Rocky Linux Admin Guide (Italian version)

A book from the Documentation Team

Version : 2024/04/28

Rocky Documentation Team

Copyright © 2023 The Rocky Enterprise Software Foundation

Table of contents

1. Licence 11
2. Imparare Linux con Rocky 12
3. Introduzione al Sistema Operativo Linux 14
   - 3.1 Che cos'è un sistema operativo? 14
   - 3.2 Generalità UNIX - GNU/Linux 16
     - 3.2.1 Storia 16
     - 3.2.2 Quota di mercato 19
     - 3.2.3 Progettazione dell'architettura 19
     - 3.2.4 La filosofia UNIX/Linux 21
   - 3.3 Le distribuzioni GNU/Linux 21
     - 3.3.1 Ambienti desktop 22
     - 3.3.2 Libero / Open source 25
   - 3.4 Aree di utilizzo 26
   - 3.5 Shell 26
     - 3.5.1 Generalità 26
     - 3.5.2 Funzionalità 27
     - 3.5.3 Principio 27
   - 3.6 Verificare le proprie Conoscenze 28
4. Comandi per gli Utenti Linux 30
   - 4.1 Generalità 30
     - 4.1.1 Gli utenti 31
     - 4.1.2 La shell 32
   - 4.2 Comandi generali 33
     - 4.2.1 comandi `apropos`, `whatis` e `man` 33
     - 4.2.2 comando `shutdown` 36
     - 4.2.3 comando `history` 36
     - 4.2.4 Autocompletamento 37
   - 4.3 Visualizzazione e Identificazione 38
     - 4.3.1 commando `clear` 38
     - 4.3.2 comando `echo` 38
     - 4.3.3 comando `date` 39
     - 4.3.4 comando `id`, `who` e `whoami` 40
   - 4.4 Albero Dei File 41
     - 4.4.1 comando `pwd` 43
     - 4.4.2 comando `cd` 44

Copyright © 2023 The Rocky Enterprise Software Foundation

4.4.3 comando `ls` 44
4.4.4 comando `mkdir` 48
4.4.5 comando `touch` 48
4.4.6 comando `rmdir` 49
4.4.7 comando `rm` 49
4.4.8 command `mv` 50
4.4.9 comando `cp` 52
4.5 Visualizzazione 53
4.5.1 comando `file` 53
4.5.2 comando `more` 53
4.5.3 comando `less` 53
4.5.4 comando `cat` 54
4.5.5 comando `tac` 55
4.5.6 comando `head` 55
4.5.7 comando `tail` 56
4.5.8 comando `sort` 56
4.5.9 comando `wc` 59
4.6 Ricerca 60
4.6.1 comando `find` 60
4.6.2 opzione `-exec` del comando `find` 60
4.6.3 comando `whereis` 61
4.6.4 command `grep` 61
4.6.5 Meta-caratteri (wildcards) 62
4.7 Reindirizzamenti e pipes 63
4.7.1 Standard input e output 63
4.7.2 Redirezione Input 64
4.7.3 Redirezione Output 65
4.7.4 Esempi di reindirizzamento 66
4.7.5 Pipes 66
4.8 Punti Speciali 67
4.8.1 comando `tee` 67
4.8.2 comandi `alias` e `unalias` 68
4.8.3 Alias e Funzioni Utili 69
4.8.4 Il carattere `;` 71
4.9 Verificare le proprie Conoscenze 71
5. Comandi avanzati per gli utenti Linux 73
5.1 comando `uniq` 73

Copyright © 2023 The Rocky Enterprise Software Foundation

5.2 comando xargs 75
5.3 pacchetto yum-utils 78
5.3.1 Comando repoquery 78
5.3.2 Comando yumdownloader 79
5.4 pacchetto psmisc 79
5.5 comando watch 80
5.6 Comando install 81
5.7 Comando tree 83
5.8 Comando stat 84
6. Editor di testo VI 87
6.1 comando vi 87
6.2 Modalità operativa 89
6.2.1 La modalità comando 89
6.2.2 La modalità inserimento 90
6.2.3 La modalità Ex 90
6.3 Muovere il cursore 90
6.3.1 Da un carattere 90
6.3.2 Dal primo carattere di una parola 91
6.3.3 Da qualsiasi posizione su una linea 91
6.4 Inserimento del testo 92
6.4.1 In relazione a un carattere 92
6.4.2 In relazione a una linea 92
6.4.3 In relazione al testo 93
6.5 Caratteri, parole e linee 93
6.5.1 Caratteri 93
6.5.2 Parole 94
6.5.3 Linee 94
6.5.4 Annullare un'azione 95
6.5.5 Annulla la cancellazione 96
6.6 comando EX 96
6.6.1 Numeri di riga del file 96
6.6.2 Cerca una stringa 96
6.6.3 Sostituisci una stringa 98
6.6.4 Operazioni sui file 98
6.6.5 Operazioni sui file 99
6.7 Altre funzioni 100
6.7.1 comando vimtutor 100

Copyright © 2023 The Rocky Enterprise Software Foundation

7. Gestione utenti 101
7.1 Generale 101
7.2 Gestione del gruppo 102
7.2.1 comando `groupadd` 102
7.2.2 Comando `groupmod` 103
7.2.3 comando `groupdel` 104
7.2.4 file `/etc/group` 105
7.2.5 file `/etc/gshadow` 106
7.3 Gestione utenti 107
7.3.1 Definizione 107
7.3.2 comando `useradd` 107
7.3.3 comando `usermod` 110
7.3.4 comando `userdel` 112
7.3.5 file `/etc/passwd` 112
7.3.6 file `/etc/shadow` 113
7.4 Proprietari dei file 115
7.4.1 Comandi di modifica 115
7.4.2 comando `chgrp` 116
7.5 Gestione degli utenti 117
7.5.1 comando `gpasswd` 117
7.5.2 comando `id` 117
7.5.3 comando `newgrp` 118
7.6 Protezione 119
7.6.1 commando `passwd` 119
7.6.2 comando `chage` 120
7.7 Gestione avanzata 122
7.7.1 file `/etc/default/useradd` 122
7.7.2 file `/etc/login.defs` 123
7.7.3 `/etc/skel` directory 124
7.8 Cambiamento di identità 125
7.8.1 comando `su` 125
8. File System 127
8.1 Partizionamento 127
8.1.1 Convenzioni di denominazione per i nomi dei file del dispositivo 129
8.1.2 Numero di partizione del dispositivo 129
8.1.3 comando `parted` 130
8.1.4 comando `cfdisk` 131

Copyright © 2023 The Rocky Enterprise Software Foundation

8.2 Gestore di volumi logici (LVM) 132
8.2.1 Il Meccanismo di Scrittura di LVM 134
8.2.2 Comandi LVM per la gestione dei volumi 135
8.2.3 Comandi LVM per visualizzare le informazioni sui volumi 137
8.2.4 Preparazione dei supporti fisici 138
8.3 Struttura di un file system 138
8.3.1 comando `mkfs` 139
8.3.2 Boot sector 140
8.3.3 Super block 140
8.3.4 Tabella degli inode 141
8.3.5 Data block 142
8.3.6 Riparazione del file system 143
8.4 Organizzazione di un file system 143
8.4.1 file `/etc/fstab` 146
8.4.2 Comandi di gestione del montaggio 147
8.5 Convenzione di denominazione dei file 148
8.5.1 Dettagli del nome di un file 149
8.6 Attributi dei file 153
8.6.1 Permessi di base di file e directory 153
8.6.2 Tipo di utente corrispondente all'autorizzazione di base 154
8.6.3 Gestione degli attributi 154
8.7 Diritti predefiniti e maschera 157
8.7.1 comando `umask` 158
9. Gestione dei processi 160
9.1 Generalità 160
9.2 Visualizzazione dei processi 161
9.3 Tipi di processi 163
9.4 Autorizzazioni e diritti 163
9.5 Gestione dei processi 164
9.5.1 La priorità di un processo 164
9.5.2 Modalità di funzionamento 165
9.6 Controlli per la gestione dei processi 165
9.6.1 comando `kill` 165
9.6.2 comando `nohup` 166
9.6.3 [CTRL] + [Z] 166
9.6.4 istruzione `&` 166
9.6.5 comandi `fg` e `bg` 167
9.6.6 comando `jobs` 167

Copyright © 2023 The Rocky Enterprise Software Foundation

9.6.7 comandi `nice` e `renice` 168
9.6.8 comando `top` 169
9.6.9 comandi `pgrep` e `pkill` 170
9.6.10 comando `killall` 171
9.6.11 comando `pstree` 171
9.6.12 Processi orfani e processi zombie 172
10. Backup e ripristino 174
10.1 Generalità 175
10.1.1 Il processo 175
10.1.2 Metodi di backup 175
10.1.3 Periodicità 176
10.1.4 Metodi di ripristino 176
10.1.5 Gli strumenti 176
10.1.6 Convenzione di denominazione 177
10.1.7 Contenuto di un backup 178
10.1.8 Modalità di archiviazione 178
10.2 Tape ArchiveR - `tar` 178
10.2.1 Linee guida per il ripristino 179
10.2.2 Backup con `tar` 179
10.3 *CoPy Input Output* - `cpio` 187
10.3.1 Creare un backup con il comando `cpio` 188
10.3.2 Tipo di backup 189
10.3.3 Aggiungere a un backup 189
10.3.4 Compressione di un backup 189
10.3.5 Leggere il contenuto di un backup 190
10.3.6 Ripristinare un backup 191
10.4 Utilità di Compressione - decompressione 192
10.4.1 Compressione con `gzip` 192
10.4.2 Compressione con `bunzip2` 193
10.4.3 Decompressione con `gunzip` 193
10.4.4 Decompressione con `bunzip2` 194
11. Avvio del sistema 195
11.1 Il processo di avvio 195
11.1.1 L'avvio del BIOS 195
11.1.2 Il Master boot record (MBR) 196
11.1.3 Il bootloader Grub2 196
11.1.4 Il kernel 197
11.1.5 `systemd` 197

Copyright © 2023 The Rocky Enterprise Software Foundation

11.2 Protezione del bootloader GRUB2 198
11.3 Systemd 199
11.3.1 Gestione dei servizi di sistema 201
11.3.2 Esempio di un file .service per il servizio postfix 202
11.3.3 Utilizzo degli obiettivi di sistema 202
11.3.4 Il processo `journald` 205
11.3.5 comando `journalctl` 206
12. Gestione dei compiti 208
12.1 Generalità 208
12.2 Come funziona il servizio 209
12.3 Sicurezza 209
12.3.1 I files `cron.allow` and `cron.deny` 210
12.3.2 Consentire ad un utente 210
12.3.3 Proibire ad un utente 210
12.4 Pianificazione delle attività 211
12.4.1 Il comando `crontab` 211
12.4.2 Usi di `crontab` 212
12.5 Il file `crontab` 212
12.5.1 Processo di esecuzione dell'attività 214
13. Implementazione della Rete 215
13.1 Generalità 215
13.1.1 Indirizzo MAC / Indirizzo IP 217
13.1.2 Dominio DNS 218
13.1.3 Promemoria del modello OSI 218
13.2 La denominazione delle interfacce 219
13.3 Uso del comandi `ip` 220
13.4 Il nome host 220
13.5 /etc/hosts file 221
13.6 il file `/etc/nsswitch.conf` 222
13.7 file `/etc/resolv.conf` 222
13.8 comando `ip` 223
13.9 configurazione DHCP 224
13.10 Configurazione statica 0
13.11 Routing (Instradamento) 0
13.12 Risoluzione dei nomi 0
13.13 Risoluzione dei problemi 0
13.13.1 comando `dig` 0
13.13.2 comando `getent` 0

Copyright © 2023 The Rocky Enterprise Software Foundation

13.13.3 comando `ipcalc` 0
13.13.4 comando `ss` 0
13.13.5 comando `netstat` 0
13.13.6 Conflitti di indirizzi IP o MAC 0
13.14 Configurazione a caldo 0
13.15 In sintesi 0
14. Gestione del software 0
14.1 Generalità 0
14.2 RPM: Gestione pacchetti RedHat 0
14.2.1 comando `rpm` 0
14.3 DNF: Dandified Yum 0
14.3.1 comando `dnf` 0
14.3.2 Altre utili opzioni `dnf` 0
14.3.3 Come funziona DNF 0
14.4 Moduli DNF 0
14.4.1 Cosa sono i moduli 0
14.4.2 Elenco dei moduli 0
14.4.3 Abilitazione dei Moduli 0
14.4.4 Installazione dei pacchetti dal flusso del modulo 0
14.4.5 Installazione di pacchetti dai profili di flusso del modulo 0
14.4.6 Rimozione e Ripristino del modulo o Commutazione 0
14.4.7 Disattivare un flusso di moduli 0
14.5 Il repository EPEL 0
14.5.1 Che cos'è EPEL e come si usa? 0
14.5.2 Installazione 0
14.5.3 Usare EPEL 0
14.5.4 Conclusione 0
14.6 Plugin DNF 0
14.6.1 `config-manager` plugin 0
14.6.2 `copr` plugin 0
14.6.3 `download` plugin 0
14.6.4 `needs-restart` plugin 0
14.6.5 `versionlock` plugin 0
15. Review basic permissions 0
15.1 Seven file types 0
15.2 The meaning of basic permissions 0
15.3 Special authority 0
15.3.1 ACL permissions 0

Copyright © 2023 The Rocky Enterprise Software Foundation

15.3.2 SetUID 0
15.3.3 SetGID 0
15.3.4 Sticky BIT 0
15.3.5 chattr 0
15.3.6 sudo 0

Copyright © 2023 The Rocky Enterprise Software Foundation

1. Licence

RockyLinux offers Linux courseware for trainers or people wishing to learn how to administer a Linux system on their own.

RockyLinux materials are published under Creative Commons-BY-SA. This means you are free to share and transform the material, while respecting the author's rights.

BY : **Attribution**. You must cite the name of the original author.

SA : **Share Alike**.

- Creative Commons-BY-SA licence : https://creativecommons.org/licenses/by-sa/4.0/

The documents and their sources are freely downloadable from:

- https://docs.rockylinux.org
- https://github.com/rocky-linux/documentation

Our media sources are hosted at github.com. You'll find the source code repository where the version of this document was created.

From these sources, you can generate your own personalized training material using mkdocs. You will find instructions for generating your document here.

How can I contribute to the documentation project?

You'll find all the information you need to join us on our git project home page.

We wish you all a pleasant reading and hope you enjoy the content.

Copyright © 2023 The Rocky Enterprise Software Foundation

2. Imparare Linux con Rocky

La Guida dell'Amministratore è una raccolta di documenti didattici dedicati agli Amministratori di Sistema. Possono essere utilizzati dai futuri amministratori di sistema che cercano di aggiornarsi, dagli attuali amministratori di sistema che desiderano un ripasso o da qualsiasi utente di Linux che desideri saperne di più sull'ambiente Linux, sui comandi, sui processi e altro ancora. Come tutti i documenti di questo tipo, si evolverà e si aggiornerà nel tempo.

Si inizia con Introduzione a Linux, che illustra Linux, le distribuzioni e l'intero ecosistema che ruota attorno al nostro sistema operativo.

I Comandi Utente contengono i comandi essenziali per iniziare a lavorare con Linux. Gli utenti più esperti possono consultare anche il capitolo seguente sui Comandi Avanzati di Linux.

L'editor di testo VI merita un capitolo a sé. Anche se Linux è provvisto di molti editor, VI è uno dei più potenti. Altri comandi utilizzano talvolta una sintassi identica a quella di VI (viene in mente `sed`). È quindi molto importante sapere qualcosa su VI, o almeno demistificarne le funzioni essenziali (come aprire un file, salvare, uscire o uscire senza salvare). Con l'uso dell'editor, l'utente acquisirà maggiore familiarità con le altre funzioni di VI. Un'alternativa consiste nell'usare nano, che è installato di default in Rocky Linux. Sebbene non sia altrettanto versatile, è semplice da usare, diretto e in grado di svolgere il proprio lavoro.

A questo punto possiamo addentrarci nel funzionamento del sistema Linux per scoprire come si comporta il sistema:

- Gestione Utenti
- File Systems
- Gestione dei Processi

Il Backup e il Ripristino sono informazioni essenziali per l'Amministratore di Sistema. Linux dispone di molte soluzioni software per migliorare i backup (rsnapshot, lsyncd, ecc.). È importante conoscere i componenti principali per il backup che sono presenti nel sistema operativo. In questo capitolo esamineremo due strumenti: `tar` e il meno diffuso `cpio`.

Copyright © 2023 The Rocky Enterprise Software Foundation

L'avvio del sistema è una lettura importante anche perché la gestione del sistema durante il processo di avvio si è evoluta in modo significativo negli ultimi anni con l'arrivo di systemd.

I capitoli finali trattano la gestione dei processi, l'implementazione della rete e la gestione del software, compresa l'installazione.

Copyright © 2023 The Rocky Enterprise Software Foundation

3. Introduzione al Sistema Operativo Linux

In questo capitolo imparerete a conoscere le distribuzioni GNU/Linux.

Obiettivi: In questo capitolo imparerai a:

- ✔ Descrivere le caratteristiche e le possibili architetture di un sistema operativo..
- ✔ Raccontare la storia di UNIX e GNU/Linux.
- ✔ Scegliete la distribuzione Linux più adatta alle vostre esigenze.
- ✔ Spiegare la filosofia del software libero e open-source.
- ✔ Scoprite l'utilità della shell.

🏁 **generalità**, **linux**, **distribuzioni**

Complessità: ⭐

Complexity: ⭐

Tempo di lettura: 10 minuti

3.1 Che cos'è un sistema operativo?

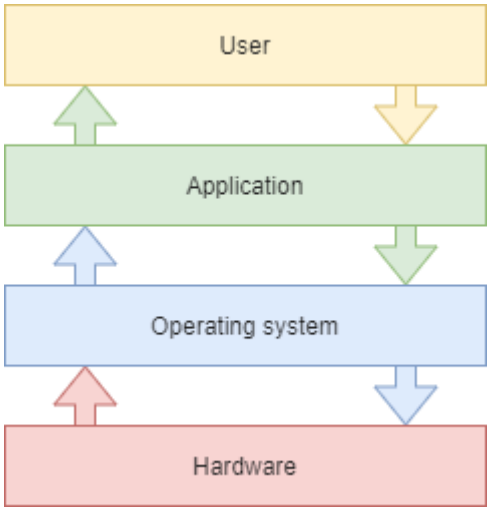
Linux, UNIX, BSD, VxWorks, Windows e MacOS sono tutti **sistemi operativi**.

> **Astrazione**
>
> Un sistema operativo è un **insieme di programmi che gestisce le risorse disponibili di un computer**.

Copyright © 2023 The Rocky Enterprise Software Foundation

Nell'ambito di questa gestione delle risorse, il sistema operativo deve:

- Gestire la memoria **fisica** o **virtuale**.
- La **memoria fisica** è costituita dalle memorie RAM e dalla memoria cache del processore, utilizzate per l'esecuzione dei programmi.
- La **memoria virtuale** è una posizione sul disco rigido (la partizione **swap**) che consente di svuotare la memoria fisica e di salvare lo stato attuale del sistema durante lo spegnimento elettrico del computer.
- Intercettare **l'accesso alle periferiche**. Raramente il software può accedere direttamente all'hardware (ad eccezione delle schede grafiche per esigenze molto specifiche).
- Fornire alle applicazioni una corretta **gestione delle attività**. Il sistema operativo è responsabile della pianificazione dei processi che occupano il processore
- **Proteggere i file** da accessi non autorizzati.
- **Raccogliere informazioni** sui programmi in uso o in fase di esecuzione.



Copyright © 2023 The Rocky Enterprise Software Foundation

3.2 Generalità UNIX - GNU/Linux

3.2.1 Storia

UNIX

- **1964 - 1968**: MULTICS (MULTiplexed Information and Computing Service) è stato sviluppato per il MIT, i Bell Labs (AT&T) e General Electric.
- **1969 — 1971**: Dopo il ritiro di Bell (1969) e poi di General Electric dal progetto, due sviluppatori, Ken Thompson e Dennis Ritchie (a cui si aggiunge in seguito Brian Kernighan), giudicando MULTICS troppo complesso, iniziano lo sviluppo di UNIX (UNiplexed Information and Computing Service). Sebbene sia stato originariamente creato in linguaggio Assembly, i creatori di UNIX svilupparono alla fine il linguaggio B e poi il linguaggio C (1971) e riscrissero completamente UNIX. Poiché è stato sviluppato nel 1970, la data di riferimento (epoch) per l'inizio del periodo di tempo dei sistemi UNIX/Linux è fissata al 01 gennaio 1970.

Il linguaggio C rimane ancora oggi uno dei linguaggi di programmazione più diffusi. Un linguaggio di basso livello, a contatto con l'hardware, che permette di adattare il sistema operativo a qualsiasi architettura di macchina dotata di un compilatore C.

UNIX è un sistema operativo aperto e in continua evoluzione che ha svolto un ruolo fondamentale nella storia dell'informatica. Costituisce la base di molti altri sistemi come Linux, BSD, MacOS e altri ancora.

UNIX è ancora rilevante al giorno d'oggi (HP-UX, AIX, Solaris, ecc.).

Progetto GNU

- **1984**: Richard Matthew Stallman ha lanciato il progetto GNU (GNU's Not Unix), che si propone di creare un sistema Unix **libero** e **aperto**, in cui gli strumenti più importanti sono: il compilatore gcc, la shell bash, l'editor Emacs e così via. GNU è un sistema operativo simile ad Unix. Lo sviluppo di GNU, iniziato nel Gennaio 1984, è noto come Progetto GNU. Molti dei programmi presenti in GNU sono rilasciati sotto l'egida del Progetto GNU; questi vengono chiamati pacchetti GNU.

Copyright © 2023 The Rocky Enterprise Software Foundation

- **1990**: Il kernel di GNU, GNU Hurd, è stato creato nel 1990 (prima della nascita di Linux).

MINIX

- **1987**: Andrew S. Tanenbaum sviluppa MINIX, un UNIX semplificato, per insegnare i sistemi operativi in modo semplice. Il signor Tanenbaum rende disponibile il codice sorgente del suo sistema operativo.

Copyright © 2023 The Rocky Enterprise Software Foundation

Linux

- **1991**: Uno studente finlandese, **Linus Torvalds**, crea un sistema operativo che gira sul suo computer personale e lo chiama Linux. Pubblica la sua prima versione 0.02, sul forum di discussione Usenet e altri sviluppatori iniziano a contribuire al miglioramento del suo sistema. Il termine Linux è un gioco di parole tra il nome del fondatore, Linus, e UNIX.
- **1993**: Viene creata la distribuzione Debian. Debian è una distribuzione non commerciale, basata sulla comunità. Sviluppato originariamente per l'uso sui server, è molto indicato per questo ruolo; tuttavia è un sistema universale, utilizzabile anche su un personal computer. Debian costituisce la base per molte altre distribuzioni, come Mint o Ubuntu.
- **1994**: la distribuzione commerciale Red Hat viene creata dalla società Red Hat, che oggi è il principale distributore del sistema operativo GNU/Linux. Red Hat supporta la versione comunitaria Fedora e, fino a poco tempo fa, la distribuzione gratuita CentOS.
- **1997**: Viene creato l'ambiente desktop KDE. Si basa sulla libreria di componenti Qt e sul linguaggio di sviluppo C++.
- **1999**: Viene creato l'ambiente desktop GNOME. Basato sulla libreria di componenti GTK+.
- **2002**: Viene creata la distribuzione Arch. Il suo tratto distintivo è che offre una rolling release (aggiornamento continuo).
- **2004**: Ubuntu viene creato dalla società Canonical (Mark Shuttleworth). È basato su Debian ma include software libero e proprietario.
- **2021**: Viene creato Rocky Linux, basato sulla distribuzione Red Hat.

Informazione

Disputa sul nome: anche se le persone sono abituate a chiamare il sistema operativo Linux verbalmente, Linux è propriamente il kernel. Non dobbiamo dimenticare lo sviluppo e il contributo del progetto GNU alla causa open source, quindi! Preferisco chiamarlo il sistema operativo GNU/Linux.

Copyright © 2023 The Rocky Enterprise Software Foundation

3.2.2 Quota di mercato

Nonostante la sua diffusione, Linux rimane relativamente sconosciuto al grande pubblico. Linux infatti è nascosto in **smartphone**, **televisori**, **internet box**, ecc. Quasi il **70% delle pagine web** servite nel mondo sono servite da un server Linux o UNIX!

Linux equipaggia poco più del **3% dei personal computer** ma più dell'**82% degli smartphone**. Il sistema operativo **Android**, ad esempio, utilizza un kernel Linux.

Linux equipaggia il 100% dei 500 supercomputer dal 2018. Un supercomputer è un computer progettato per ottenere le massime prestazioni possibili con le tecniche conosciute al momento della sua progettazione, soprattutto per quanto riguarda la velocità di calcolo.

3.2.3 Progettazione dell'architettura

- Il **kernel** è il primo componente software.
- È il cuore del sistema Linux.
- Gestisce le risorse hardware del sistema.
- Gli altri componenti software devono passarvi attraverso per accedere all'hardware.
- La **shell** è un'utilità che interpreta i comandi dell'utente e ne garantisce l'esecuzione.
- Shell principali: shell Bourne, shell C, shell Korn e shell Bourne-Again (bash).
- **Applicazioni** sono programmi utente, tra cui, a titolo esemplificativo e non esaustivo:
- Browser Internet
- Elaboratore di testi
- Fogli di calcolo

Copyright © 2023 The Rocky Enterprise Software Foundation

Multi-task

Linux appartiene alla famiglia dei sistemi operativi con condivisione del tempo. Divide il tempo di elaborazione tra diversi programmi, passando da uno all'altro in modo trasparente per l'utente. Questo implica:

- Esecuzione simultanea di più programmi.
- Distribuzione del tempo di CPU da parte dello scheduler.
- Riduzione dei problemi causati da un'applicazione interrotta.
- Prestazioni ridotte quando ci sono troppi programmi in esecuzione.

Multiutente

Lo scopo di MULTICS era quello di consentire a più utenti di lavorare su più terminali (schermo e tastiera) da un unico computer (all'epoca molto costoso). Linux, ispirandosi a questo sistema operativo, ha mantenuto la capacità di lavorare con più utenti contemporaneamente e in modo indipendente, ognuno dei quali dispone di un proprio account utente con spazio di memoria e diritti di accesso a file e software.

Multiprocessore

Linux è in grado di lavorare con computer multiprocessore o con processori multi-core.

Copyright © 2023 The Rocky Enterprise Software Foundation

Multipiattaforma

Linux è scritto in un linguaggio di alto livello che può essere adattato a diversi tipi di piattaforme durante la compilazione. Questo permette di funzionare su:

- Computer di casa (PC o laptop)
- Server (dati, applicazioni,...)
- Computer portatili (smartphone o tablet)
- Sistemi integrati (computer per auto)
- Elementi di rete attivi (router, switch)
- Elettrodomestici (TV, frigoriferi,...)

Aperto

Linux si basa su standard riconosciuti come POSIX, TCP/IP, NFS e Samba, che gli consentono di condividere dati e servizi con altri sistemi operativi.

3.2.4 La filosofia UNIX/Linux

- Trattare tutto come un file.
- Portabilità del valore.
- Fai solo una cosa e falla bene.
- KISS: Mantienilo semplice stupido (Keep It Simple Stupid).
- "UNIX è fondamentalmente un sistema operativo semplice, Ma devi essere un genio per capirne la semplicità." (**Dennis Ritchie**)
- "Unix è facile da usare. Solamente che non è chiaro con quali utenti sia amichevole." (**Steven King**)

3.3 Le distribuzioni GNU/Linux

Una distribuzione Linux è un **insieme coerente di software** assemblato attorno al kernel Linux e pronto per essere installato insieme ai componenti necessari per gestire questo software (installazione, rimozione, configurazione). Ci sono

Copyright © 2023 The Rocky Enterprise Software Foundation

distribuzioni **associative** o **comunitarie** (Debian, Rocky) o **commerciali** (RedHat, Ubuntu).

Ogni distribuzione offre uno o più **ambienti desktop**, fornisce un set di software preinstallato e una libreria di software aggiuntivo. Le opzioni di configurazione (ad esempio le opzioni del kernel o dei servizi) sono specifiche per ogni distribuzione.

Questo principio permette alle distribuzioni di essere orientate ai **principianti** (Ubuntu, Linux Mint...) o completamente personalizzabili per gli **utenti avanzati** (Gentoo, Arch); le distribuzioni possono anche essere più adatte ai **server** (Debian, Red Hat) o alle **workstation** (Fedora).

3.3.1 Ambienti desktop

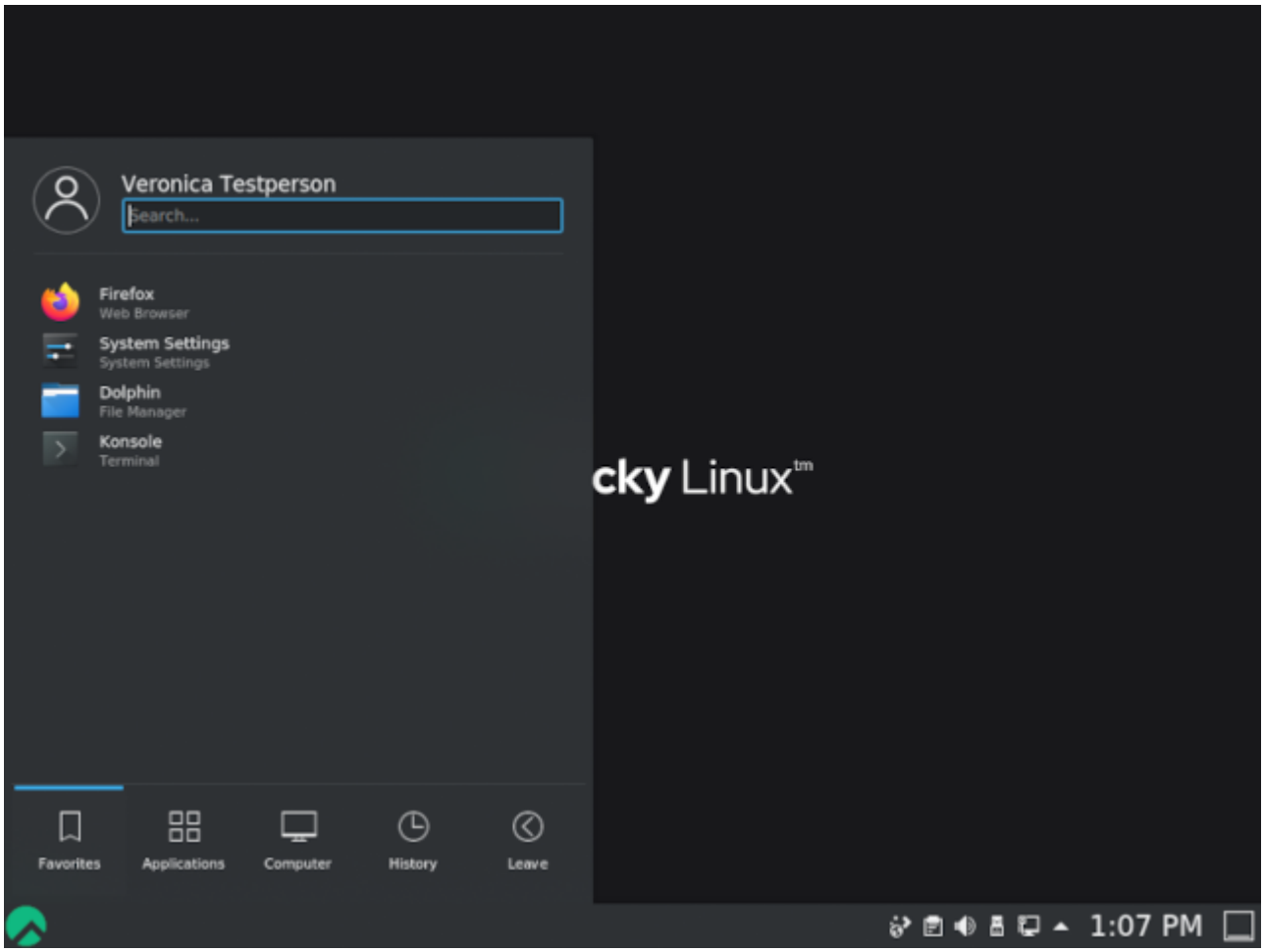
Esistono numerosi ambienti grafici, come **GNOME**, **KDE**, **LXDE**, **XFCE**, etc. Ce n'è per tutti i gusti e la loro **ergonomia** regge il confronto con i sistemi Microsoft o Apple.

Allora perché c'è così poco entusiasmo per Linux, quando questo sistema è praticamente **privo di virus**? Potrebbe essere il motivo per cui cosi tanti editori (Adobe) e produttori (Nvidia) non sono gratuiti, né forniscono una versione del proprio software o dei propri **driver** per GNU/Linux? Forse è la paura del cambiamento, o la difficoltà nel trovare dove acquistare un computer Linux, o ancora i pochi giochi distribuiti sotto Linux. Almeno quest'ultima scusa non dovrebbe essere vera a lungo, con l'avvento del motore di gioco Steam per Linux.

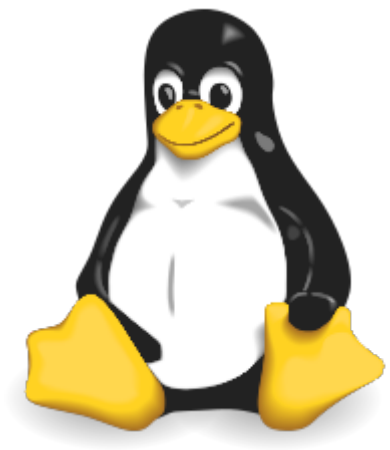
Copyright © 2023 The Rocky Enterprise Software Foundation

L'ambiente desktop di **GNOME 3** non utilizza più il concetto di desktop ma quello di GNOME Shell (da non confondere con la shell a riga di comando). Serve come desktop, come dashboard, area di notifica e selettore di finestre. L'ambiente desktop GNOME si basa sulla libreria di componenti GTK+.

Copyright © 2023 The Rocky Enterprise Software Foundation



L'ambiente desktop **KDE** si basa sulla libreria di componenti **Qt**. È tradizionalmente consigliato agli utenti che hanno familiarità con l'ambiente Windows.



Copyright © 2023 The Rocky Enterprise Software Foundation

3.3.2 Libero / Open source

Un utente di un sistema operativo Microsoft o Mac deve acquistare una licenza per utilizzare il sistema operativo. Questa licenza ha un costo, anche se di solito è trasparente (il prezzo della licenza è incluso nel prezzo del computer).

Nel mondo **GNU/Linux**, il movimento del Software Libero fornisce principalmente distribuzioni libere.

Libero non significa gratuito!

Open source: il codice sorgente è disponibile, quindi è possibile consultarlo e modificarlo a determinate condizioni.

Un software libero è necessariamente open-source, ma non è vero il contrario, poiché il software open-source si distingue per la libertà offerta dalla licenza GPL.

GNU GPL (Licenza pubblica generale GNU)

La **GPL** garantisce all'autore di un software la sua proprietà intellettuale, ma consente la modifica, la ridistribuzione o la rivendita del software da parte di terzi, a condizione che il codice sorgente sia incluso nel software. La GPL è la licenza nata dal progetto **GNU** (GNU is Not UNIX), che è stato determinante per la creazione di Linux.

Questo implica:

- La libertà di eseguire il programma, per qualsiasi scopo.
- La libertà di studiare il funzionamento del programma e di adattarlo alle proprie esigenze.
- La libertà di ridistribuire le copie.
- La libertà di migliorare il programma e di pubblicare tali miglioramenti a beneficio dell'intera comunità.

D'altra parte, anche i prodotti concessi sotto licenza GPL possono avere un costo. Non si tratta del prodotto in sé, ma della **garanzia che un team di sviluppatori continuerà a lavorarci per farlo evolvere e per risolvere gli errori, o anche per fornire assistenza agli utenti**.

Copyright © 2023 The Rocky Enterprise Software Foundation

3.4 Aree di utilizzo

Una distribuzione Linux eccelle per:

- **Server**: HTTP, e-mail, groupware, condivisione di file, ecc.
- **Sicurezza**: Gateway, firewall, router, proxy, ecc.
- **Computer centrali**: Banche, assicurazioni, industria, ecc.
- **Sistemi integrati**: Router, Internet box, SmartTV, ecc.

Linux è una scelta adatta per ospitare database o siti web, oppure come server di posta, DNS o firewall. In breve, Linux può fare praticamente tutto, il che spiega la quantità di distribuzioni specifiche.

3.5 Shell

3.5.1 Generalità

La **shell**, conosciuta anche come *command interface*, consente agli utenti di inviare comandi al sistema operativo. Oggi è meno visibile dopo l'implementazione delle interfacce grafiche, ma rimane un mezzo privilegiato sui sistemi Linux che non dispongono di interfacce grafiche e i cui servizi non sempre hanno un'interfaccia di impostazione.

Offre un vero e proprio linguaggio di programmazione che include strutture classiche (cicli, alternative) e componenti comuni (variabili, passaggio di parametri e sottoprogrammi). Permette la creazione di script per automatizzare determinate azioni (backups, creazione di utenti, monitoraggio del sistema, etc.).

