



POLITECNICO DI MILANO

Dipartimento di Elettronica e Informazione

Corso di Linguaggi Formali - Esercitazione

Introduzione rapida all'uso di Linux

Riduzione e adattamento da dispense di Carlo Brandolese

A cura di:

Daniele Paolo Scarpazza,

daniele.scarpazza@elet.polimi.it,

www.elet.polimi.it/~scarpazz

Che cosa è Unix?

- **“Unix non esiste!”**
- “unix” è una **famiglia** di sistemi operativi (AT&T Unix System V, HP-UX, Tru64, Solaris, IBM AIX, Linux, BSD, Mac OS X,)
- “Unix” è un marchio commerciale (AT&T 1970, oggi di The Open Group)

Che cosa è Linux?

- Linux è "uno dei tanti" unix
- Linux è un sistema operativo:
 - evoluto (multiutente, multitasking preemptive, a memoria protetta),
 - free (nel senso di libero),
 - stabile e affidabile,
 - in continua evoluzione,
 - con un ricco supporto per l'hardware
- Originariamente derivato da Minix a cura di Linus Torvalds

Che cosa è una distribuzione

- Linux è costituito da un kernel (disponibile in molte versioni e varianti) su cui girano molti strumenti, componenti e applicazioni
- Una "distribuzione" è un insieme di queste componenti, scelto e assemblato per voi, corredato da un programma di installazione che rende tale operazione automatico e facile
- Esistono molte distribuzioni (Red Hat, Mandrake, Slackware, Debian, SuSE, Turbolinux, ...)

Che cosa è Knoppix

- Knoppix è una distribuzione di Linux “live”, cioè:
 - non richiede di essere installata sull’hard disk (nessun ripartizionamento o riformattazione);
 - sta su un solo CD avviabile;
 - è preconfigurata per funzionare subito e bene sul massimo numero di macchine, e in rete;
 - contiene una gran quantità di software preinstallato e pronto da usare (anche OpenOffice);
 - (fatta da Klaus Knopper)

Utenti

- Nei sistemi unix-like, più utenti possono accedere concorrentemente ad un sistema; il registro degli utenti è conservato nel file `/etc/passwd`

```
root:aG4dhH1z:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:GL16gtWn:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
...
```

- Il sistema riconosce l'esistenza di **utenti** e **gruppi**. I concetti di utente e gruppo permettono di implementare la sicurezza:
 - autenticazione, riservatezza, protezione, log
 - a livello di risorsa: ciascun utente ha oppure non ha i privilegi per compiere una certa operazione su una certa risorsa e le sue azioni possono essere registrate;
- Un utente può appartenere ad uno o più gruppi
- Il superutente, denominato `root`, ha tutti i privilegi possibili.

Login

- Il processo di autenticazione si chiama **login**
- Una tipica sessione di login:

```
Red Hat Linux release 9 (Shrike)
Kernel 2.4.20 on an i686
```

```
pcsenonsrv login: scarpazza
password:
```

```
Last login: Thu Sep 25 14:56:06 2003 from cartesio.cefriel.it
[scarpazza@pcsenonsrv scarpazza]$
```

- La password non viene visualizzata (per sicurezza)
- La stringa **[scarpazza@pcsenonsrv scarpazza]\$**, detta "prompt dei comandi" indica:
 - che c'è una shell pronta a ricevere i vostri comandi
 - chi siete e dove vi trovate (macchina e directory)
- L'autenticazione non è necessaria su una distribuzione "live" come Knoppix

Comandi associati agli utenti

- **passwd**
Consente ad un utente di modificare la propria password
- **passwd <utente>**
Consente a `root` di modificare la password dell'utente <utente>
- **su [-]**
Chiede di cambiare utente e diventare `root`. Viene richiesta la password di `root`.
- **su [-] [<utente>]**
Chiede di cambiare utente, e diventare <utente>
 - 1 Indica che deve essere anche resa attiva la configurazione del nuovo utente.
- **w**
 - Riporta la lista degli utenti connessi alla macchina e indica: username, terminale, data e ora di inizio connessione, macchina remota di provenienza della connessione; esempio:

```
[scarpazza@pcsenonsrv scarpazza]$ w
 22:20:30 up 12 min,  4 users,  load average: 0.14, 0.34, 0.32
scarpazz  console    Jul  4 09:20
scarpazz  pts/2        Jul  4 09:21
pizzamig  pts/1        Jul  4 19:18  (yoga.cefriel.it)
[scarpazza@pcsenonsrv scarpazza]$
```

Comandi associati agli utenti

- Comandi utili per la gestione degli utenti:

useradd

aggiunge un nuovo utente

userdel

elimina un utente

usermod

modifica i dettagli associati ad un utente (nome, commento, home, scadenza, gruppi, shell, uid, gid)

groupadd

aggiunge un nuovo gruppo

groupdel

elimina un gruppo esistente

last

mostra le ultime connessioni effettuate sulla macchina

La shell

- La shell è l'**interprete dei comandi** cioè l'interfaccia testuale che riconosce i vostri comandi e li esegue;
- La shell può eseguire i comandi in modo interattivo (digitati "a mano") oppure in modo batch (i comandi sono memorizzati in un file di testo detto script file);
- Le shell sono come i partiti: ce ne sono tante, sono tutte più o meno equivalenti; anche se i seguaci non concorderebbero;

Sintassi generica dei comandi

- Un comando ha la seguente **sintassi generale**:

```
cmd [-o|--option] [file[...]] [;]
```

I vari campi indicano:

cmd	Nome del comando
-o	Opzioni da una sola lettera + eventuali argomenti
file[...]	Lista di nomi di file, separata da spazi
--option	Opzioni lunghe (il nome è costituito da più di una lettera)
;	separa i comandi di una lista

Nomi di file

- I comandi spesso operano su un **elenco di file**
 - Possono essere indicati **esplicitamente**
 - Possono essere indicati mediante **globbing**
- Il globbing si basa su alcuni **caratteri speciali**
 - `~/` Home directory dell'utente corrente
 - `~user` Home directory dell'utente *user*
 - `*` Sequenza di zero o più caratteri qualsiasi
 - `?` Singolo carattere qualsiasi
 - `[set]` Singolo carattere in *set*
 - `[s-e]` Singolo carattere compreso tra *s* e *e*
- Tali caratteri sono combinati a formare un **pattern**
- Il pattern è **espanso** nei nomi ad esso sovrapponibili

