

debian

Debian Reference

Osamu Aoki

Copyright © 2013-2024 Osamu Aoki

Questa guida di riferimento Debian, Debian Reference (versione 2.152) (2026-10-10 14:56:25 UTC) è pensata per fornire un'ampia panoramica del sistema Debian in qualità di guida post-installazione per l'utente. Copre molti aspetti dell'amministrazione di sistema attraverso esempi di uso di comandi di shell, pensati per chi non è sviluppatore.

Indice

1 Tutorial GNU/Linux **1**

1.1 Nozioni di base sulla console . . . 1

1.1.1 Il prompt di shell . . . 1

1.1.2 Il prompt di shell in una GUI . . . 2

1.1.3 L'account root . . . 2

1.1.4 Il prompt di shell di root . . . 3

1.1.5 Strumenti di amministrazione del sistema con interfaccia utente grafica . . . 3

1.1.6 Console virtuali . . . 3

1.1.7 Come uscire dal prompt dei comandi . . . 4

1.1.8 Come spegnere il sistema . . . 4

1.1.9 Ripristinare una console funzionante . . . 4

1.1.10 Suggerimenti per pacchetti aggiuntivi per i principianti . . . 4

1.1.11 Un account utente extra . . . 5

1.1.12 Configurazione di sudo . . . 5

1.1.13 Tempo di giocare . . . 6

1.2 File system stile Unix . . . 6

1.2.1 Nozioni di base sui file Unix . . . 7

1.2.2 Aspetti tecnici del file system . . . 8

1.2.3 Permessi del file system . . . 8

1.2.4 Controllo dei permessi per i file appena creati: umask . . . 11

1.2.5 Permessi per gruppi di utenti (gruppi) . . . 11

1.2.6 Orari . . . 13

1.2.7 Collegamenti . . . 14

1.2.8 Pipe con nome (FIFO) . . . 15

1.2.9 Socket . . . 15

1.2.10 File di device . . . 16

1.2.11 File di device speciali . . . 17

1.2.12 procfs e sysfs . . . 17

1.2.13 tmpfs . . . 17

1.3 Midnight Commander (MC) . . . 18

1.3.1 Personalizzazione di MC . . . 18
1.3.2 Avvio di MC . . . 18
1.3.3 Gestore dei file in MC . . . 19
1.3.4 Trucchetti per la riga di comando di MC . . . 19
1.3.5 L'editor interno di MC . . . 19
1.3.6 Il visualizzatore interno di MC . . . 20
1.3.7 Funzionalità di avvio automatico di MC . . . 20
1.3.8 File system virtuale di MC . . . 20
1.4 Ambiente di lavoro di base in stile Unix . . . 20
1.4.1 La shell di login . . . 21
1.4.2 Personalizzare bash . . . 21
1.4.3 Associazioni di tasti speciali . . . 22
1.4.4 Operazioni del mouse . . . 23
1.4.5 Il paginatore . . . 23
1.4.6 L'editor di testo . . . 24
1.4.7 Impostare un editor di testi predefinito . . . 24
1.4.8 Usare vim . . . 24
1.4.9 Registrare le attività della shell . . . 26
1.4.10 Comandi Unix di base . . . 26
1.5 Il semplice comando di shell . . . 28
1.5.1 Esecuzione dei comandi e variabili d'ambiente . . . 28
1.5.2 La variabile "$LANG" . . . 29
1.5.3 La variabile "$PATH" . . . 30
1.5.4 La variabile "$HOME" . . . 30
1.5.5 Opzioni della riga di comando . . . 30
1.5.6 Glob della shell . . . 31
1.5.7 Valore restituito dal comando . . . 32
1.5.8 Sequenze tipiche di comandi e ridirezione della shell . . . 32
1.5.9 Alias di comandi . . . 34
1.6 Elaborazione di testo stile Unix . . . 35
1.6.1 Strumenti Unix per il testo . . . 35
1.6.2 Espressioni regolari . . . 36
1.6.3 Espressioni di sostituzione . . . 37
1.6.4 Sostituzione globale con espressioni regolari . . . 37
1.6.5 Estrarre dati da file con tabelle di testo . . . 38
1.6.6 Frammenti di script per comandi con pipe . . . 40

2 Gestione dei pacchetti in Debian 41

2.1 Prerequisiti per la gestione dei pacchetti Debian . . . 41
2.1.1 Sistema di gestione dei pacchetti in Debian . . . 41
2.1.2 Configurazione dei pacchetti . . . 42
2.1.3 Precauzioni base . . . 42
2.1.4 Una vita di aggiornamenti senza fine . . . 43
2.1.5 Nozioni di base sugli archivi Debian . . . 44
2.1.6 Debian è al 100% software libero . . . 48
2.1.7 Dipendenze dei pacchetti . . . 49
2.1.8 Il flusso di eventi nella gestione dei pacchetti . . . 50
2.1.9 Prima risposta a problemi di gestione dei pacchetti . . . 51
2.1.10 Come scegliere i pacchetti Debian . . . 51
2.1.11 Come affrontare requisiti in conflitto . . . 52
2.2 Operazioni base per la gestione dei pacchetti . . . 52
2.2.1 Confronti `apt` - `apt-get` / `apt-cache` - `aptitude` . . . 53
2.2.2 Operazioni base per la gestione dei pacchetti dalla riga di comando . . . 54
2.2.3 Uso interattivo di aptitude . . . 55
2.2.4 Associazioni dei tasti per aptitude . . . 55
2.2.5 Viste dei pacchetti in aptitude . . . 56
2.2.6 Opzioni per i metodi di ricerca in aptitude . . . 57
2.2.7 La struttura delle espressioni regolari di aptitude . . . 58
2.2.8 Risoluzione delle dipendenze di aptitude . . . 58
2.2.9 Registri delle attività sui pacchetti . . . 58
2.3 Esempi di operazioni con aptitude . . . 58
2.3.1 Cercare pacchetti interessanti . . . 60
2.3.2 Elencare pacchetti in base alla corrispondenza del nome con espressioni regolari . . . 60
2.3.3 Sfogliare le corrispondenze ad una espressione regolare . . . 60
2.3.4 Eliminare completamente i pacchetti rimosssi . . . 60
2.3.5 Mettere ordine nello stato di installazione automatico/non automatico . . . 61
2.3.6 Aggiornamento di tutto il sistema . . . 61
2.4 Operazioni avanzate per la gestione dei pacchetti . . . 62
2.4.1 Operazioni avanzate per la gestione dei pacchetti dalla riga di comando . . . 62
2.4.2 Verifica dei pacchetti installati . . . 64
2.4.3 Salvaguardarsi da problemi coi pacchetti . . . 64
2.4.4 Cercare tra i metadati dei pacchetti . . . 65
2.5 Aspetti tecnici della gestione dei pacchetti in Debian . . . 65
2.5.1 Metadati degli archivi . . . 65
2.5.2 File ”Release” nella directory principale ed autenticità . . . 66
2.5.3 File ”Release” a livello di archivio . . . 67

2.5.4 Recuperare i metadati per un pacchetto . . . 67
2.5.5 Lo stato dei pacchetti per APT . . . 68
2.5.6 Lo stato dei pacchetti per aptitude . . . 68
2.5.7 Copie locali dei pacchetti scaricati . . . 68
2.5.8 Nomi dei file di pacchetto di Debian . . . 68
2.5.9 Il comando dpkg . . . 69
2.5.10 Il comando update-alternatives . . . 70
2.5.11 Il comando dpkg-statoverride . . . 70
2.5.12 Il comando dpkg-divert . . . 71
2.6 Ripristino da un sistema con problemi . . . 71
2.6.1 Incompatibilità con vecchie configurazioni utente . . . 71
2.6.2 Errori di cache dei dati dei pacchetti . . . 71
2.6.3 Ripristinare con il comando dpkg . . . 72
2.6.4 Installazione fallita a causa di dipendenze mancanti . . . 72
2.6.5 Pacchetti diversi con file sovrapposti . . . 72
2.6.6 Risolvere problemi negli script di pacchetto . . . 73
2.6.7 Ripristinare i dati sui pacchetti selezionati . . . 73
2.7 Suggerimenti per la gestione dei pacchetti . . . 74
2.7.1 Chi ha caricato il pacchetto? . . . 74
2.7.2 Limitare l'uso di banda di APT per gli scaricamenti . . . 74
2.7.3 Scaricare ed aggiornare automaticamente i pacchetti . . . 74
2.7.4 Updates e Backports . . . 75
2.7.5 Archivi di pacchetti esterni . . . 75
2.7.6 Pacchetti da fonti miste di archivi senza l'**uso dei pin di APT**. . . . 75
2.7.7 Modificare la versione candidata con l'**uso dei pin di APT**. . . . 77
2.7.8 Bloccare i pacchetti da installare perché "Raccomandati" . . . 78
2.7.9 Usare e aggiornare `testing` con alcuni pacchetti da `unstable` . . . 78
2.7.10 Usare e aggiornare `unstable` con alcuni pacchetti da `experimental` . . . 80
2.7.11 Retrocessione di emergenza alla versione precedente . . . 80
2.7.12 Il pacchetto equivs . . . 81
2.7.13 Fare il port di un pacchetto nel sistema stabile . . . 82
2.7.14 Server proxy per APT . . . 82
2.7.15 Ulteriori letture sulla gestione dei pacchetti . . . 83

3 Inizializzazione del sistema 84
3.1 Panoramica del processo di avvio . . . 84
3.1.1 Stadio 1: l'UEFI . . . 84
3.1.2 Stadio 2: il bootloader . . . 85
3.1.3 Stadio 3: il mini-sistema Debian . . . 86
3.1.4 Stadio 4: il normale sistema Debian . . . 87
3.2 Sistema di ripristino/recupero . . . 87
3.2.1 Sistema di recupero GRUB UEFI su USB . . . 88
3.2.2 Sistema di ripristino Linux Live su USB . . . 89
3.2.3 Sistema di ripristino Linux Live da GRUB . . . 90
3.3 Systemd . . . 90
3.3.1 Init systemd . . . 90
3.3.2 Login di systemd . . . 91
3.4 I messaggi del kernel . . . 92
3.5 I messaggi di sistema . . . 92
3.6 Gestione del sistema . . . 92
3.7 Altri strumenti di monitoraggio del sistema . . . 94
3.8 Configurazione del sistema . . . 94
3.8.1 Il nome host . . . 94
3.8.2 Il filesystem . . . 94
3.8.3 Inizializzazione delle interfacce di rete . . . 95
3.8.4 Inizializzazione del sistema cloud . . . 95
3.8.5 Esempio di personalizzazione per modificare il servizio sshd . . . 95
3.9 Il sistema udev . . . 96
3.10 L'inizializzazione dei moduli del kernel . . . 96

