Linux e la shell Bash

Gestione file e directory

- Directory di sistema
- Percorsi relativi e assoluti
- I comandi: *pwd, touch, cd, ls, mkdir, rmdir, cp, rm, mv, file, stat*
- Esercizi

Testi di riferimento:

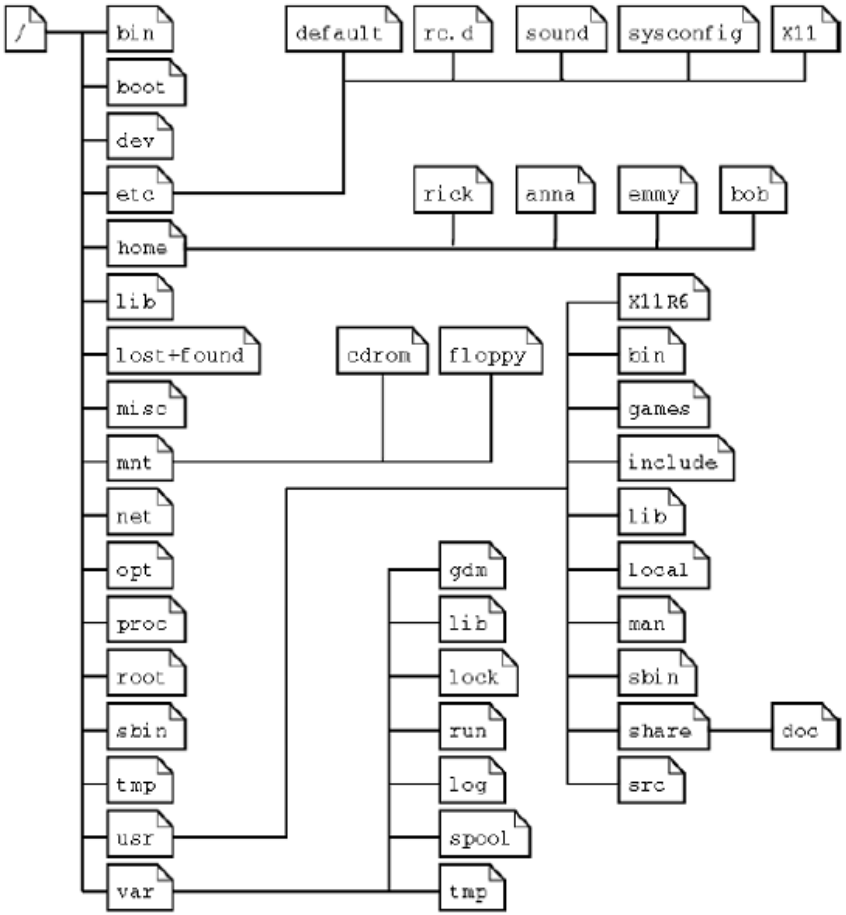
- Linux e la shell Bash http://sisinflab.poliba.it/ruta/linux/Linux e la shell Bash.pdf
- La shell Bash http://www-ictserv.poliba.it/piscitelli/doc/lab_esercitazioni_SO_no/bash_shell.pdf
- Comandi shell Bash http://sisinflab.poliba.it/giannini/SO/Comandi_shell_bash.pdf

Il file system di Unix (1/3)

- In Linux "tutto è un file"
- Le directory sono dei file che contengono tanti oggetti, cioè tanti directory entry quanti sono i file in esso contenuti
- Questo efficace meccanismo permette di:
  - Effettuare collegamenti tra le directory;
  - Attribuire specifici permessi a ciascuna directory;
  - Gestire in modo flessibile l'albero delle directory;
  - Ospitare un numero elevato di subdirectory.

Il file system di Unix (2/3)

- Il file system di Unix ha un'organizzazione gerarchica ad albero, la cui radice (root) è una directory identificata dal carattere /.