Sono disponibili diversi tipi di shell, configurabili su una piattaforma o in base alle preferenze dell'utente. Alcuni esempi sono:

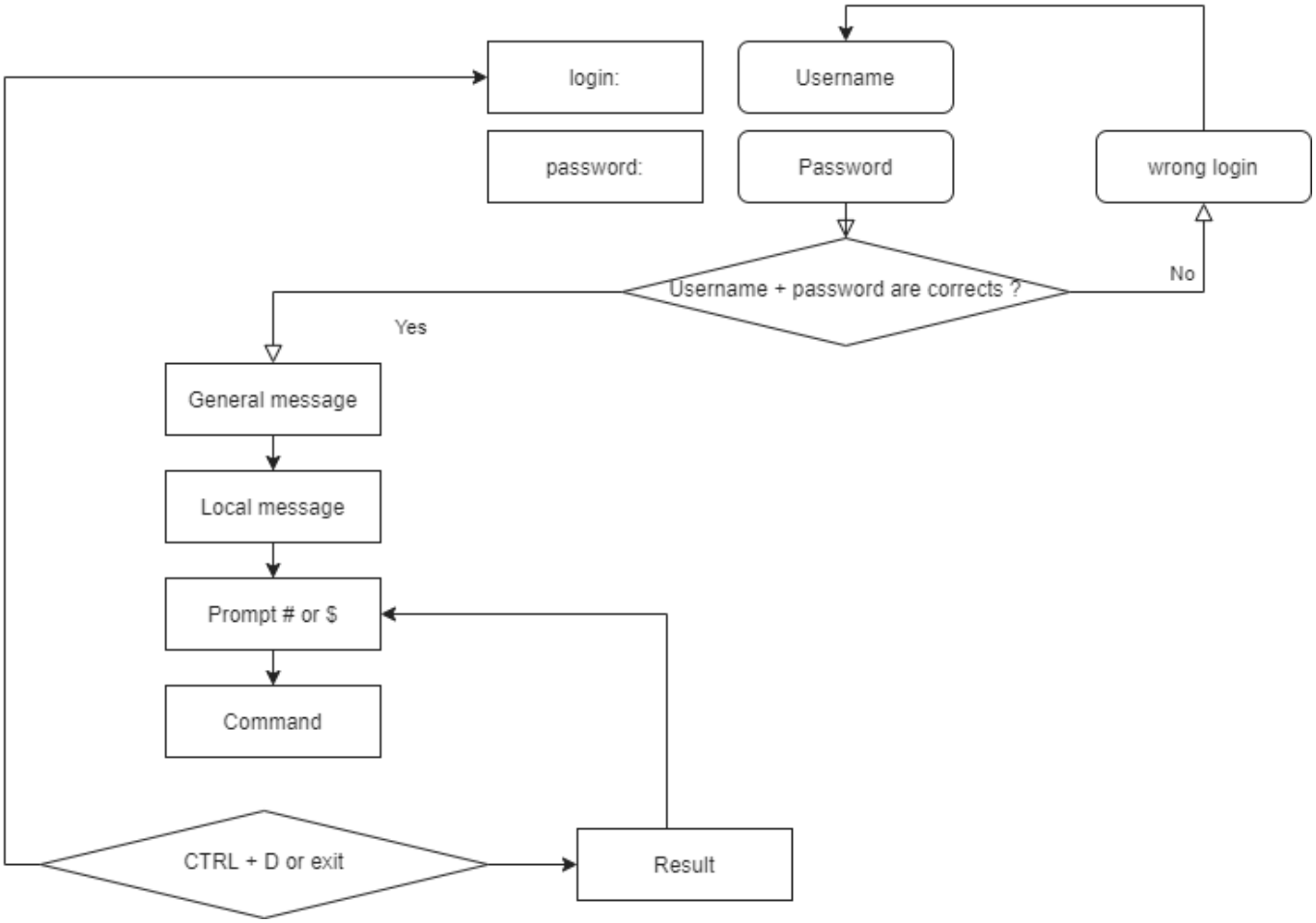
- sh, la shell standard POSIX
- csh, shell orientata ai comandi in C
- bash, Bourne-Again Shell, shell di Linux

Copyright © 2023 The Rocky Enterprise Software Foundation

3.5.2 Funzionalità

- Esecuzione del comando (verifica il comando dato e lo esegue).
- Reindirizzamento di Input/Output (restituisce i dati a un file invece di scriverli sullo schermo).
- Processo di connessione (gestisce la connessione dell'utente).
- Linguaggio di programmazione interpretato (che consente la creazione di script).
- Variabili d'ambiente (accesso a informazioni specifiche del sistema durante il funzionamento).

3.5.3 Principio



Copyright © 2023 The Rocky Enterprise Software Foundation

3.6 Verificare le proprie Conoscenze

✔ Un sistema operativo è un insieme di programmi per la gestione delle risorse disponibili di un computer:

Vero

Falso

✔ Il sistema operativo è necessario per:

Gestire la memoria fisica e virtuale

Consentire l'accesso diretto alle periferiche

Affidare la gestione dei compiti all'elaboratore

Raccogliere informazioni sui programmi utilizzati o in uso

✔ Tra queste personalità, quali hanno partecipato allo sviluppo di UNIX?

Linus Torvalds

Ken Thompson

Lionel Richie

Brian Kernighan

Andrew Stuart Tanenbaum

✔ La nazionalità originaria di Linus Torvalds, creatore del kernel Linux, è:

Svedese

Finlandese

Norvegese

Fiamminga

Francese

Copyright © 2023 The Rocky Enterprise Software Foundation

✔ Quale delle seguenti distribuzioni è la più antica:

Debian

Slackware

RedHat

Arch

✔ Il kernel di Linux è:

Multitasking

Multiutente

Multiprocessore

Multi-core

Multipiattaforma

Aperto

✔ Il software libero è necessariamente open-source?

Vero

Falso

✔ Il software Open-Source è necessariamente gratuito?

Vero

Falso

✔ Quale delle seguenti non è una shell:

Jason

Jason-Bourne shell (jbsh)

Bourne-Again shell (bash)

C shell (csh)

Korn shell (ksh)

Copyright © 2023 The Rocky Enterprise Software Foundation

4. Comandi per gli Utenti Linux

In questo capitolo imparerete i comandi di Linux e come utilizzarli.

Obiettivi : In questo capitolo, i futuri amministratori Linux impareranno come:

✔ **Muoversi** all'interno dell'albero di sistema.
✔ **Creare** un file di testo, **visualizzare** il suo contenuto e **modificarlo**.
✔ **Utilizzare** i comandi più utili di Linux.

🏁 **comandi utente**, **linux**

Conoscenza: ⭐
Complessità: ⭐

Tempo di lettura: 40 minuti

4.1 Generalità

I sistemi Linux attuali hanno utilità grafiche dedicate al lavoro di un amministratore. Tuttavia, è importante essere in grado di utilizzare l'interfaccia in modalità riga di comando per diversi motivi:

- La maggior parte dei comandi di sistema sono comuni a tutte le distribuzioni Linux, questo non è il caso degli strumenti grafici.
- Può accadere che il sistema non si avvii correttamente ma che un interprete di comando di backup rimanga accessibile.
- L'amministrazione remota viene eseguita dalla riga di comando con un terminale SSH.
- Per preservare le risorse del server, l'interfaccia grafica è installata o lanciata su richiesta.
- L'amministrazione è eseguita da scripts.

Copyright © 2023 The Rocky Enterprise Software Foundation

L'apprendimento di questi comandi consente all'amministratore di connettersi a un terminale Linux, di gestirne le risorse e i file, di identificare la stazione, il terminale e gli utenti collegati, ecc.

4.1.1 Gli utenti

L'utente di un sistema Linux è definito nel file `/etc/passwd`, da:

- Un **nome di login**, o più comunemente chiamato "login", che non può contenere spazi.
- Un identificatore numerico: **UID** (User Identifier).
- Un identificatore di gruppo: **GID** (Group Identifier).
- Un **interprete di comandi**, ad esempio una shell, che può essere diversa da un utente all'altro.
- Una **directory di connessione**, ad esempio la **directory home**.

In altri file l'utente sarà definito da:

- Una **password**, che verrà crittografata prima di essere memorizzata (`/etc/shadow`).
- Un **prompt dei comandi**, o accesso **prompt**, simbolizzato da un
- `#` per gli amministratori
- e da un `$` per gli altri utenti (`/etc/profile`).

A seconda della politica di sicurezza implementata sul sistema, la password dovrà contenere un certo numero di caratteri e soddisfare determinati requisiti di complessità.

Tra gli interpreti di comando esistenti, la **Bourne-Again Shell** (`/bin/bash`) è quella utilizzata più frequentemente. È assegnata per impostazione predefinita ai nuovi utenti. Per vari motivi, gli utenti avanzati di Linux possono scegliere shell alternative tra la Korn Shell (`ksh`), la C Shell (`csh`), etc.

La directory di accesso dell'utente è per convenzione memorizzata nella directory `/home` della workstation. Conterrà i dati personali dell'utente e i file di

Copyright © 2023 The Rocky Enterprise Software Foundation

configurazione delle sue applicazioni. Per impostazione predefinita, al login, la directory di accesso è selezionata come directory corrente.

Un'installazione di tipo workstation (con interfaccia grafica) avvia questa interfaccia sul terminale 1. Essendo Linux multiutente, è possibile connettere più utenti più volte, su diversi **terminali fisici** (TTY) o **terminali virtuali** (PTS). I terminali virtuali sono disponibili all'interno di un ambiente grafico. Un utente passa da un terminale fisico a un altro utilizzando ⎇ Alt + Fx dalla riga di comando, o utilizzando ^ Ctrl + ⎇ Alt + Fx.

4.1.2 La shell

Una volta che l'utente è collegato a una console, la shell visualizza il **prompt dei comandi**. Si comporta quindi come un ciclo infinito, ripetendo lo stesso schema a ogni istruzione inserita:

- Visualizza il prompt dei comandi.
- Lettura del comando.
- Analisi della sintassi.
- Sostituzione di caratteri speciali.
- Esecuzione del comando.
- Visualizza il prompt dei comandi.
- etc.

La sequenza di tasti ^ Ctrl + C è utilizzata per interrompere un comando in esecuzione.

L'uso di un comando segue generalmente questa sequenza:

```
comando [opzione(i)] [argomento(i)]
```

Il nome del comando è spesso in **minuscolo**.

Uno spazio separa ogni oggetto.

Copyright © 2023 The Rocky Enterprise Software Foundation

Le **opzioni abbreviate** iniziano con un trattino (`-l`), mentre le **opzioni lunghe** iniziano con due trattini (`--list`). Un doppio trattino (`--`) indica la fine dell'elenco delle opzioni.

È possibile raggruppare alcune opzioni brevi insieme:

```
ls -l -i -a
```

è equivalente a:

```
ls -lia
```

Dopo un'opzione possono esserci più argomenti:

```
$ ls -lia /etc /home /var
```

In letteratura, il termine "opzione" è equivalente al termine "parametro," che è più comunemente usato nella programmazione. Il lato opzionale di un'opzione o argomento è simboleggiata dall'inclusione in parentesi quadre `[` e `]`. Quando è possibile più di un'opzione, Una barra verticale chiamata "pipe" li separa `[a|e|i]`.

4.2 Comandi generali

4.2.1 comandi `apropos`, `whatis` e `man`

È impossibile per un amministratore a qualsiasi livello conoscere tutti i comandi e le opzioni in dettaglio. Solitamente è disponibile un manuale per tutti i comandi installati.

comando `apropos`

Il comando `apropos` ti consente di cercare per parola chiave all'interno di queste pagine dei manuali:

Opzioni	Descrizione
`-s`, `--sections list` o `--section list`	Limitato alle sezioni dei manuali.
`-a` o `--and`	Visualizza solo la voce corrispondente a tutte le parole chiave fornite.

Copyright © 2023 The Rocky Enterprise Software Foundation

Esempio:

```
$ apropos clear
clear (1)            - clear the terminal screen
clear_console (1)    - clear the console
clearenv (3)         - clear the environment
clearerr (3)         - check and reset stream status
clearerr_unlocked (3) - nonlocking stdio functions
feclearexcept (3)    - floating-point rounding and exception handling
fwup_clear_status (3) - library to support management of system firmware
updates
klogctl (3)          - read and/or clear kernel message ring buffer; set
console_loglevel
sgt-samegame (6)     - block-clearing puzzle
syslog (2)           - read and/or clear kernel message ring buffer; set
console_loglevel
timerclear (3)       - timeval operations
XClearArea (3)       - clear area or window
XClearWindow (3)     - clear area or window
XSelectionClearEvent (3) - SelectionClear event structure
```

Per trovare il comando che consentirà di cambiare la password di un account:

```
$ apropos --exact password  -a change
chage (1)            - change user password expiry information
passwd (1)           - change user password
```

comando `whatis`

Il comando `whatis` visualizza la descrizione del comando passata come argomento:

```
whatis clear
```

Esempio:

```
$ whatis clear
clear (1)            - clear the terminal screen
```


Copyright © 2023 The Rocky Enterprise Software Foundation

comando `man`

Una volta trovato con `apropos` o `whatis`, il manuale è letto da `man` ("Man è tuo amico"). Questo set di manuali è diviso in 8 sezioni, raggruppando le informazioni per argomento, la sezione predefinita è la 1:

1. Programmi o comandi eseguibili.
2. Chiamate di sistema (funzioni date dal kernel).
3. Chiamate di libreria (funzioni date dalla libreria).
4. File speciali (di solito si trovano in /dev).
5. Formati di file e convenzioni (file di configurazione come etc/passwd).
6. Giochi (come le applicazioni basate sui personaggi).
7. Varie (es. man (7)).
8. Comandi di amministrazione del sistema (di solito solo per root).
9. Routine del Kernel (non standard).

È possibile accedere alle informazioni su ciascuna sezione digitando `man x intro`, dove `x` è il numero della sezione.

Il comando:

```
man passwd
```

dirà all'amministratore le opzioni, etc, del comando passwd. Mentre:

```
man 5 passwd
```

lo informerà sui file relativi al comando.

Navigare nel manuale con le frecce `↑ Up` e `↓ Down`. Uscire dal manuale premendo il tasto `Q`.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.2.2 comando `shutdown`

Il comando `shutdown` consente di **spegnere elettronicamente** un server Linux, immediatamente o dopo un certo periodo di tempo.

```
shutdown [-h] [-r] time [message]
```

Specificare l'ora di spegnimento nel formato `hh:mm` per un'ora precisa, o `+mm` per un ritardo in minuti.

Per forzare un arresto immediato, usa la parola `now` al posto del tempo. In questo caso, il messaggio opzionale non viene inviato agli altri utenti del sistema.

Esempi:

```
[root]# shutdown -h 0:30 "Server shutdown at 0:30"
[root]# shutdown -r +5
```

Opzioni:

Opzioni	Osservazioni
`-h`	Arresta il sistema elettronicamente.
`-r`	Riavvia il sistema.

4.2.3 comando `history`

Il comando `history` visualizza la cronologia dei comandi inseriti dall'utente.

I comandi sono memorizzati nel file `.bash_history` nella directory di accesso dell'utente.

Esempio di un comando history

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ history
147 man ls
148 man history
```

Opzioni	Commenti
`-w`	Scrive la cronologia corrente nel file della cronologia
`-c`	Cancella la cronologia della sessione corrente (ma non il contenuto del file `.bash_history`).

- Manipolazione della cronologia:

Per manipolare la history, i seguenti comandi immessi dal prompt dei comandi permetteranno di:

Chiavi	Funzione
`!` + `!`	Richiama l'ultimo comando eseguito.
`!` + `N`	Richiama il comando per il suo numero nell'elenco.
`!` + `string`	Richiama il comando più recente che inizia con la stringa.
`↑ Up`	Naviga nella cronologia andando indietro nel tempo a partire dal comando più recente.
`↓ Down`	Naviga nella cronologia andando avanti nel tempo.

4.2.4 Autocompletamento

Il completamento automatico è di grande aiuto.

- Completa i comandi, i percorsi inseriti o i nomi dei file.
- Una pressione del tasto `Tab ⇥` completa la voce nel caso di una soluzione singola.
- Nel caso di più soluzioni, premere `Tab ⇥` una seconda volta per visualizzare le opzioni.

Se premendo due volte il tasto `Tab ⇥` non vengono presentate opzioni, non c'è soluzione al completamento attuale.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.3 Visualizzazione e Identificazione

4.3.1 commando `clear`

Il comando `clear` cancella il contenuto della schermata del terminale. Più precisamente, sposta la visualizzazione in modo che il prompt dei comandi si trovi in cima allo schermo, sulla prima riga.

Su un terminale fisico, il display sarà permanentemente nascosto, mentre in un'interfaccia grafica, una barra di scorrimento permetterà di tornare indietro nella cronologia del terminale virtuale.

Suggerimento

`^ Ctrl` + `L` avrà lo stesso effetto del comando `clear`

4.3.2 comando `echo`

Il comando `echo` è usato per visualizzare una stringa di caratteri.

Questo comando è più comunemente usato negli script amministrativi per informare l'utente durante l'esecuzione.

L'opzione `-n` indica nessuna stringa di output newline (di default, stringa di output newline).

```
shell > echo -n "123";echo "456"
123456

shell > echo "123";echo "456"
123
456
```

Per vari motivi, allo sviluppatore dello script potrebbe essere necessario utilizzare sequenze speciali (a partire da un carattere `\`). In questo caso, sara usata l'opzione `-e`, che consentirà l'interpretazione della sequenza.

Copyright © 2023 The Rocky Enterprise Software Foundation

Tra le sequenze usate frequentemente, possiamo menzionare:

Sequenza	Risultato
`\a`	Invia un bip sonoro
`\b`	Indietro
`\n`	Aggiunge una interruzione di linea
`\t`	Aggiunge un tab orizzontale
`\v`	Aggiunge un tab verticale

4.3.3 comando `date`

Il comando `date` visualizza la data e l'ora. Il comando ha la seguente sintassi:

```
date [-d yyyyMMdd] [format]
```

Esempi:

```
$ date
Mon May 24 16:46:53 CEST 2021
$ date -d 20210517 +%j
137
```

In quest'ultimo esempio, l'opzione `-d` visualizza una determinata data. L'opzione `+%j` formatta questa data per mostrare solo il giorno dell'anno.

Attenzione

Il formato di una data può cambiare in base al valore della lingua definito nella variabile ambientale '$LANG'.

Copyright © 2023 The Rocky Enterprise Software Foundation

La visualizzazione della data può seguire i seguenti formati:

Opzione	Formato
`+%A`	Nome completo del giorno della settimana della località (ad esempio, Domenica)
`+%B`	Nome completo del mese della località (ad esempio, Gennaio)
`+%c`	Data e ora di Locale (ad esempio, Gio Mar 3 23:05:25 2005)
`+%d`	Giorno del mese (ad es. 01)
`+%F`	Data nel formato `YYYY-MM-DD`
`+%G`	Anno
`+%H`	Ora (00..23)
`+%j`	Giorno dell'anno (001..366)
`+%m`	Numero del mese (01..12)
`+%M`	Minuto (00..59)
`+%R`	Tempo nel formato `hh:mm`
`+%s`	Secondi dal 1° gennaio 1970
`+%S`	Secondo (00..60)
`+%T`	Tempo nel formato `hh:mm:ss`
`+%u`	Giorno della settimana (`1` per Lunedì)
`+%V`	Numero della settimana (`+%V`)
`+%x`	Data nel formato `GG/MM/AAAA`

Il comando `date` consente anche di modificare la data e l'ora del sistema. In questo caso, verrà utilizzata l'opzione `-s`.

```
[root]# date -s "2021-05-24 10:19"
```

Il formato da utilizzare usando l'opzione `-s` è il seguente:

```
date -s "yyyy-MM-dd hh:mm[:ss]"
```

4.3.4 comando `id`, `who` e `whoami`

Il comando `id` viene utilizzato per visualizzare informazioni su utenti e gruppi. Per impostazione predefinita, non viene aggiunto alcun parametro utente e vengono visualizzate le informazioni dell'utente e del gruppo attualmente connessi

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ id rockstar
uid=1000(rockstar) gid=1000(rockstar) groups=1000(rockstar),10(wheel)
```

Le opzioni `-g`, `-G`, `-n` e `-u` visualizzano il gruppo principale GID, sottogruppo GIDs, nomi al posto di identificatori numerici, e l'UID dell'utente.

Il comando `whoami` visualizza il login dell'utente corrente.

Il solo comando `who` visualizza i nomi degli utenti connessi:

```
$ who
rockstar tty1   2021-05-24 10:30
root     pts/0  2021-05-24 10:31
```

Dal momento che Linux è multi-utente, è possibile che più sessioni siano aperte sulla stessa stazione, sia fisicamente che attraverso la rete. È interessante sapere quali utenti sono connessi, se non solo per comunicare con loro inviando messaggi.

- tty: rappresenta un terminale.
- pts/: rappresenta una console virtuale in un ambiente grafico con il numero dopo aver rappresentato l'istanza della console virtuale (0, 1, 2...)

L'opzione `-r` visualizza anche il runlevel (vedi capitolo "startup").

4.4 Albero Dei File

In Linux, l'albero dei file è un albero invertito, chiamato **albero gerarchico singolo**, la cui radice è la directory `/`.

La **directory corrente** è la directory in cui si trova l'utente.

La **directory di connessione** è la directory di lavoro associata all'utente. Le directory di accesso sono, per impostazione predefinita, memorizzate nella directory `/home`.

Quando l'utente accede, la directory corrente è la directory di accesso.

Copyright © 2023 The Rocky Enterprise Software Foundation

Un **percorso assoluto** fa riferimento ad un file dalla radice attraversando l'intero albero fino al livello del file:

- `/home/groupA/alice/file`

Un **percorso relativo** fa riferimento allo stesso file attraversando l'intero albero dalla directory corrente:

- `../alice/file`

Nell'esempio sopra, il "`..`" si riferisce alla directory principale della directory corrente.

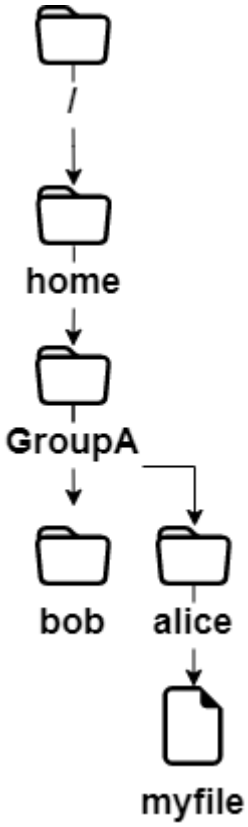
Una directory, anche se è vuota, conterrà necessariamente almeno **due riferimenti**:

- `.`: riferimento a se stessa.
- `..`: riferimento alla directory principale della directory corrente.

Un percorso relativo può quindi iniziare con `./` o `../`. Quando il percorso relativo si riferisce a una sottodirectory o ad un file nella directory corrente, il `./` è spesso omesso. L'inserimento del riferimento `./` sarà veramente richiesto solo per l'esecuzione di un file eseguibile.

Gli errori nei percorsi possono causare molti problemi: dalla creazione di cartelle o file nei luoghi sbagliati, alle eliminazioni involontarie, ecc. È quindi fortemente raccomandato di utilizzare il completamento automatico quando si immettono i percorsi.

Copyright © 2023 The Rocky Enterprise Software Foundation



Nell'esempio precedente, si vuole fornire la posizione del file `myfile` dalla directory di bob.

- In un **percorso assoluto**, la directory corrente non ha importanza. Iniziamo dalla radice e scendiamo fino alle directory `home`, `groupA`, `alice` e infine il file `myfile`: `/home/groupA/alice/myfile`.
- In un **percorso relativo**, il nostro punto di partenza è la directory corrente `bob`, saliamo di un livello con `..` (i.e., nella directpry `groupA`), poi giù nella directory di alice, e infine il file `myfile`: `../alice/myfile`.

4.4.1 comando `pwd`

Il comando `pwd` (Print Working Directory) visualizza il percorso assoluto della directory corrente.

```
$ pwd
/home/rockstar
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Per utilizzare un percorso relativo per fare riferimento a un file o a una directory, o per usare il comando `cd` per spostarsi in un'altra directory, è necessario conoscere la sua posizione nell'albero dei file.

A seconda del tipo di shell e dei diversi parametri del suo file di configurazione, il prompt del terminale (noto anche come prompt dei comandi) visualizzerà il percorso assoluto o relativo della directory corrente.

4.4.2 comando `cd`

Il comando `cd` (Cambia Directory) ti permette di cambiare la directory corrente -- in altre parole, di spostarti attraverso l'albero.

```
$ cd /tmp
$ pwd
/tmp
$ cd ../
$ pwd
/
$ cd
$ pwd
/home/rockstar
```

Come puoi vedere nell'ultimo esempio sopra, il comando `cd` senza argomenti sposta la directory corrente alla `home directory`.

4.4.3 comando `ls`

Il comando `ls` visualizza il contenuto di una directory.

```
ls [-a] [-i] [-l] [directory1] [directory2] […]
```

Esempio:

```
$ ls /home
.    ..    rockstar
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Le opzioni principali del comando `ls` sono:

Opzione	Informazione
`-a`	Visualizza tutti i file, anche quelli nascosti. I file nascosti in Linux sono quelli che iniziano con un `.`.
`-i`	Visualizza i numeri di inode.
`-l`	Utilizza un formato di elenco lungo, cioè ogni riga visualizza informazioni di formato lungo per un file o una directory.

Il comando `ls`, tuttavia, ha molte opzioni (vedi `man`):

Opzione	Informazione
`-d`	Visualizza le informazioni di una directory invece di elencare i suoi contenuti.
`-g`	Come l'opzione -l, ma non elenca il proprietario.
`-h`	Visualizza le dimensioni dei file nel formato più appropriato (byte, kilobyte, megabyte, gigabyte, ...). `h` stà per Human Readable. Deve essere utilizzato con l'opzione -l.
`-s`	Visualizza la dimensione assegnata di ogni file, in blocchi. Nel comando `ls`, la dimensione predefinita di un singolo blocco è di 1024 byte. Nel sistema operativo GNU/Linux, il "blocco" è la più piccola unità di memoria del file system e, in generale, un blocco equivale a 4096 byte. Nel sistema operativo Windows, prendendo come esempio il file system NTFS, l'unità di memorizzazione più piccola è chiamata "Cluster". La definizione del nome dell'unità di archiviazione minima può variare a seconda dei diversi file system.
`-A`	Visualizza tutti i file nella directory tranne `.` e `..`
`-R`	Visualizza il contenuto delle sottodirectory in modo ricorsivo.
`-F`	Visualizza il tipo di file. Stampa un `/` per una directory, `*` per gli eseguibili, `@` per un collegamento simbolico, e niente per un file di testo.
`-X`	Ordina i file in base alle loro estensioni.

- Descrizione delle colonne generate dall'esecuzione del comando `ls -lia`:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ ls -lia /home
78489 drwx------ 4 rockstar rockstar 4096 25 oct. 08:10 rockstar
```

Valore	Informazione
`78489`	Numero di inode.
`drwx------`	Tipo di file (`d`) e permessi (`rwx------`).
`4`	Numero di sottodirectory. (`.` e `..` incluse). Per un file, rappresenta il numero di collegamenti diretti e 1 rappresenta se stesso.
`rockstar`	Proprietà dell'utente.
`rockstar`	Proprietà del gruppo.
`4096`	Per i file, mostra la dimensione del file. Per le directory, mostra il valore fisso di 4096 byte occupati dal nome del file. Per calcolare la dimensione totale di una directory, usa `du -sh rockstar/`
`25 oct. 08:10`	Ultima data di modifica.
`rockstar`	Il nome del file (o directory).

Nota

Gli **Alias** sono spesso già posizionati nelle distribuzioni comuni.

Questo è il caso dell'alias `ll` :

```
alias ll='ls -l --color=auto'
```

Il comando `ls` ha molte opzioni. Ecco alcuni esempi avanzati di utilizzo:

- Elenca i file in `/etc` in base all'ultima modifica:

```
$ ls -ltr /etc
total 1332
-rw-r--r--.  1 root root    662 29 may   2021 logrotate.conf
-rw-r--r--.  1 root root    272 17 may.   2021 mailcap
-rw-------.  1 root root    122 12 may.  2021 securetty
...
-rw-r--r--.  2 root root     85 18 may.  17:04 resolv.conf
-rw-r--r--.  1 root root     44 18 may.  17:04 adjtime
-rw-r--r--.  1 root root    283 18 may.  17:05 mtab
```

- Elenca i file `/var` di dimensioni superiori a 1 megabyte ma inferiori a 1 gigabyte. L'esempio qui riportato utilizza i comandi avanzati `grep` con le espressioni regolari. I novizi non devono lottare troppo, ci sarà un tutorial speciale per introdurre queste espressioni regolari in futuro.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ ls -lhR /var/ | grep ^\- | grep -E "[1-9]*\.[0-9]*M"
...
-rw-r--r--. 1 apache apache 1.2M 10 may.  13:02 XB RiyazBdIt.ttf
-rw-r--r--. 1 apache apache 1.2M 10 may.  13:02 XB RiyazBd.ttf
-rw-r--r--. 1 apache apache 1.1M 10 may.  13:02 XB RiyazIt.ttf
...
```

Naturalmente, si consiglia vivamente di utilizzare il comando `find`.

```
find /var -size +1M -a -size -1024M  -a -type f  -exec ls -lh {} \;
```

- Mostra i permessi di una cartella:

Per conoscere i permessi di una cartella, ad esempio `/etc`, il seguente comando **non** sarebbe appropriato:

```
$ ls -l /etc
total 1332
-rw-r--r--.  1 root root     44 18 nov.  17:04 adjtime
-rw-r--r--.  1 root root   1512 12 janv.  2010 aliases
-rw-r--r--.  1 root root  12288 17 nov.  17:41 aliases.db
drwxr-xr-x.  2 root root   4096 17 nov.  17:48 alternatives
...
```

Il comando precedente visualizzerà il contenuto della cartella (all'interno) per impostazione predefinita. Per la cartella stessa, è possibile utilizzare l'opzione `-d`.

```
ls -ld /etc
drwxr-xr-x. 69 root root 4096 18 nov.  17:05 /etc
```

- Ordina per dimensione del file, prima il più grande:

```
ls -lhS
```

- formato ora/data con `-l`:

```
$ ls -l --time-style="+%Y-%m-%d %m-%d %H:%M" /
total 12378
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
dr-xr-xr-x. 2 root root 4096 2014-11-23 11-23 03:13 bin
dr-xr-xr-x. 5 root root 1024 2014-11-23 11-23 05:29 boot
```

- Aggiungere la barra *trailing slash* alla fine delle cartelle:

Per impostazione predefinita, il comando `ls` non visualizza l'ultima slash di una cartella. In alcuni casi, come per gli script, ad esempio, è utile visualizzarla:

```
$ ls -dF /etc
/etc/
```

- Nascondi alcune estensioni:

```
ls /etc --hide=*.conf
```

4.4.4 comando `mkdir`

Il comando `mkdir` crea una directory o un albero di directory.

```
mkdir [-p] directory [directory] [...]
```

Esempio:

```
mkdir /home/rockstar/work
```

La directory "rockstar" deve essere presente per creare la directory "work".

Altrimenti, si deve usare l'opzione `-p`. L'opzione `-p` crea le directory genitore se queste non esistono.

Pericolo

Non è consigliabile usare i nomi dei comandi Linux come directory o nomi di file.

4.4.5 comando `touch`

Il comando `touch` modifica il timestamp di un file o crea un file vuoto se il file non esiste.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
touch [-t date] file
```

Esempio:

```
touch /home/rockstar/myfile
```

Opzione	Informazione
`-t date`	Cambia la data di ultima modifica del file con la data specificata.

Formato data: `[AAAA]MMJJhhmm[ss]`

Suggerimento

Il comando `touch` è usato principalmente per creare un file vuoto, ma può essere utile, ad esempio, per i backup incrementali o differenziali. Infatti, l'unico effetto dell'esecuzione di un `touch` su un file sarà quello di forzarne il salvataggio durante il backup successivo.

4.4.6 comando `rmdir`

Il comando `rmdir` elimina una directory vuota.

Esempio:

```
rmdir /home/rockstar/work
```

Opzione	Informazione
`-p`	Rimuove la cartella genitore o le cartelle fornite, se sono vuote.

Suggerimento

Per eliminare una cartella non vuota e il suo contenuto, utilizzare il comando `rm`.

4.4.7 comando `rm`

Il comando `rm` elimina un file o una directory.

```
rm [-f] [-r] file [file] [...]
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Pericolo

Qualsiasi eliminazione di un file o directory è definitiva.

Opzioni	Informazione
`-f`	Non chiedere se eliminare.
`-i`	Chiedi se cancellare.
`-r`	Elimina una cartella e cancella ricorsivamente le sue sottocartelle.

Nota

Il comando `rm` non chiede conferma quando si eliminano i file. Tuttavia, con una distribuzione Red Hat/Rocky, `rm` chiede la conferma della cancellazione perché il comando `rm` è un `alias` del comando `rm -i`. Non sorprenderti se su un'altra distribuzione, come Debian, ad esempio, non ottieni una richiesta di conferma.

L'eliminazione di una cartella con il comando `rm`, sia che la cartella sia vuota o meno, richiede l'aggiunta dell'opzione `-r`.

La fine delle opzioni è segnalata alla shell da un doppio trattino `--`.

Nell'esempio:

```
$ >-hard-hard # To create an empty file called -hard-hard
hard-hard
[CTRL+C] To interrupt the creation of the file
$ rm -f -- -hard-hard
```

Il nome del file hard-hard inizia con un `-`. Senza l'uso del `--` la shell avrebbe interpretato il `-d` in `-hard-hard` come un'opzione.

4.4.8 command `mv`

Il comando `mv` muove e rinomina un file.

```
mv file [file ...] destination
```

Esempi:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
mv /home/rockstar/file1 /home/rockstar/file2

mv /home/rockstar/file1 /home/rockstar/file2 /tmp
```

Opzioni	Informazione
`-f`	Non chiedere conferma per la sovrascrittura del file di destinazione.
`-i`	Richiedere conferma per la sovrascrittura del file di destinazione (default).

Alcuni casi concreti ti aiuteranno a capire le difficoltà che possono sorgere:

```
mv /home/rockstar/file1 /home/rockstar/file2
```

Rinomina `file1` in `file2`. Se `file2` esiste già, sostituisci il contenuto del file con `file1`.

```
mv /home/rockstar/file1 /home/rockstar/file2 /tmp
```

Sposta `file1` e `file2` nella cartella `/tmp`.

```
mv file1 /repexist/file2
```

Sposta `file1` in `repexist` e lo rinomina `file2`.

```
mv file1 file2
```

`file1` è rinominato con `file2`.

```
mv file1 /repexist
```

Se esiste la cartella di destinazione, `file1` viene spostato in `/repexist`.

```
mv file1 /wrongrep
```

Se la directory di destinazione non esiste, `file1` viene rinominato in `wrongrep` nella directory principale.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.4.9 comando `cp`

Il comando `cp` copia un file.

```
cp file [file ...] destination
```

Esempio:

```
cp -r /home/rockstar /tmp
```

Opzioni	Informazione
`-i`	Richiesta di conferma per la sovrascrittura (default).
`-f`	Non chiedere conferma per la sovrascrittura del file di destinazione.
`-p`	Mantiene il proprietario, le autorizzazioni e il timestamp del file copiato.
`-r`	Copia una directory con i suoi file e sottodirectory.
`-s`	Crea un collegamento simbolico invece di copiare.

```
cp file1 /repexist/file2
```

`file1` viene copiato in `/repexist` con il nome `file2`.

```
cp file1 file2
```

`file1` viene copiato come `file2` in questa cartella.

```
cp file1 /repexist
```

Se esiste la directory di destinazione, `file1` viene copiato in `/repexist`.

```
cp file1 /wrongrep
```

Se la directory di destinazione non esiste, `file1` è copiato sotto il nome `wrongrep` nella directory principale.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.5 Visualizzazione

4.5.1 comando `file`

Il comando `file` visualizza il tipo di un file.

```
file file1 [files]
```

Esempio:

```
$ file /etc/passwd /etc
/etc/passwd:    ASCII text
/etc:        directory
```

4.5.2 comando `more`

Il comando `more` visualizza il contenuto di uno o più file schermata per schermata.

```
more file1 [files]
```

Esempio:

```
$ more /etc/passwd
root:x:0:0:root:/root:/bin/bash
...
```

Usando il tasto `Enter ⏎`, lo spostamento è linea per linea. Usando il tasto `Space`, lo spostamento è pagina per pagina. `/text` Ti consente di cercare la corrispondenza nel file.

4.5.3 comando `less`

Il comando `less` visualizza il contenuto di uno o più file. Il comando `less` è interattivo e ha i propri comandi per l'uso.

```
less file1 [files]
```


Copyright © 2023 The Rocky Enterprise Software Foundation

I comandi specifici per `less` sono:

Comando	Azione
H	Aiuto.
↑ Up ↓ Down → Right ← Left	Sposta su, giù di una linea, o a destra e sinistra.
Enter ⏎	Sposta giù di una riga.
Space	Sposta giù di una pagina.
⇞ Page Up e ⇟ Page Down	Sposta su o giù di una pagina.
g e G	Passa alla prima e all'ultima pagina
`/text`	Cerca il testo.
Q	Chiude il comando `less`.

4.5.4 comando `cat`

Il comando `cat` concatena il contenuto di più file e visualizza il risultato sull'output standard.

```
cat file1 [files]
```

Esempio 1 - Visualizzazione del contenuto di un file in output standard:

```
cat /etc/passwd
```

Esempio 2 - Visualizzazione del contenuto di più file in output standard:

```
cat /etc/passwd /etc/group
```

Esempio 3 - Combinare il contenuto di più file in un unico file utilizzando il reindirizzamento dell'output:

```
cat /etc/passwd /etc/group > usersAndGroups.txt
```

Esempio 4 - Visualizzazione della numerazione di linea:

```
$ cat -n /etc/profile
     1    # /etc/profile: system-wide .profile file for the Bourne shell
(sh(1))
     2    # and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
     3
     4    if [ "`id -u`" -eq 0 ]; then
     5      PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
     6    else
…
```

Example 5 - Mostra la numerazione delle righe non vuote:

```
$ cat -b /etc/profile
     1    # /etc/profile: system-wide .profile file for the Bourne shell
(sh(1))
     2    # and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

     3    if [ "`id -u`" -eq 0 ]; then
     4      PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
     5    else
…
```

4.5.5 comando `tac`

Il comando `tac` fa quasi il contrario del comando `cat`. Visualizza il contenuto di un file a partire dalla fine (che è particolarmente interessante per la lettura dei log!).

Esempio: Visualizza un file di log visualizzando prima l'ultima riga:

```
[root]# tac /var/log/messages | less
```

4.5.6 comando `head`

Il comando `head` visualizza l'inizio di un file.

```
head [-n x] file
```

Opzione	Descrizione
`-n x`	Visualizzare le prime `x` righe del file

Per impostazione predefinita (senza l'opzione `-n`), il comando `head` visualizzerà le prime 10 righe del file.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.5.7 comando `tail`

Il comando `tail` visualizza la fine di un file.