4 Autenticazione e controllo degli accessi 98
4.1 Autenticazione Unix normale . . . 98
4.2 Gestire le informazioni su account e password . . . 100
4.3 Password buone . . . 100
4.4 Creare password cifrate . . . 101
4.5 PAM e NSS . . . 101
4.5.1 File di configurazione letti da PAM e NSS . . . 102
4.5.2 La moderna gestione centralizzata del sistema . . . 102
4.5.3 "Perché GNU su non supporta il gruppo wheel" . . . 103
4.5.4 Regole più stringenti per le password . . . 103
4.6 Sicurezza dell'autenticazione . . . 104
4.6.1 Password sicure in Internet . . . 104
4.6.2 Secure Shell, shell sicura . . . 104

4.6.3 Misure aggiuntive di sicurezza per Internet . . . 105
4.6.4 Rendere sicura la password di root . . . 105
4.7 Altri controlli sugli accessi . . . 106
4.7.1 ACL (Access Control List, liste di controllo degli accessi) . . . 106
4.7.2 sudo . . . 107
4.7.3 PolicyKit . . . 107
4.7.4 Limitare l'accesso ad alcuni servizi server . . . 107
4.7.5 Funzionalità di sicurezza di Linux . . . 108

5 Impostazione della rete 109
5.1 L'infrastruttura base di rete . . . 109
5.1.1 Risoluzione dei nomi di host . . . 109
5.1.2 Il nome dell'interfaccia di rete . . . 111
5.1.3 L'intervallo degli indirizzi di rete per la LAN . . . 112
5.1.4 Il supporto per i dispositivi di rete . . . 112
5.2 La configurazione moderna della rete per il desktop . . . 112
5.2.1 Strumenti grafici di configurazione della rete . . . 113
5.3 La configurazione moderna della rete senza GUI . . . 113
5.4 La configurazione moderna della rete per il cloud . . . 114
5.4.1 La configurazione moderna della rete per il cloud con DHCP . . . 114
5.4.2 La configurazione moderna della rete per il cloud con IP statico . . . 114
5.4.3 La configurazione moderna della rete per il cloud con Network Manager . . . 114
5.5 Configurazione della rete a basso livello . . . 115
5.5.1 Comandi iproute2 . . . 115
5.5.2 Operazioni sicure a basso livello sulla rete . . . 115
5.6 Ottimizzazione della rete . . . 116
5.6.1 Trovare l'MTU ottimale . . . 116
5.6.2 Ottimizzazione di TCP per la WAN . . . 117
5.7 Infrastruttura netfilter . . . 117

6 Applicazioni per la rete 120
6.1 Browser web . . . 120
6.1.1 Falsificare la stringa User-Agent . . . 121
6.1.2 Estensione per il browser . . . 121
6.2 Il sistema di posta . . . 121
6.2.1 Nozioni di base sulla posta elettronica . . . 121
6.2.2 Limitazioni moderne ai servizi di posta . . . 122
6.2.3 Attese storiche da un servizio di posta . . . 123
6.2.4 Agente di trasporto della posta (MTA) . . . 123

6.2.4.1 La configurazione di exim4 . . . 123
6.2.4.2 La configurazione di Postfix con SASL . . . 125
6.2.4.3 La configurazione dell'indirizzo di posta . . . 126
6.2.4.4 Operazioni di base degli MTA . . . 127
6.3 Il server e le utilità per l'accesso remoto (SSH) . . . 127
6.3.1 Nozioni di base su SSH . . . 128
6.3.2 Nome utente sull'host remoto . . . 129
6.3.3 Connettersi senza password remote . . . 129
6.3.4 Gestire client SSH estranei . . . 129
6.3.5 Impostare ssh-agent . . . 130
6.3.6 Inviare un messaggio di posta da un host remoto . . . 130
6.3.7 Forwarding della porta per tunnel SMTP/POP3 . . . 130
6.3.8 Spegnere il sistema remoto su SSH . . . 130
6.3.9 Risolvere il problemi con SSH . . . 131
6.4 Server ed utilità per la stampa . . . 131
6.5 Altri server di rete . . . 131
6.6 Altri client di rete . . . 132
6.7 Diagnosi dei demoni di sistema . . . 132

7 Sistema GUI 134

7.1 Ambiente desktop GUI . . . 134
7.2 Il protocollo di comunicazione della GUI . . . 135
7.3 Infrastruttura GUI . . . 136
7.4 Applicazioni GUI . . . 137
7.5 Directory dell'utente . . . 137
7.6 Tipi di carattere . . . 137
7.6.1 Tipi di carattere base . . . 137
7.6.2 Rasterizzazione dei tipi di carattere . . . 140
7.7 Sandbox . . . 140
7.8 Desktop remoto . . . 142
7.9 Connessione a server X . . . 142
7.9.1 Connessione locale al server X . . . 142
7.9.2 Connessione remota a server X . . . 143
7.9.3 Connessione chroot a server X . . . 143
7.10 Appunti . . . 143

8 I18N e L10N 145

8.1 La localizzazione . . . 145
8.1.1 Logica alla base dell'uso della localizzazione UTF-8 . . . 145
8.1.2 La (ri)configurazione della localizzazione . . . 146
8.1.3 Codifica per i nomi di file . . . 147
8.1.4 Messaggi localizzati e documentazione tradotta . . . 147
8.1.5 Effetti della localizzazione . . . 148
8.2 L'input da tastiera . . . 148
8.2.1 L'input da tastiera per la console Linux e X Window . . . 149
8.2.2 L'input da tastiera per Wayland . . . 149
8.2.3 Il supporto per metodo di input con IBus . . . 149
8.2.4 Il supporto per metodo di input con Fcitx . . . 149
8.2.5 Un esempio per il giapponese . . . 152
8.3 L'output sul display . . . 152
8.3.1 Configurazione del terminale . . . 153
8.3.2 Caratteri dell'Asia dell'est con larghezza ambigua . . . 153

9 Suggerimenti per il sistema 154

9.1 Suggerimenti per la console . . . 154
9.1.1 Registrare le attività della shell in modo pulito . . . 154
9.1.2 Il programma screen . . . 155
9.1.3 Navigare nelle directory . . . 156
9.1.4 Wrapper per readline . . . 156
9.1.5 Strumenti per la fusione di codice sorgente . . . 156
9.2 Personalizzare vim . . . 157
9.2.1 Personalizzare vim con le funzionalità interne . . . 157
9.2.2 Personalizzare vim con pacchetti esterni . . . 159
9.3 Registrazione e presentazione di dati . . . 160
9.3.1 Demoni di registro . . . 160
9.3.2 Analizzatori di registro . . . 160
9.3.3 Visualizzazione personalizzata di dati di testo . . . 161
9.3.4 Visualizzazione personalizzata di date e orari . . . 161
9.3.5 Output colorato per la shell . . . 162
9.3.6 Comandi colorati . . . 162
9.3.7 Registrare le attività dell'editor per ripetizioni complesse . . . 163
9.3.8 Registrare l'immagine grafica di un'applicazione X . . . 163
9.3.9 Registrare i cambiamenti a file di configurazione . . . 163
9.4 Monitorare, controllare ed avviare l'attività dei programmi . . . 164
9.4.1 Cronometrare un processo . . . 164

9.4.2 Priorità di schedulazione . . . 164
9.4.3 Il comando ps . . . 165
9.4.4 Il comando top . . . 165
9.4.5 Elencare i file aperti da un processo . . . 165
9.4.6 Tenere traccia delle attività di un programma . . . 165
9.4.7 Identificazione di processi in base a file o socket . . . 166
9.4.8 Ripetere un comando ad intervalli costanti . . . 166
9.4.9 Ripetere un comando su diversi file . . . 166
9.4.10 Avviare un programma dalla GUI . . . 167
9.4.11 Personalizzare i programmi da avviare . . . 168
9.4.12 Uccidere un processo . . . 169
9.4.13 Pianificare compiti da eseguire una volta sola . . . 169
9.4.14 Pianificare compiti in modo regolare . . . 169
9.4.15 Pianificare compiti da eseguire in caso di un evento . . . 171
9.4.16 Tasto Alt-SysRq . . . 171
9.5 Suggerimenti per l'amministrazione del sistema . . . 172
9.5.1 Chi è nel sistema? . . . 172
9.5.2 Avvertire tutti gli utenti . . . 172
9.5.3 Identificazione dell'hardware . . . 172
9.5.4 Configurazione dell'hardware . . . 172
9.5.5 Orario di sistema e hardware . . . 174
9.5.6 L'infrastruttura audio . . . 174
9.5.7 Disabilitare il salvaschermo . . . 175
9.5.8 Disabilitare i bip sonori . . . 175
9.5.9 Uso della memoria . . . 175
9.5.10 Verifica della sicurezza e dell'integrità del sistema . . . 176
9.6 Suggerimenti per l'archiviazione dei dati . . . 177
9.6.1 Uso dello spazio su disco . . . 177
9.6.2 Configurazione del partizionamento dei dischi . . . 177
9.6.3 Accedere alle partizioni usando UUID . . . 178
9.6.4 LVM2 . . . 178
9.6.5 Configurazione del file system . . . 179
9.6.6 Creare file system e verificarne l'integrità . . . 179
9.6.7 Ottimizzare il file system con opzioni di mount . . . 180
9.6.8 Ottimizzare il file system tramite il superblocco . . . 180
9.6.9 Ottimizzare il disco rigido . . . 181
9.6.10 Ottimizzare le unità a stato solido . . . 181
9.6.11 Usare SMART per prevedere danni ai dischi fissi . . . 181
9.6.12 Specificare una directory per l'archiviazione di dati temporanei usando $TMPDIR . . . 182