- bin: programmi binari essenziali utilizzabili da tutti gli utenti
- sbin: (superuser bin) programmi binari essenziali per l'amministrazione del sistema
- boot: file necessari all'avvio del sistema quali il kernel ed i driver
- dev: file speciali associati a dispositivi hardware
- etc: file di configurazione
- home: home directory degli utenti che hanno un account sulla macchina
- root: home directory del root user
- tmp: file temporanei
- usr: programmi installati e dati da condividere read-only
- var: file che il sistema scrive durante l'esecuzione (log, spool, cache, ...)
- lib: librerie di sistema
- opt: programmi ottenuti da installazioni non standard
- proc: file system virtuale in memoria, contiene informazioni relative al sistema aggiornate dinamicamente
- media: Punto di accesso per i dispositivi di massa (CD-ROM, pen drive, ...)
- mnt: come /media, per montaggi manuali

Il file system di Unix (3/3)

- I nomi di file e directory possono essere composti da qualsiasi sequenza di caratteri eccetto /
- Non utilizzare gli spazi
- La struttura ad albero consente di identificare univocamente un elemento del file system mediante il percorso che va dalla root directory / al file stesso. Questo percorso viene detto pathname
- Nel pathname tutte le directory del percorso vengono separate dal carattere /
- Ogni directory contiene 2 directory speciali (hard link): «.», che corrisponde alla directory stessa; «..», che corrisponde alla directory padre
- La directory padre di / è / stessa
- Un pathname nella shell si può scrivere in due modi:
  - pathname assoluto specificando il percorso completo dalla directory radice
  - pathname relativo scrivendo il percorso a partire dalla directory corrente
- E' chiaro che specificando un percorso assoluto si può indicare un qualsiasi file o directory mentre col percorso relativo si è limitati a quanto contenuto a partire dalla directory di lavoro corrente

pwd

- pwd: (print working directory) mostra sullo standard output il percorso completo (pathname assoluto) della directory di lavoro corrente.

SINTASSI

- **pwd**

ESEMPIO

- user@ubuntu:~$ pwd

  /home/user

touch

- touch: Aggiorna data ed ora di ultima referenziazione del file in argomento (di default cambia sia l'orario di accesso che di modifica), con l'orario corrente. Se il file non esiste, crea un file vuoto con data e ora corrente

SINTASSI

- **touch [opzioni] path**

FLAG

- -a Cambia l'orario di accesso
- -c Non crea alcun file
- -m Cambia l'orario di modifica

ESEMPIO

- touch myfile
  - Se myfile esiste aggiorna l'orario di ultimo accesso e di ultima modifica con quello corrente, altrimenti crea myfile
- touch –cm myfile
  - Se myfile esiste aggiorna l'orario di ultima modifica, se non esiste non fa nulla

cd

- cd: (change directory) cambia la directory corrente in quella specificata.

SINTASSI

- **cd [path]**

ESEMPI Directory Attuale: /home/user/Desktop/shell_linux

- cd se non si specifica la directory, la directory di lavoro diventerà la home-directory dell'utente loggato. Equivalente a: cd ~

/home/user/

- cd - La directory di lavoro diventerà quella precedentemente utilizzata, questa è memorizzata nella variabile d'ambiente OLDPWD.

/home/user/Desktop/shell_linux

- cd . La directory di lavoro resterà quella corrente.

/home/user/Desktop/shell_linux

- cd .. La directory di lavoro diventerà quella padre di quella corrente.

/home/user/Desktop

ls 1/2

- ls: (list) lista il contenuto della directory corrente o di quella specificata.
- Se utilizzato senza argomenti il comando ls visualizza il contenuto della directory corrente