```
tail [-f] [-n x] file
```

Opzione	Descrizione
`-n x`	Visualizza le ultime `x` righe del file
`-f`	Visualizza le modifiche al file in tempo reale

Esempio:

```
tail -n 3 /etc/passwd
sshd:x:74:74:Privilege-separeted sshd:/var/empty /sshd:/sbin/nologin
tcpdump::x:72:72::/:/sbin/nologin
user1:x:500:500:grp1:/home/user1:/bin/bash
```

Con l'opzione `-f`, le informazioni di modifica del file verranno sempre emesse a meno che l'utente non esca dallo stato di monitoraggio con `^ Ctrl` + `C`. Questa opzione è molto utilizzata per tracciare i file di log (i registri) in tempo reale.

Senza l'opzione `-n`, il comando `tail` mostra le ultime 10 righe del file.

4.5.8 comando `sort`

Il comando `sort` ordina le linee di un file.

Permette di ordinare il risultato di un comando o il contenuto di un file in un determinato ordine, numerico, alfabetico, per dimensione (KB, MB, GB) o in ordine inverso.

```
sort [-k] [-n] [-u] [-o file] [-t] file
```

Esempio:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ sort -k 3,4 -t ":" -n /etc/passwd
root:x:0:0:root:/root:/bin/bash
adm:x:3:4:adm:/var/adm/:/sbin/nologin
```

Opzione	Descrizione
`-k`	Specifica le colonne da separare. È possibile specificare più colonne.
`-n`	Richiede un ordinamento numerico.
`-o file`	Salva l'ordinamento nel file specificato.
`-t`	Specificare un delimitatore, che richiede che i contenuti del file corrispondente siano contenuti di colonne regolarmente delimitate, altrimenti non possono essere ordinati correttamente.
`-r`	Inverte l'ordine del risultato. Usato insieme all'opzione `-n` per ordinare dal più grande al più piccolo.
`-u`	Rimuovi i duplicati dopo l'ordinamento. Equivalente a `sort file uniq`.

Il comando `sort` ordina il file solo sullo schermo. Il file non è modificato dall'ordinamento. Per salvare l'ordinamento, utilizzare l'opzione `-o` o un reindirizzamento dell'output `>`.

Per impostazione predefinita, i numeri sono ordinati in base al loro carattere. Pertanto, "110" sarà prima di "20", che a sua volta sarà prima di "3". L'opzione `-n` deve essere specificata in modo che i blocchi di caratteri numerici siano ordinati in base al loro valore.

Il comando `sort` inverte l'ordine dei risultati, con l'opzione `-r`:

```
$ sort -k 3 -t ":" -n -r /etc/passwd
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
```

In questo esempio, il comando `sort` ordinerà il contenuto del file `/etc/passwd` dal più grande uid (identificatore utente) al più piccolo.

Alcuni esempi avanzati di utilizzazione del comando `sort`:

- Mischiando i valori

Il comando `sort` permette anche di mescolare i valori con l'opzione `-R`:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
sort -R /etc/passwd
```

- Ordinamento degli indirizzi IP

Un amministratore di sistema si trova ben presto di fronte all'elaborazione di indirizzi IP dai log dei suoi servizi, come SMTP, VSFTP o Apache. Questi indirizzi vengono tipicamente estratti con il comando `cut`.

Ecco un esempio con il file `dns-client.txt`:

```
192.168.1.10
192.168.1.200
5.1.150.146
208.128.150.98
208.128.150.99
```

```
$ sort -nr dns-client.txt
208.128.150.99
208.128.150.98
192.168.1.200
192.168.1.10
5.1.150.146
```

- Ordinamento dei file mediante la rimozione dei duplicati

Il comando `sort` sa come rimuovere i duplicati dall'output del file utilizzando l'opzione `-u`.

Ecco un esempio con il file `colours.txt`:

```
Red
Green
Blue
Red
Pink
```

```
$ sort -u colours.txt
Blue
Green
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
Pink
Red
```

- Ordinamento dei file in base alle dimensioni

Il comando `sort` sa come riconoscere le dimensioni dei file, da comandi come `ls` con l'opzione `-h`.

Ecco un esempio con il file `size.txt`:

```
1.7G
18M
69K
2.4M
1.2M
4.2G
6M
124M
12.4M
4G
```

```
$ sort -hr size.txt
4.2G
4G
1.7G
124M
18M
12.4M
6M
2.4M
1.2M
69K
```

4.5.9 comando `wc`

Il comando `wc` conteggia il numero di righe, parole e/o byte di un file.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
wc [-l] [-m] [-w] file [files]
```

Opzione	Descrizione
`-c`	Conta il numero di byte.
`-m`	Conta il numero di caratteri.
`-l`	Conta il numero di linee.
`-w`	Conta il numero di parole.

4.6 Ricerca

4.6.1 comando `find`

Il comando `find` cerca la posizione di file o cartelle.

```
find directory [-name name] [-type type] [-user login] [-date date]
```

Poiché le opzioni del comando `find` sono numerose, è meglio fare riferimento al comando `man`.

Se la cartella di ricerca non è specificata, il comando `find` effettuerà la ricerca dalla cartella corrente.

Opzione	Descrizione
`-perm permissions`	Cerca i file in base ai loro permessi.
`-size size`	Ricerca dei file in base alle dimensioni.

4.6.2 opzione `-exec` del comando `find`

È possibile utilizzare l'opzione `-exec` del comando `find` per eseguire un comando su ogni riga del risultato:

```
find /tmp -name *.txt -exec rm -f {} \;
```

Il comando precedente cerca tutti i file nella cartella `/tmp` denominati `*.txt` e li elimina.

Copyright © 2023 The Rocky Enterprise Software Foundation

Comprendere l'opzione `-exec`

Nell'esempio precedente, il comando `find` costruirà una stringa che rappresenta il comando da eseguire.

Se il comando `find` trova tre file chiamati `log1.txt`, `log2.txt` e `log3.txt`, allora il comando `find` costruirà la stringa sostituendo le parentesi nella stringa `rm -f {} \;` con uno dei risultati della ricerca, e lo farà tante volte quanti sono i risultati.

Questo ci darà:

```
rm -f /tmp/log1.txt ; rm -f /tmp/log2.txt ; rm -f /tmp/log3.txt ;
```

Il carattere `;` è un carattere speciale della shell che deve essere protetto da `\` per evitare che venga interpretato troppo presto dal comando `find` (e non nel `-exec`).

Suggerimento

`$ find /tmp -name *.txt -delete` fa la stessa cosa.

4.6.3 comando `whereis`

Il comando `whereis` ricerca i file relativi a un comando.

```
whereis [-b] [-m] [-s] command
```

Esempio:

```
$ whereis -b ls
ls: /bin/ls
```

Opzione	Descrizione
`-b`	Esegue la ricerca solo del file binario.
`-m`	Ricerca solo per le pagine man.
`-s`	Ricerca solo per file sorgente.

4.6.4 command `grep`

Il comando `grep` cerca una stringa in un file.

```
grep [-w] [-i] [-v] "string" file
```

Esempio:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ grep -w "root:" /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

Opzione	Descrizione
`-i`	Ignora le maiuscole della stringa cercata.
`-v`	Esclude le linee contenenti la stringa.
`-w`	Cerca la parola esatta.

Il comando `grep` restituisce la riga completa contenente la stringa cercata.

- Il carattere speciale `^` viene utilizzato per cercare una stringa all'inizio di una riga.
- Il carattere speciale `$` cerca una stringa alla fine di una riga.

```
grep -w "^root" /etc/passwd
```

Nota

Questo comando è molto potente e si consiglia di consultarne il manuale. Ha molti derivati.

È possibile cercare una stringa in un albero di file con l'opzione `-R`.

```
grep -R "Virtual" /etc/httpd
```

4.6.5 Meta-caratteri (wildcards)

I Meta-caratteri sostituiscono uno o più caratteri (o anche un'assenza di caratteri) durante una ricerca. Questi meta-caratteri sono anche noti come wildcards.

Possono essere combinati.

Il carattere `*` sostituisce una stringa composta da qualsiasi carattere. Il carattere `*` può anche rappresentare un'assenza di caratteri.

```
$ find /home -name "test*"
/home/rockstar/test
/home/rockstar/test1
/home/rockstar/test11
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
/home/rockstar/tests
/home/rockstar/test362
```

I Meta-caratteri consentono ricerche più complesse sostituendo tutto o parte di una parola. Sostituiscono semplicemente le incognite con questi caratteri speciali.

Il carattere `?` sostituisce un singolo carattere, qualunque esso sia.

```
$ find /home -name "test?"
/home/rockstar/test1
/home/rockstar/tests
```

Le parentesi quadre `[` e `]` sono usate per specificare i valori che un singolo carattere può assumere.

```
$ find /home -name "test[123]*"
/home/rockstar/test1
/home/rockstar/test11
/home/rockstar/test362
```

Nota

Circondare sempre le parole contenenti metacaratteri con `"` per evitare che vengano sostituite dai nomi dei file che soddisfano i criteri.

Attenzione

Non confondete i meta-caratteri della shell con i meta-caratteri delle espressioni regolari. Il comando `grep' utilizza meta-caratteri delle espressioni regolari.

4.7 Reindirizzamenti e pipes

4.7.1 Standard input e output

Sui sistemi UNIX e Linux, ci sono tre flussi standard. Consentono ai programmi, attraverso la libreria `stdio.h`, di inviare e ricevere informazioni.

Questi flussi sono chiamati canale X o descrittore di file X.

Copyright © 2023 The Rocky Enterprise Software Foundation

Per impostazione predefinita:

- la tastiera è il dispositivo di input per il canale 0, chiamato **stdin** ;
- lo schermo è il dispositivo di uscita per i canali 1 e 2, chiamati **stdout** e **stderr**.

stderr riceve i flussi di errore restituiti da un comando. Gli altri flussi sono diretti a **stdout**.

Questi flussi puntano ai file delle periferiche, ma poiché tutto è un file in UNIX/Linux, i flussi di I/O possono essere facilmente indirizzati ad altri file. Questo principio è la forza della shell.

4.7.2 Redirezione Input

È possibile reindirizzare il flusso di input da un altro file con il carattere `<` o `<<`. Il comando leggerà il file invece della tastiera:

```
ftp -in serverftp << ftp-commands.txt
```

> **Nota**
>
> Solo i comandi che richiedono l'input da tastiera saranno in grado di gestire il reindirizzamento dell'input.

Il reindirizzamento dell'ingresso può anche essere utilizzato per simulare l'interattività dell'utente. Il comando leggerà il flusso di ingresso finché non incontrerà la parola chiave definita dopo il reindirizzamento dell'ingresso.

Questa funzione è utilizzata per i comandi interattivi negli script:

```
$ ftp -in serverftp << END
user alice password
put file
bye
END
```


Copyright © 2023 The Rocky Enterprise Software Foundation

La parola chiave `END` può essere sostituita da qualsiasi parola.

```
$ ftp -in serverftp << STOP
user alice password
put file
bye
STOP
```

La shell esce dal comando `ftp` quando riceve una linea contenente solo la parola chiave.

> **Attenzione**
>
> La parola chiave finale, qui `END` o `STOP`, deve essere l'unica parola sulla linea e deve essere all'inizio della linea.

Il reindirizzamento dello standard input è usato raramente, perché la maggior parte dei comandi accetta un nome di file come argomento.

Il comando `wc` potrebbe essere usato in questo modo:

```
$ wc -l .bash_profile
27 .bash_profile # the number of lines is followed by the file name
$ wc -l < .bash_profile
27 # returns only the number of lines
```

4.7.3 Redirezione Output

L'output standard può essere reindirizzato ad altri file usando il carattere `>` o `>>`.

Il semplice reindirizzamento `>` sovrascrive il contenuto del file di output:

```
date +%F > date_file
```

Quando viene utilizzato il carattere `>>`, indica che il risultato del comando viene aggiunto al contenuto del file.

```
date +%F >> date_file
```

In entrambi i casi, il file viene creato automaticamente quando non esiste.

Copyright © 2023 The Rocky Enterprise Software Foundation

L'output di errore standard può anche essere reindirizzato ad un altro file. Questa volta sarà necessario specificare il numero del canale (che può essere omesso per i canali 0 e 1):

```
ls -R / 2> errors_file

ls -R / 2>> errors_file
```

4.7.4 Esempi di reindirizzamento

Reindirizzamento di 2 output verso 2 file:

```
ls -R / >> ok_file 2>> nok_file
```

Reindirizzamento di 2 output a un singolo file:

```
$ ls -R / >> ok_file 2>> nok_file
```

Reindirizzamento del *stderr* a un "pozzo senza fondo" (`/dev/null`):

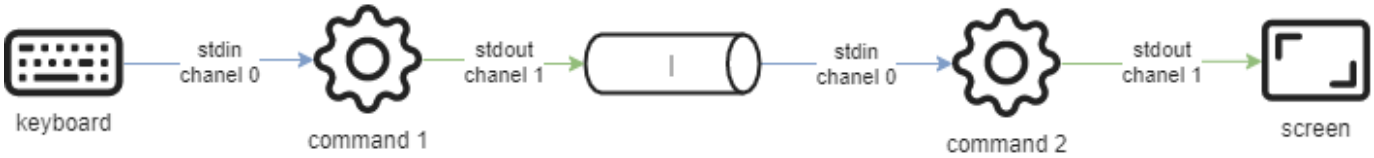
```
ls -R / 2>> /dev/null
```

Quando entrambi i flussi di uscita vengono reindirizzati, nessuna informazione viene visualizzata sullo schermo. Per utilizzare sia il reindirizzamento dell'uscita che il mantenimento della visualizzazione, è necessario utilizzare il comando `tee`.

4.7.5 Pipes

Una **pipe** è un meccanismo che consente di collegare l'output standard di un primo comando all'ingresso standard di un secondo comando.

Questa comunicazione è unidirezionale ed è fatta con il simbolo `|`. Il simbolo della pipe `|` è ottenuto premendo il tasto `⇧ Shift` + `|` contemporaneamente.



Copyright © 2023 The Rocky Enterprise Software Foundation

Tutti i dati inviati dal controllo a sinistra della pipe attraverso il canale di uscita standard vengono inviati al canale di ingresso standard del controllo a destra.

I comandi particolarmente utilizzati dopo una pipe sono i filtri.

- Esempi:

Mostra solo l'inizio:

```
ls -lia / | head
```

Mostra solo la fine:

```
ls -lia / | tail
```

Ordina il risultato:

```
ls -lia / | sort
```

Conta il numero di parole / caratteri:

```
ls -lia / | wc
```

Cerca una stringa nel risultato:

```
ls -lia / | grep fichier
```

4.8 Punti Speciali

4.8.1 comando `tee`

Il comando `tee` viene utilizzato per reindirizzare l'output standard di un comando a un file mantenendo la visualizzazione sullo schermo.

Viene combinato con la pipe `|` per ricevere come input l'output del comando da reindirizzare:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
ls -lia / | tee fic

cat fic
```

L'opzione `-a` aggiunge al file invece di sovrascriverlo.

4.8.2 comandi `alias` e `unalias`

L'uso di **alias** è un modo per chiedere alla shell di ricordare un particolare comando con le sue opzioni e di dargli un nome.

Per esempio:

```
ll
```

sostituirà il comando:

```
ls -l
```

Il comando `alias` elenca gli alias per la sessione corrente. Gli alias sono stabiliti per impostazione predefinita sulle distribuzioni Linux. Qui, gli alias per un server Rocky Linux:

```
$ alias
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-
tilde'
```

Gli alias sono definiti solo temporaneamente, per il tempo della sessione utente.

Per un uso permanente, devono essere creati nel:

- file `.bashrc` nella directory di accesso dell'utente;
- file `/etc/bashrc` per tutti gli utenti.

Copyright © 2023 The Rocky Enterprise Software Foundation

Attenzione

Occorre prestare particolare attenzione all'uso degli alias, che possono essere potenzialmente pericolosi! Ad esempio, un alias creato all'insaputa dell'amministratore:

```
alias cd='rm -Rf'
```

Il comando `unalias` ti consente di eliminare gli alias.

Per eliminare un singolo alias:

```
unalias ll
```

Per eliminare tutti gli alias:

```
unalias -a
```

Per disabilitare temporaneamente un alias, la combinazione è `\<nome alias>`.

Ad esempio se digitiamo:

```
type ls
```

potrebbe restituire quanto segue:

```
ls is an alias to « ls -rt »
```

Ora che questo è noto, possiamo vedere i risultati dell'utilizzo dell'alias o disabilitarlo in una volta con il carattere `\` eseguendo il seguente:

```
$ ls file*   # order by time
file3.txt  file2.txt  file1.txt
$ \ls file*  # order by name
file1.txt  file2.txt  file3.txt
```

4.8.3 Alias e Funzioni Utili

- alias di `grep`.

Copyright © 2023 The Rocky Enterprise Software Foundation

Colora il risultato del comando `grep`: `alias grep='grep --color=auto'`

- funzione `mcd`

È comune creare una cartella e poi spostarsi al suo interno:
`mcd() { mkdir -p "$1"; cd "$1"; }`

- funzione `cls`

Spostarsi in una cartella ed elencarne il contenuto: `cls() { cd "$1"; ls; }`

- funzione `backup`

Crea una copia di backup di un file: `backup() { cp "$1"{,.bak}; }`

- funzione `extract`

Estrae qualsiasi tipo di archivio:

```
extract () {
  if [ -f $1 ] ; then
    case $1 in
      *.tar.bz2) tar xjf $1 ;;
      *.tar.gz) tar xzf $1 ;;
      *.bz2) bunzip2 $1 ;;
      *.rar) unrar e $1 ;;
      *.gz) gunzip $1 ;;
      *.tar) tar xf $1 ;;
      *.tbz2) tar xjf $1 ;;
      *.tgz) tar xzf $1 ;;
      *.zip) unzip $1 ;;
      *.Z) uncompress $1 ;;
      *.7z) 7z x $1 ;;
      *)
        echo "'$1' cannot be extracted via extract()" ;;
    esac
  else
    echo "'$1' is not a valid file"
  fi
}
```

- Se `l'alias cmount` restituisce quanto segue: `alias cmount="mount | column -t"`

Si può quindi usare `cmount` per mostrare tutti i mount del sistema in colonne come la seguente: `[root]# cmount`

Copyright © 2023 The Rocky Enterprise Software Foundation

che restituisce il nostro filesystem montato nel seguente formato:

```
/dev/simfs  on  /                                        type  simfs
(rw,relatime,usrquota,grpquota)
proc        on  /proc                                    type  proc
(rw,relatime)
sysfs       on  /sys                                     type  sysfs
(rw,relatime)
none        on  /dev                                     type  devtmpfs
(rw,relatime,mode=755)
none        on  /dev/pts                                 type  devpts
(rw,relatime,mode=600,ptmxmode=000)
none        on  /dev/shm                                 type  tmpfs
(rw,relatime)
none        on  /proc/sys/fs/binfmt_misc                 type  binfmt_misc
(rw,relatime)
```

4.8.4 Il carattere `;`

Il carattere `;` concatena i comandi.

Una volta che l'utente preme `Enter ⏎`, i comandi vengono eseguiti in sequenza nell'ordine di inserimento.

```
ls /; cd /home; ls -lia; cd /
```

4.9 Verificare le proprie Conoscenze

✔ Cosa definisce un utente sotto Linux? (7 risposte)

✔ Cosa caratterizza un'opzione lunga per un comando?

✔ Quali comandi ti permettono di cercare aiuto su un comando?

`google`

`chuck --norris`

`info`

`apropos`

`whatis`

Copyright © 2023 The Rocky Enterprise Software Foundation

✔ Quale comando ti permette di vedere la cronologia di un utente?

✔ Quale comando ti permette di cercare del testo in un file?

`find`

`grep`

✔ Quale comando ti permette di cercare un file?

`find`

`grep`

✔ Quale comando reindirizza il flusso di errore di un comando a un nuovo file `errors.log`?

`ls -R / 2> errors.log`

`ls -R / 2>> errors.log`

`ls -R / 2> errors.log 2>&1`

Copyright © 2023 The Rocky Enterprise Software Foundation

5. Comandi avanzati per gli utenti Linux

I comandi avanzati offrono una maggiore personalizzazione e controlli in situazioni più specialistiche una volta acquisita familiarità con i comandi di base.

Obiettivi : In questo capitolo, i futuri amministratori Linux impareranno:

✔ alcuni comandi utili non trattati nel capitolo precedente. ✔ alcuni comandi avanzati.

🏁 **comandi utente**, **Linux**

Conoscenza: ⭐ **Complessità**: ⭐ ⭐ ⭐

Tempo di lettura: 20 minuti

5.1 comando `uniq`

Il comando `uniq` è un comando molto potente, usato con il comando `sort`, soprattutto per l'analisi dei file di registro. Ti consente di ordinare e visualizzare le voci rimuovendo i duplicati.

Per illustrare il funzionamento del comando `uniq`, utilizziamo un file `firstnames.txt` contenente un elenco di nomi:

```
antoine
xavier
steven
patrick
xavier
antoine
antoine
steven
```

> **Nota**
>
> `uniq` richiede che il file di input sia ordinato perché confronta solo le righe consecutive.

Copyright © 2023 The Rocky Enterprise Software Foundation

Senza argomenti, il comando `uniq` non visualizza le righe identiche che si susseguono nel file `firstnames.txt`:

```
$ sort firstnames.txt | uniq
antoine
patrick
steven
xavier
```

Per visualizzare solo le righe che appaiono solo una volta, utilizzare l'opzione `-u`:

```
$ sort firstnames.txt | uniq -u
patrick
```

Al contrario, per visualizzare solo le righe che compaiono almeno due volte nel file, utilizzare l'opzione `-d`:

```
$ sort firstnames.txt | uniq -d
antoine
steven
xavier
```

Per eliminare semplicemente linee che appaiono solo una volta, utilizzare l'opzione `-D`:

```
$ sort firstnames.txt | uniq -D
antoine
antoine
antoine
steven
steven
xavier
xavier
```

Infine, contare il numero di occorrenze di ciascuna linea, utilizzare l'opzione `-c`:

```
$ sort firstnames.txt | uniq -c
      3 antoine
      1 patrick
      2 steven
      2 xavier
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ sort firstnames.txt | uniq -cd
      3 antoine
      2 steven
      2 xavier
```

5.2 comando `xargs`

Il comando `xargs` consente la costruzione e l'esecuzione delle linee di comando da input standard.

Il comando `xargs` legge lo spazio bianco o gli argomenti delimitati da linefeed dall'ingresso standard, ed esegue il comando (`/bin/echo` per impostazione predefinita.) una o più volte utilizzando gli argomenti iniziali seguiti dagli argomenti letti dall'ingresso standard.

Un primo e più semplice esempio sarebbe il seguente:

```
$ xargs
use
of
xargs
<CTRL+D>
use of xargs
```

Il comando `xargs` attende un input dallo standard input **stdin**. Sono state inserite tre linee. La fine dell'input dell'utente è specificata in `xargs` dalla sequenza di tasti `^ Ctrl` + `D`. `xargs` esegue quindi il comando predefinito `echo` seguito dai tre argomenti corrispondenti all'input dell'utente, vale a dire:

```
$ echo "use" "of" "xargs"
use of xargs
```

È possibile specificare un comando da far eseguire a `xargs`.

Nell'esempio seguente, `xargs` eseguirà il comando `ls -ld` sul set di cartelle specificate nell'input standard:

```
$ xargs ls -ld
/home
/tmp
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
/root
<CTRL+D>
drwxr-xr-x. 9 root root 4096  5 avril 11:10 /home
dr-xr-x---. 2 root root 4096  5 avril 15:52 /root
drwxrwxrwt. 3 root root 4096  6 avril 10:25 /tmp
```

In pratica, il comando `xargs` esegue il comando `ls -ld /home /tmp /root`.

Cosa succede se il comando da eseguire non accetta argomenti multipli, come nel caso del comando `find`?

```
$ xargs find /var/log -name
*.old
*.log
find: paths must precede expression: *.log
```

Il comando `xargs` tenta di eseguire il comando `find` con più argomenti dietro l'opzione `-name`, questo causa la generazione di un errore in `find`:

```
$ find /var/log -name "*.old" "*.log"
find: paths must precede expression: *.log
```

In questo caso, il comando `xargs` deve essere costretto ad eseguire il comando `find` più volte (una volta per riga immessa come ingresso standard). L'opzione `-L` Seguito da un **intero** consente di specificare il numero massimo di voci da elaborare con il comando contemporaneamente:

```
$ xargs -L 1 find /var/log -name
*.old
/var/log/dmesg.old
*.log
/var/log/boot.log
/var/log/anaconda.yum.log
/var/log/anaconda.storage.log
/var/log/anaconda.log
/var/log/yum.log
/var/log/audit/audit.log
/var/log/anaconda.ifcfg.log
/var/log/dracut.log
/var/log/anaconda.program.log
<CTRL+D>
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Per specificare entrambi gli argomenti sulla stessa riga, utilizzare l'opzione `-n 1`:

```
$ xargs -n 1 find /var/log -name
*.old *.log
/var/log/dmesg.old
/var/log/boot.log
/var/log/anaconda.yum.log
/var/log/anaconda.storage.log
/var/log/anaconda.log
/var/log/yum.log
/var/log/audit/audit.log
/var/log/anaconda.ifcfg.log
/var/log/dracut.log
/var/log/anaconda.program.log
<CTRL+D>
```

Caso di esempio di un backup con un `tar` basato su una ricerca:

```
$ find /var/log/ -name "*.log" -mtime -1 | xargs tar cvfP /root/log.tar
$ tar tvfP /root/log.tar
-rw-r--r-- root/root      1720 2017-04-05 15:43 /var/log/boot.log
-rw-r--r-- root/root    499270 2017-04-06 11:01 /var/log/audit/audit.log
```

La caratteristica speciale del comando `xargs` è che posiziona l'argomento di input alla fine del comando chiamato. Questo funziona molto bene con l'esempio sopra riportato dal momento che i file passati formano l'elenco dei file da aggiungere all'archivio.

Utilizzando l'esempio del comando `cp`, per copiare un elenco di file in una directory, questo elenco di file verrà aggiunto alla fine del comando... ma ciò che il comando `cp` si aspetta alla fine del comando è la destinazione. Per farlo, si può usare l'opzione `-I` per inserire gli argomenti di input in un punto diverso dalla fine della riga.

```
find /var/log -type f -name "*.log" | xargs -I % cp % /root/backup
```

L'opzione `-I` consente di specificare un carattere (il carattere `%` nell'esempio precedente) in cui verranno inseriti i file di input di `xargs`.

Copyright © 2023 The Rocky Enterprise Software Foundation

5.3 pacchetto `yum-utils`

Il pacchetto `yum-utils` è una raccolta di utilità, realizzate per `yum` da vari autori, che ne rendono più facile e potente l'uso.

> **Nota**
>
> Mentre `yum` è stato sostituito da `dnf` in Rocky Linux 8, il nome del pacchetto è rimasto `yum-utils`, sebbene possa essere installato anche come `dnf-utils`. Queste sono le classiche utilities YUM implementate come shims CLI sopra a DNF per mantenere la retrocompatibilità con `yum-3`.

Ecco alcuni esempi di queste utilità.

5.3.1 Comando `repoquery`

Il comando `repoquery` viene utilizzato per interrogare i pacchetti nel repository.

Esempi di utilizzo:

- Visualizzare le dipendenze di un pacchetto (può essere un pacchetto software che è stato installato o non installato), equivalente a `dnf deplist <nome-pacchetto>`

```
repoquery --requires <package-name>
```

- Mostra i file forniti da un pacchetto installato (non funziona per i pacchetti che non sono installati), equivalente a `rpm -ql <package-name>`

```
$ repoquery -l yum-utils
/etc/bash_completion.d
/etc/bash_completion.d/yum-utils.bash
/usr/bin/debuginfo-install
/usr/bin/find-repos-of-install
/usr/bin/needs-restarting
/usr/bin/package-cleanup
/usr/bin/repo-graph
/usr/bin/repo-rss
/usr/bin/repoclosure
/usr/bin/repodiff
/usr/bin/repomanage
/usr/bin/repoquery
/usr/bin/reposync
/usr/bin/repotrack
/usr/bin/show-changed-rco
/usr/bin/show-installed
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
/usr/bin/verifytree
/usr/bin/yum-builddep
/usr/bin/yum-config-manager
/usr/bin/yum-debug-dump
/usr/bin/yum-debug-restore
/usr/bin/yum-groups-manager
/usr/bin/yumdownloader
…
```

5.3.2 Comando `yumdownloader`

Il comando `yumdownloader` scarica i pacchetti RPM dai repository. Equivalente a `dnf scaricare --downloadonly --downloaddir ./ package-name`

> **Nota**
>
> Questo comando è molto utile per creare rapidamente un repository locale di alcuni rpm!

Esempio: `yumdownloader` scaricherà il pacchetto rpm *repoquery* e tutte le sue dipendenze:

```
$ yumdownloader --destdir /var/tmp --resolve samba
or
$ dnf download --downloadonly --downloaddir /var/tmp  --resolve  samba
```

Opzioni	Commenti
`--destdir`	I pacchetti scaricati verranno memorizzati nella cartella specificata.
`--resolve`	Scarica anche le dipendenze del pacchetto.

5.4 pacchetto `psmisc`

Il pacchetto `psmisc` contiene utilità per la gestione dei processi di sistema:

- `pstree`: il comando `pstree` visualizza i processi correnti sul sistema in una struttura ad albero.
- `killall`: il comando `killall` invia un segnale di kill a tutti i processi identificati dal nome.
- `fuser`: il comando `fuser` Identifica il `PID` di processi che utilizzano i file o i file system specificati.

Copyright © 2023 The Rocky Enterprise Software Foundation

Esempi:

```
$ pstree
systemd─┬─NetworkManager───2*[{NetworkManager}]
        ├─agetty
        ├─auditd───{auditd}
        ├─crond
        ├─dbus-daemon───{dbus-daemon}
        ├─firewalld───{firewalld}
        ├─lvmetad
        ├─master─┬─pickup
        │        └─qmgr
        ├─polkitd───5*[{polkitd}]
        ├─rsyslogd───2*[{rsyslogd}]
        ├─sshd───sshd───bash───pstree
        ├─systemd-journal
        ├─systemd-logind
        ├─systemd-udevd
        └─tuned───4*[{tuned}]
```

```
# killall httpd
```

Arresta i processi (opzione `-k`) che accedono al file `/etc/httpd/conf/httpd.conf`:

```
# fuser -k /etc/httpd/conf/httpd.conf
```

5.5 comando `watch`

Il comando `watch` esegue regolarmente un comando e visualizza il risultato nel terminale a schermo intero.

L'opzione `-n` consente di specificare il numero di secondi tra ogni esecuzione del comando.

> **Nota**
>
> Per uscire dal comando `watch`, è necessario digitare i tasti: `^ Ctrl` + `C` per terminare il processo.

Esempi:

- Visualizza la fine del file `/etc/passwd` ogni 5 secondi:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
watch -n 5 tail -n 3 /etc/passwd
```

Risultato:

```
Every 5.0s: tail -n 3 /etc/
passwd
rockstar.rockylinux.lan: Thu Jul  1 15:43:59 2021

sssd:x:996:993:User for sssd:/:/sbin/nologin
chrony:x:995:992::/var/lib/chrony:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
```

- Monitoraggio del numero di file in una cartella:

```
watch -n 1 'ls -l | wc -l'
```

- Mostra un orologio:

```
watch -t -n 1 date
```

5.6 Comando `install`

Contrariamente a quanto potrebbe suggerire il nome, il comando `install` non viene usato per installare nuovi pacchetti.

Questo comando combina la copia dei file (`cp`) e la creazione di cartelle (`mkdir`), con la gestione dei diritti (`chmod`, `chown`) e altre utili funzionalità (come i backup).

```
install source dest
install -t directory source [...]
install -d directory
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Opzioni:

Opzioni	Osservazioni
`-b` o `--backup[=suffix]`	crea un backup del file di destinazione
`-d`	tratta gli argomenti come nomi di cartelle
`-D`	crea tutti i componenti principali, prima di copiare SOURCE in DEST
`-g` e `-o`	imposta la proprietà
`-m`	imposta le autorizzazioni
`-p`	preservare i timestamp dei file sorgente
`-t`	copia tutti gli argomenti di origine nella directory

Nota

Esistono opzioni per la gestione del contesto SELinux (vedere la pagina del manuale).

Esempi:

Creare una cartella con l'opzione `-d`:

```
install -d ~/samples
```

Copiare un file da una posizione di origine in una cartella:

```
install src/sample.txt ~/samples/
```

Questi due ordini si sarebbero potuti eseguire con un unico comando:

```
$ install -v -D -t ~/samples/ src/sample.txt
install: creating directory '~/samples'
'src/sample.txt' -> '~/samples/sample.txt'
```

Questo comando consente di risparmiare tempo. Combinatelo con la gestione dei proprietari, dei gruppi di proprietari e dei diritti per migliorare e ridurre i tempi:

```
sudo install -v -o rocky -g users -m 644 -D -t ~/samples/ src/sample.txt
```

Nota

In questo caso è necessario `sudo` per apportare modifiche alle proprietà.

Copyright © 2023 The Rocky Enterprise Software Foundation

È anche possibile creare un backup dei file esistenti grazie all'opzione `-b`:

```
$ install -v -b -D -t ~/samples/ src/sample.txt
'src/sample.txt' -> '~/samples/sample.txt' (archive: '~/samples/sample.txt~')
```

Come si può vedere, il comandodi `install` crea un file di backup con una tilde `~` aggiunta al nome del file originale.

Il suffisso può essere specificato grazie all'opzione `-S`:

```
$ install -v -b -S ".bak" -D -t ~/samples/ src/sample.txt
'src/sample.txt' -> '~/samples/sample.txt' (archive: '~/samples/
sample.txt.bak')
```

5.7 Comando `tree`

Espande i file o le cartelle della directory in una struttura ad albero.

opzioni	descrizione
`-a`	Vengono elencati tutti i file
`-h`	Stampa le dimensioni in un modo più leggibile per l'utente
`-u`	Visualizza il proprietario del file o il numero UID
`-g`	Visualizza il proprietario del gruppo di file o il numero GID
`-p`	Stampa le protezioni per ciascun file

Per esempio:

```
$ tree -hugp /etc/yum.repos.d/
/etc/yum.repos.d/
├── [-rw-r--r-- root     root      1.6K]  epel-modular.repo
├── [-rw-r--r-- root     root      1.3K]  epel.repo
├── [-rw-r--r-- root     root      1.7K]  epel-testing-modular.repo
├── [-rw-r--r-- root     root      1.4K]  epel-testing.repo
├── [-rw-r--r-- root     root       710]  Rocky-AppStream.repo
├── [-rw-r--r-- root     root       695]  Rocky-BaseOS.repo
├── [-rw-r--r-- root     root      1.7K]  Rocky-Debuginfo.repo
├── [-rw-r--r-- root     root       360]  Rocky-Devel.repo
├── [-rw-r--r-- root     root       695]  Rocky-Extras.repo
├── [-rw-r--r-- root     root       731]  Rocky-HighAvailability.repo
├── [-rw-r--r-- root     root       680]  Rocky-Media.repo
├── [-rw-r--r-- root     root       680]  Rocky-NFV.repo
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
├── [-rw-r--r-- root     root       690]  Rocky-Plus.repo
├── [-rw-r--r-- root     root       715]  Rocky-PowerTools.repo
├── [-rw-r--r-- root     root       746]  Rocky-ResilientStorage.repo
├── [-rw-r--r-- root     root       681]  Rocky-RT.repo
└── [-rw-r--r-- root     root      2.3K]  Rocky-Sources.repo

0 directories, 17 files
```

5.8 Comando `stat`

Il comando `stat` visualizza lo stato di un file o di un file system.

```
$ stat /root/anaconda-ks.cfg
  File: /root/anaconda-ks.cfg
  Size: 1352            Blocks: 8          IO Block: 4096   regular file
Device: 10302h/66306d   Inode: 2757097     Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2024-01-20 13:04:57.012033583 +0800
Modify: 2023-09-25 14:04:48.524760784 +0800
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
 Change: 2024-01-24 16:37:34.315995221 +0800
  Birth: 2
```

- `File`: Visualizza il percorso del file.
- `Size`: Visualizza le dimensioni del file in byte. Se si tratta di una directory, visualizza i 4096 byte fissi occupati dal nome della directory.
- `Blocks`: Visualizza il numero di blocchi allocati. Attenzione, prego! Le dimensioni di ogni blocco in questo comando sono di 512 byte. La dimensione predefinita di ciascun blocco in `ls -ls` è di 1024 byte.
- `Device` - Numero del dispositivo in notazione decimale o esadecimale.
- `Inode`: L'inode è un numero ID univoco che il kernel Linux assegna a un file o a una directory.
- `Links`: Numero di collegamenti diretti. I collegamenti diretti sono talvolta detti collegamenti fisici.
- `Access`: L'ora dell'ultimo accesso ai file e alle directory, ovvero `atime` in GNU/Linux.
- `Modify`: La data dell'ultima modifica di file e directory, ovvero `mtime` in GNU/Linux.
- `Change`: La data dell'ultima modifica della proprietà, ad esempio `ctime` in GNU/Linux.
- `Birth`: Data di origine (data della creazione). In alcuni documenti è abbreviato come `btime` o `crtime`. Per visualizzare la data di creazione è necessario che la versione del file system e del kernel sia superiore a una determinata versione.

Per i file:

atime: Dopo aver effettuato l'accesso al contenuto del file con comandi quali `cat`, `less`, `more` e `head`, l'`atime` del file può risultare aggiornato. Prestare attenzione! L'`atime` del file non viene aggiornato in tempo reale e, per motivi di prestazioni, deve attendere un certo periodo di tempo prima di poter essere visualizzato. **mtime**: La modifica del contenuto del file può aggiornare il `mtime` del file (come l'aggiunta o la sovrascrittura del contenuto del file tramite reindirizzamento); poiché la dimensione del file è una proprietà del file, anche il `ctime` del file verrà aggiornato simultaneamente. **ctime**: La modifica del proprietario, del gruppo, dei

Copyright © 2023 The Rocky Enterprise Software Foundation

permessi, della dimensione del file e dei collegamenti (soft e hard link) del file aggiornerà ctime.

Per le cartelle:

atime: Dopo aver usato il comando `cd` per entrare in una nuova directory in cui non si è mai acceduto prima, è possibile aggiornare e correggere l'`atime` di quella directory. **mtime**: L'esecuzione di operazioni quali la creazione, l'eliminazione e la ridenominazione di file in questa directory aggiornerà l'`mtime` e il `ctime` della directory. **ctime**: Quando i permessi, il proprietario, il gruppo e così via di una directory cambiano, il `ctime` della directory viene aggiornato.