9.6.13 Espandere lo spazio di archiviazione utilizzabile con LVM . . . 182
9.6.14 Espandere lo spazio di archiviazione utilizzabile montando un'altra partizione . . . 182
9.6.15 Espandere lo spazio di archiviazione utilizzabile montando un collegamento ad un'altra directory 182
9.6.16 Espandere lo spazio di archiviazione utilizzabile montando un'altra directory da sovrapporre . 183
9.6.17 Espandere lo spazio di archiviazione utilizzabile usando collegamenti simbolici . . . 183
9.7 Immagine del disco . . . 183
9.7.1 Creare un file con un'immagine di disco . . . 183
9.7.2 Scrivere direttamente sul disco . . . 184
9.7.3 Montare un file con un'immagine di disco . . . 184
9.7.4 Pulire un file con un'immagine di disco . . . 186
9.7.5 Creare un file con immagine di disco vuoto . . . 186
9.7.6 Creare un file con un'immagine ISO9660 . . . 187
9.7.7 Scrivere direttamente sul CD/DVD-R/RW . . . 187
9.7.8 Montare un file con un'immagine ISO9660 . . . 188
9.8 I dati binari . . . 188
9.8.1 Visualizzare e modificare dati binari . . . 188
9.8.2 Manipolare file senza montare i dischi . . . 188
9.8.3 Dati ridondanti . . . 189
9.8.4 recupero di file dati ed analisi forensi . . . 189
9.8.5 Suddividere un file grande in file più piccoli . . . 190
9.8.6 Pulire il contenuto di file . . . 190
9.8.7 File fittizi . . . 190
9.8.8 Cancellare un intero disco fisso . . . 191
9.8.9 Cancellare area inutilizzate di un disco fisso . . . 191
9.8.10 De-cancellare file cancellati ma ancora aperti . . . 192
9.8.11 Cercare tutti i collegamenti fisici . . . 192
9.8.12 Consumo invisibile dello spazio su disco . . . 192
9.9 Suggerimenti per la cifratura dei dati . . . 193
9.9.1 Cifratura di dischi removibili con dm-crypt/LUKS . . . 193
9.9.2 Montare dischi removibili con dm-crypt/LUKS . . . 194
9.10 Il kernel . . . 194
9.10.1 Parametri del kernel . . . 194
9.10.2 Header del kernel . . . 194
9.10.3 Compilare il kernel ed i moduli relativi . . . 195
9.10.4 Compilare i sorgenti del kernel: il metodo raccomandato dal Team del Kernel di Debian . . . 196
9.10.5 Driver per hardware e firmware . . . 196
9.11 Sistema virtualizzato . . . 197
9.11.1 Strumenti per virtualizzazione ed emulazione . . . 197
9.11.2 Fasi del processo di virtualizzazione . . . 199
9.11.3 Montare il file immagine di disco virtuale . . . 199
9.11.4 Sistema chroot . . . 200
9.11.5 Sistemi desktop multipli . . . 201

10 Gestione dei dati **202**

10.1 Condividere, copiare ed archiviare . . . 202
10.1.1 Strumenti di archiviazione e compressione . . . 203
10.1.2 Strumenti di copia e sincronizzazione . . . 203
10.1.3 Esempi di invocazione per archivi . . . 203
10.1.4 Esempi di invocazione per la copia . . . 205
10.1.5 Esempi di invocazione per la selezione di file . . . 206
10.1.6 Supporti di archiviazione . . . 207
10.1.7 Supporti di archiviazione removibili . . . 208
10.1.8 Scelta del file system per la condivisione di dati . . . 209
10.1.9 Condividere dati attraverso una rete . . . 211
10.2 Backup e ripristino . . . 211
10.2.1 Politica di backup e ripristino . . . 211
10.2.2 Suite con utilità di backup . . . 213
10.2.3 Suggerimenti per i backup . . . 214
10.2.3.1 Backup con GUI . . . 214
10.2.3.2 Backup attivato da eventi di mount . . . 215
10.2.3.3 Backup attivato da un evento timer . . . 215
10.3 Infrastruttura di sicurezza dei dati . . . 216
10.3.1 Gestione delle chiavi per GnuPG . . . 217
10.3.2 Usare GnuPG su file . . . 218
10.3.3 Usare GnuPG con Mutt . . . 218
10.3.4 Usare GnuPG con Vim . . . 218
10.3.5 Somme di controllo MD5 . . . 220
10.3.6 Portachiavi per le password . . . 220
10.4 Strumenti per la fusione di codice sorgente . . . 220
10.4.1 Estrarre differenze da file sorgenti . . . 220
10.4.2 Fondere aggiornamenti per file sorgenti . . . 222
10.4.3 Unione (merge) interattiva . . . 222
10.5 Git . . . 222
10.5.1 Configurazione del client Git . . . 222
10.5.2 Comandi Git di base . . . 224
10.5.3 Suggerimenti per Git . . . 224
10.5.4 Documenti di consultazione per Git . . . 225
10.5.5 Altri sistemi di controllo delle versioni . . . 225

11 Conversione di dati **228**
11.1 Strumenti di conversione di dati testuali . . . 228
11.1.1 Convertire un file di testo con iconv . . . 228
11.1.2 Controllare se un file è in UTF-8 con iconv . . . 230
11.1.3 Convertire nomi di file con iconv . . . 230
11.1.4 Conversione del carattere di fine riga . . . 230
11.1.5 Conversione di tabulazioni . . . 231
11.1.6 Editor con auto-conversione . . . 231
11.1.7 Estrazione del testo puro . . . 232
11.1.8 Evidenziare e formattare dati in puro testo . . . 233
11.2 Dati XML . . . 233
11.2.1 Suggerimenti base per XML . . . 233
11.2.2 Elaborazione XML . . . 234
11.2.3 Estrazione di dati XML . . . 235
11.2.4 Pulizia di dati XML . . . 235
11.3 Impaginazione . . . 237
11.3.1 Impaginazione roff . . . 237
11.3.2 TeX/LaTeX . . . 237
11.3.3 Fare una bella stampa di una pagina di manuale . . . 238
11.3.4 Creare una pagina di manuale . . . 238
11.4 Dati stampabili . . . 239
11.4.1 Ghostscript . . . 239
11.4.2 Unire due file PS o PDF . . . 239
11.4.3 Utilità per dati stampabili . . . 240
11.4.4 Stampare con CUPS . . . 240
11.5 Conversione dei dati di posta . . . 241
11.5.1 Nozioni di base sui dati di posta . . . 241
11.6 Strumenti per dati grafici . . . 242
11.6.1 Strumenti per dati grafici (metapacchetto) . . . 242
11.6.2 Strumenti per dati grafici (GUI) . . . 242
11.6.3 Strumenti per dati grafici (CLI) . . . 242
11.7 Conversioni di dati vari . . . 245

12 Programmazione **246**
12.1 Script shell . . . 246
12.1.1 Compatibilità con la shell POSIX . . . 247
12.1.2 Parametri di shell . . . 247
12.1.3 Costrutti condizionali della shell . . . 249
12.1.4 Cicli di shell . . . 249

12.1.5 Variabile d'ambiente della shell . . . 250
12.1.6 La sequenza di elaborazione della riga di comando di shell . . . 250
12.1.7 Programmi di utilità per script di shell . . . 251
12.2 Creazione di script in linguaggi interpretati . . . 253
12.2.1 Fare il debug di codice di linguaggi interpretati . . . 253
12.2.2 Programma GUI con script di shell . . . 253
12.2.3 Azioni personalizzate per un gestore di file con GUI . . . 254
12.2.4 Pazzie con corti script Perl . . . 254
12.3 Scrivere codice in linguaggi compilati . . . 255
12.3.1 C . . . 255
12.3.2 Semplice programma in C (gcc) . . . 256
12.3.3 Flex - un Lex migliorato . . . 256
12.3.4 Bison - Yacc migliorato . . . 256
12.4 Strumenti di analisi statica del codice . . . 258
12.5 Debug . . . 258
12.5.1 Esecuzione base di gdb . . . 258
12.5.2 Fare il debug di pacchetti Debian . . . 260
12.5.3 Ottenere un backtrace . . . 261
12.5.4 Comandi gdb avanzati . . . 261
12.5.5 Controllare le dipendenze dalle librerie . . . 261
12.5.6 Strumenti per tracciamento di chiamate dinamiche . . . 262
12.5.7 Fare il debug di errori X . . . 262
12.5.8 Strumenti per rilevazione di memory leak . . . 262
12.5.9 Disassemblatore di binari . . . 262
12.6 Strumenti di compilazione . . . 263
12.6.1 Make . . . 263
12.6.2 Autotools . . . 264
12.6.2.1 Compilare ed installare un programma . . . 264
12.6.2.2 Disinstallare un programma . . . 264
12.6.3 Meson . . . 265
12.7 Web . . . 265
12.8 Traduzione di codice sorgente . . . 266
12.9 Creare pacchetti Debian . . . 266

A Appendice **267**

A.1 Il labirinto Debian . . . 267
A.2 Storia del copyright . . . 267
A.3 Formato del documento . . . 268

Elenco delle tabelle

1.1 Elenco di pacchetti con interessanti programmi in modalità testuale . . . 5
1.2 Elenco di pacchetti con documentazione interessante . . . 5
1.3 Elenco degli usi delle directory principali . . . 8
1.4 Elenco dei valori possibili per il primo carattere nell'output di "`ls -l`" . . . 9
1.5 Permessi in notazione numerica per i comandi chmod(1) . . . 10
1.6 Esempi di valori di **umask** . . . 11
1.7 Elenco dei principali gruppi forniti dal sistema per accesso ai file . . . 12
1.8 Elenco dei principali gruppi forniti dal sistema per l'esecuzione di particolari comandi . . . 13
1.9 Elenco dei tipi di data . . . 13
1.10 Elenco di file dei device speciali . . . 17
1.11 Le associazioni dei tasti di MC . . . 19
1.12 La reazione al tasto Invio in MC . . . 20
1.13 Elenco di programmi shell . . . 21
1.14 Elenco di associazioni di tasti per bash . . . 22
1.15 Elenco di funzioni del mouse e relative azioni dei tasti in Debian . . . 23
1.16 Elenco di combinazioni di tasti base di Vim . . . 25
1.17 Elenco di comandi Unix di base . . . 27
1.18 Le 3 parti del valore di localizzazione . . . 29
1.19 Elenco di localizzazioni raccomandate . . . 29
1.20 Elenco di valori di "`$HOME`" . . . 30
1.21 Modelli di glob della shell . . . 31
1.22 Codici di uscita dei comandi . . . 32
1.23 Idiomi per comandi di shell . . . 33
1.24 Descrittori di file predefiniti . . . 34
1.25 Metacaratteri per BRE e ERE . . . 36
1.26 L'espressione di sostituzione . . . 37
1.27 Elenco di frammenti di script per comandi con pipe . . . 40

2.1 Elenco degli strumenti Debian di gestione dei pacchetti . . . 42
2.2 Elenco dei siti con l'archivo Debian . . . 45

2.3 Elenco delle aree dell'archivio Debian . . . 46
2.4 Relazione tra suite e nome in codice . . . 47
2.5 Elenco dei siti web importanti per la risoluzione di problemi con un pacchetto specifico . . . 51
2.6 Operazioni base di gestione dei pacchetti dalla riga di comando usando aptitude(8) e apt-get(8) /apt-cache(8) . . . 54
2.7 Opzioni degne di nota per il comando aptitude(8) . . . 55
2.8 Elenco delle associazioni di tasti per aptitude . . . 56
2.9 Elenco delle viste di aptitude . . . 57
2.10 Organizzazione delle viste standard dei pacchetti . . . 57
2.11 Elenco delle regole per espressioni regolari di aptitude . . . 59
2.12 File di registro per le attività sui pacchetti . . . 60
2.13 Elenco delle operazioni avanzate per la gestione dei pacchetti . . . 63
2.14 Il contenuto dei metadati dell'archivio Debian . . . 65
2.15 La struttura dei nomi dei pacchetti Debian . . . 68
2.16 I caratteri utilizzabili all'interno di ciascuna porzione del nome dei pacchetti Debian . . . 69
2.17 I file degni di nota creati da `dpkg` . . . 70
2.18 Elenco di valori di priorità di pin che esemplificano la tecnica d'**uso dei pin di APT**. . . . 78
2.19 Elenco degli strumenti proxy specifici per l'archivio Debian . . . 82

