

Uso di Linux

Lezione III

Scopo della lezione

- Richiamare le principali funzionalità di un sistema operativo
- Esemplificarle descrivendo la loro implementazione in Linux
- Stabilire un linguaggio comune



Peculiarità di Linux

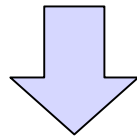
- E' un sistema operativo radicato sul concetto di shell (o terminale): l'interfaccia con cui l'utente dialoga con l'elaboratore si basa su un terminale a caratteri

```
xterm
4 -rw----- 1 malchiod users      500 Jan 31  2002 MyCodes
24 -rw-r--r-- 1 malchiod users  24064 Oct  4 18:08 NEURALGrid.ppt
968 -rw-r--r-- 1 malchiod users  985198 Oct  4 18:08 NEURALGrid.zip
228 -rw-r--r-- 1 malchiod users  226816 Mar 22  2002 Narrowing2.doc
12904 -rw-r--r-- 1 malchiod users 13192708 Oct  1  2001 NetBeans-release321.tar.gz
4 drwxr-xr-x 2 malchiod users    4096 Sep 13 15:41 Nuova cartella
32 -rw-r--r-- 1 malchiod users   29184 Nov 21  2001 OREST-Synthesis.doc
4 drwxr-xr-x 6 malchiod users    4096 May 13 17:14 ORESTEIA
56 -rw-r--r-- 1 malchiod users   52736 Sep 27 09:56 OrariIGTurno2.xls
76 -rw-r--r-- 1 malchiod users   72980 Jul 24 17:57 OreDario.gnumeric
40 -rw-r--r-- 1 malchiod users   38400 Jun 20  2001 OreDario.xls
4 -rw-r--r-- 1 malchiod users    2624 May 23  2001 OreDario.xls.gnumeric
28 -rw-r--r-- 1 malchiod users   27916 Jul 15 09:52 P15B233A6.png
464 -rw-r--r-- 1 malchiod users  469263 Jun 22  2001 PAC_meditating.rtf
276 -rw-r--r-- 1 malchiod users  274623 Jun 22  2001 PAC_meditating_on_emotions
16 -rw-r--r-- 1 malchiod web     13146 Nov 27  2001 PACm.zip
4 drwxr-xr-x 8 malchiod users    4096 Mar 26  2002 PHYSTA
32 -rw-r--r-- 1 malchiod users   32768 Dec 10  2001 PROPOSED AGENDA FOR THE ATELIER ON technical issues.doc
32 -rw-r--r-- 1 malchiod users   31744 Oct 30  2001 PacM.doc
364 -rw-r--r-- 1 malchiod users  367220 Aug 29  2001 Popo-1.jpg
144 -rw-r--r-- 1 malchiod users  139776 Sep  5 16:54 PosterTemplate.doc
1836 -rw-r--r-- 1 malchiod users 1875745 Apr 10  2001 PresMath.nb
256 -rw-r--r-- 1 malchiod users  256474 Apr 10  2001 PresMath.zip
line 122
```

Storia di Linux

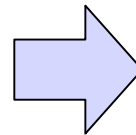
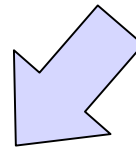
<http://ragib.hypermart.net/linux>

Andrew Tanenbaum
(MINIX)



Linus Torwald
(Linux)

Richard Stallman
(GNU)



Linux Community

Versioni di Linux

- Le differenti versioni di Linux dipendono
 - dal **kernel** installato sul sistema
 - dalla **distribuzione** (Slackware, Red Hat, Debian, Mandrake, Suse, ...)

Funzionalità di un s.o.