SINTASSI

- **ls [opzioni] [path]**

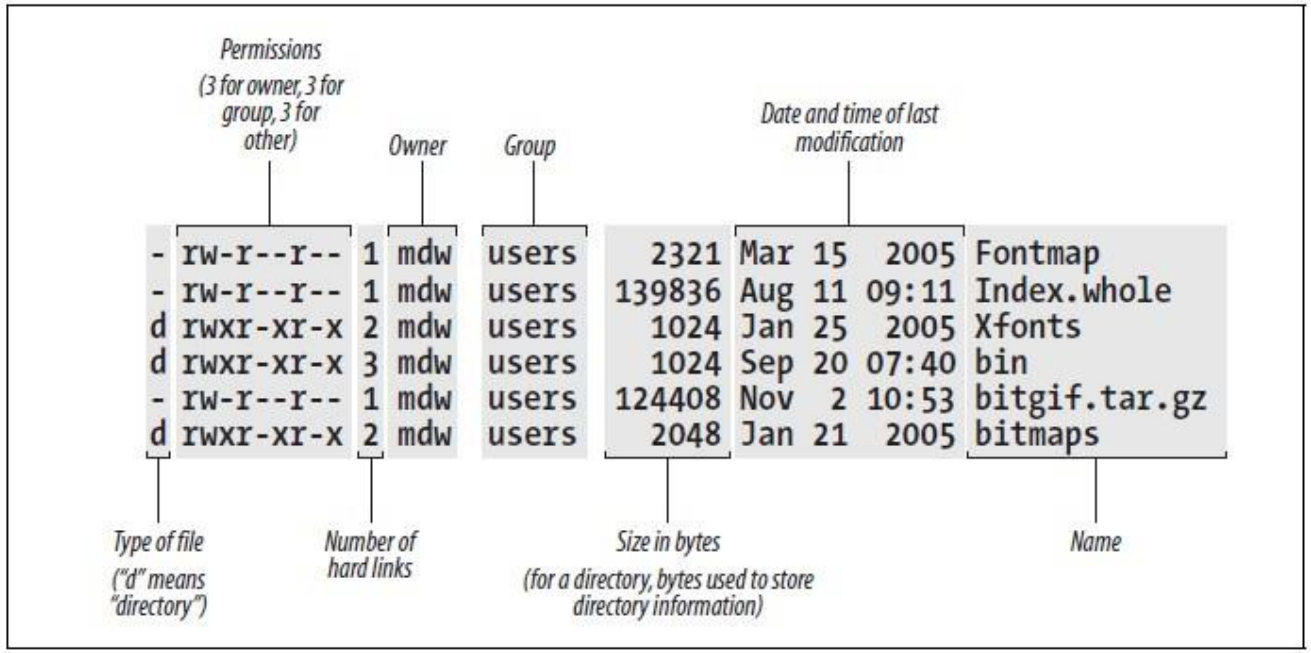
FLAG

- -a (all) mostra tutti i file, anche i file nascosti (che iniziano con il carattere ".")
- -F associa un identificatore per ogni tipo di file in coda al nome (* file eseguibili, / directory, @ per i link simbolici, | per i FIFO)
- -r inverte l'ordinamento alfabetico di visualizzazione
- -R visualizza in modo ricorsivo il contenuto delle sottodirectory
- -t ordina i file per data di ultima modifica, dal più recente
- -i visualizza l'inode del file

ls 2/2

- -u ordina i file per data di ultimo accesso, dal più recente
- -1 visulizza l'elenco in un'unica colonna.
- -l mostra informazioni dettagliate sui file in maniera tabellare:

  [Access Control List] [numero di link al file] [utente] [gruppo] [dimensione in byte] [data e ora ultima modifica] [nome file]



mkdir

- mkdir: (make directory) crea una nuova directory

SINTASSI

- **mkdir [opzioni] nomeDirectory**

FLAG

- -m ACL crea una directory specificando l'ACL
- -v attiva la modalità verbose

ESEMPI

- mkdir Scrivania/prova
- mkdir Scrivania/prova1 Scrivania/prova2
- mkdir –m 766 Scrivania/prova3
- Come verificare contestualmente che la creazione sia andata a buon fine?
  - mkdir –vm 766 prova4

rmdir

- rmdir: (remove directory) cancella una directory, la directory deve essere vuota

SINTASSI

- **rmdir [opzioni] nomeDirectory_1 [nomeDirectory_n]**

FLAG

- --ignore-fail-on-non-empty non mostra a video il messaggio di errore in caso di fallimento
- -v attiva la modalità verbose

cp

- cp: (copy) copia il file (o una directory vuota) dal percorso di partenza a quello di destinazione.
  - se il file di destinazione non esiste viene creato
  - se il file di destinazione esiste viene sovrascritto
  - il percorso di destinazione può comprendere anche il nuovo nome del file/file-directory, in quest'ultimo caso si effettuerà contemporaneamente la copia e la rinomina del file/file-directory di partenza
  - la copia di cartelle non vuote va effettuata in maniera ricorsiva