3.1 Elenco di bootloader . . . 85
3.2 Il significato della voce di menu della porzione soprastante di `/boot/grub/grub.cfg` . . . 86
3.3 Elenco di utilità di avvio per il sistema Debian . . . 88
3.4 Elenco dei livelli di errore del kernel . . . 92
3.5 Elenco di tipici esempi di comandi per `systemd` . . . 92
3.6 Elenco di tipici esempi di comandi `systemctl` . . . 93
3.7 Elenco di altri esempi di comandi per il monitoraggio in `systemd` . . . 94

4.1 I 3 importanti file di configurazione per pam_unix(8) . . . 98
4.2 Il contenuto della seconda voce di "`/etc/passwd`" . . . 99
4.3 Elenco di comandi per gestire informazioni su account . . . 100
4.4 Elenco di strumenti per generare password . . . 101
4.5 Elenco dei pacchetti degni di nota per i sistemi PAM e NSS . . . 101
4.6 Elenco di file di configurazione letti da PAM e NSS . . . 102
4.7 Elenco di servizi e porte sicuri e non sicuri . . . 104
4.8 Elenco di strumenti per fornire misure aggiuntive di sicurezza . . . 105

5.1 Elenco degli strumenti di configurazione della rete . . . 110
5.2 Elenco di intervalli di indirizzi di rete . . . 112
5.3 Tabella di traduzione dai comandi obsoleti `net-tools` ai nuovi comandi `iproute2` . . . 115
5.4 Elenco di comandi di rete a basso livello . . . 115

5.5 Elenco degli strumenti di ottimizzazione della rete . . . 116
5.6 Linee guida di base per il valore di MTU ottimale . . . 117
5.7 Elenco di strumenti per firewall . . . 118

6.1 Elenco di browser web . . . 120
6.2 Elenco di programmi di posta (MUA) . . . 122
6.3 Elenco di pacchetti base relativi ai server di trasporto della posta . . . 124
6.4 Elenco delle pagine di manuale di Postfix importanti . . . 125
6.5 Elenco dei file di configurazione correlati all'indirizzo di posta . . . 126
6.6 Elenco di operazioni base degli MTA . . . 127
6.7 Elenco dei server e delle utilità per l'accesso remoto . . . 128
6.8 Elenco dei file di configurazione per SSH . . . 129
6.9 Elenco di esempi di avvio di client SSH . . . 129
6.10 Elenco di client SSH per altre piattaforme . . . 130
6.11 Elenco di server e utilità di stampa . . . 131
6.12 Elenco di altri server applicativi di rete . . . 132
6.13 Elenco di altri client applicativi di rete . . . 133
6.14 Elenco di RFC popolari . . . 133

7.1 Lista di ambienti desktop . . . 134
7.2 Elenco di pacchetti importanti per l'infrastruttura GUI . . . 136
7.3 Elenco di applicazioni GUI degne di nota . . . 138
7.4 Elenco di tipi di carattere TrueType e OpenType degni di nota . . . 139
7.5 Elenco di ambienti per i tipi di carattere degli di nota e pacchetti correlati . . . 140
7.6 Elenco di ambienti sandbox degni di nota e pacchetti correlati . . . 141
7.7 Elenco di server per l'accesso remoto degni di nota . . . 142
7.8 Elenco di metodi di connessione al server X . . . 142
7.9 Elenco di programmi correlati con la manipolazione degli appunti a caratteri . . . 144

8.1 Elenco di pacchetti di IBus e dei suoi plugin . . . 150
8.2 Elenco di pacchetti di Fcitx5 e dei suoi plugin . . . 151

9.1 Elenco di programmi che supportano le attività in console . . . 154
9.2 Elenco di associazioni di tasti per screen . . . 156
9.3 Informazioni sull'inizializzazione di `vim` . . . 160
9.4 Elenco di analizzatori del registro di sistema . . . 161
9.5 Esempi di visualizzazione di date e orari per il comando "`ls -l`" con il **valore dello stile per gli orari** 162
9.6 Elenco di strumenti di manipolazione di immagini grafiche . . . 163
9.7 Elenco di pacchetti che possono registrare la cronologia della configurazione . . . 163
9.8 Elenco di strumenti per monitorare e controllare l'attività dei programmi. . . . 164

9.9 Elenco di valori di nice per la priorità di schedulazione . . . 165
9.10 Elenco degli stili per il comando ps . . . 165
9.11 Elenco dei segnali usati comunemente con il comando kill . . . 170
9.12 Elenco di tasti per il comando SAK degni di nota . . . 171
9.13 Elenco di strumenti per l'identificazione dell'hardware . . . 173
9.14 Elenco di strumenti di configurazione dell'hardware . . . 173
9.15 Elenco di pacchetti relativi all'audio . . . 175
9.16 Elenco di comandi per disabilitare il salvaschermo . . . 175
9.17 Elenco di dimensioni della memoria riportate . . . 176
9.18 Elenco di strumenti per verificare la sicurezza e l'integrità del sistema . . . 177
9.19 Elenco di pacchetti di gestione delle partizioni dei dischi . . . 178
9.20 Elenco di pacchetti di gestione dei file system . . . 180
9.21 Elenco di pacchetti che visualizzano e modificano dati binari . . . 189
9.22 Elenco di pacchetti per manipolare file senza montare i dischi . . . 189
9.23 Elenco di strumenti per aggiungere dati ridondanti a file . . . 189
9.24 Elenco di pacchetti per recupero di file dati ed analisi forensi. . . . 190
9.25 Elenco di utilità per la cifratura dei dati . . . 193
9.26 Elenco di pacchetti chiave da installare per la ricompilazione del kernel in un sistema Debian . . . 195
9.27 Elenco di strumenti di virtualizzazione . . . 198

10.1 Elenco di strumenti di archiviazione e compressione . . . 204
10.2 Elenco di strumenti di copia e sincronizzazione . . . 205
10.3 Elenco di possibili scelte per il file system di dispositivi di archiviazione removibili con scenari di uso tipici 210
10.4 Elenco dei servizi di rete da scegliere in base allo scenario di uso tipico . . . 211
10.5 Elenco di suite con utilità di backup . . . 213
10.6 Elenco di strumenti per l'infrastruttura di sicurezza dei dati . . . 216
10.7 Elenco di comandi per GNU Privacy Guard per la gestione delle chiavi . . . 217
10.8 Elenco dei significati dei codici di fiducia . . . 217
10.9 Elenco di comandi GNU Privacy Guard per file . . . 219
10.10 Elenco di strumenti per la fusione di codice sorgente . . . 221
10.11 Elenco di pacchetti e comandi relativi a Git . . . 223
10.12 Principali comandi Git . . . 225
10.13 Suggerimenti per Git . . . 226
10.14 Elenco di altri strumenti per sistemi di controllo delle versioni . . . 227

11.1 Elenco di strumenti di conversione di dati testuali . . . 228
11.2 Elenco dei valori delle codifiche e loro uso . . . 229
11.3 Elenco di stili per EOL per differenti piattaforme . . . 231
11.4 Elenco di comandi di conversione di TAB dai pacchetti `bsdmainutils` e `coreutils` . . . 231

11.5 Elenco di strumenti per estrarre dati in testo puro . . . 232
11.6 Elenco di strumenti per evidenziare dati in testo puro . . . 233
11.7 Elenco di entità predefinite per XML . . . 234
11.8 Elenco di strumenti XML . . . 235
11.9 Elenco di strumenti DSSSL . . . 235
11.10 Elenco di strumenti di estrazione di dati XML . . . 236
11.11 Elenco di strumenti per belle stampe XML . . . 236
11.12 Elenco di strumenti per impaginazione . . . 237
11.13 Elenco di pacchetti che aiutano a creare una pagina man . . . 238
11.14 Elenco di interpreti PostScript Ghostscript . . . 239
11.15 Elenco di utilità per dati stampabili . . . 240
11.16 Elenco di pacchetto che aiutano a convertire dati di posta . . . 241
11.17 Elenco di strumenti per dati grafici (metapacchetto) . . . 242
11.18 Elenco di strumenti per dati grafici (GUI) . . . 243
11.19 Elenco di strumenti per dati grafici (CLI) . . . 244
11.20 Elenco di strumenti di conversione di dati vari . . . 245

12.1 Elenco di bashismi tipici . . . 247
12.2 Elenco di parametri di shell . . . 248
12.3 Elenco di espansioni di parametri di shell . . . 248
12.4 Elenco di sostituzioni chiave di parametri di shell . . . 248
12.5 Elenco di operatori per paragonare file in espressioni condizionali . . . 249
12.6 Elenco di operatori per paragonare stringhe in espressioni condizionali . . . 250
12.7 Elenco di pacchetti contenenti piccoli programmi di utilità per script di shell . . . 252
12.8 Elenco di pacchetti relativi ad interpreti . . . 252
12.9 Elenco di programmi per dialoghi . . . 253
12.10 Elenco di pacchetti relativi al compilatore . . . 255
12.11 Elenco di generatori di parser LALR compatibili con Yacc . . . 256
12.12 Elenco di strumenti per l’analisi statica del codice . . . 259
12.13 Elenco di pacchetti relativi al debug . . . 259
12.14 Elenco di comandi gdb avanzati . . . 262
12.15 Elenco di strumenti per rilevazione di memory leak . . . 263
12.16 Elenco di pacchetti relativi alla compilazione . . . 263
12.17 Elenco di variabili automatiche di make . . . 264
12.18 Elenco di espansioni delle variabili di make . . . 264
12.19 Elenco di strumenti per la traduzione di codice sorgente . . . 266

Sommario

Questo testo è libero; lo si può ridistribuire e/o modificare nei termini della GNU General Public License in qualsiasi versione aderente alle Linee guida Debian per il Software Libero (DFSG).

Prefazione

Questa guida di riferimento Debian Reference (versione 2.152) (2026-10-10 14:56:25 UTC) è pensata per fornire un'ampia panoramica dell'amministrazione di un sistema Debian in qualità di guida post-installazione per l'utente.

Si rivolge ad un lettore che abbia voglia di imparare gli script di shell, ma che non è pronto a leggere tutti i sorgenti C per scoprire come funzioni il sistema GNU/Linux.

Per istruzioni sull'installazione, vedere:

- Guida all'installazione di Debian GNU/Linux per l'attuale sistema stable
- Guida all'installazione di Debian GNU/Linux per l'attuale sistema testing

Note legali

Non viene fornita alcuna garanzia. Tutti i marchi registrati appartengono ai rispettivi proprietari.