- **Gestione dell'interfaccia**
- Gestione dell'hardware
- **Gestione degli utenti**
- **Gestione degli appunti (clipboard)**
- Gestione della memoria
- **Gestione dei processi**
- **Gestione del file system**
- Gestione della rete
- **Programmi di utilità**

Ruolo dell'interfaccia grafica

- A differenza di Windows, in Linux l'interfaccia grafica è una componente **opzionale** (si possono anche installare differenti interfacce)
- Comune a tutti i sistemi è invece la possibilità di interfacciarsi con l'utente tramite dei comandi impartiti usando solamente la tastiera e un opportuno interprete (**shell**)

Shell

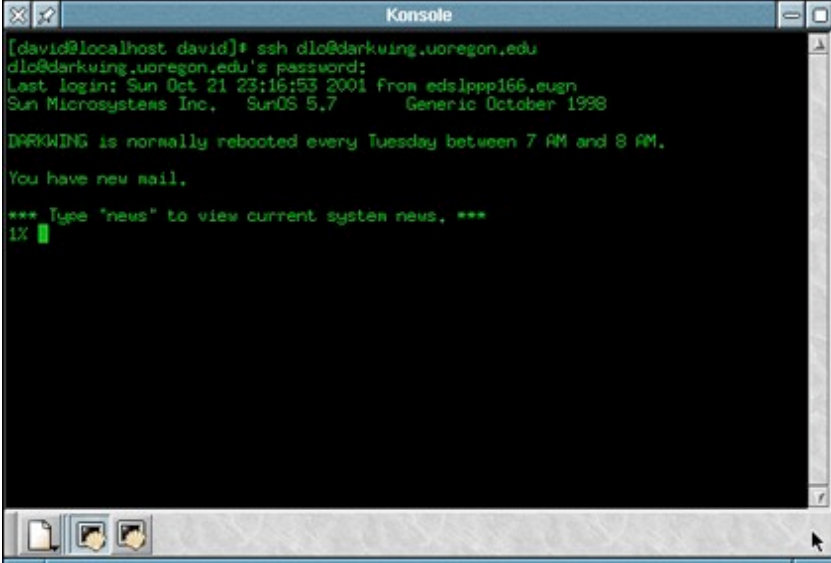
- Viene chiamato shell un particolare interprete per i comandi che l'utente inserisce in un terminale a caratteri
 - sh (shell)
 - Csh (C shell)
 - Tcsh (TC shell)
 - Ksh (Korne shell)
 - Bash (Bourne Again Shell)

Shell

- Compiti di una shell
 - Interpretare ed eseguire i comandi impartiti dall'utente
 - Eseguire procedure consistenti in un particolare flusso (e.g. una sequenza) di comandi

La shell

- Una shell indica all'utente la propria disponibilità ad accettare comandi dall'utente visualizzando un messaggio (prompt)
- Il prompt è personalizzabile e può includere informazioni riguardo alla directory corrente, alla data e ora corrente, al sistema cui si è collegati, ...



```
[david@localhost david]$ ssh dlo@darkwing.uoregon.edu
dlo@darkwing.uoregon.edu's password:
Last login: Sun Oct 21 23:16:53 2001 from eds1ppp166.eugn
Sun Microsystems Inc. SunOS 5.7 Generic October 1998

DARKWING is normally rebooted every Tuesday between 7 AM and 8 AM.

You have new mail.

*** Type "news" to view current system news. ***
IX █
```

GUI in Linux

- E' possibile installare un ambiente grafico (XFree), su cui possono essere eseguiti differenti window manager
 - mwm, twm, ...
 - Gnome (GNU Object Model Environment)
 - KDE (K Desktop Environment)
- Ognuno interpreta una serie di operazioni effettuate dall'utente per poi eseguire delle corrispondenti applicazioni del s.o.

Attenzione!

- In questa lezione ci focalizzeremo soprattutto sulle operazioni che si possono effettuare utilizzando un terminale a caratteri

Il desktop: una panoramica

- Lo schermo dell'elaboratore viene gestito utilizzando la metafora di una scrivania (desktop)
- Sul desktop è possibile memorizzare oggetti, rappresentati da icone
- Alcune icone identificano oggetti particolari e sono inserite dal s.o. (vedi il cestino)
- Altre icone identificano applicazioni o documenti e sono gestite dall'utente
- Il desktop è una parte del file system (vedi oltre)

