

Gestione file e directory: il file system

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**
 - Entry point per 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!

Link

- **Hard link**
 - Un nome (in una directory) che punta a un i-node puntato anche da altri
- **Soft 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 altri file system
 - Un file viene rimosso quando tutti i suoi hard link sono stati rimossi

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
- r ordine inverso (alfabetico o temporale)
- F appende carattere per indicare i file particolari (/ * @)
- R elenca anche i file nelle sottodirectory

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

Visualizzazione di file di testo

cat file1 file2 ...

concatena i file sullo std output

head [-n] file1 file2

visualizza le prima *n* righe

tail [-+nrf] file1 file2 ...

Visualizza le ultime (con + salta le prime) 10 righe

-r visualizza in ordine inverso

-f rilegge continuamente il file

-n visualizza (salta) le ultime (prime) *n* righe

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

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

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)

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)

	padrone	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

Protezioni simboliche: una stringa di tre caratteri

ugo a + - = rwxst

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

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
```

Protezioni standard

umask *maschera*

Per definire la maschera delle protezioni

- Il comando **umask** senza argomento mostra i permessi che sono NEGATI quando si crea un file (la maschera delle protezioni)
- Esempio:
umask 027
Nega tutti i permessi agli “altri” e i permessi di scrittura al “gruppo”

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)
- user utente
- group gruppo
- newer file
- atime, mtime, ctime [+/-] giorni
- print
- size [+/-] blocchi

Confronto di file

```
diff [-opzioni] file1 file2
```

```
diff [-opzioni] 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

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
```

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

-a modifica accesso

-m ultima modifica