Il sistema Debian è, di per sé, un'entità in continuo movimento; ciò rende difficile alla sua documentazione essere aggiornata e corretta. Sebbene come base per questo testo sia stata usata la versione `testing` al momento della sua stesura, alcuni contenuti potrebbero essere obsoleti al momento della sua lettura.

Si consideri questo documento come un riferimento secondario. Esso non sostituisce nessuna guida autorevole. L'autore e i collaboratori non si assumono responsabilità per le conseguenze di errori, omissioni o ambiguità presenti in questo documento.

Cosa è Debian

Il Progetto Debian è un'associazione di individui che hanno fatto causa comune per creare un sistema operativo libero. La sua distribuzione è caratterizzata dai seguenti aspetti.

- Impegno per la libertà del software: Il Contratto sociale Debian e le Linee Guida Debian per il Software Libero (DFSG).
- Sforzo volontario non pagato, distribuito, basato su Internet https://www.debian.org.
- Vasto numero di pacchetti software di alta qualità pre-compilati.
- Attenzione alla stabilità e alla sicurezza, con facile accesso agli aggiornamenti di sicurezza.
- Attenzione ad aggiornamenti senza problemi ai pacchetti più recenti dei software negli archivi `testing`.
- Vasto numero di architetture hardware supportate.

I vari pezzi di Software Libero in Debian provengono da GNU, Linux, BSD, X, ISC, Apache, Ghostscript, Common Unix Printing System , Samba, GNOME, KDE, Mozilla, LibreOffice, Vim, TeX, LaTeX, DocBook, Perl, Python, Tcl, Java, Ruby, PHP, Berkeley DB, MariaDB, PostgreSQL, SQLite, Exim, Postfix, Mutt, FreeBSD, OpenBSD, Plan 9 e molti altri progetti software liberi indipendenti. Debian integra questa diversità del Software Libero in un unico sistema.

Informazioni su questo documento

Linee guida

Nella stesura di questo documento sono state seguite le seguenti linee guida.

- Fornire una panoramica ed evitare casi particolari. (**Vista d'insieme**.)
- Mantenere il testo **conciso e semplice**.
- Non reinventare la ruota. (Usare rimandi alle **risorse esistenti**.)
- Concentrarsi sugli strumenti senza interfaccia utente grafica e sulla console. (Usare **esempi shell**.)
- Essere obiettivi. (Usare popcon, ecc.)

Suggerimento
Si è cercato di spiegare aspetti gerarchici e i più bassi livelli di funzionamento del sistema.

Prerequisiti



avvertimento
Il lettore deve fare lo sforzo di cercare risposte da solo andando oltre questa documentazione. Questo documento fornisce solo degli efficaci punti di partenza.

Il lettore deve cercare le soluzioni da solo dalle fonti primarie.

- Il sito Debian all'indirizzo https://www.debian.org per le informazioni generiche.
- La documentazione nella directory "`/usr/share/doc/`*`nome_pacchetto`*".
- La **pagina man** in stile Unix: "`dpkg -L` *`nome_pacchetto`* `|grep '/man/man.*/'`".
- La **pagina info** in stile GNU: "`dpkg -L` *`nome_pacchetto`* `|grep '/info/'`".
- Le segnalazioni di bug: https://bugs.debian.org/*nome_pacchetto* .
- Il Debian Wiki all'indirizzo https://wiki.debian.org/ per gli aspetti mutevoli e gli argomenti specifici.
- Le specifiche Single UNIX Specification dalla Pagina web di The UNIX System dell'Open Group.
- L'enciclopedia libera Wikipedia all'indirizzo https://www.wikipedia.org/ .
- The Debian Administrator's Handbook
- Gli HOWTO dal The Linux Documentation Project (TLDP)

Nota
Per una documentazione dettagliata può essere necessario installare il corrispondente pacchetto di documentazione il cui nome ha il suffisso "`-doc`" .

Convenzioni

Questo documento fornisce informazioni attraverso il seguente stile semplificato di rappresentazione con esempi di comandi per la shell bash(1).

```
# command-in-root-account
$ command-in-user-account
```

Questi prompt di shell distinguono gli account usati e corrispondono a ciò che si ottiene impostando le variabili d'ambiente in questo modo: "`PS1='\$'`" e "`PS2=' '`". Questi valori sono stati scelti per aumentare la leggibilità del documento e non sono quelli tipici di una reale installazione del sistema.

Tutti gli esempi di comandi sono nella localizzazione inglese "`LANG=en_US.UTF8`". Non aspettarsi che negli esempi siano tradotte le stringhe segnaposto come *`command-in-root-account`* e *`command-in-user-account`*. Questa è una scelta intenzionale per mantenere tutti gli esempi tradotti aggiornati.

Nota
Si veda il significato delle variabili d'ambiente "`$PS1`" e "`$PS2`" in bash(1).

Le **azioni** che gli amministratori di sistema devono eseguire sono scritte all'infinito, ad esempio "Premere il tasto Invio dopo ogni stringa di comando digitata nella shell."

La colonna **descrizione** ed altre simili in tabelle possono contenere un **sintagma nominale** che segue le convenzioni per le descrizioni brevi dei pacchetti e che perde l'articolo iniziale quali "un" ed "il". Le colonne possono, in alternativa, contenere un **verbo al presente** seguendo la convenzione usata nella descrizione breve dei comandi nelle pagine man. Questa scelta potrà apparire strana per qualcuno, ma fa parte della scelta deliberata dell'autore di mantenere più semplice possibile lo stile di questa documentazione. In entrambi i casi l'iniziale non è maiuscola e manca il punto finale seguendo le dette convenzioni per le descrizioni brevi.

Nota
I nomi propri, inclusi i nomi dei comandi mantengono la corretta iniziale maiuscola/minuscola indipendentemente dalla loro posizione.

Un **breve comando** citato all'interno di un paragrafo di testo viene reso dal carattere a spaziatura fissa racchiuso tra virgolette doppie come in "`aptitude safe-upgrade`".

Dati testuali da un file di configurazione citati all'interno di un paragrafo di testo vengono resi dal carattere a spaziatura fissa racchiuso tra virgolette doppie come in "`deb-src`".

Un **comando** viene reso dall'uso del carattere a spaziatura fissa per il suo nome, seguito eventualmente dal numero di sezione della sua pagina man tra parentesi, come in bash(1). Ci si senta incoraggiati ad ottenere ulteriori informazioni digitando quanto segue.

```
$ man 1 bash
```

Una **pagina man** viene resa dall'uso del carattere a spaziatura fissa per il suo nome seguito dal numero della sua sezione nelle pagine man racchiuso tra parentesi, come in sources.list(5). Ci si senta incoraggiati ad ottenere ulteriori informazioni digitando quanto segue.

```
$ man 5 sources.list
```

Una **pagina info** viene resa dal suo comando corrispondente in carattere a spaziatura fissa racchiuso tra virgolette doppie, come in "`info make`". Ci si senta incoraggiati ad ottenere ulteriori informazioni digitando quanto segue.

```
$ info make
```

Un **nome di file** viene reso dal carattere a spaziatura fissa tra virgolette doppie, come in "`/etc/passwd`". Per quanto riguarda i file di configurazione ci si senta incoraggiati ad ottenere ulteriori informazioni digitando quanto segue.

```
$ sensible-pager "/etc/passwd"
```

Un **nome di directory** viene reso dal carattere a spaziatura fissa tra virgolette doppie, come in «`/etc/apt/`». Ci si senta incoraggiati ad esplorare il suo contenuto digitando quanto segue.

```
$ mc "/etc/apt/"
```

Un **nome di pacchetto** viene reso dal carattere a spaziatura fissa, come in `vim`. Ci si senta incoraggiati ad ottenere ulteriori informazioni digitando quanto segue.

```
$ dpkg -L vim
$ apt-cache show vim
$ aptitude show vim
```

La posizione di una **documentazione** può essere indicata con il nome di file in carattere a spaziatura fissa tra virgolette doppie, come in "`/usr/share/doc/base-passwd/users-and-groups.txt.gz`" e "`/usr/share/doc/base-passw` oppure dal suo URL, come in https://www.debian.org. Ci si senta incoraggiati a leggere la documentazione digitando quanto segue.

```
$ zcat "/usr/share/doc/base-passwd/users-and-groups.txt.gz" | sensible-pager
$ sensible-browser "/usr/share/doc/base-passwd/users-and-groups.html"
$ sensible-browser "https://www.debian.org"
```

Una **variabile d'ambiente** viene resa dal suo nome preceduto dal segno "$" in carattere a spaziatura fissa tra virgolette doppie, come in "`$TERM`". Ci si senta incoraggiati ad ottenere il suo valore attuale digitando quanto segue.

```
$ echo "$TERM"
```

Popcon

I dati da popcon sono usati come misura oggettiva della popolarità di ciascun pacchetto. Sono stati scaricati il 2026-10-07 05:10:21 UTC e contengono i dati totali inviati da 291550 rapporti su 224298 pacchetti binari e 25 architetture.

Nota
Notare che l'archivio `amd64 unstable` contiene attualmente solo 78072 pacchetti. I dati di popcon contengono rapporti inviati da molte installazioni su sistemi vecchi.

Il numero popcon, preceduto da "V:" che sta per "voti", è calcolato così: "1000 * (rapporti popcon inviati con il pacchetto eseguito di recente sul PC)/(numero totale di rapporti popcon inviati)".

Il numero popcon, preceduto da "I:" che sta per "installazioni", è calcolato così: "1000 * (rapporti popcon inviati con il pacchetto installato sul PC)/(numero totale di rapporti popcon inviati)".

Nota
I numeri popcon non dovrebbero essere considerati una misura assoluta dell'importanza dei pacchetti. Ci sono molti fattori che possono influenzare le statistiche. Per esempio, alcuni sistemi che partecipano a popcon possono avere montato directory quali "`/usr/bin`" con l'opzione "`noatime`" per migliorare le prestazioni di sistema, influenzando di fatto negativamente i "voti" per quel sistema.

La dimensione dei pacchetti

Anche i dati sulla dimensione dei pacchetti sono usati come misura oggettiva per ciascun pacchetto. Sono basati sulla "`Installed-Size:`" riportata dal comando "`apt-cache show`" o "`aptitude show`" (attualmente sull'architettura `amd64` per la versione `unstable`). La dimensione è riportata in KiB (Kibibyte = unità di 1024 byte).

Nota
Una piccola dimensione per un pacchetto può indicare che esso è, nella versione `unstable` un pacchetto fittizio che installa, attraverso le dipendenze, altri pacchetti con contenuti sostanziosi. Il pacchetto fittizio permette una transizione facile o una suddivisione di pacchetto.