Gestione dell'interfaccia

- Ogni applicazione è generalmente associata a una finestra (e.g., il terminale)
- Sul desktop si possono sovrapporre più finestre (i.e. più applicazioni)
- Quando si clicca su una finestra questa viene portata in primo piano (**focus**) e l'applicazione relativa può accettare input
- Una finestra può essere in stato normale, massimizzato o minimizzato

Gestione dell'interfaccia

- Le operazioni fondamentali di un'applicazione vengono organizzate in una struttura gerarchica detta menù, inserita sotto il titolo della finestra
- Le operazioni tipiche di un particolare contesto (la parte di un'applicazione) sono raggruppate in un menù contestuale, accessibile facendo click con il pulsante destro

Gestione degli utenti

- Un utente è un'**entità** (fisica oppure immateriale) in grado di **utilizzare** alcune **risorse** di un elaboratore. Ad esempio
 - persone;
 - gruppi di persone;
 - servizi gestiti da particolari programmi.
- L'accesso a queste risorse può avvenire
 - a livello **locale**, accedendo fisicamente all'elaboratore;
 - a livello **remoto**, tramite una rete telematica.

Multiutenza

- Rappresenta la possibilità per più utenti di accedere alle risorse di un medesimo elaboratore
 - simultaneamente
 - in tempi differenti

Multiutenza in Linux

- Linux supporta entrambi i tipi di multiutenza: è possibile definire
 - quali utenti possono accedere al sistema
 - in che tempi possono accedervi
 - quali risorse possono utilizzare
- utenti (anche fisici) differenti possono accedere concorrentemente allo stesso sistema

Multiutenza in Linux (2)

- Negli elaboratori usati come server gli utenti accedono in remoto, tipicamente a risorse limitate
 - spazio su disco
 - posta, ...
- Negli elaboratori usati come client gli utenti accedono in locale, tipicamente all'intero sistema
 - noi ci occuperemo di questo ambito

Login

- È la procedura che precede l'utilizzo, da parte di un utente, di una generica risorsa in un elaboratore.

L'utente

- dichiara la propria identità, specificando un nome (login name, o login) che lo identifica univocamente rispetto agli altri utenti
- Certifica (eventualmente) la propria identità, specificando una parola chiave (o password) di accesso che è ignota agli altri utenti
- Se login e password sono validi, l'elaboratore fornisce la risorsa all'utente.

Login in Linux

- Tutti i sistemi linux prevedono la possibilità di effettuare un login in modalità testuale
- Esistono particolari programmi deputati a gestire le comunicazioni testuali a/da un sistema sia trasmettendo le informazioni in chiaro (**telnet**) che cifrandole (**ssh**)

La finestra di login

- Sostituisce la procedura di login testuale per l'accesso fisico al sistema (se è installato un wm. e il sistema è opportunamente configurato)
- Compare ogni volta che il sistema è pronto ad accettare un nuovo utente
- L'utente inserisce login e password negli appositi campi (al posto dei caratteri della password vengono visualizzati degli asterischi)



Utenti in Linux

- Il comando `who` elenca gli utenti attualmente connessi al sistema
- `who am i` (o `whoami`) ritorna l'utente che sta eseguendo la shell
- Il comando `w` indica, per ogni utente collegato, informazioni aggiuntive come il tempo di idle e le applicazioni che sta eseguendo

```
[16:01]cazzola@thor:~>w
```

USER	TTY	FROM	LOGIN@	IDLE	JCPU	PCPU	WHAT
cazzola	pts/0	:0	Sun 2pm	1:38m	3.00s	3.00s	-csh
cazzola	pts/3	belgarion	2:55pm	0.00s	0.30s	0.03s	w

```
[16:01]cazzola@thor:~>who
```

```
cazzola vc/1 Sep 14 13:59
```

```
cazzola pts/3 Sep 17 14:55 (belgarion)
```

```
[16:01]cazzola@thor:~>who am i
```

```
thor.disi.unige.it!cazzola pts/3 Sep 17 14:55 (belgarion)
```

Logout

- Complementare al login, è la procedura con cui un utente termina di utilizzare delle risorse che si rendono in questo modo disponibili ad altri utenti
- Viene effettuata selezionando la voce “logout” nel menù
- `exit` o `Ctrl+D` permettono di chiudere una shell.