Suggerimento

- Se si crea un nuovo file o una nuova directory, i suoi `atime`, `mtime` e `ctime` sono esattamente gli stessi
- Se il contenuto del file viene modificato, l'`mtime` e il `ctime` del file verranno inevitabilmente aggiornati.
- Se viene creato un nuovo file nella directory, `atime`, `ctime` e `mtime` della directory verranno aggiornati simultaneamente.
- Se viene aggiornato l'`mtime` di una directory, deve essere aggiornato anche il `ctime` di quella directory.

Copyright © 2023 The Rocky Enterprise Software Foundation

6. Editor di testo VI

In questo capitolo imparerai come lavorare con il VIsual editor.

Obiettivi : In questo capitolo, i futuri amministratori Linux impareranno come:

✔ Utilizzare i comandi principali dell'editor VI;
✔ Modificare un testo con l'editor VI.

🏁 **comandi utente**, **linux**

Conoscenze: ⭐
Complessità: ⭐ ⭐

Tempo di lettura: 20 minuti

Visual (**VI**) è un editor di testo molto popolare sotto Linux, nonostante la sua limitata ergonomia. È infatti un editor interamente in modalità testo: ogni azione viene eseguita con una chiave sulla tastiera o sui comandi dedicati.

Molto potente, è soprattutto molto pratico poiché è nel complesso minimo per le applicazioni di base. È quindi accessibile in caso di guasto del sistema. La sua *universalità* (è presente su tutte le distribuzioni Linux e sotto UNIX) lo rende uno strumento *fondamentale* per l'amministratore.

Le sue funzionalità sono:

- Inserire, eliminare, modificare il testo;
- Copiare parole, linee o blocchi di testo;
- Cercare e sostituire i caratteri.

6.1 comando `vi`

Il comando `vi` Apre l'editor di testo *VI*.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
vi [-c command] [file]
```

Esempio:

```
vi /home/rockstar/file
```

Opzione	Informazione
`-c command`	Esegue VI specificando un comando all'apertura

Se il file esiste nella posizione menzionata dal percorso, viene letta da VI che viene aperto in modalita **comandi**.

Se il file non esiste, VI apre un file vuoto e una pagina vuota viene visualizzata sullo schermo. Quando il file viene salvato, prenderà il nome specificato con il comando.

Se il comando. `vi` viene eseguito senza specificare un nome di file, VI apre un file vuoto e una pagina vuota viene visualizzata sullo schermo. Quando il file viene salvato, VI chiederà un nome di file.

L'editor `vim` prende l'interfaccia e le funzioni di VI con molti miglioramenti.

```
vim [-c command] [file]
```

Tra questi miglioramenti, l'utente ha l'evidenziazione della sintassi, che è molto utile per la modifica di script di shell o file di configurazione.

Durante una sessione, VI utilizza un file buffer in cui registra tutte le modifiche apportate dall'utente.

Nota

Finché l'utente non ha salvato il suo lavoro, il file originale non viene modificato.

All'avvio, VI è in modalità *comando*.

Suggerimento

Una riga di testo è terminata premendo `INVIO` ma se lo schermo non è abbastanza largo, VI rende le interruzioni di linea automatiche, configurazione *wrap* per impostazione predefinita. Queste interruzioni di linea potrebbero non essere desiderate, questa è la configurazione *nowrap*.

Copyright © 2023 The Rocky Enterprise Software Foundation

Per uscire da VI, dalla modalità Comando, premi `:` quindi digita:

- `q` per uscire senza salvare (*quit*);
- `w` per salvare il tuo lavoro (*write*);
- `wq` (*write quit*) o `x` (*eXit*) per salvare e uscire.

Per forzare l'uscita senza conferma, devi aggiungere *!* ai precedenti comandi.

Per forzare l'uscita senza conferma, devi aggiungere *!* ai precedenti comandi.

> ⚠ **Attenzione**
>
> Non c'è alcun backup periodico, quindi è necessario ricordarsi di salvare il lavoro regolarmente.

6.2 Modalità operativa

In VI, ci sono 3 modalità di lavoro:

- La modalità *comando*;
- La modalità *inserimento*;
- La modalità *ex*.

La filosofia di VI è quella di alternare tra la modalità *comando* e la modalità *inserimento*.

La terza modalità, *ex*, è una modalità di comando del piè di pagina modello vecchio editor di testo.

6.2.1 La modalità comando

Questa è la modalità predefinita quando si avvia VI. Per accedervi da una qualsiasi delle altre modalità, basta premere il tasto `ESC`.

In questo momento, tutta la digitazione della tastiera viene interpretata come comandi e le azioni corrispondenti vengono eseguite. Questi sono essenzialmente comandi per la modifica del testo (copia, incolla, annulla, ...).

I comandi non vengono visualizzati sullo schermo.

Copyright © 2023 The Rocky Enterprise Software Foundation

6.2.2 La modalità inserimento

Questa è la modalità di modifica del testo. Per accedervi dalla modalità *comando*, devi premere tasti speciali che eseguiranno un'azione oltre a modificare la modalità.

Il testo non viene inserito direttamente nel file ma in una zona buffer nella memoria. Le modifiche sono efficaci solo quando il file viene salvato.

6.2.3 La modalità Ex

Questa è la modalità di modifica del file. Per accedervi, è necessario prima passare alla modalità *comando*, quindi inserire il comando *ex* frequentemente preceduto dal carattere `:`.

Il comando è convalidato premendo il tasto ENTER.

6.3 Muovere il cursore

In modalità *comando*, ci sono diversi modi per spostare il cursore.

Il mouse non è attivo in un ambiente di testo ma lo è in un ambiente grafico, è possibile spostarlo carattere per carattere, ma i collegamenti esistono per andare più veloci.

VI rimane in modalità *comando* dopo aver spostato il cursore.

Il cursore è posizionato sotto il carattere desiderato.

6.3.1 Da un carattere

- Sposta uno o `n` caratteri a sinistra:

← , n ← , h o n h

- Sposta uno o `n` caratteri a destra:

→ , n → , l o n l

- Sposta uno o `n` caratteri su:

Copyright © 2023 The Rocky Enterprise Software Foundation

[↑], [n][↑], [k] o [n][k]

- Sposta uno o `n` caratteri giù:

[↓], [n][↓], [j] o [n][j]

- Passare alla fine della linea:

[$] o [END]

- Passare all'inizio della linea:

[0] o [POS1]

6.3.2 Dal primo carattere di una parola

Le parole sono costituite da lettere o numeri. Caratteri di punteggiatura e apostrofi separarano le parole.

Se il cursore è nel mezzo di una parola [w] muove alla parola successiva, [b] si posiziona all'inizio della parola.

Se la linea è finita, VI va automaticamente alla riga successiva.

- Muoversi di una o `n` parole a destra:

[w] o [n][w]

- Muoversi di una o `n` parole a sinistra:

[b] o [n][b]

6.3.3 Da qualsiasi posizione su una linea

- Passa all'ultima riga di testo:

[G]

- Passa alla linea `n`:

Copyright © 2023 The Rocky Enterprise Software Foundation

`n` `G`

- Passa alla prima riga dello schermo:

`H`

- Passa alla linea centrale dello schermo:

`M`

- Passa all'ultima riga dello schermo:

`L`

6.4 Inserimento del testo

In modalità *comando*, ci sono diversi modi per inserire il testo.

VI passa alla modalità *inserimento* dopo l'inserimento di una di queste chiavi.

> **Nota**
>
> VI passa alla modalità *inserzione*. Quindi dovrai premere il tasto `ESC` per tornare alla modalità *comando*.

6.4.1 In relazione a un carattere

- Inserimento del testo prima di un carattere:

`i` (*insert*)

- Inserimento del testo dopo di un carattere:

`a` (*append*)

6.4.2 In relazione a una linea

- Inserimento del testo all'inizio di una linea:

`I`

- Inserimento del testo alla fine di una linea:

Copyright © 2023 The Rocky Enterprise Software Foundation

A

6.4.3 In relazione al testo

- Inserimento del testo prima di una linea:

0

- Inserimento del testo dopo una linea:

o

6.5 Caratteri, parole e linee

VI consente la modifica del testo mediante la gestione di:

- caratteri,
- parole,
- linee.

In ogni caso è possibile :

- eliminare,
- sostituire,
- copiare,
- tagliare,
- incollare.

Queste operazioni sono fatte in modalità *comando*.

6.5.1 Caratteri

- Elimina uno o `n` caratteri:

x o n x

- Sostituisce un carattere con un altro:

Copyright © 2023 The Rocky Enterprise Software Foundation

`r` `character`

- Sostituisce più di un carattere con altri:

`R` `characters` `ESC`

> **Nota**
>
> Il comando `R` passa alla modalità *sostituzione*, che è una sorta di modalità *inserzione*.

6.5.2 Parole

- Cancella (taglia) una o `n` parole:

`d` `w` o `n` `d` `w`

- Copia una o `n` parole:

`y` `w` o `n` `y` `w`

- Incolla una parola una volta o `n` volte dopo il cursore:

`p` o `n` `p`

- Incolla una parola una volta o `n` volte prima del cursore:

`P` o `n` `P`

- Sostituisci una parola:

`c` `w` *`word`* `ESC`

> **Suggerimento**
>
> È necessario posizionare il cursore sotto il primo carattere della parola da tagliare (o copiare) altrimenti VI taglierà (o copiare) solo la parte della parola tra il cursore e la fine. Per eliminare una parola è sufficiente tagliarla. Se non è incollata dopo, il buffer viene svuotato e la parola è cancellata.

6.5.3 Linee

- Elimina (taglia) una o `n` linee:

Copyright © 2023 The Rocky Enterprise Software Foundation

`d` `d` o `n` `d` `d`

- Copia una o `n` linee:

`p` o `n` `p`

- Incolla ciò che è stato copiato o cancellato una volta o `n` volte dopo la linea corrente:

`P` o `n` `P`

- Incolla ciò che è stato copiato o cancellato una volta o `n` volte prima della linea corrente:

`P` o `n` `P`

- Elimina (taglia) dall'inizio della linea alla posizione del cursore:

`d` `0`

- Elimina (taglia) dalla posizione del cursore alla fine della linea:

`d` `$`

- Copia dall'inizio della linea alla posizione del cursore:

`y` `0`

- Copia dalla posizione del cursore alla fine della linea:

`y` `$`

- Elimina (taglia) il testo dalla linea corrente:

`d` `L` o `d` `G`

- Copia il testo dalla linea corrente:

`y` `L` o `y` `G`

6.5.4 Annullare un'azione

- Annulla l'ultima azione:

Copyright © 2023 The Rocky Enterprise Software Foundation

`u`

- Annulla le azioni sulla linea corrente:

`U`

6.5.5 Annulla la cancellazione

- Annulla una cancellazione

`Ctrl+r`

6.6 comando EX

La modalità *Ex* ti consente di agire sul file (salvataggio, disposizione, opzioni, ...). È dentro la modalità *Ex* che vengono inseriti i comandi di ricerca e sostituzione. I comandi vengono visualizzati nella parte inferiore della pagina e devono essere convalidati con il tasto `ENTER`.

Per passare alla modalità *Ex*, dalla modalità *comando*, digita `:`.

6.6.1 Numeri di riga del file

- Mostra/nascondi numerazione:

`:set nu` e il più lungo `:set number`

`:set nonu` e il più lungo `:set nonumber`

6.6.2 Cerca una stringa

- Cerca una stringa dalla posizione del cursore:

`/string`

- Cerca una stringa prima della posizione del cursore:

`?string`

- Vai alla prossima occorrenza trovata:

Copyright © 2023 The Rocky Enterprise Software Foundation

`n`

- Vai alla precedente occorrenza trovata:

`N`

Ci sono caratteri jolly per facilitare la ricerca in VI.

- `[]` : Cerca una gamma di caratteri o un singolo carattere i cui valori possibili sono specificati.

Esempio:

`/[Ww]ord` : ricerca *word* o *Word*

`/[1-9]word` : ricerca *1word*, *2word* ... `x`*word* dove `x` è un numero

- `^` : Cerca una stringa dall'inizio della linea.

Esempio:

`/^Word`

- `$` : Cerca una stringa fino al termine della linea.

Esempio:

`/Word$`

- `.` : Cerca una parola con una lettera sconosciuta.

Esempio:

`/W.rd` : ricerca *Word*, *Ward*...

- `*` : Cerca uno o più caratteri, qualunque cosa siano.

Esempio:

`/W*d`

Nota: se si desidera ignorare le maiuscole e le minuscole (temporanee) durante la corrispondenza delle stringhe, digitare `:set ic.`

Copyright © 2023 The Rocky Enterprise Software Foundation

6.6.3 Sostituisci una stringa

Dalla prima all'ultima riga del testo, sostituisci la stringa cercata con la stringa specificata:

```
:1,$ s/search/replace
```

Nota: si può anche usare `:0,$s/search/replace` per specificare l'inizio assoluto del file.

Dalla linea `n` alla linea `m`, sostituisci la stringa cercata con la stringa specificata:

```
:n,m s/search/replace
```

Per impostazione predefinita, solo la prima occorrenza trovata di ciascuna linea è sostituita. Per forzare la sostituzione di ogni occorrenza, devi aggiungere `/g` alla fine del comando:

```
:n,m s/search/replace/g
```

Sfoglia un intero file per sostituire la stringa cercata con la stringa specificata:

```
:w
```

6.6.4 Operazioni sui file

- Cancellare una riga vuota

```
:w file
```

- Elimina le righe con numeri di riga da n a m

```
:n,m w file
```

- Elimina la riga in cui si trova la stringa

```
:g/string/d
```

- Elimina una riga che non contiene una stringa

Copyright © 2023 The Rocky Enterprise Software Foundation

`:r file`

- Cancella tutte le righe che iniziano con #

`:q`

Il **g** qui sta per **globale**.

6.6.5 Operazioni sui file

- Chiudi l'elaborazione di un file senza salvare:

`:w`

- Esci dalla modifica di un file che è stato modificato durante la sessione ma non salvato:

`:w file`

- Esci dal file e salva:

`:n,m w file`

- Ricarica l'ultima registrazione del file:

`e!`

- Incolla il contenuto in un altro file dopo la posizione del cursore:

`:r file`

- Chiudi l'elaborazione di un file senza salvare:

`:q`

- Esci dalla modifica di un file che è stato modificato durante la sessione ma non salvato:

`:q!`

- Esci dal file e salva:

`:wq` o `:x`

Copyright © 2023 The Rocky Enterprise Software Foundation

6.7 Altre funzioni

È possibile eseguire VI specificando le opzioni da caricare per la sessione. Per fare questo, devi usare l'opzione `-c`:

```
vi -c "set nu" /home/rockstar/file
```

È anche possibile inserire il comando *Ex* in un file chiamato `.exrc` inserito nella directory di accesso dell'utente. Ad ogni avvio di VI o VIM, i comandi saranno letti e applicati.

6.7.1 comando `vimtutor`

C'è un tutorial per imparare come usare VI. Ed è accessibile con il comando `vimtutor`.

```
vimtutor
```


Copyright © 2023 The Rocky Enterprise Software Foundation

7. Gestione utenti

In questo capitolo imparerai come gestire gli utenti.

Obiettivi: In questo capitolo, i futuri amministratori di Linux impareranno a:

✔ aggiungere, eliminare o modificare un **gruppo** ;
✔ aggiungere, eliminare o modificare un **utente** ;
✔ comprendere i file associati agli utenti e ai gruppi e scoprire come gestirli;
✔ cambiare il *proprietario* o il *proprietario del gruppo* di un file;
✔ *proteggere* un account utente;
✔ cambiare identità.

🏁 **utenti**

Conoscenza: ⭐⭐
Complessità: ⭐⭐

Tempo di lettura: 30 minuti

7.1 Generale

Ogni utente deve avere un gruppo, che è chiamato il gruppo **primario dell'utente**.

Diversi utenti possono far parte dello stesso gruppo.

Gruppo diversi dal gruppo primario sono chiamati **gruppi supplementari**.

> **Nota**
>
> Ogni utente ha un gruppo primario e può essere invitato in uno o più gruppi supplementari.

I gruppi e gli utenti sono gestiti dai loro identificatori numerici unici `GID` e `UID`.

- `UID` : *User IDentifier*. ID utente univoco.
- `GID` : *Group IDentifier*. Identificatore di gruppo univoco.

Copyright © 2023 The Rocky Enterprise Software Foundation

Sia UID che GID sono riconosciuti dal kernel, il che significa che il Super Admin non è necessariamente l'utente **root**, purché l'utente **uid=0** sia il Super Admin.

I file relativi agli utenti/gruppi sono:

- /etc/passwd
- /etc/shadow
- /etc/group
- /etc/gshadow
- /etc/skel/
- /etc/default/useradd
- /etc/login.defs

> **Pericolo**
>
> Dovresti sempre usare i comandi di amministrazione invece di modificare manualmente i file.

> **Nota**
>
> Alcuni comandi in questo capitolo richiedono i diritti di amministratore. Per convenzione, si specifica il comando `sudo` quando i comandi devono essere eseguiti con diritti di amministratore. Affinché gli esempi funzionino correttamente, assicuratevi che il vostro account abbia il permesso di usare il comando `sudo`.

7.2 Gestione del gruppo

File modificati, righe aggiunte:

- `/etc/group`
- `/etc/gshadow`

7.2.1 comando `groupadd`

Il comando `groupadd` aggiunge un gruppo al sistema.

```
groupadd [-f] [-g GID] group
```

Esempio:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
sudo groupadd -g 1012 GroupeB
```

Opzione	Descrizione
`-g GID`	Definisce il `GID` del gruppo da creare.
`-f`	Il sistema sceglie un `GID` se quello specificato dall'opzione `-g` esiste già.
`-r`	Crea un gruppo di sistema con un `GID` tra `SYS_GID_MIN` e `SYS_GID_MAX`. Queste due variabili sono definite in `/etc/login.defs`.

Regole di denominazione del gruppo:

- Nessun accento o carattere speciale;
- Diverso dal nome di un utente o file di sistema esistenti.

Nota

In **Debian**, l'amministratore dovrebbe usare, tranne che negli script destinati a essere trasferiti a tutte le distribuzioni Linux, i comandi `addgroup` e `delgroup` come specificato in `man`:

```
$ man addgroup
DESCRIPTION
adduser and addgroup add users and groups to the system according to command line options and configuration information
in /etc/adduser.conf. They are friendlier front ends to the low-level tools like useradd, groupadd and usermod programs,
by default, choosing Debian policy conformant UID and GID values, creating a home directory with skeletal configuration,
running a custom script, and other features.
```

7.2.2 Comando `groupmod`

Il comando `groupmod` consente di modificare un gruppo esistente sul sistema.

```
groupmod [-g GID] [-n nom] group
```

Esempio:

```
sudo groupmod -g 1016 GroupP

sudo groupmod -n GroupC GroupB
```

Opzione	Osservazioni
`-g GID`	Nuovo `GID` del gruppo da modificare.
`-n name`	Nuovo nome.

È possibile cambiare il nome di un gruppo, il `GID` o entrambi simultaneamente.

Copyright © 2023 The Rocky Enterprise Software Foundation

Dopo la modifica, i file appartenenti al gruppo hanno un `GID` sconosciuto. Devono essere riassegnati al nuovo `GID`.

```
sudo find / -gid 1002 -exec chgrp 1016 {} \;
```

7.2.3 comando `groupdel`

Il comando `groupdel` elimina un gruppo esistente sul sistema.

```
groupdel group
```

Esempio:

```
sudo groupdel GroupC
```

Suggerimento

Quando si elimina un gruppo, possono verificarsi due condizioni:

- Se un utente ha un gruppo primario unico e si lancia il comando `groupdel` su quel gruppo, verrà indicato che c'è un utente specifico sotto il gruppo e che non può essere cancellato.
- Se un utente appartiene a un gruppo supplementare (non il gruppo primario dell'utente) e tale gruppo non è il gruppo primario di un altro utente del sistema, il comando `groupdel` eliminerà il gruppo senza ulteriori richieste.

Esempi:

```
$ sudo useradd test
$ id test
uid=1000(test) gid=1000(test) group=1000(test)
$ sudo groupdel test
groupdel: cannot remove the primary group of user 'test'

$ sudo usermod -g users -G test test
$ id test
uid=1000(test) gid=100(users) group=100(users),1000(test)
$ sudo groupdel testb
```

Suggerimento

Quando si elimina un utente con il comando `userdel -r`, viene eliminato anche il gruppo primario corrispondente. Il nome del gruppo primario di solito corrisponde al nome dell'utente.

Copyright © 2023 The Rocky Enterprise Software Foundation

Suggerimento

Ogni gruppo ha un `GID` univoco. Un gruppo può essere utilizzato da più utenti come gruppo supplementare. Per convenzione, il GID del super amministratore è 0. I GIDS riservati ad alcuni servizi o processi sono 201-999, chiamati gruppi di sistema o pseudo gruppi di utenti. Il GID per gli utenti è solitamente maggiore o uguale a 1000. Questi sono relativi a /etc/login.defs, di cui parleremo più tardi.

```
# Comment line ignored
shell > cat  /etc/login.defs
MAIL_DIR        /var/spool/mail
UMASK           022
HOME_MODE       0700
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7
UID_MIN                  1000
UID_MAX                 60000
SYS_UID_MIN               201
SYS_UID_MAX               999
GID_MIN                  1000
GID_MAX                 60000
SYS_GID_MIN               201
SYS_GID_MAX               999
CREATE_HOME     yes
USERGROUPS_ENAB yes
ENCRYPT_METHOD SHA512
```

Suggerimento

Poiché un utente fa necessariamente parte di un gruppo, è meglio creare i gruppi prima di aggiungere gli utenti. Pertanto, un gruppo potrebbe non avere alcun membro.

7.2.4 file `/etc/group`

Questo file contiene le informazioni sul gruppo (separate da `:`).

```
$ sudo tail -1 /etc/group
GroupP:x:516:patrick
  (1)  (2)(3)   (4)
```

- 1: Nome del gruppo.
- 2: La password del gruppo è identificata da una `x`. La password del gruppo è memorizzata in `/etc/gshadow`.
- 3: GID.
- 4: Utenti supplementari del gruppo (escluso l'utente primario unico).

Nota

Copyright © 2023 The Rocky Enterprise Software Foundation

Ogni riga nel file `/etc/group` corrisponde a un gruppo. Le informazioni principali dell'utente sono memorizzate in `/etc/passwd`.

7.2.5 file `/etc/gshadow`

Questo file contiene le informazioni di sicurezza sui gruppi (separate da `:`).

```
$ sudo grep GroupA /etc/gshadow
GroupA:$6$2,9,v...SBn160:alain:rockstar
   (1)      (2)            (3)      (4)
```

- 1: Nome del gruppo.
- 2: Password criptata.
- 3: Nome dell'amministratore del gruppo.
- 4: Utenti supplementari del gruppo (escluso l'utente primario unico).

> ⚠ **Attenzione**
>
> Il nome del gruppo in **/etc/group** e **/etc/gshadow** deve corrispondere uno a uno, cioè ogni riga del file **/etc/group** deve avere una riga corrispondente nel file **/etc/gshadow**.

Un `!` nella password indica che è bloccata. In questo modo, nessun utente può utilizzare la password per accedere al gruppo (poiché i membri del gruppo non ne hanno bisogno).

Copyright © 2023 The Rocky Enterprise Software Foundation

7.3 Gestione utenti

7.3.1 Definizione

Un utente è definito come segue nel file `/etc/passwd`:

- 1: Login name;
- 2: Identificazione della password, `x` indica che l'utente ha una password, la password criptata è memorizzata nel secondo campo di `/etc/shadow`;
- 3: UID;
- 4: GID del gruppo primario;
- 5: Commenti;
- 6: Home directory;
- 7: Shell (`/bin/bash`, `/bin/nologin`, ...).

Esistono tre tipi di utenti:

- **root(uid=0)**: l'amministratore di sistema;
- **system users(uid è uno dei 201~999)**: Utilizzato dal sistema per gestire i diritti di accesso alle applicazioni;
- **utente normale (uid>=1000)**: Altro account per accedere al sistema.

File modificati e righe aggiunte:

- `/etc/passwd`
- `/etc/shadow`

7.3.2 comando `useradd`

Il comando `useradd` aggiunge un utente.

```
useradd [-u UID] [-g GID] [-d directory] [-s shell] login
```

Esempio:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
sudo useradd -u 1000 -g 1013 -d /home/GroupC/carine carine
```

Opzione	Descrizione
`-u UID`	`UID` dell'utente da creare.
`-g GID`	`GID` del gruppo primario. Il `GID` qui può anche essere un `nome di un gruppo`.
`-G GID1, [GID2]...`	`GID` dei gruppi supplementari. Il `GID` qui può anche essere un `nome di un gruppo`. È possibile specificare molti gruppi supplementari separati da virgole.
`-d directory`	Crea la directory home.
`-s shell`	Specifica la shell dell'utente.
`-c COMMENT`	Aggiunge un commento.
`-U`	Aggiunge l'utente a un gruppo con lo stesso nome che viene creato nello stesso momento. Se non viene specificato, la creazione di un gruppo con lo stesso nome avviene durante la creazione dell'utente.
`-M`	Non crea la home directory dell'utente.
`-r`	Crea un account di sistema.

Alla sua creazione, l'account non ha una password ed è bloccato.

Per sbloccare l'account è necessario assegnare una password.

Quando si richiama il comando `useradd` senza alcuna opzione, vengono impostate le seguenti impostazioni predefinite per il nuovo utente:

- Crea una home directory con lo stesso nome;
- Crea un gruppo primario con lo stesso nome;
- La shell predefinita è bash;
- I valori UID e GID del gruppo primario dell'utente vengono automaticamente detratti. Questo è di solito un valore unico tra 1000 e 60.000.

Nota

Le impostazioni e i valori predefiniti si ottengono dai seguenti file di configurazione:

`/etc/login.defs` e `/etc/default/useradd`

```
$ sudo useradd test1

$ tail -n 1 /etc/passwd
test1:x:1000:1000::/home/test1:/bin/bash

$ tail -n 1 /etc/shadow
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
test1:!!:19253:0:99999:7
:::

$ tail -n 1 /etc/group ; tail -n 1 /etc/gshadow
test1:x:1000:
test1:!::
```

Regole di denominazione dell'account:

- Niente accenti, lettere maiuscole o caratteri speciali;
- Anche se è possibile utilizzare un nome utente in maiuscolo in RockyLinux, non lo raccomandiamo;
- Opzionale: imposta le opzioni `-u`, `-g`, `-d` e `-s` alla creazione.
- Diverso dal nome di un gruppo o file di sistema esistente;
- Il nome utente può contenere fino a 32 caratteri.

Attenzione

L'albero delle home directory deve essere già stato creato tranne che per l'ultima directory.

L'ultima directory viene creata dal comando `useradd`, che sfrutta l'opportunità per copiare i file da `/etc/skel` al suo interno.

Un utente può appartenere a diversi gruppi oltre a quello principale.

Esempio:

```
sudo useradd -u 1000 -g GroupA -G GroupP,GroupC albert
```

Nota

In **Debian**, è necessario specificare l'opzione `-m` per forzare la creazione della directory di login o impostare la variabile `CREATE_HOME` nel file `/etc/login.defs`. In tutti i casi, l'amministratore dovrebbe usare i comandi `adduser` e `deluser` come specificato in `man`, tranne che negli script destinati ad essere trasferiti a tutte le distribuzioni Linux:

```
$ man useradd
DESCRIPTION
    **useradd** is a low-level utility for adding users. On Debian, administrators should usually use **adduser(8)**
     instead.
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Valore predefinito per la creazione di un utente

Modifica del file `/etc/default/useradd`.

```
useradd -D [-b directory] [-g group] [-s shell]
```

Esempio:

```
sudo useradd -D -g 1000 -b /home -s /bin/bash
```

Opzione	Descrizione
`-D`	Imposta i valori predefiniti per la creazione dell'utente.
`-b directory`	Imposta la directory di accesso predefinita.
`-g group`	Imposta il gruppo predefinito.
`-s shell`	Imposta la shell predefinita.
`-f`	Imposta il numero di giorni dopo la scadenza della password prima di disabilitare l'account.
`-e`	Imposta la data di disabilitazione dell'account.

7.3.3 comando `usermod`

Il comando `usermod` consente di modificare un utente.

```
usermod [-u UID] [-g GID] [-d directory] [-m] login
```

Esempio:

```
sudo usermod -u 1044 carine
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Opzioni identiche al comando `useradd`.

Opzione	Descrizione
`-m`	Associato all'opzione `-d`. Sposta il contenuto della vecchia directory di login in quella nuova. Se la vecchia home directory non esiste, la creazione di una nuova home directory non avviene; la creazione della nuova home directory avviene se non esiste.
`-l login`	Modifica il nome di accesso. Dopo aver modificato il nome di accesso, è anche necessario modificare il nome della directory home per abbinarlo.
`-e YYYY-MM-DD`	Modifica la data di scadenza dell'account.
`-L`	Blocca l'account in modo permanente. Cioè, aggiunge un `!` all'inizio del campo della password `/etc/shadow`.
`-U`	Sblocca l'account.
`-a`	Aggiunge i gruppi supplementari dell'utente, che devono essere usati insieme all'opzione `-G`.
`-G`	Modifica i gruppi supplementari dell'utente e sovrascrive i gruppi supplementari precedenti.

Suggerimento

Per essere modificato, un utente deve essere disconnesso e non avere processi in corso.

Dopo la modifica dell'identificatore, i file appartenenti all'utente hanno un `UID` sconosciuto. Deve essere riassegnato al nuovo `UID`.

Dove `1000` è il vecchio `UID` e `1044` è il nuovo. Gli esempi sono i seguenti:

```
sudo find / -uid 1000 -exec chown 1044: {} \;
```

Blocco e sblocco dell'account utente, gli esempi sono i seguenti:

```
$ usermod -L test1
$ grep test1 /etc/shadow
test1:!
$6$n.hxglA.X5r7X0ex$qCXeTx.kQVmqsPLeuvIQnNidnSHvFiD7bQTxU7PLUCmBOcPNd5meqX6AEKSQvC
:::

$ usermod -U test1
```

La differenza tra l'opzione `-aG` e l'opzione `-G` può essere spiegata dal seguente esempio:

```
$ sudo useradd test1
$ sudo passwd test1
$ sudo groupadd groupA ; sudo groupadd groupB ; sudo groupadd groupC ; sudo
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
groupadd groupD
$ id test1
uid=1000(test1) gid=1000(test1) groups=1000(test1)

$ sudo gpasswd -a test1 groupA
$ id test1
uid=1000(test1) gid=1000(test1) groups=1000(test1),1002(groupA)

$ sudo usermod -G groupB,groupC test1
$ id test1
uid=1000(test1) gid=1000(test1) groups=1000(test1),1003(groupB),1004(groupC)

$ sudo usermod -aG groupD test1
$ id test1
uid=1000(test1) gid=1000(test1) groups=1000(test1),1003(groupB),1004(groupC),
1005(groupD)
```

7.3.4 comando `userdel`

Il comando `userdel` consente di eliminare l'account di un utente.

```
sudo userdel -r carine
```

Opzione	Descrizione
`-r`	Cancella la directory home dell'utente e i file di posta che si trovano nella directory `/var/spool/mail/`

Suggerimento

Per essere eliminato, un utente deve essere disconnesso e non avere processi in esecuzione.

Il comando `userdel` rimuove le righe corrispondenti in `/etc/passwd`, `/etc/shadow`, `/etc/group`, `/etc/gshadow`. Come detto in precedenza, `userdel -r` cancellerà anche il corrispondente gruppo primario dell'utente.

7.3.5 file `/etc/passwd`

Questo file contiene le informazioni utente (divise da `:`).

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ sudo head -1 /etc/passwd
root:x:0:0:root:/root:/bin/bash
(1)(2)(3)(4)(5)  (6)    (7)
```

- 1: Login name;
- 2: Identificazione della password, `x` indica che l'utente ha una password, la password criptata è memorizzata nel secondo campo di `/etc/shadow`;
- 3: UID.
- 4: GID del gruppo primario;
- 5: Commenti;
- 6: Home directory;
- 7: Shell (`/bin/bash`, `/bin/nologin`, ...).

7.3.6 file `/etc/shadow`

Questo file contiene le informazioni di sicurezza degli utenti (separate da `:`).

```
$ sudo tail -1 /etc/shadow
root:$6$...:15399:0:99999:7
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
:::
 (1)    (2)  (3) (4) (5) (6)(7,8,9)
```

- 1: Nome Login.
- 2: Password criptata. Utilizza l'algoritmo di crittografia SHA512, definito dal `ENCRYPT_METHOD` di `/etc/login.defs`.
- 3: L'ora in cui la password è stata cambiata l'ultima volta, il formato timestamp, in giorni. Il cosiddetto timestamp si basa sul 1 gennaio 1970 come orario standard. Ogni volta che un giorno passa, il timestamp è +1.
- 4: Durata minima della password. Ovvero, l'intervallo di tempo tra due modifiche della password (relative al terzo campo), in giorni. Definito dal `PASS_MIN_DAYS` di `/etc/login.defs`, il valore predefinito è 0, cioè, quando cambi la password per la seconda volta, non c'è alcuna restrizione. Tuttavia, se è 5, significa che non è permesso cambiare la password entro 5 giorni, e solo dopo 5 giorni.
- 5: Durata massima della password. Cioè, il periodo di validità della password (relativo al terzo campo). Definito dal `PASS_MAX_DAYS` di `/etc/login.defs`.
- 6: Il numero di giorni di avviso prima della scadenza della password (relativo al quinto campo). Il valore predefinito è di 7 giorni, definito dal `PASS_WARN_AGE` di `/etc/login.defs`.
- 7: Numero di giorni di tolleranza dopo la scadenza della password (in relazione al quinto campo).
- 8: Tempo di scadenza dell'account, il formato del timestamp, in giorni. **Nota che la scadenza di un account differisce dalla scadenza di una password. In caso di scadenza di un account, l'utente non può effettuare il login. In caso di scadenza della password, all'utente non è consentito effettuare il login utilizzando la sua password.**
- 9: Riservato per un uso futuro.

> **Pericolo**
>
> Per ogni riga del file `/etc/passwd` deve esserci una riga corrispondente nel file `/etc/shadow`.

Per la conversione della data e dell'ora, fare riferimento al seguente formato di comando:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
# Il timestamp viene convertito in una data, "17718" indica il timestamp da
inserire.
$ date -d "1970-01-01 17718 days"

# La data viene convertita in un timestamp, "2018-07-06" indica la data da
compilare.
$ echo $(($(date --date="2018-07-06" +%s)/86400+1))
```

7.4 Proprietari dei file

Pericolo

Tutti i file appartengono necessariamente a un utente e a un gruppo.

Il gruppo primario dell'utente che crea il file è, per impostazione predefinita, il gruppo proprietario del file.

7.4.1 Comandi di modifica

comando `chown`

Il comando `chown` consente di cambiare i proprietari di un file.

```
chown [-R] [-v] login[:group] file
```

Esempi:

```
sudo chown root myfile

sudo chown albert:GroupA myfile
```

Opzione	Descrizione
`-R`	Cambia ricorsivamente i proprietari della directory e di tutti i file in essa contenuti.
`-v`	Visualizza le modifiche.

Per modificare solo l'utente proprietario:

```
sudo chown albert file
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Per modificare solo il gruppo proprietario:

```
sudo chown :GroupA file
```

Modifica dell'utente e del gruppo proprietario:

```
sudo chown albert:GroupA file
```

Nell'esempio seguente, il gruppo assegnato sarà il gruppo primario dell'utente specificato.

```
sudo chown albert: file
```

Cambia il proprietario e il gruppo di tutti i file in una directory

```
sudo chown -R albert:GroupA /dir1
```

7.4.2 comando `chgrp`

Il comando `chgrp` consente di cambiare il gruppo proprietario di un file.

```
chgrp [-R] [-v] group file
```

Esempio:

```
sudo chgrp group1 file
```

Opzione	Descrizione
`-R`	Modifica i gruppi proprietari della directory e dei suoi contenuti (ricorsivo).
`-v`	Visualizza le modifiche.

Nota

È possibile applicare a un file un proprietario e un gruppo di proprietari prendendo come riferimento quelli di un altro file:

```
chown [options] --reference=RRFILE FILE
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Per esempio:

```
chown --reference=/etc/groups /etc/passwd
```

7.5 Gestione degli utenti

7.5.1 comando `gpasswd`

Il comando `gpasswd` permette di gestire un gruppo.

```
gpasswd [option] group
```

Esempi:

```
$ sudo gpasswd -A alain GroupA
[alain]$ gpasswd -a patrick GroupA
```

Opzione	Descrizione
`-a login`	Aggiunge l'utente al gruppo. Per l'utente aggiunto, questo gruppo è un gruppo supplementare.
`-A login`	Imposta l'elenco degli utenti amministrativi.
`-d USER`	Rimuove l'utente dal gruppo.
`-M USER,...`	Imposta l'elenco dei membri del gruppo.

Il comando `gpasswd -M` agisce come una modifica, non come un'aggiunta.

```
# gpasswd GroupeA
New Password:
Re-enter new password:
```

Nota

Oltre a usare `gpasswd -a` per aggiungere utenti a un gruppo, si può anche usare `usermod -G` o `usermod -AG` menzionati prima.

7.5.2 comando `id`

Il comando `id` visualizza i nomi del gruppo di un utente.