Nota
Una dimensione di pacchetto seguita da "(*)" indica che il pacchetto non è presente nella versione `unstable` ed è stata quindi usata la dimensione del pacchetto in `experimental` al suo posto.

Segnalazioni di bug in questo documento

Se si trovano errori in questo documento segnalare i bug nel pacchetto `debian-reference` usando reportbug(1). Se possibile, includere i suggerimenti di correzione come "`diff -u` alla versione in puro testo o ai sorgenti.

Alcuni promemoria per i nuovi utenti

Alcuni promemoria per i nuovi utenti:

- Fare il backup dei propri dati
  - Vedere Sezione 10.2.
- Mantenere sicure le proprie password e chiavi di sicurezza
- KISS (keep it simple stupid), cioè mantenere le cose semplici
  - Non iper-progettare il proprio sistema
- Leggere i propri file di log
  - Il **PRIMO** errore è quello che conta
- RTFM (read the fine manual), cioè leggere il manuale
- Cercare in Internet prima di fare domande
- Non lavorare come root quando non è necessario
- Non manomettere il sistema di gestione dei pacchetti
- Non digitare nulla di cui non si capisce il significato
- Non cambiare i permessi dei file (prima di una completa revisione della sicurezza)
- Non lasciare la shell di root fino a che non si sono **TESTATE** le proprie modifiche
- Avere sempre un supporto di avvio alternativo (unità flash USB, CD-ROM, …)

Alcune citazioni per i nuovi utenti

Ecco alcune interessanti citazioni dalla mailing list di Debian che potrebbero illuminare i nuovi utenti.

- ”Questo è Unix. Ti dà abbastanza corda da impiccarti.” --- Miquel van Smoorenburg <miquels at cistron.nl>
- ”Unix È amichevole... è solo molto selettivo riguardo i suoi amici.” --- Tollef Fog Heen <tollef at add.no>

Wikipedia ha un articolo (in inglese) sulla ”Filosofia Unix” che elenca citazioni interessanti.

Capitolo 1

Tutorial GNU/Linux

Penso che imparare un sistema informatico sia come imparare una lingua straniera. Anche se le guide e la documentazione sono utili, si deve fare pratica diretta. Per aiutare il lettore a iniziare dolcemente, ho elaborato alcuni punti base.

Il potente design di Debian GNU/Linux deriva dal sistema operativo Unix, cioè un sistema operativo multiutente e multitasking. Bisogna imparare a sfruttare la potenza di queste caratteristiche e le somiglianze tra Unix e GNU/Linux.

Non bisogna scappare dai testi pensati per Unix e affidarsi solamente a testi su GNU/Linux dato che in questo modo ci si priva di molte informazioni utili.

Nota
Se si è usato tramite strumenti a riga di comando un sistema *nix per un certo tempo, probabilmente si sa già tutto ciò che viene spiegato qui. Si usi questa sezione come un ripasso e per consultazioni.

1.1 Nozioni di base sulla console

1.1.1 Il prompt di shell

All'avvio del sistema, se non è stata installato alcun ambiente con GUI come il sistema desktop GNOME o KDE. Supponiamo che il proprio nome host sia `pippo`, il prompt di login apparirà come segue.

Se è stato installato un ambiente GUI, allora ci si può comunque spostare ad un prompt di login a caratteri premendo Ctrl-Alt-F3 e si può ritornare alla GUI con Ctrl-Alt-F2 (vedere Sezione 1.1.6 in seguito per maggiori informazioni).

```
foo login:
```

Al prompt di login digitare il proprio nome utente, ad esempio `pinguino` e premere il tasto Invio, poi digitare la propria password e premere di nuovo Invio.

Nota
Nella tradizione Unix, il nome utente e la password di un sistema Debian distinguono le lettere maiuscole dalle minuscole. Il nome utente di solito viene scelto usando solo lettere minuscole. Il primo account utente viene normalmente creato durante l'installazione. Account utente aggiuntivi possono essere creati da root con adduser(8).

Il sistema si presenta con il messaggio di benvenuto memorizzato in "`/etc/motd`" (Message Of The Day, messaggio del giorno) e fornisce un prompt dei comandi.

```
Debian GNU/Linux 12 foo tty3

foo login: penguin
Password:

Linux foo 6.5.0-0.deb12.4-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.10-1~bpo12+1 (2023-11-23) ↩
     x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Wed Dec 20 09:39:00 JST 2023 on tty3
foo:~$
```

Si è ora nella shell. La shell interpreta i comandi dell'utente.

1.1.2 Il prompt di shell in una GUI

Se è stato installato un ambiente con GUI durante l'installazione, all'avvio del sistema viene presentata una schermata grafica di login. Si inseriscono il nome utente e la password per fare il login come utente non privilegiato. Usare il tasto Tab per spostarsi dal campo del nome utente a quello della password e viceversa, oppure usare il mouse e il clic principale.

Si può ottenere il prompt di shell in un ambiente GUI avviando un programma `emulatore di terminale X` come gnome-terminal(1), rxvt(1) o xterm(1). Nell'ambiente desktop GNOME può essere fatto premendo il tasto SUPER (tasto Windows) e digitando "terminal" per cercare il prompt.

In altri sistemi Desktop (come `fluxbox`), potrebbe non esserci un punto evidente per l'apertura del menu. Se si è in questa condizione, provare a fare clic, con il tasto destro, sullo sfondo della schermata del desktop e sperare che appaia un menu.

1.1.3 L'account root

L'account root viene anche indicato come superutente o utente privilegiato. Da questo account si possono eseguire i seguenti compiti amministrativi.

- Leggere, scrivere e cancellare qualsiasi file sul sistema indipendentemente dai suoi permessi.
- Impostare il proprietario e i permessi di qualunque file sul sistema.
- Impostare la password di qualsiasi utente non privilegiato nel sistema.
- Fare il login in qualsiasi account senza la password.

Questi poteri illimitati dell'account root rendono necessario essere prudenti e responsabili nel loro uso.



avvertimento
Non comunicare mai la password di root ad altri.

Nota
I permessi di un file (inclusi i device hardware come CD-ROM, ecc. che sono nient'altro che un altro file per il sistema Debian) lo possono rendere inutilizzabile o inaccessibile per gli utenti non root. Benché usare l'account root sia un metodo veloce per affrontare questo tipo di situazione, la sua soluzione dovrebbe essere l'impostazione degli appropriati permessi e gruppi proprietari per il file (vedere Sezione 1.2.3).

1.1.4 Il prompt di shell di root

Ecco alcuni metodi di base per ottenere il prompt di shell per root usando la password di root.

- Inserire `root` al prompt di login a caratteri.
- Digitare "`su -l`" al prompt di shell di un utente qualsiasi.
  - L'ambiente dell'utente attuale non viene in questo caso preservato.
- Digitare "`su`" al prompt di shell di un utente qualsiasi.
  - L'ambiente dell'utente attuale viene in questo caso parzialmente preservato.

1.1.5 Strumenti di amministrazione del sistema con interfaccia utente grafica

Quando il menu del desktop non avvia gli strumenti con interfaccia grafica per l'amministrazione di sistema con i privilegi appropriati, si può avviarli dal prompt di shell di root degli emulatori di terminale, quali gnome-terminal(1), rxvt(1) o xterm(1). Vedere Sezione 1.1.4 e Sezione 7.9.



avvertimento
Non avviare mai il gestore di sessioni/display manager della GUI dall'account root inserendo `root` al prompt di un display manager come gdm3(1).
Non eseguire mai in X Window programmi con interfaccia utente grafica non fidati da remoto quando sono visualizzate informazioni critiche, dato che potrebbero spiare lo schermo X.

1.1.6 Console virtuali

Nel sistema Debian standard, ci sono sei console a caratteri in stile VT100 disponibili tra cui ci si può spostare per avviare una shell di comando direttamente sull'host Linux. A meno che non si sia in un ambiente con interfaccia grafica, si può passare da una console virtuale all'altra usando simultaneamente il `tasto_Alt_sinistro` e uno dei tasti `F1` — `F6`. Ogni console a caratteri permette un login indipendente nell'account e offre un ambiente multiutente. L'ambiente multiutente è una bellissima caratteristica di Unix e ci si può abituare presto a dipendere da esso.

Se si è nell'ambiente con GUI, si accede alla console a caratteri 3 premendo i tasti `Ctrl-Alt-F3`, cioè premendo contemporaneamente il `tasto_Ctrl_sinistro`, il `tasto_Alt_sinistro` e il `tasto_F3`. Si può tornare all'ambiente GUI, che di solito è in esecuzione sulla console virtuale 2, premendo `Alt-F2`.

In alternativa ci si può spostare in un'altra console virtuale, per esempio la console 3, dalla riga di comando.

```
# chvt 3
```

1.1.7 Come uscire dal prompt dei comandi

Si digita `Ctrl-D`, cioè il `tasto_Ctrl_sinistro` e il `tasto_d` vengono premuti contemporaneamente al prompt dei comandi per chiuderà l'attività della shell. Se si è nella console a caratteri, in questo modo si ritorna al prompt di login. Anche se questi caratteri di controllo vengono indicati con le lettere maiuscole, come "control D", non è necessario premere il tasto Maiusc. Per indicare `Ctrl-D` viene anche usata l'espressione abbreviata `^D`. In alternativa si può digitare "exit".

Se si sta usando x-terminal-emulator(1) , in questo modo si può chiudere la finestra dell'emulatore di terminale (`x-terminal-emulator`).

1.1.8 Come spegnere il sistema

Esattamente come ogni altro SO moderno in cui le operazioni su file comportano una cache dei dati in memoria per migliorare le prestazioni, il sistema Debian ha bisogno della appropriata procedura di spegnimento prima che l'alimentazione possa essere staccata in modo sicuro. Questo serve a preservare l'integrità dei file, forzando la scrittura su disco di tutti i cambiamenti avvenuti in memoria. Se è disponibile il controllo software dell'alimentazione, la procedura di spegnimento automaticamente toglie l'alimentazione al sistema. (In caso contrario può essere necessario tener premuto per alcuni secondi il tasto di accensione/spegnimento.)

Si può spegnere il sistema dalla riga di comando nella normale modalità multiutente.

```
# shutdown -h now
```

Si può spegnere il sistema dalla riga di comando nella modalità single-user.

```
# poweroff -i -f
```

Vedere Sezione 6.3.8.

1.1.9 Ripristinare una console funzionante

Quando, dopo aver fatto qualcosa di strano come "`cat` *`un-qualche-file-binario`*", lo schermo impazzisce digitare "`reset`" al prompt dei comandi. Mentre lo si digita il comando potrebbe non essere visualizzato. Si può anche usare "`clear`" per pulire lo schermo.