Utenti in Linux

- Ogni utente appartiene a uno o più gruppi di utenti
- Un gruppo è tipicamente associato a una risorsa cui gli appartenenti hanno accesso (e.g. la possibilità di utilizzare un masterizzatore DVD)

Gestione dei processi

- Un processo è l'immagine di un programma in esecuzione
- Tipicamente un sistema operativo esegue apparentemente più processi in modo simultaneo, eseguendo ciclicamente una porzione di ognuno di essi
- Ogni processo viene identificato in modo univoco da un numero di processo (PID), che gli viene assegnato (dal sistema operativo) all'atto della sua esecuzione

Processi in Linux

- Un programma viene eseguito scrivendo il nome dell'applicazione e premendo invio:
 - in foreground (il nome non è seguito da &): la shell viene bloccata fino a quando il processo non termina
 - in background (il nome è seguito da &): mentre il processo viene eseguito la shell continua ad essere disponibile all'utente
- Passaggio da foreground a background
 - Si ottiene premendo Ctrl+Z + bg
- Passaggio da background a foreground
 - Si ottiene eseguendo dalla shell il comando fg

Monitoraggio dei processi

- Viene effettuato dal comando ps

```
laren:~$ ps -fu malchiod
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
malchiod	30342	10969	0	10:02	pts/8	00:00:21	pine
malchiod	11983	1	0	10:28	?	00:00:00	xterm
malchiod	11985	11983	0	10:28	pts/37	00:00:00	bash
malchiod	15130	11985	0	13:55	pts/37	00:00:00	ps -fu malchiod

Terminazione dei processi

- Un processo può essere terminato:
 - premendo Ctrl+C (se il processo è in foreground)
 - usando il comando kill seguito dal PID

```
laren:~$ kill -9 <pid>
```

forza la terminazione di un processo

- Esempi di Segnale

```
laren:~$ kill -l
```

1) SIGUP	2) SIGINT	3) SIGQUIT	4) SIGILL
5) SIGTRAP	6) SIGIOT	7) SIGBUS	8) SIGFPE
9) SIGKILL	10) SIGUSR1	11) SIGSEGV	12) SIGUSR2
13) SIGPIPE	14) SIGALRM	15) SIGTERM	17) SIGCHLD
18) SIGCONT	19) SIGSTOP	20) SIGSTP	21) SIGTTIN
22) SIGTTOU	23) SIGURG	24) SIGXCPU	25) SIGXFSZ
26) SIGVTALRM	27) SIGPROF	28) SIGWINCH	29) SIGIO
30) SIGPWR			

Pipe

- Un pipe è un canale che permette a due processi di comunicare, in modo che l'output del primo venga utilizzato come input dal secondo
- Viene eseguito scrivendo i due comandi separandoli con un carattere di barra verticale (|). Esempio: `ps | more`

Gestione della memoria

- Linux alloca direttamente la memoria alle varie applicazioni, in modo trasparente all'utente
- È possibile verificare quanta memoria è disponibile e quanta è allocata ad ogni processo eseguendo il comando `top`

```
laren:~$ top
5:16pm up 5:22, 13 users, load average: 0.62, 0.39, 0.31
102 processes: 97 sleeping, 2 running, 0 zombie, 3 stopped
CPU states: 61.2% user, 10.0% system, 0.0% nice, 28.6% idle
Mem: 127908K av, 123816K used, 4092K free, 53804K shrd, 3776K buff
Swap: 261804K av, 17856K used, 243948K free 22780K
      cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	LIB	%CPU	%MEM	TIME
28498	malchiod	3	0	1084	1084	856	R	0	1.1	0.8	0:00 top
3	root	0	0	0	0	0	SW	0	0.1	0.0	0:22
	kupdate										
1	root	0	0	116	72	52	S	0	0.0	0.0	0:05 init

Gestione del file system

- Il file system è la componente del sistema operativo preposta alla gestione delle informazioni memorizzate permanentemente, che risiedono tipicamente su disco
- Le componenti di un file system sono
 - Il **file**
 - La **directory**