```
id USER
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Esempio:

```
$ sudo id alain
uid=1000(alain) gid=1000(GroupA) groupes=1000(GroupA),1016(GroupP)
```

7.5.3 comando `newgrp`

Il comando `newgrp` può selezionare un gruppo, dai gruppi supplementari dell'utente, come nuovo gruppo primario **temporaneo**. Il comando `newgrp` ogni volta che viene cambiato il gruppo primario di un utente, crea una nuova **child shell** (child process). Fate attenzione! **child shell** e **sub shell** sono diverse.

```
newgrp [secondarygroups]
```

Esempio:

```
$ sudo useradd test1
$ sudo passwd test1
$ sudo groupadd groupA ; sudo groupadd groupB
$ sudo usermod -G groupA,groupB test1
$ id test1
uid=1000(test1) gid=1000(test1) groups=1000(test1),1001(groupA),1002(groupB)
$ echo $SHLVL ; echo $BASH_SUBSHELL
1
0

$ su - test1
$ touch a.txt
$ ll
-rw-rw-r-- 1 test1 test1 0 10   7 14:02 a.txt
$ echo $SHLVL ; echo $BASH_SUBSHELL
1
0

# Generate a new child shell
$ newgrp groupA
$ touch b.txt
$ ll
-rw-rw-r-- 1 test1 test1  0 10   7 14:02 a.txt
-rw-r--r-- 1 test1 groupA 0 10   7 14:02 b.txt
$ echo $SHLVL ; echo $BASH_SUBSHELL
2
0
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
# You can exit the child shell using the `exit` command
$ exit
$ logout
$ whoami
root
```

7.6 Protezione

7.6.1 commando `passwd`

Il comando `passwd` viene utilizzato per gestire una password.

```
passwd [-d] [-l] [-S] [-u] [login]
```

Esempi:

```
sudo passwd -l albert

sudo passwd -n 60 -x 90 -w 80 -i 10 patrick
```

Opzione	Descrizione
`-d`	Rimuove in modo permanente la password. Solo per root (uid=0).
`-l`	Blocca in modo permanente l'account utente. Solo per root (uid=0).
`-S`	Visualizza lo stato dell'account. Solo per root (uid=0).
`-u`	Sblocca in modo permanente l'account utente. Solo per root (uid=0).
`-e`	Fa scadere definitivamente la password. Solo per root (uid=0).
`-n DAYS`	Definisce la durata minima della password. Cambiamento permanente. Solo per root (uid=0).
`-x DAYS`	Definisce la durata massima della password. Cambiamento permanente. Solo per root (uid=0).
`-w DAYS`	Definisce il tempo di avviso prima della scadenza. Cambiamento permanente. Solo per root (uid=0).
`-i DAYS`	Definisce il ritardo prima della disattivazione quando la password scade. Cambiamento permanente. Solo per root (uid=0).

Utilizzare `password -l`, cioè aggiungere "!!" all'inizio del campo password dell'utente corrispondente nel file `/etc/shadow`.

Esempio:

- Alain cambia la sua password:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
[alain]$ passwd
```

- root cambia la password di Alain

```
sudo passwd alain
```

> **Nota**
>
> Il comando `passwd` è a disposizione degli utenti per cambiare la propria password (viene richiesta la vecchia password). L'amministratore può modificare le password di tutti gli utenti senza limitazioni.

Dovranno rispettare le restrizioni di sicurezza.

Quando si gestiscono gli account utente tramite script di shell, può essere utile impostare una password predefinita dopo la creazione dell'utente.

Questo può essere fatto passando la password al comando `passwd`.

Esempio:

```
sudo echo "azerty,1" | passwd --stdin philippe
```

> **Attenzione**
>
> La password viene inserita in chiaro, `passwd` si occupa di crittografarla.

7.6.2 comando `chage`

Il comando `chage` modifica le informazioni sulla scadenza della password utente.

```
chage [-d date] [-E date] [-I days] [-l] [-m days] [-M days] [-W days] [login]
```

Esempio:

Copyright © 2023 The Rocky Enterprise Software Foundation

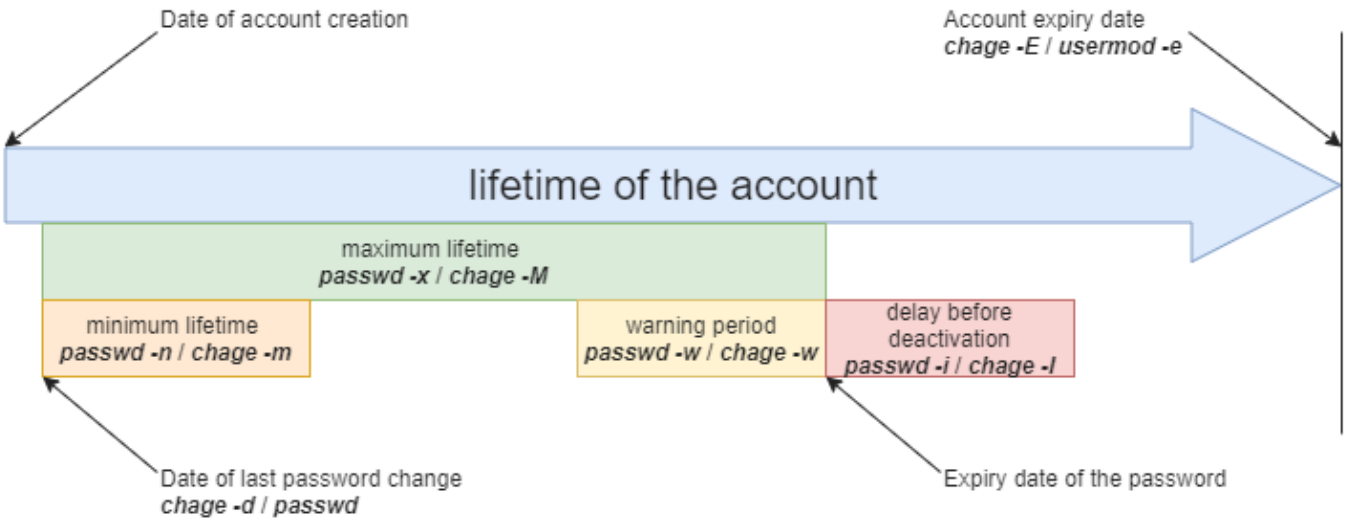
```
sudo chage -m 60 -M 90 -W 80 -I 10 alain
```

Opzione	Descrizione
`-I DAYS`	Definisce i giorni di ritardo prima della disattivazione, a password scaduta. Cambiamento permanente.
`-l`	Visualizza i dettagli della politica.
`-m DAYS`	Definisce la durata minima della password. Cambiamento permanente.
`-M DAYS`	Definisce la durata massima della password. Cambiamento permanente.
`-d LAST_DAY`	Definisce il numero di giorni dall'ultima modifica della password. È possibile utilizzare lo stile di data e ora dei giorni o lo stile YYYY-MM-DD. Cambiamento permanente.
`-E EXPIRE_DATE`	Definisce la data di scadenza dell'account. È possibile utilizzare lo stile di data e ora dei giorni o lo stile YYYY-MM-DD. Cambiamento permanente.
`-W WARN_DAYS`	Definisce il numero di giorni di avviso prima della scadenza. Cambiamento permanente.

Esempi:

```
# Il comando `chage` offre anche una modalità interattiva.
$ sudo chage philippe

# L'opzione `-d' forza la modifica della password all'accesso.
$ sudo chage -d 0 philippe
```



Copyright © 2023 The Rocky Enterprise Software Foundation

7.7 Gestione avanzata

File di configurazione:

- `/etc/default/useradd`
- `/etc/login.defs`
- `/etc/skel`

> **Nota**
>
> La modifica del file `/etc/default/useradd` viene eseguita con il comando `useradd`.
>
> Gli altri file devono essere modificati con un editor di testo.

7.7.1 file `/etc/default/useradd`

Questo file contiene le impostazioni predefinite dei valori.

> **Suggerimento**
>
> Quando si crea un utente, se le opzioni non sono specificate, il sistema utilizza i valori di default definiti in `/etc/default/useradd`.

Questo file è modificato dal comando `useradd -D` (`useradd -D` inserito senza nessun'altra opzione visualizza il contenuto del file `/etc/default/useradd`).

```
Shell > grep -v ^# /etc/default/useradd
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

Parametri	Commento
`GROUP`	Definisce il GID del gruppo primario predefinito.
`HOME`	Definisce il percorso della directory del livello superiore della home directory dell'utente comune.
`INACTIVE`	Numero di giorni di tolleranza dopo la scadenza della password. Corrisponde al 7° campo del file `/etc/shadow`. Il valore `-1` indica che il periodo di tolleranza è disattivato.
`EXPIRE`	Definisce la data di scadenza dell'account. Corrisponde all'ottavo campo del file `/etc/shadow`.
`SHELL`	Definisce l'interprete dei comandi.
`SKEL`	Definisce la struttura della directory di login.
`CREATE_MAIL_SPOOL`	Definisce la creazione della cassetta postale in `/var/spool/mail/`.

Se non si ha bisogno di un gruppo primario con lo stesso nome quando si creano gli utenti, si può fare in questo modo:

```
Shell > useradd -N test2
Shell > id test2
uid=1001(test2) gid=100(users) groups=100(users)
```

7.7.2 file `/etc/login.defs`

```
# Comment line ignored
shell > cat  /etc/login.defs
MAIL_DIR        /var/spool/mail
UMASK           022
HOME_MODE       0700
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7
UID_MIN                  1000
UID_MAX                 60000
SYS_UID_MIN               201
SYS_UID_MAX               999
GID_MIN                  1000
GID_MAX                 60000
SYS_GID_MIN               201
SYS_GID_MAX               999
CREATE_HOME     yes
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
USERGROUPS_ENAB yes
ENCRYPT_METHOD SHA512
```

`UMASK 022`: Ciò significa che il permesso di creare un file è 755 (rwxr-xr-x). Tuttavia, per motivi di sicurezza, GNU/Linux non prevede il permesso **x** per i file appena creati. Questa restrizione si applica a root (uid=0) e agli utenti ordinari (uid>=1000). Per esempio:

```
Shell > touch a.txt
Shell > ll
-rw-r--r-- 1 root root     0 Oct  8 13:00 a.txt
```

`HOME_MODE 0700`: I permessi della directory home di un utente ordinario. Non si applica alla directory home di root.

```
Shell > ll -d /root
dr-xr-x---. 10 root root 4096 Oct  8 13:12 /root

Shell > ls -ld /home/test1/
drwx------ 2 test1 test1 4096 Oct  8 13:10 /home/test1/
```

`USERGROUPS_ENAB yes`: "Quando si elimina un utente con il comando `userdel -r`, viene eliminato anche il gruppo primario corrispondente." Perché? Questo è il motivo.

7.7.3 `/etc/skel` directory

Quando viene creato un utente, vengono creati la sua home directory e i suoi file di ambiente. I file della directory `/etc/skel/` sono i modelli di file necessari per creare gli utenti.

Questi file vengono copiati automaticamente dalla directory `/etc/skel`.

- `.bash_logout`
- `.bash_profile`
- `.bashrc`

Tutti i file e le directory collocati in questa directory saranno copiati nell'albero utente al momento della loro creazione.

Copyright © 2023 The Rocky Enterprise Software Foundation

7.8 Cambiamento di identità

7.8.1 comando `su`

Il comando `su` consente di cambiare l'identità dell'utente connesso.

```
su [-] [-c command] [login]
```

Esempi:

```
$ sudo su - alain
[albert]$ su - root -c "passwd alain"
```

Opzione	Descrizione
`-`	Carica l'ambiente completo dell'utente.
`-c` comando	Esegue il comando sotto l'identità dell'utente specificato.

Se il login non è specificato, sarà `root`.

Gli utenti standard dovranno digitare la password per la nuova identità.

Suggerimento

You can use the `exit` / `logout` command to exit users who have been switched. Si noti che dopo il cambio di utente non esiste una nuova `child shell` o `sub shell`, ad esempio:

```
$ whoami
root
$ echo $SHLVL ; echo $BASH_SUBSHELL
1
0

$ su - test1
$ echo $SHLVL ; echo $BASH_SUBSHELL
1
0
```

Attenzione! `su` e `su -` sono diversi, come mostrato nell'esempio seguente:

```
$ whoami
test1
$ su root
$ pwd
/home/test1

$ env
...
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
USER=test1
PWD=/home/test1
HOME=/root
MAIL=/var/spool/mail/test1
LOGNAME=test1
...
```

```
$ whoami
test1
$ su - root
$ pwd
/root

$ env
...
USER=root
PWD=/root
HOME=/root
MAIL=/var/spool/mail/root
LOGNAME=root
...
```

Quindi, quando si vuole cambiare utente, ricordarsi di non dimenticare il `-`. Poiché non vengono caricati i file delle variabili d'ambiente necessarie, potrebbero verificarsi problemi nell'esecuzione di alcuni programmi.

Copyright © 2023 The Rocky Enterprise Software Foundation

8. File System

In questo capitolo imparerai come lavorare con i filesystems.

Obiettivi: in questo capitolo i futuri amministratori Linux impareranno a:

✔ gestire le partizioni su disco;
✔ utilizzare LVM per un migliore utilizzo delle risorse del disco;
✔ fornire agli utenti un filesystem e gestire i diritti di accesso.

e scoprire anche:

✔ come è organizzata la struttura ad albero in Linux;
✔ i diversi tipi di file offerti e come lavorare con essi;

🏁 **hardware**, **disco**, **partizione**, **lvm**, **linux**

Conoscenza: ⭐⭐
Complessità: ⭐⭐

Tempo di lettura: 20 minuti

8.1 Partizionamento

Il partizionamento consente l'installazione di diversi sistemi operativi, poiché è impossibile far convivere più sistemi operativi sulla stessa unità logica. Il partizionamento consente anche di separare i dati dal punto di vista logico (sicurezza, ottimizzazione degli accessi, ...).

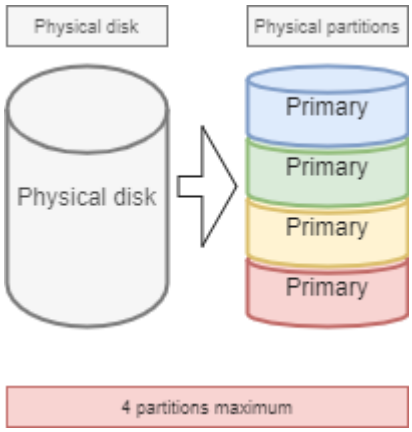
La divisione del disco fisico in volumi partizionati è registrata nella tabella delle partizioni, memorizzata nel primo settore del disco (MBR: *Master Boot Record*).

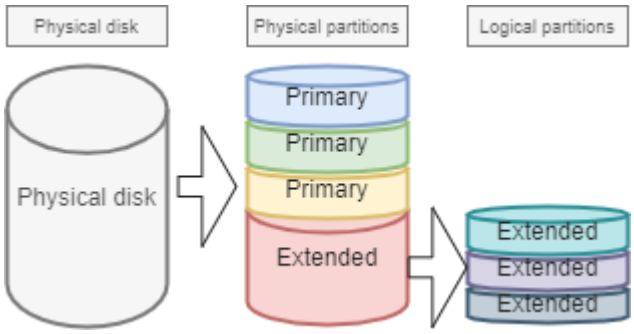
Copyright © 2023 The Rocky Enterprise Software Foundation

Per i formati di tabelle di partizione **MBR**, lo stesso disco fisico può essere suddiviso in un massimo di 4 partizioni:

- *Partizione primaria* (o partizione principale)
- *Partizione estesa*

> **Attenzione**
>
> Per ogni disco fisico può esserci una sola partizione estesa, cioè un disco fisico può avere nella tabella delle partizioni MBR fino a un massimo di:
>
> 1. Tre partizioni primarie più una partizione estesa
> 2. Quattro partizioni primarie
>
> La partizione estesa non può scrivere dati e formattare e può contenere solo partizioni logiche. Il disco fisico più grande che può essere riconosciuto dalla tabella delle partizioni MBR è di **2 TB**.





Copyright © 2023 The Rocky Enterprise Software Foundation

8.1.1 Convenzioni di denominazione per i nomi dei file del dispositivo

Nel mondo di GNU/Linux, tutto è un file. Per i dischi, vengono riconosciuti dal sistema come:

Hardware	Nome del file del dispositivo
Disco rigido IDE	/dev/hd[a-d]
Disco rigido SCSI/SATA/USB	/dev/sd[a-z]
Unità ottica	/dev/cdrom or /dev/sr0
Floppy disk	/dev/fd[0-7]
Stampante (25 pin)	/dev/lp[0-2...]
Stampante (USB)	/dev/usb/lp[0-15]
Mouse	/dev/mouse
Disco rigido virtuale	/dev/vd[a-z]

Il kernel Linux contiene i driver per la maggior parte dei dispositivi hardware.

Quelli che chiamiamo *dispositivi* sono i file, memorizzati senza `/dev`, che identificano i diversi hardware rilevati dalla scheda madre.

Il servizio udev è responsabile dell'applicazione delle convenzioni di denominazione (regole) e della loro applicazione ai dispositivi rilevati.

Per ulteriori informazioni, vedere qui.

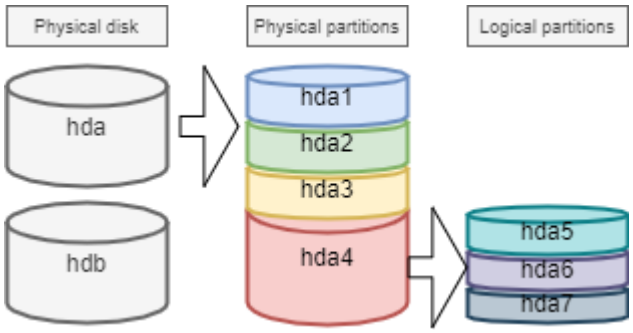
8.1.2 Numero di partizione del dispositivo

Il numero dopo il dispositivo di blocco (dispositivo di memorizzazione) indica una partizione. Per le tabelle di partizione MBR, il numero 5 deve essere la prima partizione logica.

Attenzione

Attenzione prego! Il numero di partizione citato si riferisce principalmente al numero di partizione del dispositivo a blocchi (dispositivo di archiviazione).

Copyright © 2023 The Rocky Enterprise Software Foundation



Esistono almeno due comandi per il partizionamento di un disco: `fdisk` e `cfdisk`. Entrambi i comandi hanno un menu interattivo. `cfdisk` è più affidabile e meglio ottimizzato, quindi è il migliore da usare.

L'unica ragione per usare `fdisk` è quando si vogliono elencare tutti i dispositivi logici con l'opzione `-l`. `fdisk` utilizza tabelle di partizione MBR, pertanto non è supportato per le tabelle di partizione **GPT** e non può essere usato per dischi di dimensioni superiori a **2 TB**.

```
sudo fdisk -l
sudo fdisk -l /dev/sdc
sudo fdisk -l /dev/sdc2
```

8.1.3 comando `parted`

Il comando `parted` (*partition editor*) è in grado di partizionare un disco senza gli inconvenienti di `fdisk`.

Il comando `parted` può essere usato sia dalla riga di comando che in modo interattivo. Dispone inoltre di una funzione di recupero in grado di riscrivere una tabella di partizione cancellata.

```
parted [-l] [device]
```

Come interfaccia grafica, c'è il completissimo strumento `gparted`: *Gnome PARtition EDitor*.

Il comando `gparted -l` elenca tutti i dispositivi logici di un computer.

Copyright © 2023 The Rocky Enterprise Software Foundation

Il comando `gparted`, se eseguito senza argomenti, mostrerà una modalità interattiva con le sue opzioni interne:

- `help` o un comando errato visualizzerà queste opzioni.
- `print all` in questa modalità avrà lo stesso risultato di `gparted -l` dalla riga di comando.
- `quit` per tornare al prompt.

8.1.4 comando `cfdisk`

Il comando `cfdisk` viene utilizzato per gestire le partizioni.

```
cfdisk device
```

Esempio:

```
$ sudo cfdisk /dev/sda
                                 Disk: /dev/sda
               Size: 16 GiB, 17179869184 bytes, 33554432 sectors
                       Label: dos, identifier: 0xcf173747
    Device        Boot       Start        End    Sectors   Size   Id Type
>>  /dev/sda1     *           2048    2099199    2097152     1G   83 Linux
    /dev/sda2              2099200   33554431   31455232    15G   8e Linux LVM
 lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
 x Partition type: Linux (83)                                                        x
 x     Attributes: 80                                                                x
 xFilesystem UUID: 54a1f5a7-b8fa-4747-a87c-2dd635914d60                              x
 x     Filesystem: xfs                                                               x
 x     Mountpoint: /boot (mounted)                                                   x
 mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
     [Bootable]  [ Delete ]  [ Resize ]  [  Quit  ]  [  Type  ]  [  Help  ]
     [  Write ]  [  Dump  ]
```


Copyright © 2023 The Rocky Enterprise Software Foundation

La preparazione, senza *LVM*, dei supporti fisici avviene in cinque fasi:

- Impostazione del disco fisico;
- Partizionamento dei volumi (divisione del disco, possibilità di installare più sistemi, ...);
- Creazione dei file system (permette al sistema operativo di gestire i file, la struttura ad albero, i diritti, ...);
- Montaggio dei file system (registrazione del file system nella struttura ad albero);
- Gestire l'accesso degli utenti.

8.2 Gestore di volumi logici (LVM)

Logical **V**olume **M**anager (*LVM]*)

La partizione creata dalla **partizione standard** non può regolare dinamicamente le risorse del disco rigido; una volta montata la partizione, la capacità è completamente fissa, questo vincolo è inaccettabile per il server. Anche se la partizione standard può essere espansa o ridotta forzatamente con alcuni mezzi tecnici, è facile che si verifichi una perdita di dati. LVM può risolvere questo problema molto bene. LVM è disponibile in Linux dalla versione 2.4 del kernel e le sue caratteristiche principali sono:

- Capacità del disco più flessibile;
- Movimento di dati online;
- Dischi in modalità *stripe*;
- Volumi Mirrored (ricopiati);
- Istantanee del volume_(snapshot_).

Il principio di LVM è molto semplice:

- tra il disco fisico (o la partizione del disco) e il file system viene aggiunto un livello di astrazione logica
- unire più dischi (o partizioni di dischi) in un Gruppo di Volumi**(VG**)
- eseguire le operazioni di gestione del disco su di essi attraverso una funzionalità chiamata Logical Volume**(LV**).

Copyright © 2023 The Rocky Enterprise Software Foundation

Il supporto fisico: Il supporto di memorizzazione di LVM può essere l'intero disco rigido, una partizione del disco o un array RAID. Il dispositivo deve essere convertito, o inizializzato, in LVM Physical Volume (**PV**), prima di poter eseguire ulteriori operazioni.

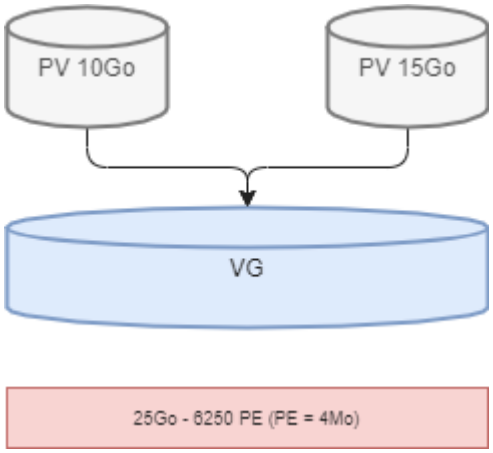
PV (Physical Volume): Il blocco logico di archiviazione di base di LVM. Per creare un volume fisico, è possibile utilizzare una partizione del disco o il disco stesso.

VG (Volume Group): Simile ai dischi fisici di una partizione standard, un VG è costituito da uno o più PV.

LV (Logical Volume): Simile alle partizioni del disco rigido nelle partizioni standard, il LV è costruito sopra il VG. È possibile impostare un file system su LV.

PE: L'unità di memoria più piccola che può essere allocata in un volume fisico, per impostazione predefinita **4MB**. È possibile specificare una dimensione aggiuntiva.

LE: La più piccola unità di memoria che può essere allocata in un Logical Volume. Nella stessa VG, PE e LE sono uguali e corrispondono uno a uno.



Lo svantaggio è che se uno dei volumi fisici va fuori servizio, tutti i volumi logici che utilizzano questo volume fisico vanno persi. È necessario utilizzare LVM sui dischi raid.

Nota

LVM è gestito solo dal sistema operativo. Pertanto il *BIOS* ha bisogno di almeno una partizione senza LVM per l'avvio.

Copyright © 2023 The Rocky Enterprise Software Foundation

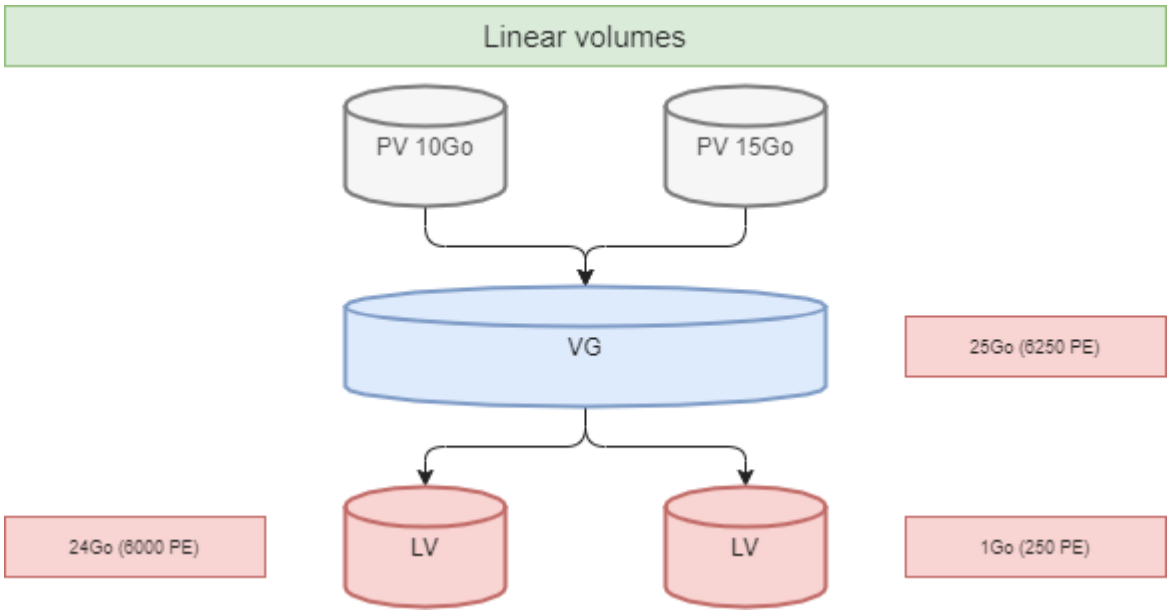
Informazione

Nel disco fisico, l'unità di memorizzazione più piccola è il **sector**; nel file system, l'unità di memorizzazione più piccola di GNU/Linux è il **block**, che è chiamato **cluster** nel sistema operativo Windows; nel RAID, l'unità di memorizzazione più piccola è il **chunk**.

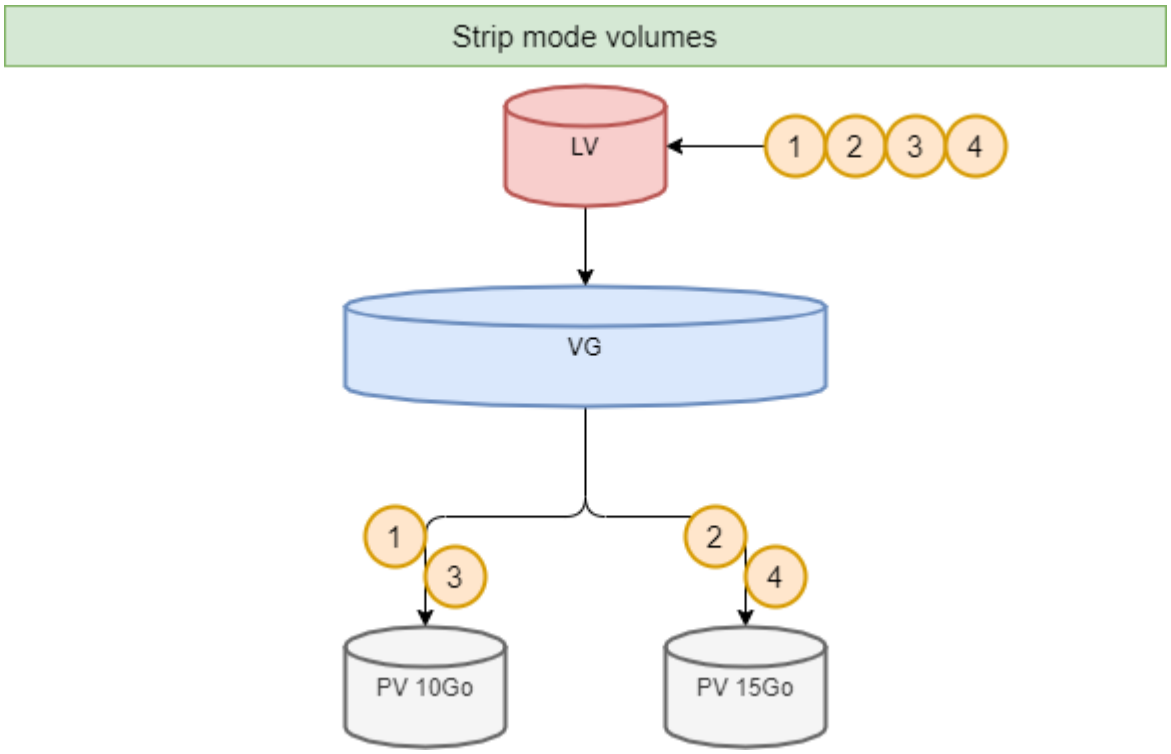
8.2.1 Il Meccanismo di Scrittura di LVM

Per l'archiviazione dei dati in **LV** esistono diversi meccanismi di memorizzazione, due dei quali sono:

- Volumi lineari;
- Volumi in modalità *stripe*;
- Volumi Mirrored.



Copyright © 2023 The Rocky Enterprise Software Foundation



8.2.2 Comandi LVM per la gestione dei volumi

I principali comandi rilevanti sono i seguenti:

Elemento	PV	VG	LV
scan	pvscan	vgscan	lvscan
create	pvcreate	vgcreate	lvcreate
display	pvdisplay	vgdisplay	lvdisplay
remove	pvremove	vgremove	lvremove
extend		vgextend	lvextend
reduce		vgreduce	lvreduce
informazioni sintetiche	pvs	vgs	lvs

comando `pvcreate`

Il comando `pvcreate` viene utilizzato per creare volumi fisici. Trasforma le partizioni (o i dischi) di Linux in volumi fisici.

```
pvcreate [-opzioni] partizione
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Esempio:

```
[root]# pvcreate /dev/hdb1
pvcreate -- physical volume « /dev/hdb1 » successfully created
```

È anche possibile utilizzare un disco intero (il che facilita l'aumento delle dimensioni del disco in ambienti virtuali, ad esempio).

```
[root]# pvcreate /dev/hdb
pvcreate -- physical volume « /dev/hdb » successfully created

# It can also be written in other ways, such as
[root]# pvcreate /dev/sd{b,c,d}1
[root]# pvcreate /dev/sd[b-d]1
```

Opzione	Descrizione
`-f`	Forza la creazione del volume (disco già trasformato in volume fisico). Usare con estrema cautela.

comando `vgcreate`

Il comando `vgcreate` viene utilizzato per creare gruppi di volumi. Raggruppa uno o più volumi fisici in un gruppo di volumi.

```
vgcreate <VG_name> <PV_name...> [opzione]
```

Esempio:

```
[root]# vgcreate volume1 /dev/hdb1
…
vgcreate – volume group « volume1 » successfully created and activated

[root]# vgcreate vg01 /dev/sd{b,c,d}1
[root]# vgcreate vg02 /dev/sd[b-d]1
```

comando `lvcreate`

Il comando `lvcreate` crea volumi logici. Il file system viene quindi creato su questi volumi logici.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
lvcreate -L size [-n name] VG_name
```

Esempio:

```
[root]# lvcreate –L 600M –n VolLog1 volume1
lvcreate -- logical volume « /dev/volume1/VolLog1 » successfully created
```

Opzione	Descrizione
`-L size`	Imposta la dimensione del volume logico in K, M o G.
`-n name`	Imposta il nome del LV. File speciale creato in `/dev/nome_volume` con questo nome.
`-l numero`	Imposta la percentuale della capacità del disco rigido da utilizzare. È possibile utilizzare anche il numero di PE. Un PE equivale a 4 MB.

Informazione

Dopo aver creato un volume logico con il comando `lvcreate`, la regola di denominazione del sistema operativo è - `/dev/VG_name/LV_name`, questo tipo di file è un soft link (altrimenti noto come link simbolico). Il file di collegamento punta a file come `/dev/dm-0` e `/dev/dm-1`.

8.2.3 Comandi LVM per visualizzare le informazioni sui volumi

comando `pvdisplay`

Il comando `pvdisplay` consente di visualizzare le informazioni sui volumi fisici.

```
pvdisplay /dev/PV_name
```

Esempio:

```
[root]# pvdisplay /dev/PV_name
```

comando `vgdisplay`

Il comando `vgdisplay` consente di visualizzare le informazioni sui gruppi di volumi.

```
vgdisplay VG_name
```

Esempio:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
[root]# vgdisplay volume1
```

comando `lvdisplay`

Il comando `lvdisplay` consente di visualizzare le informazioni sui volumi logici.

```
lvdisplay /dev/VG_name/LV_name
```

Esempio:

```
[root]# lvdisplay /dev/volume1/VolLog1
```

8.2.4 Preparazione dei supporti fisici

La preparazione con LVM del supporto fisico si articola come segue:

- Impostazione del disco fisico
- Partizione dei volumi
- **Volume fisico LVM**
- **Gruppi di volumi LVM**
- **Volumi logici LVM**
- Creazione di file system
- Montaggio dei file system
- Gestire l'accesso degli utenti

8.3 Struttura di un file system

Un **FS** del *file system* è responsabile delle seguenti azioni:

- Protezione dei diritti di accesso e modifica dei file;
- Manipolazione dei file: creazione, lettura, modifica e cancellazione;
- Individuazione dei file sul disco;
- Gestione dello spazio della partizione.

Copyright © 2023 The Rocky Enterprise Software Foundation

Il sistema operativo Linux è in grado di utilizzare diversi file system (ext2, ext3, ext4, FAT16, FAT32, NTFS, HFS, BtrFS, JFS, XFS, ...).

8.3.1 comando `mkfs`

Il comando `mkfs` (make file system) consente di creare un file system Linux.

```
mkfs [-t fstype] filesys
```

Esempio:

```
[root]# mkfs -t ext4 /dev/sda1
```

Opzione	Descrizione
`-t`	Indica il tipo di file system da utilizzare.

Attenzione

Senza un file system non è possibile utilizzare lo spazio su disco.

Ogni file system ha una struttura identica su ogni partizione. Un **Boot Sector** e un **Super block** inizializzati dal sistema, quindi una **Inode table** e un **Data block** inizializzati dall'amministratore.

Nota

L'unica eccezione è la partizione **swap**.

Copyright © 2023 The Rocky Enterprise Software Foundation

8.3.2 Boot sector

Il boot sector è il primo settore del supporto di memorizzazione avviabile, ovvero cilindro 0, traccia 0, settore 1 (1 settore equivale a 512 byte). Si compone di tre parti:

1. MBR (master boot record): 446 byte.
2. DPT (tabella di partizione del disco): 64 byte.
3. BRID (boot record ID): 2 byte.

Elemento	Descrizione
MBR	Memorizza il "boot loader" (o "GRUB"); carica il kernel, passa i parametri; fornisce un'interfaccia di menu all'avvio; trasferisce a un altro loader, ad esempio quando sono installati più sistemi operativi.
DPT	Registra lo stato della partizione dell'intero disco.
BRID	Determina se il dispositivo è utilizzabile per l'avvio.

8.3.3 Super block

La dimensione della tabella del **Super block** è definita al momento della creazione. È presente in ogni partizione e contiene gli elementi necessari per il suo utilizzo.

Descrive il File System:

- Nome del Volume Logico;
- Nome del File System;
- Tipo di File System;
- Stato del File System;
- Dimensione del File System;
- Numero di blocchi liberi;
- Puntatore all'inizio dell'elenco dei blocchi liberi;
- Dimensione dell'elenco di inode;
- Numero ed elenco degli inode liberi.

Copyright © 2023 The Rocky Enterprise Software Foundation

Una copia viene caricata nella memoria centrale non appena il sistema viene inizializzato. Questa copia viene aggiornata non appena viene modificata e il sistema la salva periodicamente (comando `sync`).

Quando il sistema si arresta, copia anche questa tabella in memoria nel suo blocco.

8.3.4 Tabella degli inode

La dimensione della **tabella degli inode** è definita al momento della sua creazione ed è memorizzata nella partizione. È costituito da record, chiamati inode, corrispondenti ai file creati. Ogni record contiene gli indirizzi dei blocchi di dati che compongono il file.

> **Nota**
>
> Il numero di inode è unico all'interno di un file system.

Una copia viene caricata nella memoria centrale non appena il sistema viene inizializzato. Questa copia viene aggiornata non appena viene modificata e il sistema la salva periodicamente (comando `sync`).

Quando il sistema si arresta, copia anche questa tabella in memoria nel suo blocco.

Un file è gestito dal suo numero di inode.

> **Nota**
>
> La dimensione della tabella degli inode determina il numero massimo di file che il FS può contenere.

Copyright © 2023 The Rocky Enterprise Software Foundation

Informazioni presenti nella *tabella degli inode*:

- Numero di inode;
- Tipo di file e autorizzazioni di accesso;
- Numero di identificazione del proprietario;
- Numero di identificazione del gruppo proprietario;
- Numero di collegamenti in questo file;
- Dimensione del file in byte;
- Data dell'ultimo accesso al file;
- Data dell'ultima modifica del file;
- Data dell'ultima modifica dell'inode (= creazione);
- Tabella di diversi puntatori (tabella dei blocchi) ai blocchi logici contenenti i pezzi del file.

8.3.5 Data block

La sua dimensione corrisponde al resto dello spazio disponibile della partizione. Quest'area contiene i cataloghi corrispondenti a ciascuna directory e i blocchi di dati corrispondenti ai contenuti dei file.

Per garantire la coerenza del file system, un'immagine del superblocco e della tabella degli inode viene caricata in memoria (RAM) al momento del caricamento del sistema operativo, in modo che tutte le operazioni di I/O vengano effettuate attraverso queste tabelle di sistema. Quando l'utente crea o modifica i file, questa immagine di memoria viene aggiornata per prima. Il sistema operativo deve quindi aggiornare regolarmente il superblocco del disco logico `(` comando `sync`).

Queste tabelle vengono scritte sul disco rigido quando il sistema viene spento.

Attenzione

In caso di arresto improvviso, il file system potrebbe perdere la sua consistenza e causare la perdita di dati.

Copyright © 2023 The Rocky Enterprise Software Foundation

8.3.6 Riparazione del file system

È possibile verificare la consistenza di un file system con il comando `fsck`.

In caso di errori, vengono proposte soluzioni per riparare le incongruenze. Dopo la riparazione, i file che rimangono senza voci nella tabella degli inode vengono allegati alla cartella `/lost+found` dell'unità logica.

comando `fsck`

Il comando `fsck` è uno strumento di controllo e riparazione dell'integrità in modalità console per i file system di Linux.