1.1.10 Suggerimenti per pacchetti aggiuntivi per i principianti

Sebbene anche l'installazione minima del sistema Debian, senza alcun ambiente desktop, fornisca le funzionalità Unix di base è una buona idea installare con apt-get(8) alcuni pacchetti aggiuntivi per la riga di comando e i terminali a caratteri basati su curses, come `mc` e `vim` con cui i principianti possono fare i primi passi. Eseguire i comandi seguenti.

```
# apt-get update
 ...
# apt-get install mc vim sudo aptitude
 ...
```

Se questi pacchetti sono già installati, nessun nuovo pacchetto sarà installato.

Potrebbe essere una buona idea leggere un po' di documentazione.

Si possono installare alcuni di questi pacchetti usando i comandi seguenti.

```
# apt-get install package_name
```

pacchetto	popcon	dimensione	descrizione
mc	V:40, I:176	1590	Gestore di file testuale a tutto schermo
sudo	V:761, I:869	6814	Programma per garantire privilegi di root limitati agli utenti
vim	V:83, I:338	4164	Editor di testi Unix Vi IMproved, un editor di testi per programmatori (versione standard)
vim-tiny	V:59, I:980	1867	Editor di testi Unix Vi IMproved, un editor di testi per programmatori (versione compatta)
emacs-nox	V:3, I:12	46564	Emacs dal progetto GNU, l'editor di testi estensibile basato su Lisp
w3m	V:10, I:129	2853	Browser Web testuali
gpm	V:8.3, I:9.0	524	Taglia e Incolla in stile Unix nella console testuale (demone)

Tabella 1.1: Elenco di pacchetti con interessanti programmi in modalità testuale

pacchetto	popcon	dimensione	descrizione
doc-debian	I:886	185	Documentazione del Progetto Debian, (FAQ Debian) ed altri documenti
debian-policy	I:6.8	5147	Manuale Debian Policy e documenti correlati
developers-reference	V:0.2, I:2.2	2657	Linee guida ed informazioni per gli sviluppatori Debian
debmake-doc	I:0.33	11803	Guida Debian per il Manutentore
debian-history	I:0.50	6513	Storia del Progetto Debian
debian-faq	I:885	791	FAQ Debian

Tabella 1.2: Elenco di pacchetti con documentazione interessante

1.1.11 Un account utente extra

Se non si vuole usare il proprio account utente principale per le esercitazioni descritte in seguito, si può creare un account utente per esercitarsi, ad esempio `pesce` con il comando seguente.

```
# adduser fish
```

Rispondere a tutte le domande.

In questo modo si crea un nuovo account chiamato `pesce`. Dopo aver fatto pratica si può rimuover questo account e la sua directory home digitando quanto segue.

```
# deluser --remove-home fish
```

Nei sistemi non Debian e Debian specializzati, le attvità descritte sopra devono usare invece le utilità a basso livello useradd(8) e userdel(8).

1.1.12 Configurazione di sudo

Per la tipica postazione di lavoro di un unico utente, come il sistema desktop Debian su un PC portatile, è frequente l'uso di una semplice configurazione di sudo(8), come qui di seguito descritto, per permettere all'utente non privilegiato, ad esempio `pinguino`, di ottenere privilegi di amministratore con la sola propria password, senza quella dell'utente root.

```
# echo "penguin  ALL=(ALL) ALL" >> /etc/sudoers
```

In alternativa è uso comune usare la configurazione seguente per permettere all'utente non privilegiato, ad esempio `pinguino`, di ottenere privilegi di amministratore senza alcuna password.

```
# echo "penguin  ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers
```

Questo trucco dovrebbe essere usato esclusivamente per le postazioni di lavoro con un solo utente, in cui l'utente è anche l'amministratore.



avvertimento
Non impostare account per utenti regolari su postazioni di lavoro multi-utente in questo modo perché ciò sarebbe un grosso problema per la sicurezza del sistema.



Attenzione
Nell'esempio precedente la password e l'account di `pinguino` richiedono altrettanta protezione della password e account dell'utente root.
I privilegi di amministrazione in questo contesto sono forniti a qualcuno che è autorizzato ad eseguire i compiti di amministrazione del sistema sulla macchina. Non dare mai tali privilegi ad un manager del dipartimento di Amministrazione della propria azienda o al proprio capo, a meno che non siano autorizzati e capaci.

Nota
Per fornire privilegi di accesso a specifici device e file, si dovrebbe considerare l'uso dei **gruppi** per dare un accesso limitato, invece di usare i privilegi di `root` attraverso sudo(8).
Con una configurazione più attenta e precisa, sudo(8) può garantire privilegi amministrativi limitati ad altri utenti in un sistema condiviso, senza rendere nota la password di root. Questo può aiutare la tracciabilità su host con più di un amministratore, in modo che si possa dire chi ha fatto cosa. D'altra parte si potrebbe anche volere che nessuno abbia tali privilegi.

1.1.13 Tempo di giocare

Si è ora pronti a giocare con il sistema Debian senza rischi fintanto che si usa l'account utente non privilegiato.

Ciò è possibile perché il sistema Debian è, anche subito dopo l'installazione predefinita, configurato con i permessi dei file corretti che impediscono agli utenti non privilegiati di danneggiare il sistema. Ci possono essere naturalmente ancora dei punti deboli che possono essere sfruttati, ma chi si preoccupa di tali cose non dovrebbe leggere questa sezione ma piuttosto il manuale Securing Debian.

Si può imparare il sistema Debian, come sistema *nix nelle sezioni seguenti.

- Sezione 1.2 (concetti di base)
- Sezione 1.3 (sopravvivere)
- Sezione 1.4 (lavoro di base)
- Sezione 1.5 (uso della shell)
- Sezione 1.6 (elaborazione di testi)

1.2 File system stile Unix

In GNU/Linux ed altri sistemi operativi *nix, i file sono organizzati in directory. Tutti i file e le directory sono organizzati in un unico grande albero che ha la sua radice in "`/`". Viene chiamato albero perché il file system, se viene disegnato, ha l'aspetto di un albero messo però sottosopra.

Questi file e directory possono essere sparsi su diversi dispositivi. mount(8) serve per attaccare il file system trovato su un qualche dispositivo al grande albero dei file. Al contrario, umount(8) lo stacca. Nei kernel Linux recenti mount(8)

con alcune opzioni può collegare parte di un albero dei file in qualche altra posizione o può montare file system come condivisi, privati, slave o non-collegabili. Le opzioni di montaggio supportate per ciascun file system sono disponibili in "`/usr/share/doc/linux-doc-*/Documentation/filesystems/`".

Nei sistemi Unix vengono chiamate **directory** quelle che in altri sistemi sono chiamate **cartelle**. Notare anche che non esiste in nessun sistema Unix un concetto di **unità**, come "`A:`". C'è un solo unico file system e tutto vi è incluso. Questo è un grandissimo vantaggio rispetto a Windows.

1.2.1 Nozioni di base sui file Unix

Ecco alcune nozioni di base sui file Unix.

- Nei nomi dei file si distingue tra **maiuscole e minuscole**. I file "`MIOFILE`" e "`MioFile`" sono cioè file diversi.
- La **directory root** indica la radice del file system, indicata semplicemente con "`/`". Non la si confonda con la directory home dell'utente root: "`/root`".
- Ogni directory ha un nome che può contenere qualsiasi lettera o simbolo **tranne"`/`"**. La directory radice è un'eccezione; il suo nome è "`/`" (pronunciato "slash" o "directory root/radice") e il suo nome non può essere cambiato.
- Ogni file o directory è indicato da un **nome di file pienamente qualificato**, un **nome file assoluto** o un **percorso**, fornendo la sequenza di directory attraverso le quali si deve passare per raggiungerlo. I tre termini sono sinonimi.
- Tutti i **nomi di file pienamente qualificati** iniziano con la directory "`/`" e c'è un carattere "`/`" tra ciascuna directory o file nel nome del file. Il primo carattere "`/`" è la directory di più alto livello e gli altri "`/`" separano le directory successive fino a che non si raggiunge l'ultima voce che è il nome del file stesso. I termini usati possono creare confusione. Prendere in considerazione come esempio il seguente **nome file pienamente qualificato** come esempio: "`/usr/share/keytables/us.map.gz`"; anche se ci si riferisce al solo nome base, "`us.map.gz`" come al suo nome file.
- La directory radice ha un certo numero di rami, come "`/etc/`" e "`/usr/`". Queste sottodirectory a loro volta si diramano in ulteriori sottodirectory, come "`/etc/systemd/`" e "`/usr/local/`". Viste tutte insieme vengono chiamate **albero delle directory**. Si può pensare ad un nome file assoluto come ad un percorso dalla base dell'albero ("`/`") alla punta di un ramo (un file). Si può anche sentir parlare dell'albero delle directory come se fosse un albero **genealogico** che comprende tutti i discendenti di un'unica figura detta directory radice ("`/`): le sottodirectory hanno perciò dei **genitori** e un percorso mostra gli antenati completi di un file. Ci sono anche percorsi relativi che iniziano da una qualche posizione che non sia la directory radice. Si dovrebbe tenere a mente che la directory "`../`" si riferisce alla directory genitore. Questa terminologia si applica anche ad altre strutture simili a quella delle directory, come le strutture di dati gerarchici.
- Non c'è alcun nome speciale di percorso di directory che corrisponda ad un dispositivo fisico, come il disco fisso. Questa è una differenza rispetto a RT-11, CP/M, OpenVMS, MS-DOS, AmigaOS e Microsoft Windows, in cui il percorso contiene una porzione con il nome di dispositivo come "`C:\`". (Esistono tuttavia directory che si riferiscono ai dispositivi fisici come a parte del file system normale. Vedere Sezione 1.2.2.)

Nota
Benché si **possa** usare quasi qualsiasi lettera o simbolo nel nome di un file, in pratica farlo non è una bella idea. È meglio evitare qualsiasi carattere che ha spesso un significato particolare sulla riga di comando, inclusi spazi, tabulazioni, a capo e altri caratteri speciali: `{ } ( ) [ ] ' ` " \ / > < | ; ! # & ^ * % @ $`. Se si vuole separare delle parole all'interno di un nome file, buone alternative sono un punto, un trattino ed il segno di sottolineatura. Si può anche scrivere in maiuscolo ogni parola, "`InQuestoModo`". Gli utenti Linux esperti tendono ad evitare l'uso degli spazi nei nomi dei file.

Nota
La parola "root" può significare l'"utente root" o la "directory root". Il contesto in cui il termine viene usato dovrebbe rendere chiaro il suo significato.

Nota
La parola **percorso** non è usata solamente per un **nome di file pienamente qualificato** come descritto in precedenza, ma anche per il **percorso di ricerca dei comandi**. Il significato è di solito reso chiaro dal contesto.