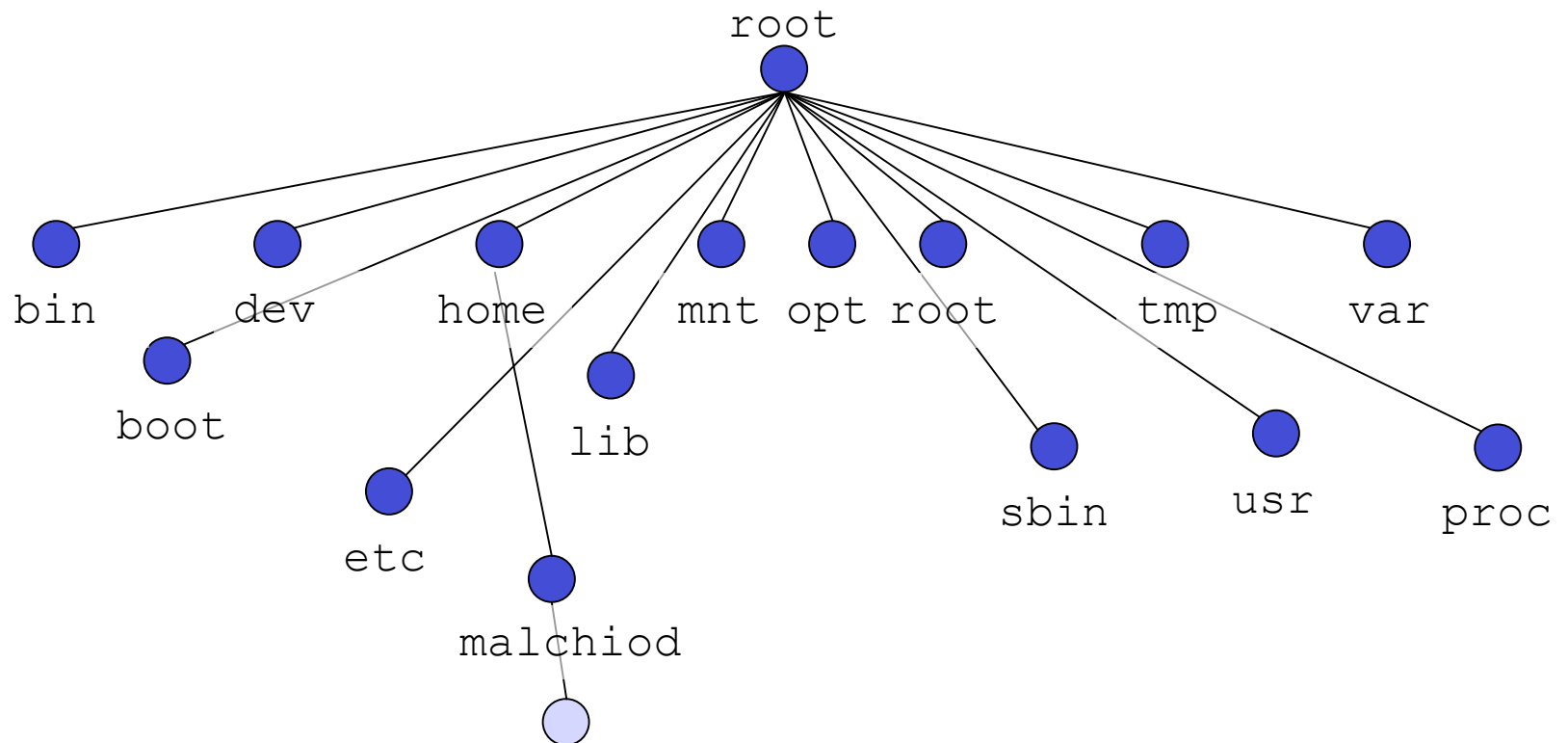
File e directory

- Un file memorizza permanentemente una serie di informazioni aventi unità logica
 - Una applicazione (elaboratore testi, visualizzatore di clip multimediali, ...)
 - Dei dati (una relazione, un video musicale, ...)
- Una directory (o cartella, o folder) è un contenitore di oggetti, che possono essere file o altre directory