Accesso alla Documentazione

- Linux fornisce due diversi sistemi di help relativi ai comandi: `man` e `info`
- `man [n] cmd`
 - `n` Sezione dei manuali cui fare riferimento
 - `cmd` Comando di cui mostrare il manuale
- `info [cmd [section]]`
 - `cmd` Comando di cui mostrare il manuale
 - `section` Sezione nel manuale del comando
- storicamente, `info` è arrivato dopo `man` e presenta funzionalità più avanzate
- comparare il risultato di volta in volta;

Documentazione in linea

```
[scarpazza@pcsenonsrv scarpazza]$ man useradd
```

```
USERADD(8)
```

```
USERADD(8)
```

NAME

useradd - Create a new user or update default new user information

SYNOPSIS

```
useradd [-c comment] [-d home_dir]
          [-e expire_date] [-f inactive_time]
          [-g initial_group] [-G group[,...]]
          [-m [-k skeleton_dir] | -M] [-p passwd]
          [-s shell] [-u uid [ -o]] [-n] [-r] login
```

```
useradd -D [-g default_group] [-b default_home]
          [-f default_inactive] [-e default_expire_date]
          [-s default_shell]
```

DESCRIPTION

Creating New Users

When invoked without the **-D** option, the **useradd** command creates a new user account using the values specified on the command line and the default value from the system. The new user account will be entered into the system files as needed, the home directory will be created, and initial files copied, depending on the command line options. The options which apply to the **useradd** command are:

-c comment

The new user's password file comment field.

Documentazione in linea

```
[scarpazza@pcsenonsrv scarpazza]$ info latex
File: latex.info, Node: Top, Next: Overview, Prev: (dir), Up: (dir)

LaTeX2e is a document preparation system implemented as a macro package for
Donald E. Knuth's TeX typesetting program.

LaTeX was originally conceived by Leslie Lamport.

This is edition 1.6 of the LaTeX2e documentation.

* Menu:

* Overview::          What is LaTeX?
* Commands::         Commands within a LaTeX document.
* Parameters::      The command line.
* Command Index::   An alphabetical list of LaTeX commands.
* Concept Index::   An alphabetical list of concepts.
```

```
--zz-Info: (latex.info.gz)Top, 18 lines --All-- Subfile: latex.info-1.gz-----
```

Il filesystem

- Il **File System** fornisce all'utente una visione logica di tutti i dispositivi: hard disk, CD, floppy, ramdisk, mouse, scanner, porte varie, ...
- La struttura di un file system si basa su due concetti fondamentali: file e directory
- Sistemi operativi diversi utilizzano diverse implementazioni di file system

File

- Ad un file sono associati gli attributi:
 - Nome: E' un nome simbolico con cui ci si riferisce ad esso, può contenere una estensione che indica il tipo
 - Locazione: E' un puntatore alla posizione fisica sul dispositivo
 - Dimensione: Dimensione dei dati
 - Date: Indicano il momento della creazione, ultima modifica o ultimo accesso al file
 - Proprietari: L'utente e il gruppo che possiedono il file
 - Diritti: Indica quali operazioni possono essere eseguite su quel file

Comandi

- **ls [-alRl] [<percorso>]**

- Mostra l'elenco dei file nella directory *path* o in quella corrente

- **-a** Mostra anche i file "invisibili", il cui nome inizia con un . (punto).
- **-l** Mostra le informazioni associate a ciascun file
- **-R** Mostra il contenuto della directory *path* (o quella corrente) e tutte le sottodirectory.
- **-1** Mostra un solo file per ogni linea

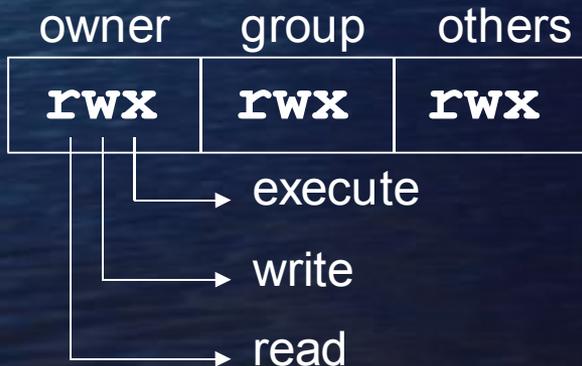
```
[scarpazza@pcsenonsrv scarpazza]$ ls
pippo                pluto                paperino
minnie

[scarpazza@pcsenonsrv scarpazza]$ ls -a
.                    ..                    pippo
pluto                paperino              minnie

[scarpazza@pcsenonsrv scarpazza]$ ls -la
drwxr-xr-x   2 root    root      512 Jun 21 13:56 .
drwxrwxrwx  15 root    root     1024 Jul 11 12:58 ..
lrwxr--r--   1 root    root    13520 Jun 10  1998 pippo -> /etc/pippo
drw-rw-rw-   1 root    root     80115 Jun 21 13:48 pluto
-rwxr--r--   1 root    root    64563 Jun 21 13:46 paperino
-rwxr--r--   1 root    root     8891 Jun 21 13:46 minnie
```