```
fsck [-sACVRTNP] [ -t fstype ] filesys
```

Esempio:

```
[root]# fsck /dev/sda1
```

Per controllare la partizione root, è possibile creare un file `forcefsck` e riavviare o eseguire `shutdown` con l'opzione `-F`.

```
[root]# touch /forcefsck
[root]# reboot
or
[root]# shutdown –r -F now
```

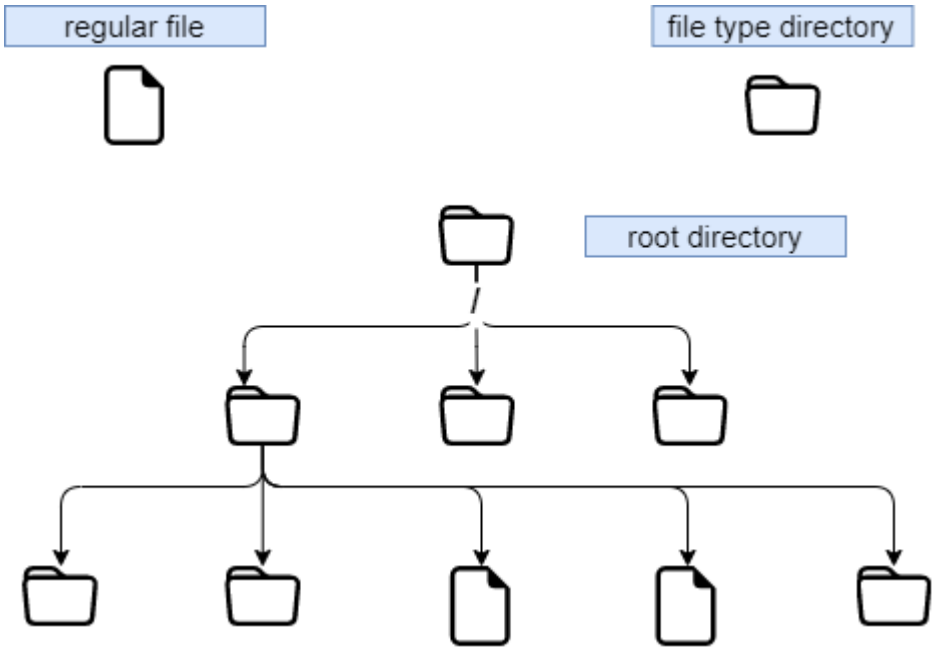
Attenzione

La partizione da controllare deve essere smontata.

8.4 Organizzazione di un file system

Per definizione, un file system è una struttura ad albero di directory costruita a partire da una directory principale (un dispositivo logico può contenere un solo file system).

Copyright © 2023 The Rocky Enterprise Software Foundation



Nota

In Linux tutto è un file.

Documento di testo, directory, binario, partizione, risorsa di rete, schermo, tastiera, kernel Unix, programma utente, ...

Copyright © 2023 The Rocky Enterprise Software Foundation

Linux rispetta lo standard **FHS***(Filesystems Hierarchy Standard*) (vedi `man hier`) che definisce i nomi delle cartelle e i loro ruoli.

Directory	Funzionalità	Abbreviazione di
`/`	Contiene directory speciali	
`/boot`	File relativi all'avvio del sistema	
`/sbin`	Comandi necessari per l'avvio e la riparazione del sistema	*binari di sistema*
`/bin`	Eseguibili dei comandi di base del sistema	*binari*
`/usr/bin`	Comandi di amministrazione del sistema	
`/lib`	Librerie condivise e moduli del kernel	*librerie*
`/usr`	Tutto ciò che non è necessario per il funzionamento minimo del sistema	*Risorse di sistema UNIX*
`/mnt`	Per il montaggio di SF temporanei	*mount*
`/media`	Per il montaggio di supporti rimovibili	
`/misc`	Per montare la directory condivisa del servizio NFS.	
`/root`	Directory di accesso dell'amministratore	
`/home`	Dati utente	
`/tmp`	File temporanei	*temporanei*
`/dev`	File speciali del dispositivo	*dispositivo*
`/etc`	File di configurazione e scripts	*configurazione del testo modificabile*
`/opt`	Specifico per le applicazioni installate	*opzionale*
`/proc`	File system virtuale che rappresenta diversi processi	*processi*
`/var`	File variabili vari	*variabili*
`/sys`	File system virtuale, simile a /proc	
`/run`	Questo è /var/run	
`/srv`	Service Data Directory	*servizio*

- Per eseguire un montaggio o uno smontaggio, a livello di albero, non ci si deve trovare sotto il suo punto di montaggio.
- Il montaggio su una directory non vuota non cancella il contenuto. È solo nascosta.
- Solo l'amministratore può eseguire il montaggio.
- I punti di montaggio da montare automaticamente all'avvio devono essere inseriti in `/etc/fstab`.

Copyright © 2023 The Rocky Enterprise Software Foundation

8.4.1 file `/etc/fstab`

Il file `/etc/fstab` viene letto all'avvio del sistema e contiene i montaggi da eseguire. Ogni file system da montare è descritto su una singola riga, con i campi separati da spazi o tabulazioni.

> **Nota**
>
> Le righe vengono lette in sequenza (`fsck`, `mount`, `umount`).

```
/dev/mapper/VolGroup-lv_root   /         ext4    defaults        1   1
UUID=46….92                    /boot     ext4    defaults        1   2
/dev/mapper/VolGroup-lv_swap   swap      swap    defaults        0   0
tmpfs                          /dev/shm  tmpfs   defaults        0   0
devpts                         /dev/pts  devpts  gid=5,mode=620  0   0
sysfs                          /sys      sysfs   defaults        0   0
proc                           /proc     proc    defaults        0   0
  1                              2         3        4            5   6
```

Colonna	Descrizione
1	Dispositivo del file system (`/dev/sda1`, UUID=..., ...)
2	Nome del punto di montaggio, **percorso assoluto** (eccetto **swap**)
3	Tipo di filesystem (ext4, swap, ...)
4	Opzioni speciali per il montaggio (`default`, `ro`, ...)
5	Abilitare o disabilitare la gestione del backup (0:non eseguito il backup, 1:eseguito il backup)
6	Ordine di controllo quando si controlla lo SF con il comando `fsck` (0:nessun controllo, 1:priorità, 2:non priorità)

Il comando `mount -a` consente di eseguire il montaggio automatico in base al contenuto del file di configurazione `/etc/fstab`; le informazioni montate vengono poi scritte in `/etc/mtab`.

Copyright © 2023 The Rocky Enterprise Software Foundation

Attenzione

Solo i punti di montaggio elencati in `/etc/fstab` saranno montati al riavvio. In generale, si sconsiglia di scrivere le unità flash USB e i dischi rigidi rimovibili nel file `/etc/fstab`, perché quando il dispositivo esterno viene scollegato e riavviato, il sistema segnalerà che il dispositivo non può essere trovato, con conseguente mancato avvio. Quindi cosa dovrei fare? Montaggio temporaneo, ad esempio:

```
Shell > mkdir /mnt/usb
Shell > mount -t  vfat  /dev/sdb1  /mnt/usb

# Read the information of the USB flash disk
Shell > cd /mnt/usb/

# When not needed, execute the following command to pull out the USB flash disk
Shell > umount /mnt/usb
```

Informazione

È possibile fare una copia del file `/etc/mtab` o copiare il suo contenuto in `/etc/fstab`. Se si desidera visualizzare l'UUID del numero di partizione del dispositivo, digitare il seguente comando: `lsblk -o name,uuid`. UUID è l'abbreviazione di `Universally Unique Identifier`.

8.4.2 Comandi di gestione del montaggio

comando di `montaggio`

Il comando `mount` consente di montare e visualizzare le unità logiche nella struttura.

```
mount [-option] [device] [directory]
```

Esempio:

```
[root]# mount /dev/sda7 /home
```

Opzione	Descrizione
`-n`	Imposta il montaggio senza scrivere in `/etc/mtab`.
`-t`	Indica il tipo di file system da utilizzare.
`-a`	Monta tutti i filesystem menzionati in `/etc/fstab`.
`-r`	Monta il file system in sola lettura (equivalente a `-o ro`).
`-w`	Monta il file system in lettura/scrittura, per impostazione predefinita (equivalente a `-o rw`).
`-o`	Argomento seguito da un elenco di opzioni separate da virgole (`remount`, `ro`, ...).

Copyright © 2023 The Rocky Enterprise Software Foundation

> **Nota**
>
> Il solo comando `mount` visualizza tutti i file system montati.

comando `umount`

Il comando `umount` viene utilizzato per smontare le unità logiche.

```
umount [-option] [device] [directory]
```

Esempio:

```
[root]# umount /home
[root]# umount /dev/sda7
```

Opzione	Descrizione
`-n`	Imposta la rimozione del montaggio senza scrivere in `/etc/mtab`.
`-r`	Rimonta in sola lettura se `umount` fallisce.
`-f`	Forza la rimozione del montaggio.
`-a`	Rimuove i montaggi di tutti i filesystem menzionati in `/etc/fstab`.

> **Nota**
>
> Durante lo smontaggio, non si deve rimanere al di sotto del punto di montaggio. In caso contrario, viene visualizzato il seguente messaggio di errore: `device is busy`.

8.5 Convenzione di denominazione dei file

Come in ogni sistema, per potersi orientare nella struttura ad albero e nella gestione dei file, è importante rispettare le regole di denominazione dei file.

- I file sono codificati con 255 caratteri;
- È possibile utilizzare tutti i caratteri ASCII;
- Le lettere maiuscole e minuscole sono differenziate;
- La maggior parte dei file non ha il concetto di estensione. Nel mondo GNU/Linux, la maggior parte delle estensioni dei file non è richiesta, tranne alcune (ad esempio, .jpg, .mp4, .gif, ecc.).

Copyright © 2023 The Rocky Enterprise Software Foundation

I gruppi di parole separati da spazi devono essere racchiusi tra virgolette:

```
[root]# mkdir "working dir"
```

Nota

Sebbene non ci sia nulla di tecnicamente sbagliato nel creare un file o una directory con uno spazio al suo interno, è generalmente una "best practice" da evitare e sostituire ogni spazio con un trattino basso.

Nota

Il **.** all'inizio del nome del file serve solo a nasconderlo a un semplice `ls`.

Esempi di convenzione sull'estensione dei file:

- `.c`: file sorgente in linguaggio C;
- `.h`: file di intestazione C e Fortran;
- `.o`: file oggetto in linguaggio C;
- `.tar`: file di dati archiviati con l'utilità `tar`;
- `.cpio`: file di dati archiviati con l'utilità `cpio`;
- `.gz`: file di dati compressi con l'utilità `gzip`;
- `.tgz`: file di dati archiviati con l'utilità `tar` e compressi con l'utilità `gzip`;
- `.html`: pagina web.

8.5.1 Dettagli del nome di un file

Copyright © 2023 The Rocky Enterprise Software Foundation

```
[root]# ls -liah /usr/bin/passwd
266037 -rwsr-xr-x 1 root 59K mars 22 2019 /usr/bin/passwd
1 2 3 4 5 6 7 8 9
```

Parte	Descrizione
1	Numero di inode
2	Tipo di file (primo carattere del blocco di 10), "-" significa che si tratta di un file ordinario.
3	Diritti di accesso (ultimi 9 caratteri del blocco di 10)
4	Se si tratta di una directory, questo numero rappresenta il numero di sottodirectory presenti nella directory, comprese quelle nascoste. Se si tratta di un file, indica il numero di collegamenti, quando il numero è 1, ovvero, c'è un solo collegamento, quello relativo al file stesso.
5	Nome del proprietario
6	Nome del gruppo
7	Dimensione (byte, kilo, mega)
8	Data dell'ultimo aggiornamento
9	Nome del file

Nel mondo GNU/Linux esistono sette tipi di file:

Tipi di file	Descrizione
-	Rappresenta un file ordinario. Compresi i file di testo semplice (ASCII); file binari (binario); file in formato dati (dati); vari file compressi.
d	Rappresenta un file di directory.
b	Rappresenta un file di dispositivo a blocchi. Include tutti i tipi di dischi rigidi, unità USB e così via.
c	Rappresenta un file di dispositivo di caratteri. Dispositivo di interfaccia della porta seriale, come il mouse, la tastiera, ecc.
s	Rappresenta un file socket. Si tratta di un file appositamente utilizzato per la comunicazione di rete.
p	Rappresenta un file pipe. È un tipo di file speciale. Lo scopo principale è quello di risolvere gli errori causati da più programmi che accedono a un file contemporaneamente. FIFO è l'abbreviazione del first-in-first out.
l	I file soft link, chiamati anche file di collegamento simbolico, sono simili ai collegamenti di Windows. File di collegamento rigido, noto anche come file di collegamento fisico.

Descrizione supplementare della directory

In ogni directory ci sono due file nascosti: **.** e **..**. È necessario utilizzare `ls -al` per visualizzarli, ad esempio:

```
# . Indica che nella directory corrente, ad esempio, è necessario eseguire uno
script in una directory, di solito:
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
Shell > ./scripts

# .. rappresenta la directory un livello sopra la directory corrente, ad
esempio:
Shell > cd /etc/
Shell > cd ..
Shell > pwd
/

# Per una directory vuota, la sua quarta parte deve essere maggiore o uguale a
2. Perché ci sono "." e ".."
Shell > mkdir /tmp/t1
Shell > ls -ldi /tmp/t1
1179657 drwxr-xr-x 2 root root 4096 Nov 14 18:41 /tmp/t1
```

File speciali

Per comunicare con le periferiche (dischi rigidi, stampanti...), Linux utilizza dei file di interfaccia chiamati file speciali (*device file* o *special file*). Consentono l'identificazione da parte delle periferiche.

Questi file sono speciali perché non contengono dati, ma specificano la modalità di accesso per comunicare con il dispositivo.

Sono definiti in due modalità:

- modalità **a blocchi**;
- modalità a **carattere**.

```
# Block device file
Shell > ls -l /dev/sda
brw-------   1   root  root  8, 0 jan 1 1970 /dev/sda

# Character device file
Shell > ls -l /dev/tty0
crw-------   1   root  root  8, 0 jan 1 1970 /dev/tty0
```


Copyright © 2023 The Rocky Enterprise Software Foundation

File di comunicazione

Si tratta dei file pipe (*pipes*) e *socket*.

- I **file Pipe** passano le informazioni tra i processi tramite FIFO_(First In, First Out_). Un processo scrive informazioni transitorie su un file *pipe* e un altro le legge. Dopo la lettura, le informazioni non sono più accessibili.
- I **file socket** consentono la comunicazione bidirezionale tra processi (su sistemi locali o remoti). Utilizzano un *inode* del file system.

File di collegamento

Questi file danno la possibilità di assegnare diversi nomi logici allo stesso file fisico. Viene quindi creato un nuovo punto di accesso al file.

Esistono due tipi di file di collegamento:

- File di collegamento soft, chiamato anche file di collegamento simbolico;
- File di collegamento hard, chiamato anche file di collegamento fisico.

Le loro caratteristiche principali sono:

Tipi di link	Descrizione
file soft link	Una scorciatoia simile a quella di Windows. Ha un permesso di 777 e punta al file originale. Quando il file originale viene eliminato, il file collegato e il file originale vengono visualizzati in rosso.
File hard link	Il file originale ha lo stesso numero di *inode* del file collegato. Ha lo stesso numero *inode* del file collegato fisicamente. Possono essere aggiornati in modo sincrono, includendo il contenuto del file e quando è stato modificato. Non è possibile attraversare le partizioni, né i file system. Non può essere utilizzato per le directory.

Esempi specifici sono i seguenti:

```
# Permissions and the original file to which they point
Shell > ls -l /etc/rc.locol
lrwxrwxrwx 1 root root 13 Oct 25 15:41 /etc/rc.local -> rc.d/rc.local

# When deleting the original file. "-s" represents the soft link option
Shell > touch /root/Afile
Shell > ln -s /root/Afile /root/slink1
Shell > rm -rf /root/Afile
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
[root@Ansible ~]# ls -l /root/slink1
lrwxrwxrwx 1 root root 11 11月 14 18:05 /root/slink1 -> /root/Afile
[root@Ansible ~]#
```

```
Shell > cd /home/paul/
Shell > ls –li letter
666 –rwxr--r-- 1 root root … letter

# The ln command does not add any options, indicating a hard link
Shell > ln /home/paul/letter /home/jack/read

# The essence of hard links is the file mapping of the same inode number in
different directories.
Shell > ls –li /home/*/*
666 –rwxr--r-- 2 root root … letter
666 –rwxr--r-- 2 root root … read

# If you use a hard link to a directory, you will be prompted:
Shell > ln  /etc/  /root/etc_hardlink
ln: /etc: hard link not allowed for directory
```

8.6 Attributi dei file

Linux è un sistema operativo multiutente in cui il controllo dell'accesso ai file è essenziale.

Questi controlli sono funzioni di:

- permessi di accesso ai file ;
- utenti (*ugo Users Groups Others*).

8.6.1 Permessi di base di file e directory

La descrizione dei **permessi dei file** è la seguente:

Permessi dei file	Descrizione
r	Leggi. Consente la lettura di un file (`cat`, `less`, ...) e la copia di un file (`cp`, ...).
w	Scrivere. Consente di modificare il contenuto del file (`cat`, `>>`, `vim`, ...).
x	Eseguire. Considera il file come un file **eseguibile**(binario o script).
-	Nessun diritto

Copyright © 2023 The Rocky Enterprise Software Foundation

La descrizione dei **permessi delle directory** è la seguente:

Autorizzazioni delle directory	Descrizione
r	Leggi. Consente di leggere il contenuto di una directory (`ls -R`).
w	Scrivere. Permette di creare ed eliminare file/directory in questa directory, come i comandi `mkdir`, `rmdir`, `rm`, `touch` e così via.
x	Eseguire. Consente la discesa nella directory(`cd`).
-	Nessun diritto

Informazione

Per i permessi di una directory, `r` e `x` di solito appaiono contemporaneamente. Lo spostamento o la ridenominazione di un file dipende dal fatto che la directory in cui si trova il file abbia il permesso `w`, così come l'eliminazione di un file.

8.6.2 Tipo di utente corrispondente all'autorizzazione di base

Tipo utente	Descrizione
u	Proprietario
g	Gruppo proprietario
o	Altri utenti

Informazione

In alcuni comandi è possibile designarli tutti con **a** (*all*). **a = ugo**.

8.6.3 Gestione degli attributi

La visualizzazione dei diritti si effettua con il comando `ls -l`. Si tratta degli ultimi 9 caratteri di un blocco di 10. Più precisamente 3 volte 3 caratteri.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
[root]# ls -l /tmp/myfile
-rwxrw-r-x  1  root  sys  ... /tmp/myfile
  1  2  3       4     5
```

Parte	Descrizione
1	Permessi del proprietario (**u**ser), qui `rwx`
2	Permessi del gruppo proprietario (**g**roup), qui `rw-`
3	Permessi di altri utenti (**o**thers), qui `r-x`
4	Proprietario del file
5	Gruppo proprietario del file

Per impostazione predefinita, il *proprietario* di un file è colui che lo ha creato. Il *gruppo* del file è il gruppo del proprietario che ha creato il file. Gli *altri* sono quelli che non rientrano nei casi precedenti.

Gli attributi vengono modificati con il comando `chmod`.

Solo l'amministratore e il proprietario di un file possono modificarne i diritti.

comando `chmod`

Il comando `chmod` consente di modificare i permessi di accesso a un file.

```
chmod [opzione] modalità file
```

Opzione	Osservazione
`-R`	Modifica ricorsivamente i permessi della directory e di tutti i file in essa contenuti.

Attenzione

I diritti dei file e delle directory non sono dissociati. Per alcune operazioni, sarà necessario conoscere i diritti della directory contenente il file. Un file protetto da scrittura può essere cancellato da un altro utente, purché i diritti della directory che lo contiene consentano a questo utente di eseguire questa operazione.

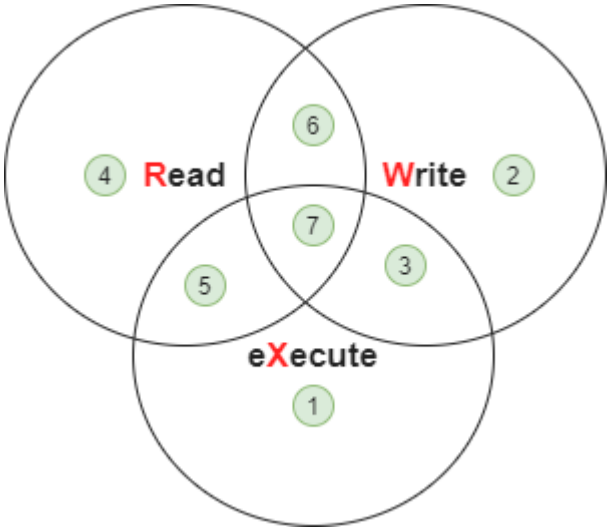
L'indicazione della modalità può essere una rappresentazione ottale (ad esempio `744`) o una rappresentazione simbolica ([`ugoa`][`+=-`][`rwxst`]).

Copyright © 2023 The Rocky Enterprise Software Foundation

RAPPRESENTAZIONE OTTALE (O NUMERICA):

Numero	Descrizione
4	r
2	s
1	x
0	-

Sommare i tre numeri per ottenere un'autorizzazione di tipo utente. ad es. **755=rwxr-xr-x**.



user			group			other		
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

Rights 777

user			group			other		
r	w	x	r	-	-	-	-	x
4	2	1	4	-	-	-	-	1

Rights 741

Informazione

A volte si vedrà `chmod 4755`. Il numero 4 si riferisce al permesso speciale **set uid**. I permessi speciali non verranno per il momento ampliati in questa sede, ma solo come nozioni di base.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
[root]# ls -l /tmp/fil*
-rwxrwx--- 1 root root … /tmp/file1
-rwx--x--- 1 root root … /tmp/file2
-rwx--xr-- 1 root root … /tmp/file3

[root]# chmod 741 /tmp/file1
[root]# chmod -R 744 /tmp/file2
[root]# ls -l /tmp/fic*
-rwxr----x 1 root root … /tmp/file1
-rwxr--r-- 1 root root … /tmp/file2
```

RAPPRESENTAZIONE SIMBOLICA

Questo metodo può essere considerato come un'associazione "letterale" tra un tipo di utente, un operatore e i diritti.

Who?	
User	u
Group	g
Other	o
All	a

Operation	
add	+
remove	-
replace	=

Rights	
Read	r
Write	w
eXecute	x

```
[root]# chmod -R u+rwx,g+wx,o-r /tmp/file1
[root]# chmod g=x,o-r /tmp/file2
[root]# chmod -R o=r /tmp/file3
```

8.7 Diritti predefiniti e maschera

Quando un file o una directory viene creata, ha già dei permessi.

- Per una directory: `rwxr-xr-x` o *755*.
- Per un file: `rw-r-r-` o *644*.

Questo comportamento è definito dalla **maschera predefinita**.

Il principio è quello di rimuovere il valore definito dalla maschera al massimo dei diritti senza il diritto di esecuzione.

Copyright © 2023 The Rocky Enterprise Software Foundation

Per una directory :

Max rights		Umask		Effective rights
777	-	022	=	755
rwxrwxrwx		----w--w-		rwxr-xr-x

Per un file, i diritti di esecuzione vengono rimossi:

Effective rights		EXecution rights		File rights
755	-	111	=	644
rwxr-xr-x		--x--x--x		rw-rw-rw-

Informazione

Il file `/etc/login.defs' definisce l'UMASK predefinito, con un valore di **022**. Ciò significa che il permesso di creare un file è 755 (rwxr-xr-x). Tuttavia, per motivi di sicurezza, GNU/Linux non prevede il permesso **x** per i file appena creati; questa restrizione si applica a root (uid=0) e agli utenti ordinari (uid>=1000).

```
# root user
Shell > touch a.txt
Shell > ll
-rw-r--r-- 1 root root     0 Oct  8 13:00 a.txt
```

8.7.1 comando `umask`

Il comando `umask` consente di visualizzare e modificare la maschera.

```
umask [opzione] [modalità]
```

Esempio:

```
$ umask 033
$ umask
0033
$ umask -S
u=rwx,g=r,o=r
$ touch umask_033
$ ls -la  umask_033
-rw-r--r-- 1 rockstar rockstar 0 nov.   4 16:44 umask_033
$ umask 025
$ umask -S
u=rwx,g=rx,o=w
$ touch umask_025
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ ls -la  umask_025
-rw-r---w- 1 rockstar rockstar 0 nov.   4 16:44 umask_025
```

Opzione	Descrizione
`-S`	Visualizzazione simbolica dei diritti dei file.

Attenzione

`umask` non influisce sui file esistenti. `umask -S` visualizza i diritti dei file (senza il diritto di esecuzione) dei file che verranno creati. Quindi, non è la visualizzazione della maschera utilizzata per sottrarre il valore massimo.

Nota

`umask` modifica la maschera fino alla disconnessione.

Informazione

Il comando `umask` appartiene ai comandi incorporati di bash, quindi quando si usa `man umask`, vengono visualizzati tutti i comandi incorporati. Se si desidera visualizzare solo la guida di `umask`, è necessario utilizzare il comando `help umask`.

Per mantenere il valore, è necessario modificare i seguenti file di profilo:

Per tutti gli utenti:

- `/etc/profile`
- `/etc/bashrc`

Per un particolare utente:

- `~/.bashrc`

Quando viene scritto il file di cui sopra, esso sovrascrive il parametro **UMASK** di `/etc/login.defs`. Se si desidera migliorare la sicurezza del sistema operativo, è possibile impostare umask a **027** o **077**.

Copyright © 2023 The Rocky Enterprise Software Foundation

9. Gestione dei processi

In questo capitolo si apprenderà come lavorare con i processi.

Obiettivi : In questo capitolo, futuri amministratori Linux impareranno come:

✔ Riconoscere il `PID` e il `PPID` di un processo;
✔ Visualizzare e cercare processi;
✔ Gestire i processi.

🏁 **processi**, **linux**

Conoscenza: ⭐⭐
Complessità: ⭐

Tempo di lettura: 20 minuti

9.1 Generalità

Un sistema operativo è costituito da processi. Questi processi vengono eseguiti in un ordine specifico e sono correlati. Ci sono due categorie di processi, quelli focalizzati sull'ambiente utente e quelli focalizzati sull'ambiente hardware.

Quando viene eseguito un programma, Il sistema creerà un processo posizionando i dati del programma e il codice in memoria e creando una **runtime stack**. Un processo è un'istanza di un programma con un ambiente di processore associato (contatore ordinale, registri, ecc.) e un ambiente di memoria.

Ogni processo ha:

- un *PID* : ***P**rocess **ID**entifier*, un identificatore di processo unico
- un *PPID* : ***P**arent **P**rocess **ID**entifier*, identificatore univoco del processo genitore

Copyright © 2023 The Rocky Enterprise Software Foundation

Da filiazioni successive, il processo `init` è il padre di tutti i processi.

- Un processo è sempre creato da un processo genitore
- Un processo genitore può avere più processi figlio

C'è una relazione genitore/figlio tra i processi. Un processo figlio è il risultato di un genitore che chiama la primitiva *fork()* e duplica il suo codice per creare un figlio. Il *PID* del processo figlio viene restituito al processo genitore in modo che possa comunicare. Ogni processo figlio ha l'identificatore del suo processo genitore, il *PPID*.

Il numero *PID* rappresenta il processo al momento dell'esecuzione. Quando il processo finisce, il numero è di nuovo disponibile per un altro processo. Eseguendo lo stesso comando più volte produrrà un diverso *PID* ogni volta.!!! Note "Nota"

```
I processi non devono essere confusi con i _threads_. Ogni processo ha il suo
contesto di memoria (risorse e spazio degli indirizzi), mentre i _thread_ dello
stesso processo condividono questo contesto.
```

9.2 Visualizzazione dei processi

Il comando `ps` visualizza lo stato dei processi in esecuzione.

```
ps [-e] [-f] [-u login]
```

Esempio:

```
# ps -fu root
```

Opzione	Descrizione
`-e`	Visualizza tutti i processi.
`-f`	Visualizza ulteriori informazioni.
`-u` login	Visualizza i processi dell'utente.

Copyright © 2023 The Rocky Enterprise Software Foundation

Alcune opzioni aggiuntive:

Opzione	Descrizione
`-g`	Visualizza i processi nel gruppo.
`-t tty`	Visualizza i processi in esecuzione dal terminale.
`-p PID`	Visualizza le informazioni del processo.
`-H`	Visualizza le informazioni in una struttura ad albero.
`-I`	Visualizza ulteriori informazioni.
`--sort COL`	Ordina il risultato secondo una colonna.
`--headers`	Visualizza l'intestazione di ogni pagina del terminale.
`--format "%a %b %c"`	Personalizza il formato di visualizzazione dell'uscita.

Senza un'opzione specificata, il comando `ps` visualizza solo i processi in esecuzione sul terminale corrente.

Il risultato viene visualizzato nelle seguenti colonne:

```
# ps -ef
UID  PID PPID C STIME  TTY TIME      CMD
root 1   0    0 Jan01  ?   00:00/03  /sbin/init
```

Colonna	Descrizione
`UID`	Utente proprietario.
`PID`	Identificatore di processo.
`PPID`	Identificatore del processo genitore.
`C`	Priorità del processo.
`STIME`	Data e ora di esecuzione.
`TTY`	Terminale di esecuzione.
`TIME`	Durata di elaborazione.
`CMD`	Comando eseguito.

Il comportamento del controllo può essere completamente personalizzato:

```
# ps -e --format "%P %p %c %n" --sort ppid --headers
 PPID   PID COMMAND          NI
    0     1 systemd           0
    0     2 kthreadd          0
    1   516 systemd-journal   0
    1   538 systemd-udevd     0
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
     1   598 lvmetad           0
     1   643 auditd           -4
     1   668 rtkit-daemon      1
     1   670 sssd              0
```

9.3 Tipi di processi

Il processo dell'utente:

- è iniziato da un terminale associato a un utente
- accede alle risorse tramite richieste o daemons

I processi di sistema sono quindi chiamati daemons (***D**isk **A**nd **E**xecution **MON**itor*)

- è iniziato dal sistema
- non è associato a nessun terminale, ed è di proprietà di un utente di sistema (spesso `root`);
- è caricato al momento dell'avvio, risiede in memoria, e sta aspettando una chiamata
- è solitamente identificato dalla lettera `d` associato al nome del processo

I processi di sistema sono quindi chiamati daemons (***D**isk **A**nd **E**xecution **MON**itor*).

9.4 Autorizzazioni e diritti

Le credenziali dell'utente vengono passate al processo creato quando viene eseguito un comando.

Per impostazione predefinita., l'attuale `UID` e `GID` (del processo) sono quindi identici al **effettivo** `UID` e `GID` (il `UID` e `GID` dell'utente che ha eseguito il comando).

Quando un `SUID` (e/o `SGID`) è impostato su un comando, l'attuale `UID` (e/o `GID`) diventa quello del proprietario (e/o gruppo proprietario) del comando e non più quello dell'utente o del gruppo di utenti che ha emesso il comando. Effettivo e reale **UIDs** sono quindi **differenti**.

Copyright © 2023 The Rocky Enterprise Software Foundation

Ogni volta che si accede a un file, il sistema controlla i diritti del processo in base ai suoi effettivi identificatori.

9.5 Gestione dei processi

Un processo non può essere eseguito indefinitamente, perchè questo sarebbe a discapito di altri processi in esecuzione e impedirebbe il multitasking.

Pertanto, il tempo di elaborazione totale disponibile viene suddiviso in piccoli intervalli e ogni processo (con una priorità) accede al processore in modo sequenziale. Il processo prenderà diversi stati durante la sua vita tra gli stati:

- pronto: in attesa della disponibilità del processo
- in esecuzione: accede al processore
- sospeso: aspettando un I/O (input/output);
- fermato: aspettando un segnale da un altro processo
- zombie: richiesta di distruzione
- morto: il padre del processo chiude il suo processo figlio

La sequenza di chiusura del processo è la seguente:

1. Chiusura dei file aperti
2. Rilascio della memoria usata
3. Invio di un segnale ai processi genitore e figlio

Quando un genitore termina, i suoi figli diventano orfani. Vengono quindi adottati dal processo `init`, che provvederà a distruggerli.

9.5.1 La priorità di un processo

Il processore lavora in time sharing, con ogni processo che occupa una quantità di tempo del processore.

I processi sono classificati per priorità, il cui valore varia da **-20** (la priorità più alta) a **+19** (la priorità più bassa).

La priorità predefinita di un processo è **0**.

Copyright © 2023 The Rocky Enterprise Software Foundation

9.5.2 Modalità di funzionamento

I processi possono essere eseguiti in due modi:

- **sincrona**: l'utente perde l'accesso alla shell durante l'esecuzione del comando. Il prompt dei comandi riappare alla fine dell'esecuzione del processo.
- **asincrona**: il processo viene elaborato in background. Il prompt dei comandi viene visualizzato di nuovo immediatamente.

I vincoli della modalità asincrona:

- il comando o lo script non devono attendere l'input della tastiera
- il comando o lo script non devono restituire alcun risultato sullo schermo
- lasciare che la shell termini il processo

9.6 Controlli per la gestione dei processi

9.6.1 comando `kill`

Il comando `kill` invia un segnale di arresto a un processo.

```
kill [-signal] PID
```

Esempio:

```
$ kill -9 1664
```

Codice	Segnale	Descrizione
`2`	*SIGINT*	Arresto immediato del processo
`9`	*SIGKILL*	Interruzione del processo (CTRL + D)
`15`	*SIGTERM*	Arresto pulito del processo
`18`	*SIGCONT*	Riprendere il processo
`19`	*SIGSTOP*	Sospendere il processo

I segnali sono i mezzi di comunicazione tra i processi. Il comando `kill` invia un segnale a un processo.

Copyright © 2023 The Rocky Enterprise Software Foundation

Suggerimento

L'elenco completo dei segnali presi in considerazione dal comando `kill` è disponibile digitando il comando :

```
$ man 7 signal
```

9.6.2 comando `nohup`

`nohup` consente il lancio di un processo indipendentemente da una connessione.

```
comando nohup
```

Esempio:

```
$ nohup myprogram.sh 0</dev/null &
```

`nohup` ignora il segnale `SIGHUP` inviato quando un utente si disconnette.

Domanda

`nohup` gestisce l'output e l'error standard ma non l'input standard, da cui il reindirizzamento di questo input a `/dev/null`.

9.6.3 [CTRL] + [Z]

Premendo la combinazione `CTRL` + `Z` contemporaneamente, il processo sincrono è temporaneamente sospeso. L'accesso al prompt viene ripristinato dopo aver visualizzato il numero del processo che è stato appena sospeso.

9.6.4 istruzione `&`

La dichiarazione `&` esegue il comando in modo asincrono (il comando viene quindi chiamato *job*) e visualizza il numero di *job*. L'accesso al prompt viene quindi restituito.

Esempio:

```
$ time ls -lR / > list.ls 2> /dev/null &
[1] 15430
$
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Il numero *job* è ottenuto durante l'elaborazione in background e viene visualizzato in parentesi quadre, seguito dal numero di `PID`.

9.6.5 comandi `fg` e `bg`

Il comando `fg` mette il processo in primo piano:

```
$ time ls -lR / > list.ls 2>/dev/null &
$ fg 1
time ls -lR / > list.ls 2/dev/null
```

mentre il comando `bg` lo colloca in background:

```
[CTRL]+[Z]
^Z
[1]+ Stopped
$ bg 1
[1] 15430
$
```

Se è stato messo in background quando è stato creato con l'argomento `&` o più tardi con la combinazione `CTRL` + `Z`, un processo può essere riportato in primo piano con il comando `fg` e il suo numero di lavoro.

9.6.6 comando `jobs`

Il comando `jobs` visualizza l'elenco dei processi in esecuzione in background e specifica il loro numero di lavoro.

Esempio:

```
$ jobs
[1]- Running    sleep 1000
[2]+ Running    find / > arbo.txt
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Le colonne rappresentano:

1. numero di lavoro
2. l'ordine in cui i processi sono in esecuzione
3. un `+` : Il processo selezionato per impostazione predefinita per i comandi `fg` e `bg` quando non viene specificato un numero di processo
4. a `-` : Questo processo è il processo successivo che prende il `+`
5. *Running* (processo in esecuzione) o *Stopped* (processo sospeso).
6. il comando

9.6.7 comandi `nice` e `renice`

Il comando `nice` consente l'esecuzione di un comando specificando la sua priorità.

```
comando nice priority
```

Esempio:

```
$ nice -n+15 find / -name "file"
```

A differenza di `root`, un utente standard può solo ridurre la priorità di un processo. Saranno accettati solo valori tra +0 e +19.

Suggerimento

Quest'ultima limitazione può essere eliminata per utente o per gruppo modificando il file `/etc/security/limits.conf`.

Il comando `renice` ti consente di modificare la priorità di un processo di esecuzione.

```
renice priority [-g GID] [-p PID] [-u UID]
```

Esempio:

```
$ renice +15 -p 1664
```


Copyright © 2023 The Rocky Enterprise Software Foundation

| Opzione | Descrizione | | ------- | ---------- | | `-g` | `GID` del gruppo proprietario del processo. | | `-p` | `PID` del processo. | | `-u` | `UID` del proprietario del processo. |

Il comando `renice` agisce sui processi già in esecuzione. È quindi possibile modificare la priorità di un processo specifico, ma anche di diversi processi appartenenti a un utente o un gruppo.

Suggerimento

Il comando `pidof`, associato al comando `xargs` (vedi il corso Comandi avanzati), permette di applicare una nuova priorità in un singolo comando:

```
$ pidof sleep | xargs renice 20
```

9.6.8 comando `top`

Il comando `top` visualizza i processi e il loro consumo di risorse.

```
$ top
PID  USER PR NI ... %CPU %MEM  TIME+    COMMAND
2514 root 20 0       15    5.5 0:01.14   top
```

Colonna	Descrizione
`PID`	Identificatore del processo.
`USER`	Utente proprietario.
`PR`	Priorità del processo.
`NI`	Valore di Nice.
`%CPU`	Carico del processore.
`%MEM`	Carico di memoria.
`TIME+`	Tempo di utilizzo del processore.
`COMMAND`	Comando eseguito.

Il comando `top` permette di controllare i processi in tempo reale e in modalità interattiva.

Copyright © 2023 The Rocky Enterprise Software Foundation

9.6.9 comandi `pgrep` e `pkill`

Il comando `pgrep` cerca i processi in esecuzione per un nome di processo e visualizza il *PID* che soddisfa i criteri di selezione sull'output standard.

Il comando `pkill` invia a ogni processo il segnale specificato (per impostazione predefinita *SIGTERM*).