Gerarchia del file system

- Un file system ha quindi una struttura gerarchica (in particolare, ad albero) nella quale
 - I file sono le estremità (foglie)
 - Le directory sono i nodi rimanenti
 - Esiste un'unica directory (radice) cui tutta la struttura fa capo

Gerarchia del file system



Programma Laboratorio Programmazione

File system

- `/bin`: eseguibili
- `/boot`: file per l'esecuzione del s.o.
- `/dev`: device driver
- `/etc`: file di configurazione
- `/home`: home directory degli utenti
- `/lib`: librerie di sistema
- `/mnt`: mount point per file system esterni
- `/opt`: packages e add-on
- `/root`: directory dell'amministratore
- `/sbin`: eseguibili di sistema
- `/tmp`: file temporanei
- `/usr`: applicazioni
- `/var`: file variabili (ad es. log, aree di spool)
- `/proc`: processi in esecuzione ed informazioni sul sistema

File system strutturati

- La possibilità di inserire directory nelle directory fornisce l'opportunità di creare dei file system strutturati, in cui le informazioni sono memorizzate in modo ordinato
- In file system ben strutturati risulta più facile localizzare ed accedere alle informazioni in modo veloce

Pathname

- Ogni file all'interno del file system è individuato in base alla sua posizione nell'albero, cioè al cammino (**pathname** o **path**) che si deve percorrere per raggiungerlo partendo dalla radice
- se voglio indicare il programma del corso di informatica devo specificare il cammino:

```
root+home+malchiod+Programma Informatica  
Generale
```

File system in Linux

- La radice della struttura gerarchica è la **root directory**, indicata dal carattere slash (/)
- I nodi di un pathname sono separati dal carattere slash (/)
- Ogni disco dell'elaboratore è individuato da un **mount point** nella struttura gerarchica

Visualizzazione di directory

- Il comando `pwd` visualizza il pathname della directory in cui si è attualmente posizionati (directory corrente)

```
laren:~$ pwd  
/home/malchiod
```

- Quando si esegue un terminale, la directory corrente è automaticamente impostata alla propria home directory

Visualizzazione di directory

- Il comando `ls` permette di visualizzare il contenuto della directory corrente
- Se dopo `ls` si specifica il nome di una directory, vengono visualizzati i contenuti di quest'ultima
 - `.` indica la directory corrente
 - `..` indica la directory che contiene la directory corrente

Path assoluti e relativi

- Il nome di una directory può essere indicato specificando
 - Un **pathname assoluto**, specificando tutto il percorso a partire dalla root directory
 - Un **pathname relativo**, specificando uno dei possibili percorsi a partire dalla directory corrente

Spostamento tra directory

- Il comando `cd` permette di cambiare la directory corrente
 - Se non vengono specificati argomenti, ci si posiziona nella home directory
 - Se viene specificato come argomento il pathname (assoluto o relativo) di una directory, ci si posiziona in quest'ultima

Esercizio

- Esplorate il file system

Operazioni con le directory

- Creazione: tramite il comando `mkdir`
- Eliminazione: tramite il comando `rmdir` (eseguito solo se la directory è vuota)
- Eliminazione di una directory e del suo contenuto: tramite il comando `rm -rf`
- Altre operazioni (spostamento, variazione del nome, ...): tramite gli stessi comandi visti per i file (vedi oltre)

Wildcard

- Nello specificare i pathname di file o directory, è possibile indicare un'espressione contenente dei **caratteri jolly** (o **wildcard**)
 - * indica una qualunque sequenza di caratteri
 - ? indica un qualunque carattere
 - [] indica un qualunque carattere appartenente alla sequenza indicata tra le parentesi

Esempi

	Diventa
m^*	m, ma, mb, ..., maa, mab, ...
$m?lo$	malo, mblo, mclo, mdlo, ...
$m[aeiou]lo$	malo, melo, milo, molo, mulo