Protezione

- I **diritti** sono formati da 3 gruppi di 3 bit:
 - In ciascun gruppo:
 - il primo bit (r) indica se il file può essere letto
 - il secondo bit (w) indica se il file può essere scritto
 - il terzo bit (x) indica se il file può essere eseguito
 - i tre gruppi:
 - il primo gruppo (u) indica i diritti dell'utente proprietario
 - il secondo (g) i diritti degli appartenenti al gruppo proprietario
 - il terzo (o) indica i diritti di tutti gli altri utenti



owner	group	others
rwX	rwX	rwX
111	101	101
7	5	5

Comandi

- **chmod mode file ...**
 - Cambia i diritti di accesso ai file specificati
 - Notazione ottale: `uuu ggg ooo`
 - `uuu`: diritti del proprietario
 - `ggg`: diritti del gruppo
 - `ooo`: diritti degli altri
 - Notazione simbolica: `{ugoa} {+=-} {rwx}`
 - `{ugoa}`: Gruppo di permessi proprietario, gruppo, altri o tutti
 - `{+=-}`: Operazione aggiunge, fissa o rimuove
 - `{rwx}`: Tipo di permesso: lettura scrittura o esecuzione
- **chown [-R] owner[:group] file ...**
 - Modifica il proprietario di un file
 - `-R`: Ricorsivo. Modifica anche i file nelle sottodirectory
 - `owner`: Specifica la username del nuovo proprietario
 - `group`: Specifica il gruppo del nuovo proprietario
- **chgrp [-R] group file ...**
 - Modifica il gruppo di un file
 - `-R`: Ricorsivo. Modifica anche i file nelle sottodirectory
 - `group`: Specifica il gruppo del nuovo proprietario

Comandi

- **pwd**
 - Mostra il path assoluto della directory corrente
- **cd [path]**
 - Cambia la directory corrente
 - Con un argomento, rende *path* la directory corrente
 - Senza argomenti, rende corrente la home directory dell'utente
 - Esistono alcuni nomi speciali di directory:
 - `.` Directory corrente
 - `..` Directory padre della directory corrente
 - `/` Root directory
 - `~/` Home directory dell'utente
 - `~user` Home directory dell'utente `user`.
 - Diversi livelli di directory sono separati dal carattere / (slash)
 - I pathname che iniziano con / sono detti assoluti mentre in tutti gli altri casi si parla di pathname relativi

Comandi

- **cp** [-r] [-p] [-i] [-f] *src dest*

- Copia uno o più file

- -r Ricorsiva. Copia una directory e tutto il suo contenuto
- -p Conserva gli attributi del file originale (proprietario, gruppo, diritti di accesso e data di creazione e accesso)
- -i Interattivo. Chiede conferma prima di sovrascrivere un file
- -f Forza la copia di uno o più file anche se la destinazione esiste

- Sorgente e destinazione possono essere:

- *file1 file2* Copia *file1* in *file2*
- *file... dest* Copia tutti i file nella directory *dest*
- *dir... dest* Copia le directory nella directory *dest* (solo con l'opzione -r)

- **mkdir** [-p] *dir*

- Crea una nuova directory. La directory padre di *dir* deve esistere

- -p Crea la nuova directory *dir* e tutte le directory intermedie che sono necessarie

Comandi

- **mv [-i|-f] src dest**

- Sposta uno o più file. L'utente deve avere i diritti di scrittura sui file sorgenti, in quanto devono essere cancellati

- **-i** Interattivo. Chiede conferma prima di sovrascrivere un file
- **-f** Forza lo spostamento

- Sorgente e destinazione possono essere:

- *file1 file2* Sposta il *file1* in *file2* (anche directory)
- *file... dest* Sposta i file nella directory *dest*
- *dir... dest* Sposta le directory nella directory *dest*

- **touch [-a|-m] [-r ref] [file...]**

- Modifica le date di accesso o di modifica di uno o più file

- Se un file non esiste viene creato

- **-a** Cambia la data di accesso usando la data corrente
- **-m** Cambia la data di modifica usando la data corrente

- **-r ref** Usa la data del file *ref* invece della data corrente

Comandi

- **rm [-i|-f] [-r dir...] [file...]**
 - Rimuove uno o più file
 - **-i** Interattivo. Chiede conferma prima di eliminare un file
 - **-f** Forza la rimozione
 - **-r dir...** Elimina le directory ed il loro contenuto
- **rmdir [-p] dir...**
 - Rimuove una o più directory. Le directory devono essere vuote o, al più, contenere solo altre directory
 - **-p** Rimuove la directory *dir* e la sua directory padre

Comandi

- **du** **[-k]** **[-s]** **[-a]** **dir...**
 - Mostra lo stato di utilizzo dei dischi
 - **-k** Mostra le informazioni in kilobytes
 - **-s** Mostra solo il totale
 - **-a** Mostra le informazioni per tutti i file
- **df** **[-b]** **[-l]** **[dir|device]**
 - Mostra lo spazio disponibile sui vari file system
 - **-b** Lo spazio è espresso in kilobytes
 - **-l** Mostra i dati relativi ai soli file system locali

```
[scarpazza@pcsenonsrv scarpazza]$ df -b -l
Filesystem          kbytes    used   avail capacity  Mounted on
/proc                0         0       0         0%    /proc
/dev/hda6           1192748  187923  945188    17%    /
/dev/hda7           1986439  757868 1168978    40%    /usr
/dev/fd0              1440      0      1440    100%    /mnt/floppy
swap                 298376    592   297784     1%    swap
```

Link

- Soft link o symbolic link

- È un riferimento ad un altro file o ad un'altra directory
- Rimuovendo un link il file originale non viene rimosso
- Rimuovendo un file gli eventuali link divengono inconsistenti
- Il comando `ls` mostra i symbolic link usando una `l` come primo carattere degli attributi del file

```
[scarpazza@pcsenonsrv scarpazza]$ ls -l
lrwxrwxrwx   1 scarpazz      8 Jul 14 14:38 link1 -> original
lrwxrwxrwx   1 scarpazz  8 Jul 14 14:38 link2 -> original
-rw-r--r--   1 scarpazz  5 Jul 14 14:37 original
[scarpazza@pcsenonsrv scarpazza]$ rm link1
[scarpazza@pcsenonsrv scarpazza]$ ls -l
lrwxrwxrwx   1 scarpazz  8 Jul 14 14:38 link2 -> original
-rw-r--r--   1 scarpazz  5 Jul 14 14:37 original
[scarpazza@pcsenonsrv scarpazza]$ rm original
[scarpazza@pcsenonsrv scarpazza]$ ls -l
lrwxrwxrwx   1 scarpazz  8 Jul 14 14:38 link2 -> original
[scarpazza@pcsenonsrv scarpazza]$ cat link2
cat: cannot open link2
[scarpazza@pcsenonsrv scarpazza]$
```

