



DICo

Sistemi
Operativi

Bruschi
Monga

Esercizio
memorie di
massa

Shell e file
system
File system

Unix power
tools
find
Archivi

Sistemi Operativi¹

Mattia Monga

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

a.a. 2011/12



DICo

Sistemi
Operativi

Bruschi
Monga

Esercizio
memorie di
massa

Shell e file
system
File system

Unix power
tools
find
Archivi

Lezione XV: Unix power tools e primi esperimenti col kernel



- 1 Creare un disco virtuale
- 2 Partizionare il disco
- 3 Creare il file system
- 4 Montare il file system

1 Creare un disco virtuale

```
1 qemu-img create prova.img 10M
2
3 dd if=/dev/zero of=prova.img bs=1024 count=10240
4
5 qemu -hda minix -hdb prova.img
```

2 Partizionare il disco (con part) c0d1: primo controller IDE (c0), secondo disco (d1)

3 Creare il file system mkfs /dev/c0d1p0

4 Montare il file system mount /dev/c0d1p0 /mnt

Montare un file system



DICo

Sistemi
Operativi

Bruschi
Monga

Esercizio
memorie di
massa

Shell e file
system
File system

Unix power
tools
find
Archivi

Il file system di UNIX è un albero (in realtà un DAG, perché come si vedrà ci possono essere dei link “trasversali”) con una sola radice.

L’introduzione di un nuovo pezzo di file system (per esempio contenuto in un dischetto, una chiavetta USB, un disco di rete) si dice **montare** il file system. Occorre scegliere un punto di montaggio: il file system diventerà un sottoalbero la cui radice è tale punto (eventuali sottoalberi già esistenti non saranno più accessibili)

```
mount /dev/fd0 /mnt
```

- 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 |

Ad 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

Esercizio
memorie di
massa

Shell e file
system

File system

Unix power
tools

find

Archivi



- 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`)

Esercizio
memorie di
massa

Shell e file
system

File system

Unix power
tools

find

Archivi



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`

Esercizio
memorie di
massa

Shell e file
system
File system

Unix power
tools
find
Archivi



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.



- 1 Trovare il file piú “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

Esercizio
memorie di
massa

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'archiviatore 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 ma assenti nel setup di MINIX di base, con `gzip` o `bzip2`

I file `.zip` sono archivi compressi.

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

| Prog. (sez. man) | Descrizione |
|------------------|---|
| od (1) | dump files in octal and other formats |
| 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 |
| test (1) | check file types and compare values |
| tee (1) | read from standard input and write to standard output ... |
| basename (1) | strip directory and suffix from filenames |
| dirname (1) | strip non-directory suffix from file name |
| sed (1) | stream editor for filtering and transforming text |

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



- 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 MINIX
- 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.



- [http://www.gnu.org/software/fileutils/
fileutils.html](http://www.gnu.org/software/fileutils/fileutils.html)