

Shell di UNIX

12/05/10

1

Parte I

I Comandi di base

12/05/10

2

Una sessione di lavoro

- **Apertura di una finestra di shell**
 - nei sistemi a finestre è sufficiente clickare sull'icona corrispondente
- **Fine di una sessione**
 - CTRL-d, exit, logout (dipende dall'interprete dei comandi)
- **NOTA:** all'interno della shell i caratteri maiuscoli sono diversi dai minuscoli!

I comandi in UNIX

- Sintassi, in generale, di un comando UNIX
comando [-opzioni] argomenti
- I comandi troppo lunghi possono essere continuati sulla riga successiva battendo “\” come ultimo carattere della riga
- Si possono dare più comandi sulla stessa riga separandoli con “;” (saranno eseguiti in sequenza)
comando1 ; comando2 ; ...
- Si possono dare comandi in “background” tra loro e rispetto la shell con “&”
comando1 & comando2 & ...

Il comando `ls`

- Per visualizzare il contenuto di una directory
`ls [-opzioni] file ...`

Opzioni

- a visualizza anche i file che iniziano con il punto
- l output in formato esteso
- g include/sopprime l'indicazione del proprietario
- t ordine per tempo di modifica del file (altrimenti si usa ordine alfabetico)
- r ordine inverso (alfabetico o temporale)
- R elenca anche i file nelle sottodirectory

12/05/10

5

Manipolazione di file

`cp [-fir] src1 src2 ... dest`

copia uno o più file

`rm [-fir] file1 file2 ...`

cancella i file elencati

`mv [-fi] file1 file2 ... dest`

sposta uno o più file/cambia il nome di un file

- f non chiede mai conferma (attenzione!!!)
- i chiede conferma per ciascun file
- r opera ricorsivamente nelle sottodirectory

12/05/10

6

Manipolazione di directory

cd *directory*

cambia la directory in quella indicata

pwd

mostra path directory corrente

mkdir *directory*

crea la directory specificata

rmdir *dir1 dir2 ...*

cancella una o più directory (devono essere vuote)

rm -rf *dir1 dir2 ...*

cancella una o più directory (anche se piene) 7

Esempi

- Listing dei files:
 - > ls
 - > ls -l
 - > ls -a
 - > ls -al
 - > ls -l /bin
 - > ...
- Creazione/rimozione di directory:
 - > mkdir d1
 - > rmdir d1
- Copia il file f1 in f2:
 - > cp f1 f2

Esempi

- Sposta/rinomina il file f1 in f2:
 - > mv f1 f2
- cp e mv come primo argomento possono prendere una lista di file; in tal caso il secondo argomento deve essere una directory:
 - > cp f1 f2 f3 d1 (copia f1, f2, f3 nella directory d1)

Visualizzazione di file di testo

cat file1 file2 ...

concatena i file sul flusso di standard output

head [-n] file1 file2

visualizza le prima *n* righe

tail [-nrf] file1 file2 ...

-r visualizza in ordine inverso

-f rilegge continuamente il file

-n visualizza le ultime *n* righe

Informazioni sul sistema

- Ogni utente è identificato dal suo login (UID) ed appartiene a uno o più gruppi (GID)
- Per avere informazioni sugli utenti o sul sistema:
 - **whoami**
 - **who am i**
 - **who**
 - **w**
 - **id**
 - **groups**
 - **finger**
 - **uname**
 - **passwd**
 - **su**
 - **date**

Help in linea

- Tutti i comandi di UNIX sono documentati in linea
 - **man comando**

Organizzano in sezioni corrispondenti ad argomenti

1. Commands
2. System Calls
3. Library Functions
4. Administrative Files
5. Miscellaneous Information
6. Games
7. I/O and Special Files
8. Maintenance Commands

Help in linea

- Oltre al man

apropos *chiave*

elenca le pagine del manuale contenente *chiave*

whatis *comando*

indica le sezioni del manuale in cui si trova *comando*

I metacaratteri

I metacaratteri in Unix

- La shell Unix riconosce alcuni caratteri speciali, chiamati metacaratteri, che possono comparire nei comandi.
- 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.

Esempio

- `user> ls *.java`
Albero.java div.java ProvaAlbero.java
AreaTriangolo.java EasyIn.java ProvaAlbero1.java
AreaTriangolo1.java IntQueue.java
- Il metacarattere `*` all'interno di un pathname è un'abbreviazione per un nome di file. Il pathname `*.java` viene espanso dalla shell con tutti i nomi di file che terminano con l'estensione `.java`. Il comando `ls` fornisce quindi la lista di tutti e soli i file con tale estensione.

Abbreviazione del Pathname

- I seguenti metacaratteri, chiamati *wildcard* sono usati per abbreviare il nome di un file in un pathname:

Metacarattere	Significato
*	stringa di 0 o più caratteri
?	singolo carattere

Esempi

- `user> 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*.)
- `user> ls *.*`
- `user> ls /dev/tty?`
(`/dev/ttya /dev/ttyb`)

I path

- . è la directory corrente
- .. è la directory padre di quella corrente
- I file che iniziano con . sono nascosti
- Path assoluto = /dir1/dir2/...
 - Parte dalla radice del file system
- Path relativo = dir1/dir2/...
 - Parte dalla cartella corrente

I file

- Un solo tipo di file fisico: byte stream
- 4 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

Special file

- Ogni device 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 !