Link

- Hard link

- È una copia del descrittore di un altro file o di un'altra directory
- Rimuovendo un link il file originale non viene rimosso
- Il file non viene rimosso fino a che tutti i link non sono stati rimossi
- Il comando `ls` mostra gli hard link incrementando link count, nella seconda colonna

```
[scarpazza@pcsenonsrv scarpazza]$ ls -l
lrwxrwxrwx   3 scarpazz      8 Jul 14 14:38 link1
lrwxrwxrwx   3 scarpazz      8 Jul 14 14:38 link2
-rw-r--r--   3 scarpazz      5 Jul 14 14:37 original
[scarpazza@pcsenonsrv scarpazza]$ rm link1
[scarpazza@pcsenonsrv scarpazza]$ ls -l
lrwxrwxrwx   2 scarpazz      8 Jul 14 14:38 link2
-rw-r--r--   2 scarpazz      5 Jul 14 14:37 original
[scarpazza@pcsenonsrv scarpazza]$ rm original
[scarpazza@pcsenonsrv scarpazza]$ ls -l
lrwxrwxrwx   1 scarpazz      8 Jul 14 14:38 link2
[scarpazza@pcsenonsrv scarpazza]$ cat link2
This is the content of the file...
[scarpazza@pcsenonsrv scarpazza]$
```

Dispositivi

- Un device è un file speciale che fornisce una interfaccia comune a diversi dispositivi o dati. I device sono raccolti in `/dev` e nelle sue sottodirectory:

<code>stdin, stdout, stderr</code>	Standard input, output, error
<code>ttyX, ptyX, console</code>	Terminale fisico X, terminale virtuale X, terminale corrente
<code>null</code>	Null file (pozzo senza fondo)
<code>fd0, fd1, ...</code>	Primo, secondo floppy disk, ...
<code>hda, hdb, hdc, hdd</code>	Dischi IDE: primary master, primary slave, secondary master, secondary slave
<code>hda0, hda1, hda2, ...</code>	Prima, seconda, terza, partizione sul disco hda
<code>psaux</code>	Mouse PS/2
<code>sda, sdb, sdc, ...</code>	Primo, secondo, terzo disco SCSI
<code>sda1, sda2, sda3, ...</code>	Prima, seconda, terza partizione sul disco sda
<code>scd0, scd1, ...</code>	Primo, secondo, ... CD SCSI
<code>sg0, sg1, ...</code>	Primo, secondo, ... generico dispositivo SCSI
<code>ttyS0, ttyS1, ...</code>	Prima, seconda, ... porta seriale

Procfs

- Attraverso **/proc** è possibile accedere a numerose informazioni sullo stato corrente del sistema, esempio:
 - **/proc/cpuinfo** contiene informazioni sulla CPU;
 - **/proc/devices** major e minor di ogni dispositivo;
 - **/proc/tty/driver/serial** stato della seriale;
 - **/proc/sys/kernel/version** versione del kernel;
 - **/proc/sys/kernel/hostname** nome di rete della macchina;
 - **/proc/filesystem** tipi di filesystem noti al kernel;
 - **/proc/ide/ide1/hdc/media** dispositivo IDE sec/slave;
 - **/proc/ide/ide1/hdc/model** marca e modello del " " ;
 - **/proc** contiene una sottodirectory per ogni processo: raccoglie informazioni sul processo (immagine dell'eseguibile, stato, memoria, paginazione, directory di lavoro, descrittori file aperti, thread);

Organizzazione del file system

- E' prevista una organizzazione standard del file system:
 - / Root directory
 - /bin/ Raccoglie gli eseguibili e i comandi standard
 - /dev/ Contiene i device
 - /etc/ Contiene file di configurazione
 - /etc/passwd Password file
 - /etc/mnttab Tabella dei file system che vengono montati al boot
 - /etc/rc0.d/ Contiene gli script di shutdown. I nomi degli script iniziano con `Knn` e vengono eseguiti in ordine alfabetico. Sono invocati con argomento `stop`
 - /etc/rcS.d/ Contiene gli script di boot in single user mode. I nomi degli script iniziano con `Snn` e sono invocati con argomento `start`
 - /etc/rc2.d/ Contiene gli script di boot in multi user mode
 - /etc/init.d/ Raccoglie tutti gli script di boot e shutdown. Le directory `rc` contengono link a questi file

Organizzazione del file system

- `/var/` Contiene temporanei o semi-temporanei di sistema
 - `/var/spool/` Contiene le code di stampa
 - `/var/mail/` Contiene le 'inbox' degli utenti
 - `/var/log/` Contiene i transcript delle operazioni di sistema
- `/home/` Contiene le home directory degli utenti.
- `/lib/` Contiene le librerie statiche e dinamiche sia di sistema, sia degli applicativi
- `/local/` E' il mount-point di default dei file system locali
- `/mnt/` E' il mount-point dei file system addizionali
- `/opt/` Contiene pacchetti software aggiuntivi
- `/proc/` Accesso allo stato del kernel
- `/tmp/` Directory temporanea
- `/usr/` Contiene parti addizionali del sistema operativo

Mounting

- E' l'operazione che permette di collegare un file system ad una directory, detta mount-point, di un altro file system:
 - Il mount-point è una directory
 - Il contenuto preesistente del mount-point viene nascosto
- La rimozione di un collegamento viene detta unmounting:
 - Il contenuto preesistente del mount-point è nuovamente accessibile
 - Non è possibile scollegare un file system se un processo sta operando sul file system montato
- E' possibile collegare file system condivisi attraverso una rete (NFS mounting)
- Il file `/etc/mnttab` tiene traccia dei file system montati
- Le operazioni di mounting e unmounting agiscono sul file `/etc/mnttab`

Mounting

- Il file `/etc/mnttab` contiene le seguenti informazioni:
 - `special` Il nome del file system o del dispositivo
 - `mount-point` Il path del mount-point
 - `fstype` Il tipo di file system montato
 - `options` Opzioni di montaggio, tipicamente i diritti di accesso
 - `time` l'ora a cui il file system e' stato montato
- Il file `/etc/fstab` indica i file system da montare al boot del sistema, in particolare:
 - Il nome del device da montare
 - Il nome del file system su eseguire controlli di coerenza
 - Le opzioni di controllo di coerenza
 - Il tipo di file system
 - Le opzioni con cui montare il file system
 - Indicazioni sul montaggio al boot