SINTASSI

- **cp [opzioni] pathFileOrigine pathFileDestinazione**

FLAG

- -f forza la sovrascrittura
- -i richiede interattivamente la conferma prima dell'eventuale sovrascrittura di file di destinazione esistenti
- -p mantiene le caratteristiche (permessi, etc.) del file sorgente compresa la data e l'ora di referenziazione
- -R copia ricorsiva
- -v attiva la modalità verbose, che visualizza in output quello che il sistema ha fatto in seguito al nostro comando

rm

- rm: (remove) rimuove un file (o una directory in maniera ricorsiva se non vuota -R)

SINTASSI

- **rm [opzioni] nomeFile**

FLAG

- -R cancellazione ricorsiva del contenuto di una directory e della directory stessa se non vuota
- -v attiva la modalità verbose

ESEMPI

- rm –Rv provaFull
- rm file1

mv

- mv: (move) sposta (taglia e incolla/rinomina) file o directory da un percorso di partenza ad uno di destinazione

SINTASSI

- **mv [opzioni] pathFileOrigine pathFileDestinazione**

FLAG

- -f forza la sovrascrittura
- -i richiede interattivamente la conferma di sovrascrittura
- -v attiva la modalità verbose

ESEMPI

• mv file1 file2	Rinomina file1 in file2. Se quest'ultimo esiste viene sovrascritto da file1
• mv file1 directoryA	Sposta file1 in directoryA
• mv directoryA directoryB	Sposta directoryA in directoryB se quest'ultima esiste già, viceversa rinomina directoryA in directoryB
• mv directoryA file1	Operazione non consentita se file1 è un file esistente

file

- file: Determina il tipo di uno o più file

SINTASSI

- **file [opzioni] nomeFile_1 [nomeFile_n]**

ESEMPI

- gianna@ubuntu:~$ file *
  - DivinaCommedia: UTF-8 Unicode text, with very long lines
  - Documenti: directory
  - fileempty: empty
  - exe.sh: Bourne-Again shell script, ASCII text executable
  - Immagini: directory
  - StarWars: ASCII text
  - …

Mostra il tipo di tutti i file (e directory), qualunque sia il loro nome (metacarattere "*"), contenuti nella directory corrente

stat

- stat: ad ogni file sono associate una serie di informazioni o metadati, quali la dimensione, i permessi, le date di accesso e di modifica, eccetera. Queste informazioni possono essere lette dal filesystem e vengono memorizzate in una struttura dati chiamata stat.

- Per avere informazioni dettagliate su di un file si può utilizzare il comando omonimo

SINTASSI

- **stat nomeFile**

ESEMPI

```
gianna@gianna-X550LD:~$ stat Esempi_Lezione/DivinaCommedia
  File: "Esempi_Lezione/DivinaCommedia"
  Dim.: 557222          Blocchi: 1096       Blocco di IO: 4096   file regolare
Device: 808h/2056d      Inode: 139410      Coll.: 1
Accesso: (0664/-rw-rw-r--)  Uid: ( 1000/  gianna)   Gid: ( 1000/  gianna)
Accesso  : 2016-11-23 19:59:13.492292000 +0100
Modifica : 2016-11-23 11:09:56.000000000 +0100
Cambio   : 2016-11-23 19:59:13.496294136 +0100
Creazione: -
```

Esercizi

1) Scrivere le istruzioni per spostarsi nella directory *dir12/dir121* presente nella home directory dell'utente, supponendo di essere nella directory */home/user/dir1/dir11* utilizzando il pathname relativo e il pathname assoluto.

Soluzione

- Pathname relativo
  cd ../../dir12/dir121
- Pathname assoluto
  cd /home/user/dir12/dir121

2) Spiegare il significato della seguente ACL: *lrwxr-xr-x* relativa al file *myFile*, scrivere la maschera in ottale del set di permessi e il comando per privare del permesso di esecuzione gli utenti generici della macchina.