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/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`
 - Quando tutti gli hard link che puntano allo stesso i-node sono stati rimossi allora viene rimosso l'i-node

Occupazione spazio su disco

- Per controllare l'occupazione dei dischi

df [-k -h]

Opzioni

- k mostra l'occupazione in KByte
- h mostra l'occupazione in formato "umano"

Occupazione spazio su disco

- Per vedere lo spazio (in blocchi) occupato da una directory e tutte le sottodirectory

du [-opzioni] directory ...

Opzioni

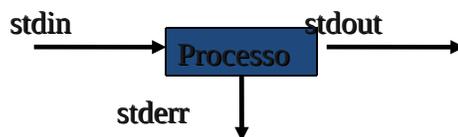
- a mostra l'occupazione di ciascun file
- s mostra solo il totale complessivo
- k mostra l'occupazione in KByte

12/05/10

25

Ri-direzione dell'I/O

- Ogni processo ha tre canali associati



- Ogni canale può essere ri-diretto
 - su file
 - su altro canale

12/05/10

26

Ri-direzione dell'I/O

comando < file

stdin da file

comando > file

stdout in file (se il file esiste già viene sovrascritto)

comando >> file

stdout aggiunto in coda al file

comando >& file

stderr e stdout in file (se il file esiste già viene sovrascritto)

comando 2> file

stderr in file (se il file esiste già viene sovrascritto)

12/05/10

27

Ri-direzione dell'I/O

comando > file 2>&1

comando &> file

stdout e stderr sullo stesso file descriptor

comando > fileout 2> fileerr

stdout di comando in fileout e stderr in fileerr

12/05/10

28

Ri-direzione dell'I/O

comando1 | comando2

pipe tra comando1 e comando2
(stdout di comando1 in stdin di
comando2)

Esempi

```
user> echo ciao a tutti >file      # ridirezione dell'output
user> more file
ciao a tutti
```

```
user> echo ciao a tutti >>file    # ridirezione dell'output (append)
user> more file
ciao a tutti
ciao a tutti
```

- Il comando `wc` (word counter) fornisce numero di linee, parole, caratteri di un file:
 - `user> wc <progetto.txt`
21 42 77
 - `user> wc <<delim # here document`
? queste linee formano il contenuto
? del testo
? delim
2 7 44
 - `user> man -s2 passwd` # ridirezione dei messaggi di errore
No entry for passwd in section(s) 2 of the manual.

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
 - CR prossima riga
 - b pagina precedente
 - */pattern* prossima pagina con *pattern*
 - *?pattern* pagina precedente con *pattern*
 - q termina programma
 - v edita file corrente

Cambio di proprietario

```
chgrp [-R] gruppo file
```

cambia il gruppo del file

```
chown [-R] utente[:gruppo] file
```

cambia proprietario [e gruppo] del file

- In entrambi i casi l'opzione -R indica di propagare il comando alle sottodirectory

Cambio protezione

chmod [-R] protezione file

Protezioni assolute: un numero di quattro cifre (il primo si può omettere)

	user	group	other
4 2 1	4 2 1	4 2 1	4 2 1
s S t	r w x	r w x	r w x

Protezioni simboliche: una stringa di tre caratteri a scelta tra quelli tra parentesi quadre
[u go a] [+ -=] [rwxst]

Simboli delle protezioni

u user

g group

o other

a all (user, group, other)

r read permission

w write permission

x execution permission

S setgid (vedi dopo)

s setuid (vedi dopo)

t sticky bit (vedi dopo)

Cambio protezione

- Esempi

- **chmod 640 prova.txt**

- Lettura/scrittura per proprietario
 - Lettura per gruppo
 - Nessun permesso per altri

- **chmod 755 dir**

- Lettura/scrittura/esecuzione per proprietario
 - Lettura/esecuzione per gruppo
 - Lettura/esecuzione per altri

- **chmod u+x filename**

- Aggiungi permesso di esecuzione per l'utente

Sticky bit

- Sticky bit (t)

- Non usato su file

- Per directory, solo il proprietario del file o root possono cancellare o rinominare i file contenuti (es. directory /tmp)

```
$ ls -ld
```

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

Setuid e setgid

- Setuid (s)
 - Per diventare temporaneamente il padrone del file
- Setgid (S)
 - Per diventare temporaneamente dello stesso gruppo del padrone del file

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

Ricerca di un file

find directory espressione

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

-name pattern (usare gli apici se si usano espressioni regolari)

-type tipo (b c d l f p)

-user utente

-group gruppo

-newer file

-atime [+/-] giorni #tempo di accesso

-mtime [+/-] giorni #tempo di modifica

Confronto di file

```
diff [-opzioni] file1 file2
diff [-opzioni] -r dir1 dir2
```

mostra le righe diverse, indicando quelle da aggiungere (a), cancellare (d) e cambiare (c)

- b ignora gli spazi a fine riga, collassa gli altri
- i ignora la differenza tra maiuscolo e minuscolo
- w ignora completamente la spaziatura

12/05/10

39

Confronto di file – Esempio

- Prova1
ciao
come va?
bene
grazie

- Prova 2
ciao
come?
bene
molto bene
grazie

- Prova 3
ciao

```
$ diff Prova1 Prova2
2c2
< come va?
---
> come?
4c4,5
< grazie
---
> molto bene
> grazie
```

```
$ diff Prova1 Prova3
2,4d1
< come va?
< bene
< grazie
```

```
$ diff Prova3 Prova1
1a2,4
> come va?
> bene
> grazie
```

12/05/10

40

Modifica di attributi di file

touch [-opzioni] [data] file ...

aggiorna data e ora dell'ultimo
accesso/modifica di un file

- se data non è specificata, usa data e ora corrente
- se il file non esiste lo crea vuoto
- opzioni
 - a cambia tempo di accesso del file
 - m cambia tempo di modifica del file