Comandi

- **mount [-F *fstype*] [-r] *special mntpoint***
 - Monta il file system *special* nella directory *mntpoint*
 - L'operazione può essere compiuta solo da `root`
 - `-F` Specifica il tipo di file system
 - `-r` Monta il file system in modalità read-only
- **umount {*special* | *mntpoint*}**
 - Rimuove un file system montato. Il file system da rimuovere può essere indicato sia fornendo il dispositivo, *special*, sia il mount point *mntpoint*
- **mountall [*table*]**
 - Effettua il montaggio dei file system specificati nella tabella *table*
 - Solo i file system indicati come montabili al boot sono effettivamente montati
 - Se l'opzione *table* è omessa, viene usata la tabella `/etc/fstab`

Comandi

- **sync**
 - Aggiorna lo stato del file system
 - In pratica effettua il flush di tutti i buffer di sistema in modo da rendere coerente lo stato del file system
- **eject [-f] [-q] [device]**
 - Espelle un dispositivo rimovibile (CD, floppy motorizzati);
 - Il dispositivo può essere espulso solo se non è in uso
 - **-f** Forza l'espulsione anche se il file system è in uso
 - **-q** Interroga il sistema per verificare se il dispositivo specificato è presente
 - Senza argomenti il comando si riferisce al dispositivo di default
 - **-d** Ritorna il nome del dispositivo di default

I processi

- Un programma descrive le operazioni che devono essere svolte dal calcolatore per mezzo di istruzioni di alto livello (sorgente) o di basso livello (codice macchina)
- Un processo rappresenta una esecuzione del codice di un programma
- Un processo è costituito da:
 - Una immagine del programma, cioè del suo codice macchina
 - Un descrittore di processo è organizzato schematicamente così:

Puntatore
Process ID
Contenuto dei registri del processore
File descriptor dei file aperti

Stato del processo
Contatore di programma
Memoria associata al processo
...

Esecuzione di un processo

- Un processo viene sempre creato da un altro processo
 - Il processo creatore si chiama processo padre o parent process
 - Il processo creato si chiama processo figlio o child process
- Nei sistemi operativi unixlike tutti i processi discendono dal processo `init`, che inizializza il sistema
- I processi possono essere in esecuzione in due modalità:
 - **Foreground** Il processo figlio eredita e ritiene tutte le risorse del padre. Il padre è sospeso in attesa della terminazione del figlio
 - **Background** Il processo figlio eredita tutte le risorse del padre. Il processo padre rimane in esecuzione. A questo punto è lo scheduler ad assegnare le risorse ai due processi.
- L'utente può specificare e/o modificare la modalità di esecuzione di un processo.

Operazioni sui processi

- Creazione

- Un processo viene creato in foreground fornendo ad una shell il nome del programma corrispondente
- Per creare un processo in background si usa il carattere `&` alla fine della linea di comando della shell

- Terminazione

- Un processo in foreground viene terminato premendo opportune combinazioni di tasti, tipicamente `CTRL-C`
- Un processo in background viene terminato inviandogli un opportuno segnale (`SIGKILL`, `SIGABRT`) tramite il comando `kill`

- Sospensione

- Un processo in foreground viene sospeso premendo opportune combinazioni di tasti, tipicamente `CTRL-Z`
- Un processo in background viene sospeso inviandogli un opportuno segnale (`SIGSTOP`) tramite i comandi `kill` e `stop`

Operazioni sui processi

- Continuazione
 - Si può forzare la continuazione di un processo sospeso utilizzando i comandi **fg** e **bg**
 - È anche possibile inviare al processo sospeso un opportuno segnale (**SIGCONT**) tramite il comando **kill**
- Duplicazione
 - Un processo può essere duplicato solo attraverso chiamate a funzioni di sistema operativo
- Analisi
 - È possibile visualizzare alcune informazioni relative ai processi utilizzando i comandi **ps**, **pstree** e **jobs**

Comandi

- **program [&]**

- Crea un nuovo processo eseguendo una immagine di program
- Normalmente il processo viene eseguito in foreground
 - & Crea il nuovo processo e lo esegue in background

- **ps [a] [u] [x]**

- Mostra le informazioni più utili riguardanti i processi in esecuzione
 - a Mostra tutte le informazioni
 - u Mostra il nome del proprietario dei processi
 - x Mostra i processi di tutti gli utenti, non solo i propri

```
[scarpazza@pcsenonsrv scarpazza]$ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.3	0.3	1376	476	?	S	10:13	0:04	init
root	2	0.0	0.0	0	0	?	SW	10:13	0:00	[keventd]
root	555	0.0	0.5	2104	632	?	S	10:14	0:00	/sbin/dhclient -1
root	608	0.0	0.4	1456	556	?	S	10:14	0:00	syslogd -m 0
root	612	0.0	0.3	1380	400	?	S	10:14	0:00	klogd -x
rpc	638	0.0	0.3	1548	436	?	S	10:14	0:00	portmap
rpcuser	657	0.0	0.4	1580	556	?	S	10:14	0:00	rpc.statd
root	707	0.0	0.3	1368	420	?	S	10:14	0:00	/usr/sbin/apmd -p
root	744	0.0	0.4	3516	504	?	S	10:14	0:00	/usr/sbin/sshd

ps esempio di esecuzione