```
pgrep process
pkill [option] [-signal] process
```

Esempi:

- Ottenere il numero del processo da `sshd`:

```
$ pgrep -u root sshd
```

- Terminare tutti i processi di `tomcat`:

```
$ pkill tomcat
```

> **Nota**
>
> Prima di terminare un processo, è meglio conoscere esattamente lo scopo del processo, altrimenti si possono verificare arresti del sistema o altri problemi imprevedibili.

Oltre a inviare segnali ai processi interessati, il comando `pkill` può anche terminare la sessione di connessione dell'utente in base al numero di terminale, come ad esempio:

```
$ pkill -t pts/1
```


Copyright © 2023 The Rocky Enterprise Software Foundation

9.6.10 comando `killall`

La funzione di questo comando è pressoché identica a quella del comando `pkill`. L'utilizzo è - `killall [option] [ -s SIGNAL | -SIGNAL ] NAME`. Il segnale predefinito è *SIGTERM*.

Opzioni	Descrizione
`-l`	elencare tutti i nomi dei segnali conosciuti
`-i`	chiedere conferma prima di terminare
`-I`	corrispondenza del nome del processo senza distinzione tra maiuscole e minuscole

Esempio:

```
$ killall tomcat
```

9.6.11 comando `pstree`

Questo comando visualizza l'avanzamento in una struttura ad albero e il suo utilizzo è - `pstree [opzione]`.

Opzione	Descrizione
`-p`	Visualizzare il PID del processo
`-n`	ordinare l'output per PID
`-h`	evidenziare il processo corrente e i suoi progenitori
`-u`	mostrare le transizioni uid

```
$ pstree -pnhu
systemd(1)─┬─systemd-journal(595)
           ├─systemd-udevd(625)
           ├─auditd(671)───{auditd}(672)
           ├─dbus-daemon(714,dbus)
           ├─NetworkManager(715)─┬─{NetworkManager}(756)
           │                     └─{NetworkManager}(757)
           ├─systemd-logind(721)
           ├─chronyd(737,chrony)
           ├─sshd(758)───sshd(1398)───sshd(1410)───bash(1411)───pstree(1500)
           ├─tuned(759)─┬─{tuned}(1376)
           │            ├─{tuned}(1381)
           │            ├─{tuned}(1382)
           │            └─{tuned}(1384)
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
            ├─agetty(763)
            ├─crond(768)
            ├─polkitd(1375,polkitd)─┬─{polkitd}(1387)
            │                       ├─{polkitd}(1388)
            │                       ├─{polkitd}(1389)
            │                       ├─{polkitd}(1390)
            │                       └─{polkitd}(1392)
            └─systemd(1401)───(sd-pam)(1404)
```

9.6.12 Processi orfani e processi zombie

processo orfano: Quando un processo genitore termina, i suoi figli sono detti orfani. Il processo di init adotta questi processi in stato speciale e la raccolta dello stato viene completata finché non vengono distrutti. Dal punto di vista concettuale, il processo di orfanizzazione non comporta alcun danno.

processo zombie: Dopo che un processo figlio ha completato il suo lavoro e viene terminato, il suo processo genitore deve chiamare la funzione di elaborazione del segnale wait() o waitpid() per ottenere lo stato di cessazione del processo figlio. Se il processo padre non lo fa, anche se il processo figlio è già uscito, conserva alcune informazioni sullo stato di uscita nella tabella dei processi di sistema. Poiché il processo padre non può ottenere le informazioni sullo stato del processo figlio, questi processi continueranno a occupare risorse nella tabella dei processi. I processi in questo stato vengono chiamati zombie.

Pericolo:

- Occupa le risorse del sistema e causa una riduzione delle prestazioni della macchina.
- Impossibile generare nuovi processi figli.

Come verificare se ci sono processi zombie nel sistema in uso?

```
$ ps -lef | awk '{print $2}' | grep Z
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Questi caratteri possono comparire in questa colonna:

- **D** - sospensione ininterrotta (di solito IO)
- **I** - Thread del kernel inattivo
- **R** - in esecuzione o eseguibile (in coda di esecuzione)
- **S** - sospensione interrompibile (attesa del completamento di un evento)
- **T** - fermato dal segnale di controllo del lavoro
- **t** - interrotto dal debugger durante il tracciamento
- **W** - paging (non più valido dal kernel 2.6.xx)
- **X** - morto (non si dovrebbe mai vedere)
- **Z** - processo defunto ("zombie"), terminato ma non recuperato dal suo genitore

Copyright © 2023 The Rocky Enterprise Software Foundation

10. Backup e ripristino

In questo capitolo imparerai come eseguire il backup e ripristinare i tuoi dati con Linux.

Obiettivi : In questo capitolo, futuri amministratori Linux impareranno come:

✔ usare i comandi `tar` e `cpio` per effettuare un backup;
✔ controllare i loro backup e ripristinare i dati;
✔ comprimere o decomprimere i loro backup.

🏁 **backup**, **ripristino**, **compressione**

Conoscenza: ⭐⭐⭐
Complessità: ⭐⭐

Tempo di lettura: 40 minuti

> **Nota**
>
> In tutto questo capitolo le strutture di comando usano "device" per specificare sia la posizione di destinazione per il backup, sia la posizione di origine durante il ripristino. Il dispositivo può essere un supporto esterno o un file locale. Dovresti sviluppare una certa confidenza con questo concetto durante lo svolgimento del capitolo, ma puoi sempre ritornare a questa nota per chiarimenti se ne hai bisogno.

Il backup risponde a una necessità di conservare e ripristinare i dati in modo sicuro ed efficace.

Il backup consente di proteggersi dai seguenti problemi:

- **Distruzione**: volontaria o involontaria. Umana o tecnica. Virus, ...
- **Cancellazione**: volontaria o involontaria. Umana o tecnica. Virus, ...
- **Integrità** : i dati diventano inutilizzabili.

Nessun sistema è infallibile, nessun umano è infallibile, quindi per evitare di perdere dati, questi devono essere salvati per poi essere in grado di ripristinarli dopo un problema.

Copyright © 2023 The Rocky Enterprise Software Foundation

Il supporto di backup dovrebbe essere tenuto in un'altra stanza (o edificio) rispetto al server in modo che un disastro non distrugga il server e i backup.

Inoltre, l'amministratore deve controllare regolarmente che i supporti siano ancora leggibili.

10.1 Generalità

Ci sono due principi, il **backup** e l'**archivio**.

- L'archivio distrugge la fonte delle informazioni dopo l'operazione.
- Il backup conserva la fonte delle informazioni dopo l'operazione.

Queste operazioni consistono nel salvare informazioni in un file, su un supporto periferico o supportato (nastri, dischi, ...).

10.1.1 Il processo

I backup richiedono molta disciplina e rigore da parte dell'amministratore di sistema. È necessario porsi le seguenti domande:

- Qual è il mezzo appropriato?
- Cosa dovrebbe essere salvato?
- Quante copie?
- Quanto durerà il backup?
- Metodo?
- Quante volte?
- Automatico o manuale?
- Dove conservarlo?
- Quanto tempo sarà conservato?

10.1.2 Metodi di backup

- **Completo**: uno o più **filesystems** sono salvati (kernel, dati, utilità, ...).
- **Parziale**: uno o più **files** sono salvati (configurazioni, directories, ...).

Copyright © 2023 The Rocky Enterprise Software Foundation

- **Differenziale**: solo i file modificati dall'ultimo backup **completo** sono salvati.
- **Incrementale**: solo i file modificati dall'ultimo backup sono salvati.

10.1.3 Periodicità

- **Pre-corrente** : in un dato momento (prima di un aggiornamento del sistema, ...).
- **Periodica**: Ogni giorno, settimana, mese, ...

> **Suggerimento**
>
> Prima di una modifica del sistema, può essere utile fare un backup. Tuttavia, non ha senso eseguire il backup dei dati ogni giorno se vengono modificati solo ogni mese.

10.1.4 Metodi di ripristino

A seconda delle utilità disponibili, sarà possibile eseguire diversi tipi di ripristini.

- **Ripristino Completo**: alberi delle directory, ...
- **Ripristino Selettivo**: parte dell'albero, files, ...

È possibile ripristinare un intero backup ma è anche possibile ripristinarne solo una parte. Tuttavia, quando si ripristina una directory, i file creati dopo il backup non vengono eliminati.

> **Suggerimento**
>
> Per ripristinare una directory come era al momento del backup, è necessario eliminare completamente il suo contenuto prima di avviare il ripristino.

10.1.5 Gli strumenti

Ci sono molte utilità per fare il backup.

- **strumenti di editor** ;
- **strumenti grafici**;
- **strumenti da riga di comando**: `tar`, `cpio`, `pax`, `dd`, `dump`, ...

Copyright © 2023 The Rocky Enterprise Software Foundation

I comandi che useremo qui sono `tar` e `cpio`.

- `tar`:
- facile da usare ;
- consente di aggiungere file a un backup esistente.
- `cpio`:
- conserva i proprietari;
- conserva gruppi, date e permessi;
- salta i file danneggiati;
- file system completo.

> **Nota**
>
> Questi comandi salvano in un formato proprietario e standardizzato.

10.1.6 Convenzione di denominazione

L'uso di una convenzione di denominazione consente di indirizzare rapidamente il contenuto di un file di backup ed evitare così ripristini pericolosi.

- nome della directory;
- utilità utilizzata;
- opzioni utilizzate;
- data.

> **Suggerimento**
>
> Il nome del backup deve essere un nome esplicito.

> **Nota**
>
> La nozione di estensione in Linux non esiste. In altre parole, il nostro uso delle estensioni qui è per l'operatore umano. Se l'amministratore di sistema vede un file `.tar.gz` o `.tgz`, per esempio, sa come gestire il file.

Copyright © 2023 The Rocky Enterprise Software Foundation

10.1.7 Contenuto di un backup

Un backup contiene in genere i seguenti elementi:

- il file;
- il nome;
- il proprietario;
- la dimensione;
- le autorizzazioni;
- data di accesso.

> **Nota**
>
> Manca il numero `inode`.

10.1.8 Modalità di archiviazione

Esistono due diverse modalità di archiviazione:

- file su disco;
- dispositivo.

10.2 Tape ArchiveR - `tar`

Il comando `tar` consente di salvare su più supporti successivi (opzioni multi-volume).

È possibile estrarre tutto o parte di un backup.

`tar` esegue implicitamente il backup in modalità relativa anche se il percorso delle informazioni di cui eseguire il backup è menzionato in modalità assoluta. Tuttavia, sono possibili backup e ripristini in modalità assoluta.

Copyright © 2023 The Rocky Enterprise Software Foundation

10.2.1 Linee guida per il ripristino

Le domande giuste da porsi sono:

- cosa: parziale o completo;
- dove: il luogo in cui i dati saranno ripristinati;
- come: assoluto o relativo.

> **Attenzione**
>
> Prima di un ripristino, è importante prendere del tempo per pensare e determinare il metodo più appropriato per evitare errori.

I ripristini vengono solitamente eseguiti dopo che si è verificato un problema che deve essere risolto rapidamente. Un ripristino scadente può, in alcuni casi, peggiorare la situazione.

10.2.2 Backup con `tar`

L'utilità predefinita per la creazione di backup su sistemi UNIX è il comando `tar`. Questi backup possono essere compressi con `bzip2`, `xz`, `lzip`, `lzma`, `lzop`, `gzip`, `compress` o `zstd`.

`tar` consente di estrarre un singolo file o una directory da un backup, visualizzarne il contenuto o convalidarne l'integrità.

Stimare le dimensioni di un backup

Il comando seguente stima la dimensione in kilobyte di un possibile file *tar*:

```
$ tar cf - /directory/to/backup/ | wc -c
20480
$ tar czf - /directory/to/backup/ | wc -c
508
$ tar cjf - /directory/to/backup/ | wc -c
428
```

> **Attenzione**
>
> Attenzione, la presenza di "-" nella riga di comando disturba `zsh`. Passa a `bash`!

Copyright © 2023 The Rocky Enterprise Software Foundation

Convenzione di denominazione per un backup `tar`

Ecco un esempio di convenzione di denominazione per un backup `tar`, sapendo che la data deve essere aggiunta al nome.

chiavi	Files	Suffisso	Funzionalità
`cvf`	`home`	`home.tar`	`/home` in modalità relativa, forma non compressa
`cvfP`	`/etc`	`etc.A.tar`	`/etc` in modalità assoluta, nessuna compressione
`cvfz`	`usr`	`usr.tar.gz`	`/usr` in modalità relativa, compressione *gzip*
`cvfj`	`usr`	`usr.tar.bz2`	`/usr` in modalità relativa, compressione *bzip2*
`cvfPz`	`/home`	`home.A.tar.gz`	`home` in modalità assoluta, compressione *gzip*
`cvfPj`	`/home`	`home.A.tar.bz2`	`home` in modalità assoluta, compressione *bzip2*
...			

Creare un backup

CREARE UN BACKUP IN MODALITÀ RELATIVA

La creazione di un backup non compresso in modalità relativa viene eseguita con le opzioni `cvf`:

```
tar c[vf] [device] [file(s)]
```

Esempio:

```
[root]# tar cvf /backups/home.133.tar /home/
```

Opzione	Descrizione
`c`	Crea un backup.
`v`	Visualizza il nome dei file elaborati.
`f`	Consente di specificare il nome del backup (supporto).

Suggerimento

Il trattino (-) davanti alle opzioni di 'tar' non è necessario!

Copyright © 2023 The Rocky Enterprise Software Foundation

CREARE UN BACKUP IN MODALITÀ ASSOLUTA

La creazione di un backup non compresso in modo esplicito in modalità assoluta viene eseguita con le opzioni `cvfP`:

```
$ tar c[vf]P [device] [file(s)]
```

Esempio:

```
[root]# tar cvfP /backups/home.133.P.tar /home/
```

Opzione	Descrizione
`P`	Crea un backup in modalità assoluta.

Attenzione

Con la chiave `P`, il percorso dei file su cui eseguire il backup deve essere inserito come **assoluto**. Se le due condizioni (chiave `P` e percorso **assoluto**) non sono indicate, il backup è in modalità relativa.

CREAZIONE DI UN BACKUP COMPRESSO CON `GZIP`

La creazione di un backup compresso con 'gzip' viene eseguita con le opzioni `cvfz`:

```
$ tar cvzf backup.tar.gz dirname/
```

Opzione	Descrizione
`z`	Comprime il backup con *gzip*.

Nota

L'estensione `.tgz` è un'estensione equivalente a `.tar.gz`.

Nota

Mantenere le chiavi `cvf` (`tvf` o `xvf`) invariate per tutte le operazioni di backup e aggiungere semplicemente la chiave di compressione alla fine delle chiavi rende il comando più facile da capire (ad esempio, `cvfz` o `cvfj`, ecc.).

Copyright © 2023 The Rocky Enterprise Software Foundation

CREAZIONE DI UN BACKUP COMPRESSO CON `BZIP`

La creazione di un backup compresso con `bzip` viene eseguita con le opzioni `cvfj`:

```
$ tar cvfj backup.tar.bz2 dirname/
```

Opzione	Descrizione
`j`	Comprime il backup con_bzip2_.

Nota

Le estensioni `.tbz` e `.tb2` sono equivalenti alle estensioni `.tar.bz2`.

COMPRESSIONE `COMPRESS`, `GZIP`, `BZIP2`, `LZIP` E `XZ`

La compressione, e di conseguenza la decompressione, avrà un impatto sul consumo di risorse (tempo e utilizzo della CPU).

Ecco una classifica della compressione di un insieme di file di testo, dal meno al più efficiente:

- compress (`.tar.Z`)
- gzip (`.tar.gz`)
- bzip2 (`.tar.bz2`)
- lzip (`.tar.lz`)
- xz (`.tar.xz`)

Aggiungere un file o una directory a un backup esistente

È possibile aggiungere uno o più elementi a un backup esistente.

```
tar {r|A}[key(s)] [device] [file(s)]
```

Per aggiungere `/etc/passwd` al backup `/backups/home.133.tar`:

```
[root]# tar rvf /backups/home.133.tar /etc/passwd
```


Copyright © 2023 The Rocky Enterprise Software Foundation

L'aggiunta di una directory è simile. Qui aggiungi `dirtoadd` a `backup_name.tar`:

```
$ tar rvf backup_name.tar dirtoadd
```

Opzione	Descrizione
`r`	Aggiunge uno o più file alla fine di un backup multimediale ad accesso diretto (disco rigido).
`A`	Aggiunge uno o più file al termine di un backup su un supporto di accesso sequenziale (nastro).

Nota

Non è possibile aggiungere file o cartelle a un backup compresso.

```
$ tar rvfz backup.tgz filetoadd
tar: Cannot update compressed archives
Try `tar --help' or `tar --usage' for more information.
```

Nota

Se il backup è stato eseguito in modalità relativa, aggiungere file in modalità relativa. Se il backup è stato eseguito in modalità assoluta, aggiungere i file in modalità assoluta.

Le modalità miste possono causare problemi durante il ripristino.

Elencare il contenuto di un backup

È possibile visualizzare il contenuto di un backup senza estrarlo.

```
tar t[key(s)] [device]
```

Opzione	Descrizione
`t`	Visualizza il contenuto di un backup (compresso o meno).

Esempi:

```
$ tar tvf backup.tar
$ tar tvfz backup.tar.gz
$ tar tvfj backup.tar.bz2
```

Quando il numero di file in un backup diventa grande, è possibile inviare in *pipe* il risultato del comando `tar` ad un *impaginatore* (`more`, `less`, `most`, ecc.):

```
$ tar tvf backup.tar | less
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Suggerimento

Per elencare o recuperare il contenuto di un backup, non è necessario menzionare l'algoritmo di compressione utilizzato quando è stato creato il backup. Cioè, un `tar tvf` è equivalente a `tar tvfj`, per leggere il contenuto, e un `tar xvf` è equivalente a `tar xvfj`, per estrarre.

Suggerimento

Controlla sempre il contenuto di un backup.

Verificare l'integrità di un backup

L'integrità di un backup può essere testata con la chiave `W` al momento della sua creazione:

```
$ tar cvfW file_name.tar dir/
```

L'integrità di un backup può essere testata con la chiave `d` dopo la sua creazione:

```
$ tar vfd file_name.tar dir/
```

Suggerimento

Aggiungendo una seconda `v` alla chiave precedente, si otterrà l'elenco dei file archiviati così come le differenze tra i file archiviati e quelli presenti nel file system.

```
$ tar vvfd  /tmp/quodlibet.tar .quodlibet/
drwxr-x--- rockstar/rockstar     0 2021-05-21 00:11 .quodlibet/
-rw-r--r-- rockstar/rockstar     0 2021-05-19 00:59 .quodlibet/queue
[…]
-rw------- rockstar/rockstar  3323 2021-05-21 00:11 .quodlibet/config
.quodlibet/config: Mod time differs
.quodlibet/config: Size differs
[…]
```

L'opzione `W` viene utilizzata anche per confrontare il contenuto di un archivio con il filesystem:

```
$ tar tvfW file_name.tar
Verify 1/file1
1/file1: Mod time differs
1/file1: Size differs
Verify 1/file2
Verify 1/file3
```


Copyright © 2023 The Rocky Enterprise Software Foundation

La verifica con l'opzione `W` non può essere eseguita con un archivio compresso. Deve essere utilizzata l'opzione `d` :

```
$ tar dfz file_name.tgz
$ tar dfj file_name.tar.bz2
```

Estrarre (*untar*) un backup

L'estrazione di un backup (*untar*) `*.tar` viene eseguito con le opzioni `xvf`:

Estrarre il file `etc/exports` dal backup `/savings/etc.133.tar` nella cartella `etc` della directory corrente:

```
$ tar xvf /backups/etc.133.tar etc/exports
```

Estrarre tutti i file dal backup compresso `/backups/home.133.tar.bz2` nella directory corrente:

```
[root]# tar xvfj /backups/home.133.tar.bz2
```

Estrarre tutti i file dal backup `/backups/etc.133.P.tar` nella loro directory originale:

```
$ tar xvfP /backups/etc.133.P.tar
```

Attenzione

Vai nel posto giusto.

Controlla il contenuto del backup.

Opzione	Descrizione
`x`	Estrarre i file dal backup, compressi o meno.

L'estrazione di un backup *tar-gzipped* (`*.tar.gz`) viene eseguita con le opzioni `xvfz` :

```
$ tar xvfz backup.tar.gz
```

L'estrazione di un *tar-bzipped* (`*.tar.bz2`) viene eseguita con le opzioni `xvfj` :

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ tar xvfj backup.tar.bz2
```

Suggerimento

Per estrarre o elencare il contenuto di un backup, non è necessario menzionare l'algoritmo di compressione utilizzato per creare il backup. Cioè, un `tar xvf` è equivalente a `tar xvfj`, per estrarre il contenuto, e un `tar tvf` è equivalente a `tar tvfj`, per elencare.

Attenzione

Per ripristinare i file nella loro cartella originale (chiave `P` di un `tar xvf`), devi aver generato il backup con il percorso assoluto. Cioè, con la chiave `P` di un `tar cvf`.

ESTRARRE SOLO UN FILE DA UN BACKUP *TAR*

Per estrarre un file specifico da un backup *tar*, specificare il nome di tale file alla fine del comando `tar xvf`.

```
$ tar xvf backup.tar /path/to/file
```

Il comando precedente estrae solo il file `/path/to/file` dal backup `backup.tar`. Questo file verrà ripristinato nella directory `/path/to/` creata, o già presente, nella directory corrente.

```
$ tar xvfz backup.tar.gz /path/to/file
$ tar xvfj backup.tar.bz2 /path/to/file
```

ESTRARRE UNA CARTELLA DA UN BACKUP *TAR*

Per estrarre una sola directory (incluse le sottodirectory e i file) da un backup, specificare il nome della directory alla fine del comando `tar xvf`.

```
$ tar xvf backup.tar /path/to/dir/
```

Per estrarre più directory, specificare ciascuno dei nomi uno dopo l'altro:

```
$ tar xvf backup.tar /path/to/dir1/ /path/to/dir2/
$ tar xvfz backup.tar.gz /path/to/dir1/ /path/to/dir2/
$ tar xvfj backup.tar.bz2 /path/to/dir1/ /path/to/dir2/
```


Copyright © 2023 The Rocky Enterprise Software Foundation

ESTRARRE UN GRUPPO DI FILE DA UN BACKUP *TAR* UTILIZZANDO ESPRESSIONI REGOLARI (*REGEX*)

Specificate un (*regex*) per estrarre i file corrispondenti al modello di selezione specificato.

Ad esempio, per estrarre tutti i file con l'estensione `.conf` :

```
$ tar xvf backup.tar --wildcards '*.conf'
```

chiavi :

- `--wildcards *.conf` corrisponde ai file con estensione `.conf`.

10.3 *CoPy Input Output* - `cpio`

Il comando `cpio` consente di salvare su più supporti successivi senza specificare alcuna opzione.

È possibile estrarre tutto o parte di un backup.

Non c'è alcuna opzione, a differenza del comando `tar`, per eseguire il backup e comprimere allo stesso tempo. Quindi è fatto in due passaggi: backup e compressione.

Per eseguire un backup con `cpio`, è necessario specificare un elenco di file di cui eseguire il backup.

Questo elenco è fornito con i comandi `find`, `ls` o `cat`.

- `find` : naviga in un albero, ricorsivo o meno;
- `ls` : elencare una directory, ricorsiva o meno;
- `cat` : legge un file contenente gli alberi delle directory o i file da salvare.

> **Nota**
>
> `ls` non può essere usato con `-l` (dettagli) o `-R` (ricorsivo).
>
> Richiede un semplice elenco di nomi.

Copyright © 2023 The Rocky Enterprise Software Foundation

10.3.1 Creare un backup con il comando `cpio`

Sintassi del comando `cpio`:

```
[files command |] cpio {-o| --create} [-options] [<file-list] [>device]
```

Esempio:

Con un reindirizzamento dell'output di `cpio`:

```
$ find /etc | cpio -ov > /backups/etc.cpio
```

Utilizzo del nome di un supporto di backup:

```
$ find /etc | cpio -ovF /backups/etc.cpio
```

Il risultato del comando `find` viene inviato come input al comando `cpio` tramite una *pipe* (carattere `|`, `AltGr` + `6`).

Qui, il comando `find /etc` restituisce un elenco di file corrispondenti al contenuto della directory `/etc` (ricorsivamente) al comando `cpio`, che esegue il backup.

Non dimenticare il segno `>` durante il salvataggio o l'opzione `F save_name_cpio`.

Opzioni	Descrizione
`-o`	Crea un backup (*output*).
`-v`	Visualizza il nome dei file elaborati.
`-F`	Indica il backup da modificare (medium).

Backup su un supporto:

```
$ find /etc | cpio -ov > /dev/rmt0
```

Il supporto può essere di vari tipi:

- unità nastro: `/dev/rmt0` ;
- una partizione: `/dev/sda5`, `/dev/hda5`, etc.

Copyright © 2023 The Rocky Enterprise Software Foundation

10.3.2 Tipo di backup

Backup con percorso relativo

```
$ cd /
$ find etc | cpio -o > /backups/etc.cpio
```

Backup con percorso assoluto

```
$ find /etc | cpio -o > /backups/etc.A.cpio
```

> ⚠ **Attenzione**
>
> Se il percorso specificato nel comando `find` è **assoluto** il backup verrà eseguito in **assoluto**.
>
> Se il percorso indicato nel comando `find` è **relativo** il backup verrà eseguito in **relativo**.

10.3.3 Aggiungere a un backup

```
[files command |] cpio {-o| --create} -A [-options] [<fic-list] {F|>device}
```

Esempio:

```
$ find /etc/shadow | cpio -o -AF SystemFiles.A.cpio
```

L'aggiunta di file è possibile solo su supporti ad accesso diretto.

Opzione	Descrizione
`-A`	Aggiunge uno o più file a un backup su disco.
`-F`	Indica il backup da modificare.

10.3.4 Compressione di un backup

- Salva **poi** comprimi

```
$ find /etc | cpio  –o > etc.A.cpio
$ gzip /backups/etc.A.cpio
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ ls /backups/etc.A.cpio*
/backups/etc.A.cpio.gz
```

- Salva **e** comprimi

```
$ find /etc | cpio –o | gzip > /backups/etc.A.cpio.gz
```

Non c'è alcuna opzione, a differenza del comando `tar`, per salvare e comprimere allo stesso tempo. Quindi è fatto in due passaggi: salvataggio e compressione.

La sintassi del primo metodo è più facile da capire e ricordare, perché viene eseguita in due passaggi.

Per il primo metodo, il file di backup viene automaticamente rinominato dall'utilità `gzip` che aggiunge `.gz` alla fine del nome del file. Allo stesso modo l'utilità `bzip2` aggiunge automaticamente `.bz2`.

10.3.5 Leggere il contenuto di un backup

Sintassi del comando `cpio` per leggere il contenuto di un backup *cpio*:

```
cpio -t [-options] [<fic-list]
```

Esempio:

```
$ cpio -tv </backups/etc.152.cpio | less
```

Opzioni	Descrizione
`-t`	Legge un backup.
`-v`	Visualizza gli attributi del file.

Dopo aver eseguito un backup, è necessario leggerne il contenuto per essere sicuri che non ci siano stati errori.

Allo stesso modo, prima di eseguire un ripristino, è necessario leggere il contenuto del backup che verrà utilizzato.

Copyright © 2023 The Rocky Enterprise Software Foundation

10.3.6 Ripristinare un backup

Sintassi del comando `cpio` per ripristinare un backup:

```
cpio {-i| --extract} [-E file] [-options] [<device]
```

Esempio:

```
$ cpio -iv </backups/etc.152.cpio | less
```

Opzioni	Descrizione
`-i`	Ripristinare un backup completo.
`-E file`	Ripristina solo i file il cui nome è contenuto nel file.
`--make-directories` o `-d`	Ricostruisce la struttura ad albero mancante.
`-u`	Sostituisce tutti i file anche se esistono.
`--no-absolute-filenames`	Permette di ripristinare un backup effettuato in modalità assoluta in modo relativo.

Attenzione

Per impostazione predefinita, al momento del ripristino, i file sul disco la cui ultima data di modifica è più recente o uguale alla data del backup non vengono ripristinati (per evitare di sovrascrivere le informazioni recenti con informazioni più vecchie).

L'opzione `u`, d'altra parte, consente di ripristinare le versioni precedenti dei file.

Esempi:

- Ripristinare un backup assoluto in modalità assoluta

```
$ cpio –ivF home.A.cpio
```

- Ripristino assoluto su una struttura ad albero esistente

L'opzione `u` consente di sovrascrivere i file esistenti nella posizione in cui avviene il ripristino.

```
$ cpio –iuvF home.A.cpio
```

- Ripristinare un backup assoluto in modalità relativa

Copyright © 2023 The Rocky Enterprise Software Foundation

L'opzione lunga `no-absolute-filenames` consente un ripristino in modalità relativa. Infatti la `/` all'inizio del percorso verrà rimossa.

```
$ cpio --no-absolute-filenames -divuF home.A.cpio
```

Suggerimento

La creazione di directory è forse necessaria, quindi l'utilizzo dell'opzione `d`

- Ripristinare un backup relativo

```
$ cpio –iv <etc.cpio
```

- Ripristino in modalità assoluta di un file o di una directory

Il ripristino di un particolare file o directory richiede la creazione di un file di elenco che deve poi essere eliminato.

```
echo "/etc/passwd" > tmp
cpio –iuE tmp -F etc.A.cpio
rm -f tmp
```

10.4 Utilità di Compressione - decompressione

L'utilizzo della compressione al momento di un backup può avere una serie di inconvenienti:

- Allunga il tempo di backup e il tempo di ripristino.
- Rende impossibile aggiungere file al backup.

Nota

È quindi meglio fare un backup e comprimerlo piuttosto che comprimerlo durante il backup.

10.4.1 Compressione con `gzip`

Il comando `gzip` comprime i dati.

Sintassi del comando `gzip`:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
gzip [options] [file ...]
```

Esempio:

```
$ gzip usr.tar
$ ls
usr.tar.gz
```

Il file riceve l'estensione `.gz`.

Mantiene gli stessi permessi e le stesse date di ultimo accesso e modifica.

10.4.2 Compressione con `bunzip2`

Anche il comando `bunzip2` comprime i dati.

Sintassi del comando `bzip2`:

```
bzip2 [options] [file ...]
```

Esempio:

```
$ bzip2 usr.cpio
$ ls
usr.cpio.bz2
```

Al nome del file viene assegnata l'estensione `.bz2`.

La compressione con `bzip2` è migliore della compressione con `gzip` ma ci vuole più tempo per eseguirla.

10.4.3 Decompressione con `gunzip`

Il comando `gunzip` decomprime i dati compressi.

Sintassi del comando `gunzip`:

```
gunzip [options] [file ...]
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Esempio:

```
$ gunzip usr.tar.gz
$ ls
usr.tar
```

Il nome del file viene troncato da `gunzip` e l'estensione `.gz` viene rimossa.

`gunzip` decomprime anche i file con le seguenti estensioni:

- `.z` ;
- `-z` ;
- `_z` .

10.4.4 Decompressione con `bunzip2`

Il comando `bunzip2` decomprime i dati compressi.

Sintassi del comando `bzip2`:

```
bzip2 [options] [file ...]
```

Esempio:

```
$ bunzip2 usr.cpio.bz2
$ ls
usr.cpio
```

Il nome del file viene troncato da `bunzip2` e l'estensione `.bz2` viene rimossa.

`bunzip2` decomprime anche il file con le seguenti estensioni:

- `-bz` ;
- `.tbz2` ;
- `tbz` .

Copyright © 2023 The Rocky Enterprise Software Foundation

11. Avvio del sistema

In questo capitolo verrà illustrato come si avvia il sistema.

Obiettivi : In questo capitolo, i futuri amministratori Linux apprenderanno:

✔ Le diverse fasi del processo di avvio;
✔ Come Rocky Linux supporta questo avvio tramite Grub2 e systemd;
✔ Come proteggere Grub2 da un attacco;
✔ Come gestire i servizi;
✔ Come accedere ai registri di log con `journald`.

🏁 **utenti**

Conoscenza: ⭐⭐
Complessità: ⭐⭐⭐

Tempo di lettura: 20 minuti

11.1 Il processo di avvio

È importante capire il processo di avvio di Linux per poter risolvere i problemi che potrebbero verificarsi.

Il processo di avvio include:

11.1.1 L'avvio del BIOS

Il **BIOS** (Basic Input/Output System) esegue il **POST** (power on self test) per rilevare, testare e inizializzare i componenti hardware del sistema.

Quindi carica il **MBR** (Master Boot Record).

Copyright © 2023 The Rocky Enterprise Software Foundation

11.1.2 Il Master boot record (MBR)

Il Master Boot Record sono i primi 512 byte del disco di avvio. Il MBR trova il dispositivo di avvio e carica il bootloader **GRUB2** in memoria passando il controllo ad esso.

I successivi 64 byte contengono la tabella delle partizioni del disco.

11.1.3 Il bootloader Grub2

Il bootloader predefinito per la distribuzione Rocky 8 è **GRUB2** (GRand Unified Bootloader). GRUB2 sostituisce il vecchio. GRUB bootloader (chiamato anche GRUB legacy).

Il file di configurazione di GRUB2 si trova in `/boot/grub2/grub.cfg` ma questo file non dovrebbe mai essere modificato direttamente.

Le impostazioni di configurazione del menu GRUB2 si trovano in `/etc/default/grub` e sono usate per generare il file `grub.cfg`.

```
# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root
rhgb quiet net.ifnames=0"
GRUB_DISABLE_RECOVERY="true"
```

Se vengono apportate modifiche a uno o più di questi parametri, deve essere eseguito il comando `grub2-mkconfig` per rigenerare il file `/boot/grub2/grub.cfg`.

```
[root] # grub2-mkconfig –o /boot/grub2/grub.cfg
```

- GRUB2 cerca l'immagine del kernel compresso (il file `vmlinuz`) nella cartella `/boot`.
- GRUB2 carica l'immagine del kernel in memoria ed estrae il contenuto del file immagine `initramfs` in una cartella temporanea in memoria usando il file system `tmpfs`.

Copyright © 2023 The Rocky Enterprise Software Foundation

11.1.4 Il kernel

Il kernel inizia il processo `systemd` con PID 1.

```
root          1      0  0 02:10 ?        00:00:02 /usr/lib/systemd/systemd --
switched-root --system --deserialize 23
```

11.1.5 `systemd`

Systemd è il genitore di tutti i processi di sistema. Legge il target del link `/etc/systemd/system/default.target` (es. `/usr/lib/systemd/system/multi-user.target`) per determinare l'obiettivo predefinito del sistema. Il file definisce i servizi da avviare.

Systemd posiziona quindi il sistema nello stato definito dall'obiettivo eseguendo le seguenti attività di inizializzazione:

1. Imposta il nome della macchina
2. Inizializza la rete
3. Inizializza SELinux
4. Mostra il banner di benvenuto
5. Inizializza l'hardware in base agli argomenti forniti al kernel al momento dell'avvio
6. Monta i file system, inclusi i file system virtuali come /proc
7. Pulisce le directory in /var
8. Avvia la memoria virtuale (swap)

Copyright © 2023 The Rocky Enterprise Software Foundation

11.2 Protezione del bootloader GRUB2

Perché proteggere il bootloader con una password?

1. Prevenire l'accesso in *Single user mode* - Se un utente malintenzionato può avviare in single user mode, diventa l'utente root.
2. Impedire l'accesso alla console di GRUB - Se un utente malintenzionato riesce a utilizzare la console Grub, può modificare la sua configurazione o raccogliere informazioni sul sistema utilizzando il comando `cat`.
3. Impedire l'accesso ai sistemi operativi insicuri. Se c'è un doppio avvio sul sistema, un utente malintenzionato può selezionare un sistema operativo come DOS che all'avvio ignora i controlli di accesso e le autorizzazioni dei file.

Per proteggere con password il bootloader GRUB2:

- Rimuovere `-unrestricted` dalla dichiarazione principale `CLASS=` nel file `/etc/grub.d/10_linux`.
- Se un utente non è stato ancora configurato, utilizzare il comando `grub2-setpassword` per fornire una password per l'utente root:

```
# grub2-setpassword
```

Un file `/boot/grub2/user.cfg` sarà creato se non era già presente. Contiene la password hashed. di GRUB2.

> **Nota**
>
> Questo comando supporta solo le configurazioni con un singolo utente root.

```
[root]# cat /boot/grub2/user.cfg
GRUB2_PASSWORD=grub.pbkdf2.sha512.10000.CC6F56....A21
```

- Ricreare il file di configurazione con il comando `grub2-mkconfig`:

```
[root]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-327.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-327.el7.x86_64.img
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
Found linux image: /boot/vmlinuz-0-rescue-f9725b0c842348ce9e0bc81968cf7181
Found initrd image: /boot/initramfs-0-rescue-
f9725b0c842348ce9e0bc81968cf7181.img
done
```

- Riavviare il sistema e controllare.

Tutte le voci definite nel menu GRUB richiederanno ora un utente e una password da inserire a ciascun avvio. Il sistema non avvierà un kernel senza l'intervento diretto dell'utente dalla console.

- Quando viene richiesto l'utente, inserire `root`;
- Quando viene richiesta una password, inserire la password fornita al comando `grub2-setpassword`.

Per proteggere solo la modifica delle voci del menu GRUB e l'accesso alla console, l'esecuzione del comando `grub2-setpassword` è sufficiente. Ci possono però essere casi in cui ci sono buone ragioni per non farlo. Questo potrebbe essere particolarmente vero in un data center remoto in cui l'inserimento di una password ogni volta che viene riavviato un server è difficile o impossibile da fare.

11.3 Systemd

Systemd è un gestore di servizi per i sistemi operativi Linux.

È sviluppato per:

- rimanere compatibile con gli script di inizializzazione del vecchio SysV,
- fornire molte funzionalità, come l'avvio parallelo dei servizi di sistema all'avvio del sistema, l'attivazione su richiesta dei demoni, il supporto per le istantanee o la gestione delle dipendenze tra i servizi.