File e directory nascosti

- Quando un file o una directory ha un nome che inizia con il carattere di punto (.), non viene elencato quando si visualizza la directory cui appartiene (a meno di non usare l'opzione `-a` del comando `ls`)

Operazioni su file

- Creazione: tramite il comando `touch`
- Apertura
 - di file eseguibili: scrivendone il nome
 - di file di dati: tramite le relative applicazioni
- Visualizzazione dei contenuti: tramite i comandi `cat`, `more` e `less`
- Spostamento e/o modifica del nome: tramite il comando `mv` (vale anche per le directory)
- Cancellazione: tramite il comando `rm`

Estensione di un file

- Linux è in grado di capire il formato dei file leggendo parte del loro contenuto (intestazione). Il comando `file` effettua tale riconoscimento
- È comunque possibile aggiungere delle estensioni significative al nome di un file

Esempi

```
laren:~$ file tesi
tesi: directory
laren:~$ file test.ppt
test.ppt: Microsoft Office Document
laren:~$ file train500.pat
train500.pat: ASCII text
laren:~$ file tq.jpg
tq.jpg: JPEG image data, JFIF standard
laren:~$ file tutto.zip
tutto.zip: Zip archive data, at least v2.0 to extract
laren:~$ file report
report: PDF document, version 1.2
```

Link

- Un collegamento (link) è un particolare tipo di file la cui funzione è esclusivamente quella di “puntare” a un altro file o directory
- L’apertura di un collegamento è equivalente all’apertura del file o directory cui fa riferimento
- Linux permette la creazione di link **soft** e **hard**, usando il comando `ln`

Soft link

- I link soft (o symbolic) sono simili ai collegamenti in Windows
- Teoricamente è possibile creare dei “riferimenti circolari”
- Spostamento, rimozione e modifica del nome di un link soft non hanno effetto sull’oggetto puntato
- L’eliminazione di un oggetto puntato rende il corrispondente link pendente

Esempio

```
laren:~$ ls
file
laren:~$ cat file
prova di collegamento
laren:~$ ln -s file link
laren:~$ ls -ls
total 4
  4 -rw-r--r-- 1 malchiod users  22 Oct 12 18:34 file
  0 lrwxrwxrwx 1 malchiod users   4 Oct 12 18:34 link -> file
laren:~$ cat link
prova di collegamento
laren:~$ mkdir directory
laren:~$ mv file directory/
laren:~$ ls -ls
total 4
  4 drwxr-xr-x 2 malchiod users 4096 Oct 12 18:35 directory
  0 lrwxrwxrwx 1 malchiod users   4 Oct 12 18:34 link -> file
laren:~$ cat link
cat: link: No such file or directory
```

Hard link

- Un hard link rappresenta un vero e proprio duplicato dell'indice di directory che punta a un file (o a una directory)
- Lo spostamento o addirittura la cancellazione di un oggetto puntato non modificano il comportamento di un hard link

Esempio

```
laren:~$ ls -ls
total 8
  4 drwxr-xr-x 2 malchiod users    4096 Oct 12 18:46 directory
  4 -rw-r--r-- 1 malchiod users    22 Oct 12 18:34 file
laren:~$ ln file link
laren:~$ ls -ls
total 12
  4 drwxr-xr-x 2 malchiod users    4096 Oct 12 18:46 directory
  4 -rw-r--r-- 2 malchiod users    22 Oct 12 18:34 file
  4 -rw-r--r-- 2 malchiod users    22 Oct 12 18:34 link
laren:~$ mv file directory/
laren:~$ cat link
prova di collegamento
laren:~$ rm directory/file
laren:~$ cat link
prova di collegamento
```

Esempio di uso dei link

- La directory `/usr/share/zoneinfo` contiene informazioni relativi ai vari fusi orari del mondo
- Il file `/etc/localtime` è un symbolic link a un file all'interno di `/usr/share/zoneinfo` relativo al fuso orario in cui l'elaboratore è installato
- Le applicazioni fanno riferimento a `/etc/localtime`