```
[root@pcsenonsrv root]$ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.3	0.3	1376	476	?	S	10:13	0:04	init
root	2	0.0	0.0	0	0	?	SW	10:13	0:00	[keventd]
root	3	0.0	0.0	0	0	?	SW	10:13	0:00	[kapmd]
root	4	0.0	0.0	0	0	?	SWN	10:13	0:00	[ksoftirqd_CPU0]
root	9	0.0	0.0	0	0	?	SW	10:13	0:00	[bdf flush]
root	5	0.0	0.0	0	0	?	SW	10:13	0:00	[kswapd]
root	555	0.0	0.5	2104	632	?	S	10:14	0:00	/sbin/dhclient -1
root	608	0.0	0.4	1456	556	?	S	10:14	0:00	syslogd -m 0
root	612	0.0	0.3	1380	400	?	S	10:14	0:00	klogd -x
rpc	638	0.0	0.3	1548	436	?	S	10:14	0:00	portmap
rpcuser	657	0.0	0.4	1580	556	?	S	10:14	0:00	rpc.statd
root	707	0.0	0.3	1368	420	?	S	10:14	0:00	/usr/sbin/apmd -p
root	744	0.0	0.4	3516	504	?	S	10:14	0:00	/usr/sbin/sshd
root	758	0.0	0.4	2068	572	?	S	10:14	0:00	xinetd -stayalive
ntp	777	0.0	1.8	2400	2392	?	SL	10:14	0:00	ntpd -U ntp
root	790	0.0	0.0	4196	40	?	S	10:14	0:00	/bin/sh /usr/bin/
mysql	820	0.0	0.0	13984	36	?	S	10:14	0:00	/usr/libexec/mysql
root	885	0.0	0.2	4528	360	?	S	10:14	0:00	/usr/libexec/postgresql
postfix	892	0.0	0.1	4604	136	?	S	10:14	0:00	pickup -l -t fifo
postfix	893	0.0	0.1	4688	240	?	S	10:14	0:00	nqmgr -l -n qmgr
root	907	0.0	0.0	21824	4	?	S	10:14	0:01	/usr/bin/spamd -d
privoxy	921	0.0	0.0	1884	4	?	S	10:14	0:00	/usr/sbin/privoxy
root	947	0.0	0.0	1416	56	?	S	10:14	0:00	gpm -t imps2 -m
root	974	0.0	0.0	19872	80	?	S	10:14	0:00	/usr/sbin/httpd
apache	1012	0.0	0.0	19944	4	?	S	10:14	0:00	/usr/sbin/httpd
apache	1013	0.0	0.0	19944	4	?	S	10:14	0:00	/usr/sbin/httpd
postgres	1050	0.0	0.4	9568	604	?	S	10:15	0:00	/usr/bin/postmast
postgres	1063	0.0	0.0	10560	4	?	S	10:15	0:00	postgres: stats b
postgres	1064	0.0	0.0	9592	4	?	S	10:15	0:00	postgres: stats c
root	1084	0.0	0.1	1424	132	?	S	10:15	0:00	cron
root	1104	0.0	0.3	8696	476	?	S	10:15	0:00	cupsd
root	1263	0.0	0.2	6076	304	?	S	10:15	0:00	ipp://hplj2 21 sc
xfs	1362	0.0	0.0	5904	68	?	S	10:15	0:00	xfs -droppriv -da
root	1386	0.0	0.4	6868	584	?	S	10:15	0:00	smbd -D
root	1396	0.0	0.4	5668	544	?	S	10:15	0:00	nmbd -D

Comandi

- **jobs [-1]**
 - Mostra la lista dei processi sospesi, figli della shell corrente
 - Ogni processo è individuato da un identificatore detto *jobid*
 - -1 Include nella lista anche il process id
- **fg [%jobid]**
 - Porta un processo in foreground
 - Senza argomenti porta in foreground il processo indicato con il simbolo '+' nella lista prodotta dal comando `jobs`
 - %*jobid* Porta in foreground il processo identificato dall'identificatore *jobid*, secondo quanto mostrato dal comando `jobs`
- **bg [%jobid]**
 - Porta un processo in background
 - Senza argomenti porta in background il processo indicato con il simbolo '+' nella lista prodotta dal comando `jobs`
 - %*jobid* Porta in background il processo identificato dall'identificatore *jobid*, secondo quanto mostrato dal comando `jobs`

Comandi

- **kill** `{-l|[-signal]}{pid|%jobid}`
 - Invia un segnale ad un processo, per default il segnale SIGKILL
 - **-l** Mostra la lista dei nomi dei segnali disponibili
 - **-signal** Invia il segnale *signal* al processo

```
[scarpazza@pcsenonsrv scarpazza]$ jobs -l
[1] + 19398 Suspended          find .
[2]   19399 Running            nedit thesis.tex
[3] - 19400 Running            xterm
[scarpazza@pcsenonsrv scarpazza]$ kill 19398
[1]   Terminated             find .
[scarpazza@pcsenonsrv scarpazza]$ kill -STOP 19399
[2] + Suspended (signal)       nedit thesis.tex
[scarpazza@pcsenonsrv scarpazza]$ jobs -l
[2] + 19399 Suspended (signal)  nedit thesis.tex
[3] - 19400 Running            xterm
[scarpazza@pcsenonsrv scarpazza]$ kill -CONT %2
[2]   nedit thesis.tex &
[scarpazza@pcsenonsrv scarpazza]$ jobs -l
[2]   19399 Running            nedit thesis.tex
[3] + 19400 Running            xterm
[scarpazza@pcsenonsrv scarpazza]$
```

Comandi

- **`stop pid | %jobid`**
 - Invia il segnale `SIGSTOP` al processo identificato `jobid`
 - È equivalente al comando `kill -STOP %jobid`
- **`time command [args]`**
 - Esegue il comando `command` con eventuali argomenti e mostra i tempi consumati: user time, system time e real time
 - Alcune implementazioni di questo comando mostrano anche la percentuale di utilizzo medio della CPU
 - Il formato di uscita dipende dalla implementazione specifica e dall shell in uso

```
[scarpazza@pcsenonsrv scarpazza]$ time find . > /dev/null
0.22u 2.21s 0:36.31 6.6%
[scarpazza@pcsenonsrv scarpazza]$ sh -c "time find tmp > /dev/null"

real      1.0
user      0.0
sys       0.1

[scarpazza@pcsenonsrv scarpazza]$
```