Soluzione

- lrwxr-xr-x → link simbolico; u: tutti i permessi (lettura, scrittura, esecuzione); g: lettura, esecuzione; o: lettura, esecuzione
- Maschera in ottale: 755 → u: 20+21+22=7 g: 20+22=5 o: 20+22=5
- chmod o-x myFile

3) Il comando ls -1F produce il seguente output

file1
file2
file3
file4/

- Spiegare cosa fa il comando ls -1F
- Indicare quale dei seguenti comandi è esatto
  mv file1 file2 file3
  mv file1 file2 file4 ← comando esatto
- Spiegare perché il comando è esatto e descriverne il funzionamento

Soluzione

- Il comando ls lista il contenuto di una directory, il flag -1 visualizza l'elenco in un'unica colonna, il flag –F aggiunge in coda al nome del file un identificatore, in questo caso file1 file2 e file3 sono file normali, file4 è una directory.
- Il comando *mv file1 file2 file4* sposta file1 e file2 nella directory file4.

4) Verificare in quale directory si trova l'utente, rinominare il file myFile1 presente nella directory in cui si trova l'utente assegnandogli il nome myFile2 spostandolo contemporaneamente nella directory padre della directory corrente dell'utente.

Soluzione
- pwd
- mv myFile1 ../myFile2

5) Usando esclusivamente la notazione numerica, si cambino i permessi di accesso al file myFile in modo tale che risultino: - r-xrw-r- -

Soluzione
- chmod 564 myFile

Esercizi

6) Dato il file *prova* dotato dei seguenti permessi di accesso - r-x r-- rwx indicare come cambiano tali permessi quando vengono lanciati i comandi:
   - chmod 755 prova
   - chmod g-r+w prova
   - chmod o-x prova

Soluzione
- r-x r-- rwx (547 u: lettura, esecuzione; g: lettura; o: lettura, scrittura, esecuzione)
- 547 → 755 : rwx r-x r-x (u: lettura, scrittura, esecuzione; g: lettura, esecuzione; o: lettura, esecuzione)
- 755 → g-r+w : rwx -wx r-x (735 u: lettura, scrittura, esecuzione; g: scrittura, esecuzione; o: lettura, esecuzione)
- 735 → o-x : rwx –wx r - - (734 u: lettura, scrittura, esecuzione; g: scrittura, esecuzione; o: lettura)

7) Siamo nella home directory dell'utente, creare il file *home* nella directory /home/user/Scrivania/, tagliarlo e incollarlo in /home/user/, visualizzare in output quello che il sistema ha fatto in seguito al nostro comando ed infine rimuovere il file.

Soluzione

- touch ./Scrivania/home
- mv -v ./Scrivania/home /home/user/
- rm ./home

8) Di seguito sono listati alcuni attributi del file *mioFile*. Scrivere l'effetto sui permessi dei seguenti comandi.
N.B. Ogni comando agisce sui permessi originali di *mioFile*.

rwxr-x--- 2 user1 groupA 4096 5 dic 17:00 mioFile

- chmod 713 mioFile
- chmod og+x-r mioFile
- chmod 134 mioFile

Soluzione

- 750 (rwxr-x---) → 713 (rwx--x-wx)
- 750 (rwxr-x---) → 711 (rwx--x--x)
- 750 (rwxr-x---) → 134 (--x-wxr--)

9) Eseguire le seguenti operazioni:
   a. All'interno della vostra home directory create 2 directory denominate *uno* e *due*.
   b. Copiate il file /etc/profile nella directory *uno*, conservandone il nome.
   c. Copiate il file /etc/profile nella directory *due* cambiandone il nome in *copia-profile*.
   d. Spostate il file *profile* nella directory *due* ed il file *copia-profile* nella directory *uno*.
   e. Cancellate i due file con uno stesso comando.
   f. Cancellate le due directory vuote.

Soluzione

Soluzione

a. mkdir uno due
b. cp /etc/profile uno
c. cp /etc/profile due/copia-profile
d. mv uno/profile due
e. mv due/copia-profile uno
f. rm uno/copia-profile due/profile
g. rmdir uno due

N.B. Se il flag –v segue il nome del comando, in output sarà visualizzato quello che il sistema ha fatto.

10) Eseguire le seguenti operazioni:

a. Copiate il file */bin/ls* nella vostra home directory specificando il pathname.
b. Eliminate i permessi di esecuzione a tutti gli utenti per la copia locale.
c. Ripristinate i permessi di esecuzione al solo proprietario del file.
d. Create una directory *test* nella vostra home directory e copiate il file *ls* al suo interno
e. Eliminate i permessi di esecuzione alla directory e provate a listare il contenuto. Perché non è concessa l'operazione?
f. Ripristinate i vecchi permessi e cancellate con un unico comando il file all'interno della directory e la directory stessa.