Home directory

- Ogni utente abilitato ad accedere al sistema ha una propria home directory contenuta in `/home` e avente come nome il proprio nome utente
- Nell'indicazione di un pathname, la home directory dell'utente attualmente connesso è abbreviabile usando il carattere tilde (`~`)

Permessi su file

- gli utenti possono specificare i seguenti **permessi di accesso**
 - **Read** (R) indica se possono essere letti i contenuti di un file o di una cartella
 - **Write** (W) indica se è possibile modificare il contenuto di un file o di una cartella
 - **Execute** (X) indica se è possibile eseguire un file o posizionarsi all'interno di una cartella

Permessi sui file (2)

- i permessi sui file sono specificabili su tre livelli
 - Relativamente all'utente che li crea
 - Relativamente agli utenti facenti parte dello stesso gruppo dell'utente che li crea
 - Relativamente ai rimanenti utenti
- Sono specificati da un terzetto di caratteri

```
laren:~$ ls -ls
      total 8
  4 drwxr-xr-x 2 malchiod users    4096 Oct 12 18:46 directory
  4 -rw-r--r-- 1 malchiod users     22 Oct 12 18:34 file
```

Esercizio

- Trovare dei file o delle directory che abbiano dei permessi che non consentono una qualche forma di accesso (e.g. in lettura o in scrittura)
- Verificare che non sia possibile effettuare l'operazione corrispondente

Documentazione

- Il comando `man` permette di visualizzare una descrizione delle funzionalità dei vari comandi utilizzabili all'interno di un terminale
- Esempi
 - `man ls`
 - `man cd`
 - `man man`

Esercizio

- Consultare la documentazione dei comandi finora visti
- In molti casi la documentazione descriverà l'uso di particolari opzioni che modificano il comportamento del comando: esercitarsi nell'uso di queste opzioni

Modifica dei permessi

- Leggere la documentazione del comando `chmod` e verificare come utilizzarlo per modificare i permessi associati a un proprio file
- Esercitarsi a modificare i permessi, verificando di volta in volta che la modifica abbia avuto luogo

Ricerche nel file system

- `find` esegue una ricerca ricorsiva a partire da una directory specificata. È una utility molto potente (è possibile specificare nomi parziali, indicare le date entro cui effettuare la ricerca, ...). Risulta però complessa da utilizzare
- `locate/slocate` esegue una ricerca su un archivio aggiornato periodicamente

Redirezione da/a file

- È possibile utilizzare dei file per
 - Memorizzare l'output (stdout) di un comando, indicando dopo di questo il carattere di maggiore (>) seguito dal nome del file
 - Due caratteri di maggiore (>>) indicano che l'output del comando deve essere aggiunto al file, che non viene sovrascritto
 - Memorizzare l'input (stdin) di un comando, indicando dopo di questo il carattere di minore (<) seguito dal nome del file
 - Anche stderr (file su cui vengono inviati gli errori) può essere reindirizzato specificando l'ampersand (>&, >>&)

Esempio

```
laren:~$ ls -lh
total 8.0k
drwxr-xr-x  2 malchiod users  4.0k Oct 12 18:47 directory
-rw-r--r--  1 malchiod users   22 Oct 12 18:34 link
laren:~$ ls -lh > output
laren:~$ ls -lh
total 12k
drwxr-xr-x  2 malchiod users  4.0k Oct 12 18:47 directory
-rw-r--r--  1 malchiod users   22 Oct 12 18:34 link
-rw-r--r--  1 malchiod users  198 Oct 13 11:47 output
laren:~$ cat output
total 8.0k
drwxr-xr-x  2 malchiod users  4.0k Oct 12 18:47 directory
-rw-r--r--  1 malchiod users   22 Oct 12 18:34 link
-rw-r--r--  1 malchiod users    0 Oct 13 11:47 output
laren:~$
```

Esercizio

- Creare nella home directory una cartella
- Creare all'interno di quest'ultima dei documenti
- Creare nella cartella una serie di altre cartelle ed esercitarsi a creare, spostare e copiare file al loro interno
- Esercitarsi a creare e spostare collegamenti
- Eseguire delle ricerche