Manipolazione dei file di testo

- Il sistema unix-like fanno grande uso di file di testo e dispongono di molte utilities per la loro manipolazione;
- Alla base di molte utilities vi sono i concetti di:
 - Espressione Regolare (ER)
 - E' una forma compatta per indicare insiemi di stringhe
 - Si basa su una grammatica semplice costituita da
 - atomi
 - operatori
 - Glob
 - E' una particolare ER usata per i nomi di file
 - Quoting
 - La shell normalmente espande alcuni caratteri speciali (^\$', '~', ...)
 - Speciali virgolette prevengono o forzano l'espansione

Espressioni Regolari

- Caratteri speciali
 - `.` Indica un carattere qualsiasi
 - `^` Indica l'inizio di una linea
 - `$` Indica la fine di una linea
 - `\` Carattere di escape. Toglie il significato speciale ad un carattere
- Stringhe, set e range
 - `xyz` Indica la stringa "`xyz`" stessa
 - `[xyz]` Indica uno dei caratteri `x`, `y` o `z`, in alternativa (set)
 - `[^xyz]` Indica tutti i caratteri tranne `x`, `y` e `z` (set)
 - `[start-end]` Indica uno dei caratteri tra `start` e `end`, secondo l'ordinamento ASCII (range)
- Sottoespressioni
 - `(RE)` Indica una intera espressione regolare. Le parentesi sono usate per usarla come sottoespressione

Espressioni Regolari

- Operatori

- $(RE)^*$ Indica RE ripetuta 0 o più volte
- $(RE)^+$ Indica RE ripetuta 1 o più volte
- $(RE)?$ Indica RE ripetuta 0 o 1 volte
- $(RE1) | (RE2)$ Indica $RE1$ o $RE2$
- $(RE1)(RE2)$ Indica la concatenazione di $RE1$ e $RE2$

- Se una ER è sovrapponibile a una stringa si ha un match

- Una RE rappresenta una classe di stringhe:

- $[a-zA-Z]$ Stringhe alfabetiche
- $[+-]?[0-9]^+$ Numeri decimali interi
- $[+-]?[0-9]^*\.[0-9]^+$ Numeri decimali con parte frazionaria
- $/*.**/$ Linea di commento nel linguaggio C
- $.*$ Qualsiasi stringa

Filtri

- Le utilities per l'elaborazione dei file di testo sono filtri
- Un filtro è un programma che:
 - Legge un file una linea alla volta
 - Elabora la linea
 - Scrive il risultato su un file
- Normalmente:
 - Il file di ingresso è lo `stdin` o un file specificato come argomento
 - Il file di uscita è lo `stdout`



grep

- **grep [-v|-n|-i|-c] *re* [*file...*]**
 - Confronta ogni linea in ingresso con l'espressione regolare *re* e mostra solo le linee che la contengono
 - **-v** Mostra le linee che non contengono *re*
 - **-n** Mostra anche nome del file di ingresso ed il numero di linea
 - **-i** Esegue il matching in maniera case-insensitive
 - **-c** Mostra solo il numero di linee che contengono *re*
 - A volte è necessario racchiudere l'espressione regolare tra apici per evitare che la shell espanda alcuni caratteri speciali
 - Solo le espressioni regolari semplici sono supportate
- **egrep [-v|-n|-i|-c] *ere* [*file...*]**
 - Supporta anche le espressioni regolari estese (ERE)
- **fgrep [-v|-n|-i|-c] *string* [*file...*]**
 - Ricerca stringhe letterali piuttosto che espressioni regolari

Varie

- **strings** *[-n len] [file...]*
 - Cerca e stampa le stringhe alfanumeriche in un generico file binario
 - Le stringhe devono avere una lunghezza di almeno 4 caratteri
 - *-n len* Considera solo stringhe composte da almeno *len* caratteri
- **cat** *[-n|-b] [file|-...]*
 - Copia i file in ingresso sullo standard output
 - *-n* Stampa il numero di linea per tutte le linee
 - *-b* Stampa il numero di linea solo per le linee non vuote
 - *-* Legge lo standard input
- **head** *[-n] file...*
 - Mostra l'inizio di uno o più file. Senza l'argomento opzionale mostra le prime dieci linee di ogni file
 - *-n* Mostra le prime *n* linee

Varie

- **tail [+n|-n] [-f] file...**
 - Mostra le ultime linee di un file. Senza argomenti opzionali mostra le ultime dieci linee, quindi termina
 - **+n** Salta le prime *n* linee del file
 - **-n** Mostra le ultime *n* linee
 - **-f** Dopo aver mostrato le linee richieste non termina ma resta in attesa di nuove linee aggiunte al file.
- **more file...**
 - Mostra uno o più file una “pagina” alla volta ed accetta comandi:
 - **SPACE** Mostra una nuova pagina
 - **nRETURN** Mostra le prossime *n* linee
 - **/pattern** Cerca la prossima occorrenza di *pattern*
 - **n** Cerca la successiva occorrenza di *pattern*
 - **v** Lancia l'editor **vi**
 - **q** Termina

Gestione degli archivi

- Linux dispone di diverse utility per la gestione di archivi compressi:
 - Creazione di archivi di file
 - `tar`
 - Compressione di file
 - `gzip` / `gunzip`
 - `compress` / `uncompress`
 - `zcat`
 - Codifica
 - `uuencode` / `uudecode`
 - `crypt`
- Nel seguito sono brevemente descritte le funzionalità e le modalità di utilizzo di tali utilities

tar

- `tar c[v]f tarfile [file...] | -I file`
- `tar t[v]f tarfile [file...]`
- `tar x[v]f tarfile [file...]`
 - Aggiunge i file specificati ad un archivio.
 - Il comando supporta alcune opzioni in tutte le forme:
 - `v` Mostra i nomi dei file che vengono processati
 - `f` Utilizza un file come archivio. Si possono specificare device.
 - I tre modi di funzionamneto sono:
 - `c` Crea l'archivio
 - `-I` specifica un file da cui leggere l'elenco dei file da includere nell'archivio, uno per linea
 - `t` Mostra il contenuto dell'archivio. Se specificato viene mostrato solo il file richiesto.
 - `x` Estrae i file dall'archivio. Se specificato viene estratto solo il file richiesto.

gzip/gunzip

- **gzip [-1..-9] file...**

- Comprime i file specificati. Ad ogni nome di file compresso viene aggiunto il suffisso **.gz**. I file originali sono rimossi

- **-1..-9** Specifica il livello di compressione. La compressione maggiore si ottiene con **-9**.