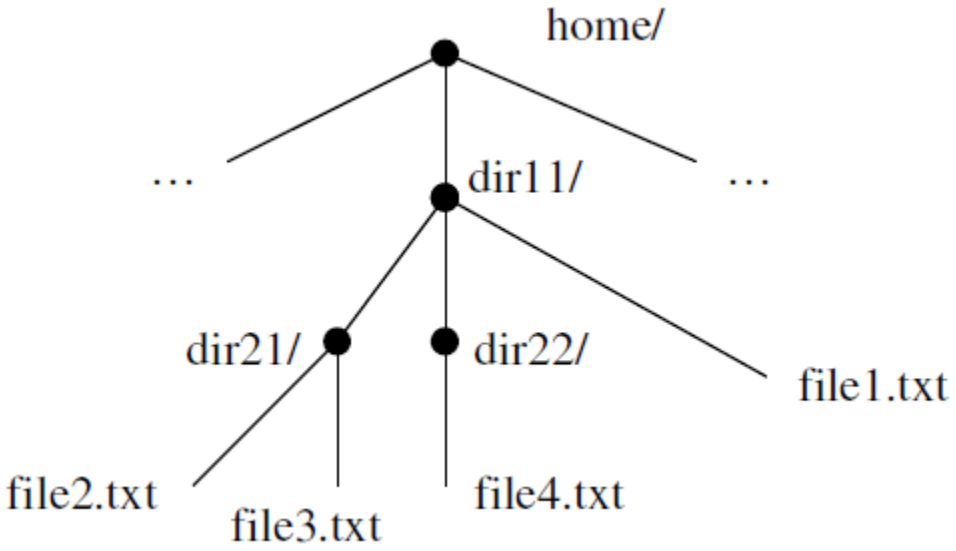
Soluzione

Soluzione

a. cp /bin/ls /home/user/
b.
   - chmod a-x ls
   - chmod ugo-x ls
   - chmod -x ls
c. chmod u+x ls
d.
   - mkdir test
   - cp ls test
e.
   - chmod –x test
   - ls test
f.
   - chmod +x test
   - rm –R test

Esercizi

11) Partendo dalla propria home ("home" nel disegno), attraverso opportuni comandi di shell, si creino le cartelle e i file necessari a riprodurre la situazione raffigurata di seguito



12) Si impostino come segue i permessi delle cartelle e dei file: si assegnino per la cartella dir11/ ed a tutto il suo contenuto diritti di lettura, scrittura ed esecuzione al proprietario, diritti di sola lettura agli utenti del gruppo e nessun diritto agli altri utenti. A questo punto, solo per il file file3.txt, si impostino invece solo diritti di lettura per il proprietario, nessun diritto per il gruppo e nessun diritto per gli altri utenti.

Soluzione

Soluzioni

11) Comandi per generare l'albero in figura
- mkdir dir11
- cd dir11
- mkdir dir21 dir22
- touch file1.txt
- cd dir21
- touch file2.txt file3.txt
- cd ../dir22
- touch file4.txt

12) Cambiamento dei permessi
- chmod -R 740 dir11
- chmod 400 dir11/dir21/file3.txt

Esercizi + Soluzione

13) Partendo dalla cartella dir11/, attraverso opportuni comandi di shell si sposti il file file4.txt nella directory dir11/ rinominandolo come file5.txt, si sposti la directory dir21/ con il suo contenuto nella directory dir22/ ed infine si cancelli la directory dir11/ con tutto il suo contenuto. Modificare i permessi di lettura/scrittura quando necessario.

Soluzione

- cd dir11
- mv dir22/file4.txt file5.txt
- mv dir21 dir22
- cd ..
- chmod u+w dir11/dir22/dir21/file3.txt
- rm -R dir11

Esercizi

14) Disegnare l'albero di directory (e di file) generato dalla seguente sequenza di comandi. La directory di partenza è la home directory dell'utente.

- mkdir uno due
- cd uno
- touch testo1
- mv testo1 ..
- cd ../due
- touch testo2
- cp ../testo1 .

Soluzione

Soluzione