



Sistemi Operativi¹

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

a.a. 2019/20

¹© 2008–19 M. Monga. Creative Commons Attribution — Condividi allo stesso modo 4.0 Internazionale. <http://creativecommons.org/licenses/by-sa/4.0/deed.it>. Immagini tratte da [2] e da Wikipedia.



Lezione XV: Shell e file system



Shell e file system

- Ogni processo (compresa la shell stessa) ha associata una *directory di lavoro* (working directory), che può essere cambiata col comando (interno alla shell) `cd`
- I programmi fondamentali per operare sul file system

<code>ls (1)</code>	list directory contents
<code>cp (1)</code>	copy files and directories
<code>rm (1)</code>	remove files or directories
<code>mv (1)</code>	move (rename) files
<code>mkdir (1)</code>	make directories
<code>rmdir (1)</code>	remove empty directories
<code>df (1)</code>	report file system disk space usage
<code>du (1)</code>	estimate file space usage
<code>pwd (1)</code>	print name of current/working directory



Permessi

A ogni file vengono associati dei *permessi*, che definiscono le azioni permesse sui dati del file

- **Read:** leggere il contenuto del file o directory
- **Write:** scrivere (cambiare) il file o directory
- **eXecute** eseguire le istruzioni contenute nel file o accedere alla directory

R	W	X	
1	1	0	6
1	0	1	5
1	0	0	4
1	1	1	7

I permessi possono essere diversi per 3 categorie di utenti del sistema:

- **User:** il “proprietario” del file
- **Group:** gli appartenenti al gruppo proprietario
- **All:** tutti gli altri



- Cambiare il proprietario
 - `chown utente[:gruppo] file`
- Cambiare il gruppo
 - `chgrp gruppo file`
- Cambiare i permessi
 - `chmod 755 file`
 - `chmod +x file`
 - `chmod a=rw file`
 - `chmod g-x file`
- (per creare un utente: `adduser`)



Il proprietario di un processo in esecuzione è normalmente *diverso* dal proprietario del file contenente un programma (e diverso ad ogni esecuzione)

- effective UID bit: il processo assume come proprietario il proprietario del file del programma
- SUID root
- `chmod 4555 file`
- `chmod u+s file`



Per selezionare file con determinate caratteristiche si usa `find`
`find percorso predicato`
Seleziona, nel sottoalbero definito dal percorso, tutti i file per cui il predicato è vero
Spesso usato insieme a `xargs`
`find percorso predicato | xargs comando`
funzionalmente equivalente a
`comando $(find percorso predicato)`
ma evita i problemi di lunghezza della riga di comando perché `xargs` si preoccupa di “spezzarla” opportunamente.



Spesso si vuole fare un'operazione per ogni file trovato con `find`. L'espressione più naturale sarebbe:

```
for i in $(find percorso predicato); do
    comando $i
done
```

Questa forma presenta due problemi: può eccedere la misura della linea di comando e non funziona correttamente se i nomi dei file contengono *spazi*

Due espressioni idiomatiche



Sistemi Operativi

Bruschi Monga

Shell e file system

File system

Unix power tools

find

Archivi

Un'alternativa è

```
find percorso predicato -print0 | xargs -0 -n 1
```

In questo modo (`-print0`) i file trovati sono separati dal carattere `0` anziché spazi e `xargs` è capace di adattarsi a questa forma.

Un'alternativa più generale che mostra la potenza del linguaggio di shell che non distingue fra comandi e costrutti di controllo di flusso (sono tutti "comandi" utilizzabili in una pipeline)

```
find percorso predicato | while read x; do
  comando $x
done
```

`read x` legge una stringa e la assegna alla variabile `x`.

287

Esercizi



Sistemi Operativi

Bruschi Monga

Shell e file system

File system

Unix power tools

find

Archivi

- 1 Trovare il file più "grosso" in un certo ramo
- 2 Copiare alcuni file (ad es. il cui nome segue un certo pattern) di un ramo in un altro mantenendo la gerarchia delle directory
- 3 Calcolare lo spazio occupato dai file di proprietà di un certo utente
- 4 Scrivere un comando che conta quanti file ci sono in un determinato ramo del filesystem

288

Archivi



Sistemi Operativi

Bruschi Monga

Shell e file system

File system

Unix power tools

find

Archivi

Un archivio *archive* è un file di file, cioè un file che contiene i byte di diversi altri file e i relativi *metadati*. (Cfr. con una directory, che è un file speciale, che sostanzialmente contiene solo l'elenco dei file)

- `ar` L'archivatore classico, generalmente utilizzato per le librerie (provare `ar t /usr/lib/i86/libc.a`)
- `tar` Tape archive, standard POSIX
`tar cvf archivio.tar lista_files`

Gli archivi possono essere compressi con `compress` o, più comunemente, con `gzip` o `bzip2`

I file `.zip` sono archivi compressi.

289

Altre utility



Sistemi Operativi

Bruschi Monga

Shell e file system

File system

Unix power tools

find

Archivi

Altre utility "standard" di cui è bene conoscere almeno l'esistenza

Prog. (sez. man)	Descrizione
<code>uniq</code> (1)	report or omit repeated lines
<code>cut</code> (1)	remove sections from each line of files
<code>tr</code> (1)	translate or delete characters
<code>dd</code> (1)	convert and copy a file
<code>tee</code> (1)	read from standard input and write to standard output ...
<code>sed</code> (1)	stream editor for filtering and transforming text
<code>seq</code> (1)	print a sequence of numbers

Inoltre è molto utile conoscere le espressioni regolari (man 7 `re_format`), usate da `grep`, `sed`, ecc.

290



Altre utility "standard" di cui è bene conoscere almeno l'esistenza

Prog. (sez. man)	Descrizione
basename (1)	strip directory and suffix from filenames
dirname (1)	strip non-directory suffix from file name
uniq (1)	report or omit repeated lines
cut (1)	remove sections from each line of files
tr (1)	translate or delete characters
dd (1)	convert and copy a file
stat (1)	display file or file system status

cd invece non è un programma, ma un comando interno della shell (che differenza fa?)



- 1 Creare un archivio tar.gz contenente tutti i file la cui dimensione è minore di 50KB
- 2 Rinominare un certo numero di file: per esempio tutti i file .png in .jpg
- 3 Creare un file da 10MB costituito da caratteri casuali (usando /dev/random) e verificare se contiene la parola JOS
- 4 Trovare l'utente che ha il maggior numero di file nel sistema
- 5 Trovare i 3 utenti che, sommando la dimensione dei loro file, occupano più spazio nel sistema.