- **gunzip file...**

- Decomprime i file specificati. I nuovi nomi di file sono quelli originali privati del suffisso **.gz**. I file compressi sono rimossi

```
[scarpazza@pcsenonsrv scarpazza]$ ls -l
...
-rwxr--r--  1 scarpazz  59100 Jul 11 12:58 pota.txt
...
[scarpazza@pcsenonsrv scarpazza]$ gzip pota.txt
[scarpazza@pcsenonsrv scarpazza]$ ls -l *.gz
-rwxr--r--  1 scarpazz  16848 Jul 11 12:58 pota.txt.gz
[scarpazza@pcsenonsrv scarpazza]$ gunzip pota.txt.gz
[scarpazza@pcsenonsrv scarpazza]$ ls -l *.txt
-rwxr--r--  1 scarpazz  59100 Jul 11 12:58 pota.txt
[scarpazza@pcsenonsrv scarpazza]$
```

crypt

- **crypt key**

- Cripta lo standard input e genera l'uscita sullo standard output
 - *key* Specifica la chiave da usare
- L'algoritmo di encryption è noto e pubblico e non fornisce un alto livello di sicurezza
- Questa utility può essere usata come filtro

```
[scarpazza@pcsenonsrv scarpazza]$ cat main.c
#include <stdio.h>
main() {
    printf( "Hello, World!\n" );
}
[scarpazza@pcsenonsrv scarpazza]$ crypt mykey < main.c
nÌ$ÊXÔóÖïëÂñË-m0ø:AXpt~CP>ÐDÎÛ}oPûÆÃOÊ¼Ûünà@òt»
[scarpazza@pcsenonsrv scarpazza]$ crypt mykey < main.c > main.c.enc
[scarpazza@pcsenonsrv scarpazza]$ ls -l
-rw-rw-rw-  1 brandole      63 Jul 31 14:25 main.c
-rw-rw-rw-  1 brandole      63 Jul 31 14:27 main.c.enc
[scarpazza@pcsenonsrv scarpazza]$
```

Redirezione e Pipelining

- I filtri e altri comandi:
 - Leggono l'ingresso dallo `standard input` (stream 0)
 - Scrivono l'uscita sullo `standard output` (stream 1)
 - Scrivono i messaggi di errore sullo `standard error` (stream 2)
- È possibile modificare questo comportamento tramite:
 - Redirezione dell'input/output
 - Un file viene letto al posto dello standard input
 - Un file viene scritto al posto dello standard output
 - Un file viene scritto al posto dello standard error
 - Pipelining
 - L'uscita di un comando viene usata come ingresso al comando seguente nella pipeline
 - L'ingresso di un comando viene prelevato dall'uscita del comando precedente nella pipeline

Redirezione

- Redirezione dello standard output:
 - `>file` Verso il file `file`. Sovrascrive
 - `>>file` Verso il file `file`. Aggiunge alla fine
- Redirezione dello standard input:
 - `<file` Dal file `file`
 - `<<word` Dallo standard input (o dallo script) fino alla parola `word`
- Redirezione dello standard error:
 - `2>file` Verso il file `file`. Sovrascrive
 - `2>>file` Verso il file `file`. Aggiunge alla fine
- Redirezione generica
 - `n>file` Lo stream `n` verso il file `file`. Sovrascrive
 - `n>>file` Lo stream `n` verso il file `file`. Aggiunge alla fine
 - `n>&m` Lo stream `n` verso lo stream `m`

Pipelining

- Una pipeline è una sequenza di comandi:

- Separati dal carattere `|`
- Eseguiti in cascata

- La seguente sintassi indica una pipeline:

```
cmd1 | cmd2 | ... cmdN
```

- In questo caso vengono usati gli standard input e gli standard output dei comandi

- Per usare lo standard error si usa la forma:

```
... | cmd1 2>&1 | cmd2 | ...
```

- In questo caso:

- Lo stream 2 del comando cmd1 viene rediretto sullo stream 1
- Lo standard output viene usato nella pipe come standard input del comando cmd2

Comandi

- Un comando semplice ha la forma:
 - `cmd;` Il comando `cmd` viene eseguito in foreground (in modo sincrono) ed il valore di uscita viene salvato nella variabile `$?`
 - `cmd&` Il comando `cmd` viene eseguito in background (in modo asincrono)
- Una list di comandi ha la forma:
 - `cmd1; cmd2; ...` I comandi `cmd1`, `cmd2`, ... vengono eseguiti in sequenza, ed il valore ritornato dall'ultimo comando viene salvato nella variabile `$?`
 - `cmd1 && cmd2 ...` Il comando `cmd1` viene eseguito. Il comando `cmd2` viene eseguito solo se il valore di ritorno di `cmd1` è uguale a zero
 - `cmd1 || cmd2 ...` Il comando `cmd1` viene eseguito. Il comando `cmd2` viene eseguito solo se il valore di ritorno di `cmd1` è diverso da zero

Dove documentarvi

- Tutta la documentazione di cui potreste avere bisogno è disponibile gratuitamente su Internet
- Imparate a reperire autonomamente la documentazione che vi serve:
 - www.linuxdoc.org
 - www.redhat.com
 - www.google.com
- spesso ci sono molti modi per portare a termine lo stesso compito;
- acquisite lo spirito critico necessario per capire di ogni documento che cosa vi può servire, che cosa è obsoleto, che cosa non si applica alla vostra distribuzione o configurazione
- Tipologie di documentazione disponibile:
 - guide guide complete, su argomenti vasti; importanti per imparare ma spesso poco aggiornate (IG, SAG, NAG, KHG, ...)
 - how-to brevi documenti; mettono rapidamente (e sbrigativamente) il lettore nelle condizioni di usare subito uno strumento;
 - FAQ domande poste di frequente su un argomento, e loro risposte; spesso contengono trucchetti d'uso, retroscena...
 - tutorial procedono passo per passo e sono pensate per lettori che non sanno nulla;