toglie gli altri
- r read permission
- w write permission
- x execution permission
- S setgid (vedi dopo)
- s setuid (vedi dopo)
- t sticky bit (vedi dopo)

Esempi di modifica dei permessi

- `chmod 640 prova.txt`
 - Lettura/scrittura per proprietario: $6 \rightarrow 4 + 2 \rightarrow 110 \rightarrow rw-$
 - Lettura per gruppo: $4 \rightarrow 100 \rightarrow r--$
 - Nessun permesso per altri: $0 \rightarrow 000 \rightarrow ---$
- `chmod 755 dir`
 - $r/w/x$ per proprietario: $7 \rightarrow 4 + 2 + 1 \rightarrow 111 \rightarrow rwx$
 - Lettura/esecuzione per gruppo: $5 \rightarrow 4 + 1 \rightarrow 101 \rightarrow r-x$
 - Lettura/esecuzione per altri: $5 \rightarrow 4 + 1 \rightarrow 101 \rightarrow r-x$
- `chmod u+x filename`
 - Aggiungi permesso di esecuzione per l'utente
- `chmod o-w filename`
 - Togli il permesso di scrittura agli altri
- `chmod g=rx`
 - Assegna il permesso di lettura ed esecuzione, *ma non di scrittura* al gruppo

Lo sticky bit

- Lo *sticky bit* è identificato dal simbolo `t`
- Non usato sui file
- Nelle directory, solo il proprietario del file o root possono cancellare o rinominare i file contenuti

Esempio: la directory `/tmp`

```
$ ls -ld /tmp
```

```
drwxrwxrwt 6 root root 1024 Aug 10 01:03 /tmp
```

Setuid e Setgid

- Setuid (s): chi esegue il file diventa temporaneamente il proprietario
- Setgid (S): chi esegue il file diventa temporaneamente dello stesso gruppo del file

Esempio: il comando `passwd`

```
$ ls -l /usr/bin/passwd
```

```
-r-s--x--x 1 root root 17700 Jun 25 2004 /usr/bin/passwd
```

Parte 3: Visualizzazione e ricerca dei file

Visualizzazione per pagine

Esistono tre comandi quasi equivalenti:

- `pg file1 file2 ...`
- `more file1 file2 ...`
- `less file1 file2 ...`

Durante la visualizzazione è possibile dare dei comandi interattivi:

`spazio` prossima pagina

`invio` prossima riga

`b` pagina precedente

`/testo` prossima pagina contenente `testo`

`?testo` pagina precedente contenente `testo`

`q` termina la visualizzazione dei file

`v` modifica il file corrente

Ricerca di file per nome o attributi

- Visita tutto l'albero sotto la directory specificata e ritorna i file che rendono vera espressione:

```
find directory espressione
```

- espressioni:

```
-name pattern (usare gli apici " " se si usano metacaratteri)
```

```
-type tipo (b c d l f)
```

```
-user utente
```

```
-group gruppo
```

```
-newer file
```

```
-atime [+/-] giorni ultimo accesso
```

```
-mtime [+/-] giorni ultima modifica
```

Esempi di uso di `find`

- Tutti i file con estensione `.txt`: `$ find . -name "*.txt"`
- Tutti i link della directory corrente: `$ find . -type l`
- Tutti i file posseduti da `pippo` in `/tmp`: `$ find /tmp -user pippo`
- Tutti i file modificati da meno di un giorno in `/var/log`: `$ find /var/log -mtime -1`
- Tutti i file modificati da più di due giorni in `/var/log`: `$ find /var/log/ -mtime +2`

Ricerca di file per contenuto

- Il comando `grep` restituisce le linee dei file che contengono un pattern specificato
`grep [options] pattern [file1 file2 ...]`
- Opzioni:
 - i ignora la distinzione fra lettere maiuscole e minuscole
 - l fornisce la lista dei file che contengono il pattern
 - n le linee in output sono precedute dal numero di linea
 - v vengono restituite solo le linee che *non* contengono il pattern
 - w vengono restituite solo le linee che contengono il pattern come parola completa

Esempi di uso di `grep`

- visualizza le linee del file `/etc/passwd` che contengono la stringa `rossi`: `$ grep rossi /etc/passwd`
- visualizza le linee del file `relazione.txt` che contengono la stringa `Nel Fiume`, ignorando le maiuscole/minuscole: `$ grep -i "Nel Fiume" relazione.txt`
- visualizza le linee del file `relazione.txt` che non contengono il carattere "a", precedute dal numero di riga:
`$ grep -nv a relazione.txt`
- visualizza i nomi dei file con estensione `.c` che contengono la parola intera `print`: `$ grep -lw print *.c`