Nota

Systemd è il sistema di inizializzazione predefinito da RedHat/CentOS 7.

Copyright © 2023 The Rocky Enterprise Software Foundation

Systemd introduce il concetto di file di unità, noti anche come unità di systemd.

Tipo	Estensione del file	Funzionalità
Unità di servizio	`.service`	Servizio di sistema
Unità di destinazione	`.target`	Un gruppo di unità systemd
Mount unit	`.automount`	Un punto di montaggio automatico per il file system

Nota

Ci sono molti tipi di unità: Device unit, Mount unit, Path unit, Scope unit, Slice unit, Snapshot unit, Socket unit, Swap unit, Timer unit.

- Systemd supporta le istantanee dello stato del sistema e il ripristino.
- Mount points possono essere configurati come target di systemd.
- All'avvio, systemd crea socket di ascolto per tutti i servizi di sistema che supportano questo tipo di attivazione e passa questi socket ai relativi servizi non appena vengono avviati. Ciò consente di riavviare un servizio senza perdere un singolo messaggio inviato dalla rete durante la sua indisponibilità. Il socket corrispondente rimane accessibile e tutti i messaggi vengono accodati.
- I servizi di sistema che utilizzano D-BUS per le comunicazioni tra processi possono essere avviati su richiesta la prima volta che vengono utilizzati da un client.
- Systemd arresta o riavvia solo i servizi in esecuzione. Le versioni precedenti (prima di RHEL7) tentavano di arrestare direttamente i servizi senza controllarne lo stato corrente.
- I servizi di sistema non ereditano alcun contesto (come le variabili di ambiente HOME e PATH). Ogni servizio opera nel proprio contesto di esecuzione.

Tutte le operazioni delle unità di servizio sono soggette a un timeout predefinito di 5 minuti per evitare che un servizio malfunzionante blocchi il sistema.

Copyright © 2023 The Rocky Enterprise Software Foundation

11.3.1 Gestione dei servizi di sistema

Le unità di servizio terminano con l'estensione di file `.service` e hanno uno scopo simile a quello degli script di init. Il comando `systemctl` viene utilizzato per `visualizzare`, `avviare`, `fermare`, `riavviare` un servizio di sistema:

systemctl	Descrizione
systemctl start *name*.service	Avviare un servizio
systemctl stop *name*.service	Arresta un servizio
systemctl restart *name*.service	Riavviare un servizio
systemctl reload *name*.service	Ricaricare una configurazione
systemctl status *name*.service	Controlla se un servizio è in esecuzione
systemctl try-restart *name*.service	Riavviare un servizio solo se è in esecuzione
systemctl list-units --type service --all	Visualizzare lo stato di tutti i servizi

Il comando `systemctl` viene utilizzato anche per `abilitare` o `disabilitare` un servizio di sistema e la visualizzazione dei servizi associati:

systemctl	Descrizione
systemctl enable *name*.service	Attivare un servizio
systemctl disable *name*.service	Disabilitare un servizio
systemctl list-unit-files --type service	Elenca tutti i servizi e i controlli se sono in esecuzione
systemctl list-dependencies --after	Elenca i servizi che si avviano prima dell'unità specificata
systemctl list-dependencies --before	Elenca i servizi che si avviano dopo l'unità specificata

Esempi:

```
systemctl stop nfs-server.service
# or
systemctl stop nfs-server
```

Per elencare tutte le unità attualmente caricate:

```
systemctl list-units --type service
```

Per elencare tutte le unità e per verificare se sono attivate:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
systemctl list-unit-files --type service
```

```
systemctl enable httpd.service
systemctl disable bluetooth.service
```

11.3.2 Esempio di un file .service per il servizio postfix

```
postfix.service Unit File
What follows is the content of the /usr/lib/systemd/system/postfix.service unit
file as currently provided by the postfix package:

[Unit]
Description=Postfix Mail Transport Agent
After=syslog.target network.target
Conflicts=sendmail.service exim.service

[Service]
Type=forking
PIDFile=/var/spool/postfix/pid/master.pid
EnvironmentFile=-/etc/sysconfig/network
ExecStartPre=-/usr/libexec/postfix/aliasesdb
ExecStartPre=-/usr/libexec/postfix/chroot-update
ExecStart=/usr/sbin/postfix start
ExecReload=/usr/sbin/postfix reload
ExecStop=/usr/sbin/postfix stop

[Install]
WantedBy=multi-user.target
```

11.3.3 Utilizzo degli obiettivi di sistema

Su Rocky8/RHEL8, il concetto di runlevel è stato sostituito dagli obiettivi systemd.

I sistemi di destinazione sono rappresentati da unità di destinazione. Le unità di destinazione terminano con l'estenzione `.target` e il loro unico scopo è di raggruppare altre unità systemd in una catena di dipendenze.

Ad esempio, l'unità `graphical.target`, che viene utilizzata per avviare una sessione grafica, inizializza i servizi di sistema come il **GNOME display manager** (`gdm.service`) o l'**accounts service** (`accounts-daemon.service`) e attiva anche l'unità `multi-user.target`.

Copyright © 2023 The Rocky Enterprise Software Foundation

Allo stesso modo, l'unità `multi-user.target` inizializza altri servizi di sistema essenziali, come **NetworkManager** (`NetworkManager.service`) o **D-Bus** (`dbus.service`) e attiva un'altra unità di destinazione denominata `basic.target` .

Unità di destinazione.	Descrizione
poweroff.target	Chiude il sistema e lo spegne
rescue.target	Attiva una shell di salvataggio
multi-user.target	Attiva un sistema multiutente senza interfaccia grafica
graphical.target	Attiva un sistema multiutente con interfaccia grafica
reboot.target	Spegne e riavvia il sistema

La destinazione predefinita

Per determinare quale obiettivo viene utilizzato per impostazione predefinita:

```
systemctl get-default
```

Questo comando cerca l'obiettivo del collegamento simbolico situato in `/etc/systemd/system/default.target` e visualizza il risultato.

```
$ systemctl get-default
graphical.target
```

Il comando `systemctl` può anche fornire un elenco di obiettivi disponibili:

```
systemctl list-units --type target
UNIT                   LOAD   ACTIVE SUB    DESCRIPTION
basic.target           loaded active active Basic System
bluetooth.target       loaded active active Bluetooth
cryptsetup.target      loaded active active Encrypted Volumes
getty.target           loaded active active Login Prompts
graphical.target       loaded active active Graphical Interface
local-fs-pre.target    loaded active active Local File Systems (Pre)
local-fs.target        loaded active active Local File Systems
multi-user.target      loaded active active Multi-User System
network-online.target  loaded active active Network is Online
network.target         loaded active active Network
nss-user-lookup.target loaded active active User and Group Name Lookups
paths.target           loaded active active Paths
remote-fs.target       loaded active active Remote File Systems
slices.target          loaded active active Slices
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
sockets.target         loaded active active Sockets
sound.target           loaded active active Sound Card
swap.target            loaded active active Swap
sysinit.target         loaded active active System Initialization
timers.target          loaded active active Timers
```

Per configurare il sistema all'utilizzo di un diverso target predefinito:

```
systemctl set-default name.target
```

Esempio:

```
# systemctl set-default multi-user.target
rm '/etc/systemd/system/default.target'
ln -s '/usr/lib/systemd/system/multi-user.target' '/etc/systemd/system/
default.target'
```

Per passare a un'unità di destinazione diversa nella sessione corrente:

```
systemctl isolate name.target
```

La **Modalità di ripristino** fornisce un ambiente semplice per riparare il sistema nei casi in cui è impossibile eseguire un normale processo di avvio.

In `modalità di ripristino`, il sistema tenta di montare tutti i file system locali e avviare diversi servizi di sistema importanti, ma non abilita un'interfaccia di rete o consente ad altri utenti di connettersi al sistema contemporaneamente.

Su Rocky 8, la `modalità di ripristino` è equivalente al vecchio `single user mode` e richiede la password di root.

Per modificare la destinazione corrente immettere `rescue mode` nella sessione corrente:

```
systemctl rescue
```

Modalità di emergenza fornisce l'ambiente più minimalista possibile e consente di riparare il sistema anche in situazioni in cui il sistema non è in grado di inserire la modalità di salvataggio. Nella modalità di emergenza, il sistema monta il file system root solo per la lettura. Non tenterà di montare qualsiasi altro file system

Copyright © 2023 The Rocky Enterprise Software Foundation

locale, non attiverà alcuna interfaccia di rete e inizializzerà alcuni servizi essenziali.

Per modificare il target corrente e immettere la modalità di emergenza nella sessione corrente:

```
systemctl emergency
```

Arresto, sospensione e ibernazione

Il comando `systemctl` sostituisce alcuni dei comandi di gestione dell'alimentazione utilizzati nelle versioni precedenti:

Vecchio comando	Nuovo comando	Descrizione
`halt`	`systemctl halt`	Spegne il sistema.
`poweroff`	`systemctl poweroff`	Arresta elettricamente il sistema.
`reboot`	`systemctl reboot`	Riavvia il sistema.
`pm-suspend`	`systemctl suspend`	Sospende il sistema.
`pm-hibernate`	`systemctl hibernate`	Iberna il sistema.
`pm-suspend-hybrid`	`systemctl hybrid-sleep`	Iberna e sospende il sistema.

11.3.4 Il processo `journald`

I file di registro possono, oltre a `rsyslogd`, essere gestiti anche dal demone `journald` che è un componente di `systemd`.

Il demone `journald` cattura i messaggi Syslog, i messaggi di registro del kernel, i messaggi dal disco RAM iniziale e dall'inizio dell'avvio, nonché i messaggi scritti nell'output standard e l'output di errore standard di tutti i servizi, quindi li indicizza e li rende disponibili all'utente.

Il formato del file di registro nativo, che è un file binario strutturato e indicizzato, migliora le ricerche e consente un funzionamento più rapido, memorizza anche le informazioni dei metadati, come i timestamp o gli ID utente.

Copyright © 2023 The Rocky Enterprise Software Foundation

11.3.5 comando `journalctl`

Il comando `journalctl` visualizza i file di registro.

```
journalctl
```

Il comando elenca tutti i file di registro generati sul sistema. La struttura di questa uscita è simile a quella utilizzata in `/var/log/messages/` ma offre alcuni miglioramenti:

- la priorità delle voci è segnata visivamente;
- i timestamp sono convertiti nella zona oraria locale del sistema;
- vengono visualizzati tutti i dati registrati, inclusi i registri rotativi;
- l'inizio di un avvio è contrassegnato da una linea speciale.

Uso del display continuo

Con il display continuo, i messaggi di registro vengono visualizzati in tempo reale.

```
journalctl -f
```

Questo comando restituisce un elenco delle dieci linee di registro più recenti. L'utilità continua quindi a funzionare e attende che avvengano nuove modifiche per visualizzarle immediatamente.

Filtrare i Messaggi

È possibile utilizzare diversi metodi di filtraggio per estrarre informazioni che si adattano a diverse esigenze. I messaggi di registro vengono spesso utilizzati per monitorare il comportamento errato del sistema. Per visualizzare le voci con una priorità selezionata o superiore:

```
journalctl -p priority
```


Copyright © 2023 The Rocky Enterprise Software Foundation

È necessario sostituire la priorità con una delle seguenti parole chiave (o un numero):

- debug (7),
- info (6),
- notice (5),
- warning (4),
- err (3),
- crit (2),
- alert (1),
- and emerg (0).

Copyright © 2023 The Rocky Enterprise Software Foundation

12. Gestione dei compiti

In questo capitolo imparerai come gestire le attività programmate.

Obiettivi : In questo capitolo, futuri amministratori Linux impareranno come:

✔ Linux si occupa della pianificazione dei compiti;
✔ limitare l'uso di `cron` a determinati utenti;
✔ pianificare le attività.

🏁 **crontab**, **crond**, **pianificazione**, **linux**

Conoscenza: ⭐⭐
Complessità: ⭐⭐

Tempo di lettura: 15 minuti

12.1 Generalità

La pianificazione delle attività è gestita con l'utilità `cron`. Essa permette l'esecuzione periodica dei compiti.

È riservata all'amministratore per le attività di sistema ma può essere utilizzata da utenti normali per attività o script a cui hanno accesso. Per accedere all'utilità `cron`, usiamo: `crontab`.

Il servizio `cron` è usato per:

- Operazioni di amministrazione ripetitive;
- Backups;
- Monitoraggio dell'attività del sistema;
- Esecuzione di un programma.

`crontab` è un'abbreviazione per **cron table**, ma può essere pensato come una tabella di programmazione attività.

Copyright © 2023 The Rocky Enterprise Software Foundation

Attenzione

Per impostare una pianificazione, il sistema deve avere l'ora locale impostata correttamente.

12.2 Come funziona il servizio

Il servizio di `cron` è gestito da un demone `crond` presente in memoria.

Per verificare il suo stato:

```
[root] # systemctl status crond
```

Suggerimento

Se il demone `crond` non è in esecuzione, dovrai inizializzarlo manualmente e/o automaticamente all'avvio. Infatti, anche se i compiti sono pianificati, non saranno lanciati.

Inizializzazione manuale del demone `crond`:

```
[root]# systemctl {status|start|restart|stop} crond
```

Initializzazione del demone `crond` all'avvio del sistema:

```
[root]# systemctl enable crond
```

12.3 Sicurezza

Per implementare una pianificazione, un utente deve disporre dell'autorizzazione all'utilizzo del servizio `cron`.

Questa autorizzazione varia in base alle informazioni contenute nei file seguenti:

- `/etc/cron.allow`
- `/etc/cron.deny`

Attenzione

Se nessuno dei file è presente, tutti gli utenti possono usare `cron`.

Copyright © 2023 The Rocky Enterprise Software Foundation

12.3.1 I files `cron.allow` and `cron.deny`

File `/etc/cron.allow`

Solo gli utenti contenuti in questo file sono autorizzati a utilizzare `cron`.

Se esiste ed è vuoto, nessun utente può usare `cron`.

> ⚠ **Attenzione**
>
> Se `cron.allow` è presente, `cron.deny` è **ignorato**.

File `/etc/cron.deny`

Gli utenti di questo file non sono autorizzati a utilizzare `cron`.

Se è vuoto, tutti gli utenti possono usare `cron`.

Per impostazione predefinita, `/etc/cron.deny` esiste ed è vuoto e `/etc/cron.allow` non esiste.

12.3.2 Consentire ad un utente

Solo **user1** sarà in grado di utilizzare `cron`.

```
[root]# vi /etc/cron.allow
user1
```

12.3.3 Proibire ad un utente

Solo **user2** non sarà in grado di usare `cron`.

```
[root]# vi /etc/cron.deny
user2
```

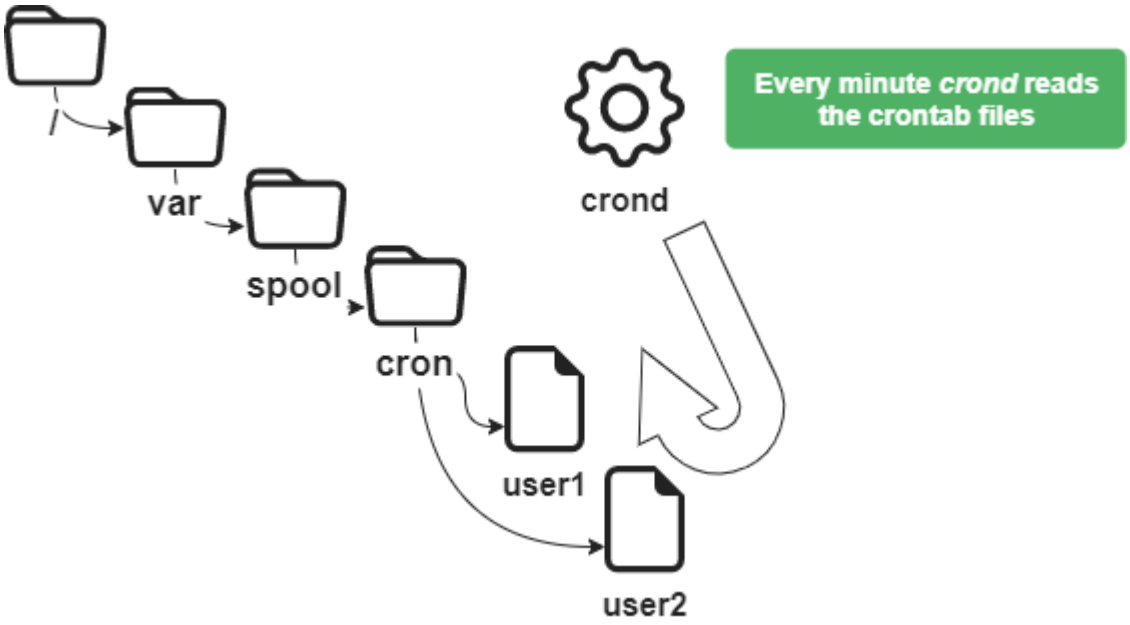
`cron.allow` non deve essere presente.

Copyright © 2023 The Rocky Enterprise Software Foundation

12.4 Pianificazione delle attività

Quando un utente pianifica un'attività, viene creato un file con il suo nome in `/var/spool/cron/`.

Questo file contiene tutte le informazioni che il `crond` deve sapere riguardo a tutte le attività create da questo utente, i comandi o i programmi da eseguire e quando eseguirli (ora, minuto, giorno ...).



12.4.1 Il comando `crontab`

Il comando `crontab` viene utilizzato per gestire il file di pianificazione.

```
crontab [-u user] [-e | -l | -r]
```

Esempio:

```
[root]# crontab -u user1 -e
```

Opzione	Descrizione
`-e`	Modifica il file di pianificazione con VI
`-l`	Visualizza il contenuto del file di pianificazione
`-u`	Imposta il nome dell'utente di cui si vuole manipolare il file di programma previsionale
`-r`	Cancella il file di pianificazione

Copyright © 2023 The Rocky Enterprise Software Foundation

> **Attenzione**
>
> `crontab` senza opzione elimina il vecchio file di pianificazione e attende che l'utente inserisca nuove righe. Devi premere `ctrl` + `d` per uscire da questa modalità di modifica.
>
> Solo `root` può utilizzare l'opzione `-u utente` per gestire il file di pianificazione di un altro utente.
>
> L'esempio sopra consente a root di pianificare un'attività per l'utente1.

12.4.2 Usi di `crontab`

Gli usi di `crontab` sono molti e includono:

- Modifiche ai file `crontab` presi in considerazione immediatamente;
- Nessun bisogno di riavviare.

D'altra parte, devono essere presi in considerazione i seguenti punti:

- Il programma deve essere autonomo;
- Fornire reindirizzamenti (stdin, stdout, stderr);
- Non è rilevante per eseguire comandi che utilizzano richieste di ingresso/uscita su un terminale.

> **Nota**
>
> È importante capire che lo scopo della programmazione è quello di eseguire i compiti automaticamente, senza la necessità di un intervento esterno.

12.5 Il file `crontab`

Il file `crontab` è strutturato in base alle seguenti regole.

- Ogni riga di questo file corrisponde a una pianificazione;
- Ogni linea ha sei campi, 5 per il tempo e 1 per l'ordine;
- Ogni campo è separato da uno spazio o da una tabulazione;
- Ogni linea termina con un ritorno a capo;
- Un `#` all'inizio della linea la commenta.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
[root]# crontab -e
10 4 1 * * /root/scripts/backup.sh
1  2 3 4 5       6
```

Campo	Descrizione	Dettaglio
1	Minuto(i)	Da 0 a 59
2	Ora(e)	Da 0 a 23
3	Giorno(i) del mese	Da 1 a 31
4	Mese dell'anno	Da 1 a 12
5	Giorno(i) della settimana	Da 0 a 7 (0=7=Domenica)
6	Compito da eseguire	Comando completo o script

Attenzione

Le attività da eseguire devono utilizzare percorsi assoluti e, se possibile, usare reindirizzamenti.

Al fine di semplificare la notazione per la definizione del tempo, è consigliabile utilizzare simboli speciali.

Wildcards	Descrizione
`*`	Indica tutti i possibili valori del campo
`-`	Indica una gamma di valori
`,`	Indica un elenco di valori
`/`	Definisce un passo

Esempi:

Script eseguito il 15 Aprile alle 10:25am:

```
25 10 15 04 * /root/scripts/script > /log/…
```

Esegui alle 11am e quindi alle 4pm di ogni giorno:

```
00 11,16 * * * /root/scripts/script > /log/…
```

Esegui ogni ora dalle 11am alle 4pm di ogni giorno:

```
00 11-16 * * * /root/scripts/script > /log/…
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Esegui ogni 10 minuti durante l'orario di lavoro:

```
*/10 8-17 * * 1-5 /root/scripts/script > /log/…
```

Per l'utente root, `crontab` ha anche alcune impostazioni speciali del tempo:

Impostazioni	Descrizione
@reboot	Esegue un comando al riavvio del sistema
@hourly	Esegue un comando ogni ora
@daily	Esegui giornalmente dopo la mezzanotte
@weekly	Esegui il comando ogni domenica dopo la mezzanotte
@monthly	Esegui il comando il primo giorno del mese subito dopo mezzanotte
@annually	Esegui il 1 gennaio subito dopo mezzanotte

12.5.1 Processo di esecuzione dell'attività

Un utente, rockstar, vuole modificare il suo file `crontab`:

1) `crond` controlla se è permesso (`/etc/cron.allow` e `/etc/cron.deny`).

2) Se lo è, accede al file `crontab` (`/var/spool/cron/rockstar`).

Ogni minuto `cron` legge il file di pianificazione.

3) Esegue le attività pianificate.

4) Riporta sistematicamente in un file di registro (`/var/log/cron`).

Copyright © 2023 The Rocky Enterprise Software Foundation

13. Implementazione della Rete

In questo capitolo imparerai come gestire e lavorare con la rete.

Obiettivi : In questo capitolo imparerai come:

- ✔ Configurare una workstation per usare DHCP;
- ✔ Configurare una workstation per utilizzare una configurazione statica;
- ✔ Configura una workstation per utilizzare un gateway;
- ✔ Configurare una workstation per utilizzare i server DNS;
- ✔ Risolvere i problemi relativi alla rete di una workstation.

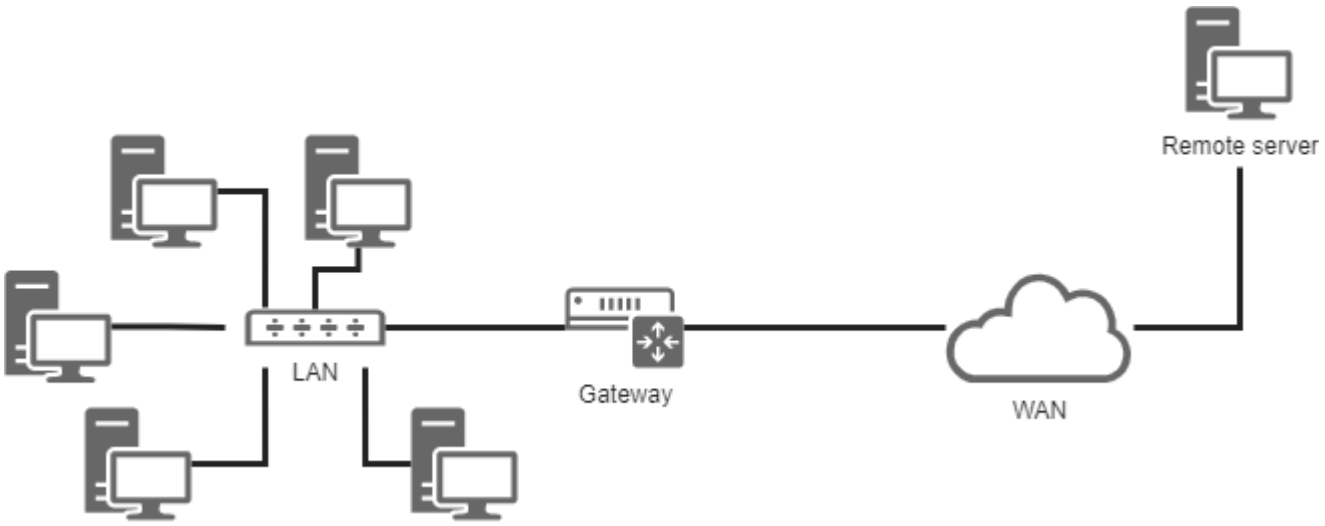
🏁 **rete**, **linux**, **ip**

Conoscenza: ⭐⭐
Complessità: ⭐⭐

Tempo di lettura: 30 minuti

13.1 Generalità

Per illustrare questo capitolo, useremo la seguente architettura.



Copyright © 2023 The Rocky Enterprise Software Foundation

Ci consentirà di prendere in considerazione:

- l'integrazione in una LAN (local area network);
- la configurazione di un gateway per raggiungere un server remoto;
- la configurazione di un server DNS e l'implementazione della risoluzione dei nomi.

I parametri minimi da definire per la macchina sono:

- il nome della macchina;
- l'indirizzo IP;
- la subnet mask.

Esempio:

- `pc-rocky`;
- `192.168.1.10`;
- `255.255.255.0`.

La notazione chiamata CIDR è sempre più frequente: 192.168.1.10/24

Gli indirizzi IP vengono utilizzati per il corretto routing dei messaggi (pacchetti). Sono divisi in due parti:

- la parte fissa, identifica la rete;
- l'identificatore dell'host nella rete.

La subnet mask è un insieme di **4 byte** destinato a isolare:

- l'indirizzo di rete (**NetID** o **SubnetID**) eseguendo un AND logico bit per bit tra l'indirizzo IP e la maschera;
- l'indirizzo dell'host. (**HostID**) eseguendo un AND logico bit per bit tra l'indirizzo IP e il complemento della maschera.

Copyright © 2023 The Rocky Enterprise Software Foundation

Ci sono anche indirizzi specifici all'interno di una rete, che devono essere identificati. Il primo indirizzo di un intervallo e l'ultimo hanno un ruolo particolare:

- Il primo indirizzo di un intervallo è l'**indirizzo di rete**. Viene utilizzato per identificare le reti e per instradare le informazioni da una rete all'altra.
- L'ultimo indirizzo di un intervallo è l'**indirizzo di trasmissione**. Viene utilizzato per trasmettere informazioni a tutte le macchine sulla rete.

13.1.1 Indirizzo MAC / Indirizzo IP

L'**indirizzo MAC** è un identificatore fisico scritto in fabbrica sul dispositivo. Questo a volte viene definito l'indirizzo hardware. Consiste di 6 byte spesso espressi in forma esadecimale (per esempio 5E:FF:56:A2:AF:15). È composto da: 3 byte dell'identificatore del produttore e 3 byte del numero di serie.

> ⚠ **Attenzione**
>
> Quest'ultima affermazione è oggi un po' meno vera con la virtualizzazione. Ci sono anche soluzioni software per cambiare l'indirizzo MAC.

Un indirizzo Internet Protocol (**IP**) è un numero di identificazione permanente o temporaneo assegnato a ciascun dispositivo collegato a una rete di computer che utilizza l'Internet Protocol. Una parte definisce l'indirizzo di rete (NetID o SubnetID a seconda dei casi), l'altra parte definisce l'indirizzo dell'host nella rete (HostID). La dimensione relativa di ciascuna parte varia in base alla (sub)mask della rete.

Un indirizzo IPv4 definisce un indirizzo su 4 byte. Per il numero di indirizzi disponibili che è vicino alla saturazione è stato creato un nuovo standard, l'IPv6 definito su 16 byte.

IPv6 è spesso rappresentato da 8 gruppi di 2 byte separati da un due punti. Gli zeri insignificanti possono essere omessi, uno o più gruppi di 4 zeri consecutivi possono essere sostituiti da un doppio due punti.

Le maschere di sottorete hanno da 0 a 128 bit. (Per esempio 21ac: 0000:0000:0611:21e0:00ba:321b:54da/64 o 21ac::611:21e0:321b:54da/64)

Copyright © 2023 The Rocky Enterprise Software Foundation

In un indirizzo web o URL (Uniform Resource Locator), un indirizzo IP può essere seguito da un due punti e dall'indirizzo della porta (che indica l'applicazione a cui i dati sono destinati). Inoltre per evitare confusione in un URL, l'indirizzo IPv6 è scritto in parentesi quadre [ ], due punti, indirizzo della porta.

Gli indirizzi IP e MAC devono essere univoci su una rete!

13.1.2 Dominio DNS

Le macchine client possono far parte di un dominio DNS (**Domain Name System**, ad esempio `mydomain.lan`).

Il nome completo del computer (**FQDN**) diventa `pc-rocky.mydomain.lan`.

Un insieme di computer può essere raggruppato in un set logico, che risolve i nomi, chiamato dominio DNS. Un dominio DNS non è, ovviamente, limitato a una singola rete fisica.

Affinché un computer faccia parte di un dominio DNS, è necessario fornire un suffisso DNS (qui `mydomain.lan`) e un server da poter interrogare.

13.1.3 Promemoria del modello OSI

Aiuto alla memoria

Per ricordare l'ordine degli strati del modello OSI, ricorda la seguente frase: **Please Do Not Touch Steven's Pet Alligator**.

Livello	Protocolli
7 - Applicazione	POP, IMAP, SMTP, SSH, SNMP, HTTP, FTP, ...
6 - Presentazione	ASCII, MIME, ...
5 - Sessione	TLS, SSL, NetBIOS, ...
4 - Trasporto	TLS, SSL, TCP, UDP, ...
3 - Rete	IPv4, IPv6, ARP, ...
2 - Collegamento dati	Ethernet, WiFi, Token Ring, ...
1 - Collegamento fisici	Cavi, fibre ottiche, onde radio, ...

Livello 1 (Fisico) supporta la trasmissione su un canale di comunicazione (Wifi, fibra ottica, cavo RJ, ecc.). Unità: il bit.

Copyright © 2023 The Rocky Enterprise Software Foundation

Livello 2 (Data Link) supporta la topologia di rete (token-ring, star, bus, etc.), divisione dei dati ed errori di trasmissione. Unità: il frame.

Livello 3 (Rete) supporta la trasmissione dati end-to-end (Routing IP = Gateway). Unità: il pacchetto.

Livello 4 (Trasporto) supporta il tipo di servizio (connesso o non connesso) crittografia e controllo del flusso. Unità: il segmento o il datagramma.

Livello 5 (Sessione) supporta la comunicazione tra due computer.

Livello 6 (Presentazione) rappresenta l'area indipendente dai dati a livello di applicazione. Essenzialmente questo livello traduce dal formato di rete al formato dell'applicazione, o dal formato dell'applicazione al formato di rete.

Layer 7 (Applicazione) rappresenta il contatto con l'utente. Fornisce i servizi offerti dalla rete: http, dns, ftp, imap, pop, smtp, etc.

13.2 La denominazione delle interfacce

lo è l'intefaccia di "**loopback**" che consente ai programmi TCP/IP di comunicare tra loro senza lasciare la macchina locale. Ciò consente di verificare se il modulo di rete **del sistema funziona correttamente** e consente anche il ping del localhost. Tutti i pacchetti che entrano attraverso localhost escono attraverso localhost. I pacchetti ricevuti corrispondono ai pacchetti inviati.

Il kernel Linux assegna i nomi delle interfacce con un prefisso specifico a seconda del tipo. Ad esempio tradizionalmente, tutte le interfacce **Ethernet**, iniziano con **eth**. Il prefisso è seguito da un numero, il primo è 0 (eth0, eth1, eth2...). Alle interfacce wifi è stato assegnato un prefisso wlan.

Sulle distribuzioni Linux Rocky 8, systemd nominerà le interfacce seguendo la nuova politica in cui "X" rappresenta un numero:

- `enoX`: dispositivi on-board
- `ensX`: slot hotplug PCI Express
- `enpXsX`: posizione fisica/geografica del connettore dell'hardware
- ...

Copyright © 2023 The Rocky Enterprise Software Foundation

13.3 Uso del comandi `ip`

Dimentica il vecchio comando `ifconfig`! Pensa `ip`!

... Note "Nota"

```
Commento per gli amministratori di vecchi sistemi Linux:

Il comando storico di gestione della rete è `ifconfig`. Questo comando è stato
sostituito dal comando `ip`, che è già ben noto agli amministratori di rete.

Il comando `ip` è l'unico comando per gestire **indirizzo IP, ARP, routing,
ecc.**.

Il comando `ifconfig` non è più installato per impostazione predefinita in
Rocky8.

È importante iniziare con le buone abitudini ora.
```

13.4 Il nome host

Il comando `hostname` visualizza o imposta il nome host del sistema

```
hostname [-f] [hostname]
```

Opzione	Descrizione
`-f`	Mostra il FQDN
`-i`	Visualizza gli indirizzi IP del sistema

Suggerimento

Questo comando viene utilizzato da vari programmi di rete per identificare la macchina.

Per assegnare un nome host, è possibile utilizzare il comando `hostname`, ma le modifiche non verranno mantenute all'avvio successivo. Il comando senza argomenti visualizza il nome host.

Per impostare il nome host, bisogna modificare il file `/etc/sysconfig/network`:

```
NETWORKING=yes
HOSTNAME=pc-rocky.mondomaine.lan
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Lo script di avvio di RedHat consulta anche il file `/etc/hosts` per risolvere il nome host del sistema.

All'avvio del sistema, Linux valuta il valore `HOSTNAME` nel file `/etc/sysconfig/network`.

Utilizza quindi il file `/etc/hosts` per valutare l'indirizzo IP principale del server e il suo nome host. E dedurre il nome di dominio DNS.

È quindi essenziale compilare questi due file prima di qualsiasi configurazione dei servizi di rete.

Suggerimento

Per sapere se questa configurazione è ben fatta, i comandi `hostname` e `hostname -f` devono rispondere con i valori previsti.

13.5 /etc/hosts file

Il file `/etc/hosts` è una tabella di mapping dei nomi host statici, che segue il seguente formato:

```
@IP <hostname>  [alias]  [# comment]
```

Esempio di un file `/etc/hosts`:

```
127.0.0.1       localhost localhost.localdomain
::1             localhost localhost.localdomain
192.168.1.10    rockstar.rockylinux.lan rockstar
```

Il file `/etc/hosts` viene ancora utilizzato dal sistema, soprattutto al momento dell'avvio quando viene determinato il nome di dominio completo del sistema (FQDN).

Suggerimento

RedHat raccomanda che sia compilata almeno una linea con il nome del sistema.

Se il servizio **DNS** (**D**domain **N**ame **S**ervice) non è presente, è necessario compilare tutti i nomi nel file hosts per ciascuno dei computer.

Copyright © 2023 The Rocky Enterprise Software Foundation

Il file `/etc/hosts` contiene una riga per voce, con l'indirizzo IP, il nome di dominio completo, quindi il nome host (in quest'ordine) e una serie di alias (alias1 alias2 ...). L'alias è opzionale.

13.6 il file `/etc/nsswitch.conf`

Il **NSS** (**N**ame **S**ervice **S**witch) consente di sostituire i file di configurazione (ad esempio `/etc/passwd`, `/etc/group`, `/etc/hosts`) con uno o più database centralizzati.

Il file `/etc/nsswitch.conf` viene utilizzato per configurare i database del servizio dei nomi.

```
passwd: files
shadow: files
group: files

hosts: files dns
```

In questo caso, Linux cercherà prima una corrispondenza del nome host (riga `hosts:`) nel file `/etc/hosts` (valore `files`) prima di interrogare il DNS (valore `dns`)! Questo comportamento può essere variato modificando il file `/etc/nsswitch.conf`.

Naturalmente, è possibile immaginare di interrogare un LDAP, MySQL o altro server configurando il servizio dei nomi per rispondere alle richieste di sistema per host, utenti, gruppi, ecc.

La risoluzione del servizio dei nomi può essere testata con il comando `getent` che vedremo più avanti in questo corso.

13.7 file `/etc/resolv.conf`

Il file `/etc/resolv.conf` contiene la configurazione della risoluzione dei nomi DNS.

```
#Generated by NetworkManager
domain mondomaine.lan
search mondomaine.lan
nameserver 192.168.1.254
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Suggerimento

Questo file è ormai storia. Non è più compilato direttamente!

Le nuove generazioni di distribuzioni hanno generalmente integrato il servizio `NetworkManager`. Questo servizio consente di gestire la configurazione in modo più efficiente, sia in modalità grafica che console.

Consente l'aggiunta di server DNS dal file di configurazione di un'interfaccia di rete. Quindi popola dinamicamente il file `/etc/resolv.conf` che non dovrebbe mai essere modificato direttamente, altrimenti le modifiche alla configurazione andranno perse al successivo avvio del servizio di rete.

13.8 comando `ip`

Il comando `ip` del pacchetto `iproute2` consente di configurare un'interfaccia e la relativa tabella di routing.

Mostra le interfacce :

```
[root]# ip link
```

Mostra le informazioni sulle interfacce:

```
[root]# ip addr show
```

Mostra le informazioni su una interfaccia :

```
[root]# ip addr show eth0
```

Mostra la tabella ARP:

```
[root]# ip neigh
```

Tutti i comandi di gestione della rete storici sono stati raggruppati sotto il comando `ip`, che è ben noto agli amministratori di rete.

Copyright © 2023 The Rocky Enterprise Software Foundation

13.9 configurazione DHCP

Il protocollo **DHCP** (**D**ynamic **H**ost **C**Control **P**rotocol) consente di ottenere una configurazione IP completa tramite la rete. Questa è la modalità di configurazione predefinita di un'interfaccia di rete sotto Rocky Linux, il che spiega perché un sistema connesso alla rete attraverso un router internet può funzionare senza ulteriori configurazioni.

La configurazione delle interfacce sotto Rocky Linux è contenuta nella cartella `/etc/sysconfig/network-scripts/`.

Per ogni interfaccia Ethernet, un file `ifcfg-ethX` consente la configurazione dell'interfaccia associata.

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
HWADDR=00:0c:29:96:32:e3
```

- Nome dell'interfaccia : (deve essere nel nome del file)

```
DEVICE=eth0
```

- Avvia automaticamente l'interfaccia:

```
ONBOOT=yes
```

- Effettuare una richiesta DHCP all'avvio dell'interfaccia:

```
BOOTPROTO=dhcp
```

- Specificare l'indirizzo MAC (opzionale ma utile quando ci sono diverse interfacce):

```
HWADDR=00:0c:29:96:32:e3
```


Copyright © 2023 The Rocky Enterprise Software Foundation