Le linee guida raccomandate per la gerarchia dei file sono descritte in dettaglio nel Filesystem Hierarchy Standard ("`/usr/share/doc/debian-policy/fhs/fhs-2.3.txt.gz`" e in hier(7)). Come inizio si dovrebbe tenere a mente quanto segue.

directory	uso della directory
`/`	la directory root
`/etc/`	file di configurazione a livello di sistema
`/var/log/`	file di registro del sistema
`/home/`	tutte le directory home degli utenti non privilegiati

Tabella 1.3: Elenco degli usi delle directory principali

1.2.2 Aspetti tecnici del file system

Nella scia della **tradizione Unix**, il sistema Debian GNU/Linux fornisce il file system in cui risiedono i dati fisici nel disco fisso e negli altri dispositivi di memorizzazione, ed inoltre le interazioni con i dispositivi hardware come gli schermi delle console e le console seriali remote vengono rappresentate in un modo unificato in "`/dev/`".

Ogni file, directory, pipe con nome (un modo per due programmi di scambiare dati) o dispositivo fisico presente in un sistema Debian GNU/Linux ha una struttura di dati chiamata inode che descrive gli attributi ad esso associati come l'utente che lo possiede (proprietario), il gruppo a cui appartiene, la data dell'ultimo accesso ad esso, ecc. L'idea di rappresentare praticamente tutto nel file system è stata un'innovazione di Unix e i kernel Linux moderni hanno sviluppato questa idea e sono andati oltre. Ora anche le informazioni sui processi in esecuzione sul computer possono essere trovate nel file system.

La rappresentazione astratta e unificata di entità fisiche e di processi interni è molto potente dato che permette di usare lo stesso comando per lo stesso tipo di operazione su molti tipi di device completamente diversi l'uno dall'altro. È anche possibile cambiare il modo in cui il kernel funziona scrivendo dati in file speciali che sono collegati ai processi in esecuzione.

Suggerimento
Per identificare la corrispondenza tra l'albero dei file e le entità fisiche, eseguire mount(8) senza opzioni.

1.2.3 Permessi del file system

I permessi del file system di sistemi *nix sono definiti e influenzano tre categorie di utenti.

- L'**utente** che è il proprietario del file (**u**).
- Gli altri utenti del **gruppo** a cui appartiene il file (**g**).
- Tutti gli **altri**utenti (**o**) a cui ci si riferisce anche con i termini "mondo" e "tutti".

Per un file, a ciascun permesso corrispondono le seguenti azioni.

- Il permesso di **lettura** (**r**) permette al proprietario di esaminare il contenuto del file.
- Il permesso di **scrittura** (**w**) permette al proprietario di modificare il file.

- Il permesso di **esecuzione** (**x**) permette al proprietario di eseguire il file come comando.

Per una directory, a ciascun permesso corrispondono le seguenti azioni.

- Il permesso di **lettura** (**r**) permette al proprietario di elencare il contenuto della directory.
- Il permesso di **scrittura** (**w**) permette al proprietario di aggiungere o rimuovere file dalla directory.
- Il permesso di **esecuzione** (**x**) permette al proprietario di accedere ai file nella directory.

In questo caso il permesso di **esecuzione** su una directory non solo significa poter leggere i file in quella directory ma anche poterne vedere gli attributi, come la dimensione e la data di modifica.

Per visualizzare le informazioni sui permessi (ed altro) di file e directory si usa ls(1). Quando chiamato con l'opzione "`-l`" mostra, nell'ordine elencato in seguito, le seguenti informazioni.

- **Tipo di file** (primo carattere).
- **Permessi** di accesso al file (nove caratteri, tre ciascuno per utente, gruppo e altri, in questo ordine).
- **Numero di collegamenti fisici** al file.
- Nome dell'**utente** proprietario del file.
- Nome del **gruppo** a cui il file appartiene.
- **Dimensione** in caratteri (byte) del file.
- **Data ed ora** del file (mtime).
- **Nome** del file.

carattere	**significato**
`-`	file normale
d	directory
`l`	collegamento simbolico
c	device a caratteri
b	device a blocchi
p	pipe con nome
s	socket

Tabella 1.4: Elenco dei valori possibili per il primo carattere nell'output di "`ls -l`"

Per cambiare il proprietario di un file da un account root si usa chown(1). Per cambiare il gruppo di un file dall'account del proprietario o da quello di root si usa chgrp(1). Per cambiare i permessi di accesso di file o directory dall'account del proprietario o da quello di root si usa chmod(1). La sintassi di base per manipolare un file `pippo` è la seguente.

```
# chown newowner foo
# chgrp newgroup foo
# chmod  [ugoa][+-=][rwxXst][,...] foo
```

Per esempio si può fare in modo che l'utente `pippo` sia il proprietario di un albero di directory condivisa dal gruppo `pluto` con i seguenti comandi.

```
# cd /some/location/
# chown -R foo:bar .
# chmod -R ug+rwX,o=rX .
```

Ci sono altri tre bit speciali di permessi.

- Il bit **set user ID** (**s** o **S** al posto del permesso **x** dell'utente).
- Il bit **set group ID** (**s** o **S** al posto del permesso **x** del gruppo).
- Il bit **sticky** (**t** o **T** al posto del permesso **x** degli altri).

Nell'output di "`ls -l`" il valore di questi bit è **maiuscolo** se il bit di permesso di esecuzione nascosto da essi è **non impostato**.

L'impostazione del bit **set user ID** per un file eseguibile permette ad un utente di eseguire quel file con l'ID del proprietario del file (per esempio **root**). In modo simile, l'impostazione del bit **set group ID** per un file eseguibile permette ad un utente di eseguire il file con l'ID del gruppo a cui appartiene il file (per esempio **root**). Poiché l'impostazione di questi bit può causare rischi in termini di sicurezza, il loro uso richiede una particolare cautela.

L'impostazione del bit **set group ID** per una directory abilita lo schema di creazione di file in stile BSD, in cui tutti i file creati nella directory appartengono al **gruppo** della directory.

L'impostazione dello **sticky bit** per una directory impedisce la rimozione di un file nella directory da parte dell'utente che non ne è il proprietario. Per preservare i contenuti di un file in directory con permessi di scrittura per tutti, come "`/tmp`" o in directory con permessi di scrittura per un gruppo, non solo si deve impostare il permesso di **scrittura** per il file, ma anche impostare lo **sticky bit** per la directory. In caso contrario, qualunque utente con i permessi di scrittura per la directory può rimuovere il file e crearne uno nuovo con lo stesso nome.

Ecco alcuni interessanti esempi di permessi dei file.

```
$ ls -l /etc/passwd /etc/shadow /dev/ppp /usr/sbin/exim4
crw------T 1 root root   108, 0 Oct 16 20:57 /dev/ppp
-rw-r--r-- 1 root root     2761 Aug 30 10:38 /etc/passwd
-rw-r----- 1 root shadow   1695 Aug 30 10:38 /etc/shadow
-rwsr-xr-x 1 root root   973824 Sep 23 20:04 /usr/sbin/exim4
$ ls -ld /tmp /var/tmp /usr/local /var/mail /usr/src
drwxrwxrwt 14 root root  20480 Oct 16 21:25 /tmp
drwxrwsr-x 10 root staff  4096 Sep 29 22:50 /usr/local
drwxr-xr-x 10 root root   4096 Oct 11 00:28 /usr/src
drwxrwsr-x  2 root mail   4096 Oct 15 21:40 /var/mail
drwxrwxrwt  3 root root   4096 Oct 16 21:20 /var/tmp
```

Esiste un metodo numerico alternativo per descrivere i permessi dei file con chmod(1); tale metodo numerico usa numeri ottali (base=8) di 3 o 4 cifre.

cifra	**significato**
1ª cifra opzionale	somma di **set user ID** (=4), **set group ID** (=2) e **sticky bit** (=1)
2ª cifra	somma dei permessi di **lettura** (=4), **scrittura** (=2) e **esecuzione** (=1) per l'**utente**
3ª cifra	come sopra, ma per il**gruppo**
4ª cifra	come sopra, ma per gli **altri**

Tabella 1.5: Permessi in notazione numerica per i comandi chmod(1)

Sembra complicato, ma è in realtà piuttosto semplice. Se si guardano le prime colonne (2-10) nell'output del comando "`ls -l`" e le si leggono come una rappresentazione binaria (base=2) dei permessi sui file (dove "-" equivale a "0" e "rwx" equivalgono a "1"), le ultime 3 cifre del valore numerico dei permessi dovrebbero apparire come la corretta rappresentazione dei permessi sui file in numerazione ottale (base=8).

Per esempio, provare a fare quanto segue.

```
$ touch foo bar
$ chmod u=rw,go=r foo
$ chmod 644 bar
$ ls -l foo bar
-rw-r--r-- 1 penguin penguin 0 Oct 16 21:39 bar
-rw-r--r-- 1 penguin penguin 0 Oct 16 21:35 foo
```

Suggerimento
Se si deve aver accesso alle informazioni visualizzate da "`ls -l`" in script di shell, si dovrebbero usare i comandi pertinenti, come test(1), stat(1) e readlink(1). Si possono usare anche i comandi interni della shell come "`[`" o "`test`".

1.2.4 Controllo dei permessi per i file appena creati: umask

I permessi che vengono applicati ad un file o una directory appena creati sono limitati dal comando interno della shell `umask`. Vedere dash(1), bash(1) e builtins(7).

```
(file permissions) = (requested file permissions) & ~(umask value)
```

umask	permessi dei file creati	permessi delle directory create	uso
`0022`	`-rw-r--r--`	`-rwxr-xr-x`	scrivibile solo dall'utente
`0002`	`-rw-rw-r--`	`-rwxrwxr-x`	scrivibile solo dal gruppo

Tabella 1.6: Esempi di valori di **umask**

Il sistema Debian usa, in modo predefinito, uno schema UPG (User Private Group, gruppo privato dell'utente). Ogni volta che viene aggiunto un nuovo utente al sistema, viene creato un UPG; questo ha lo stesso nome dell'utente per il quale è stato creato e quell'utente è l'unico membro dell'UPG. Lo schema UPG rende sicura l'impostazione della umask a `0002` dato che ogni utente ha il proprio gruppo privato. (In alcune varianti di Unix è abbastanza comune impostare tutti gli utenti normali come appartenenti ad un unico gruppo **`users`** ed è in quel caso una buona idea impostare umask a `0022` per ragioni di sicurezza.)

Suggerimento
Abilitare UPG inserendo "`umask 002`" nel file `~/.bashrc`.

1.2.5 Permessi per gruppi di utenti (gruppi)



avvertimento
Assicurarsi di salvare le modifiche non ancora salvate prima di fare il riavvio o azioni simili.

Si può aggiungere l'utente `pinguino` ad un gruppo `uccelli` in due passi:

- Cambiare la configurazione del gruppo usando uno dei modi seguenti:
  - Eseguire "`sudo usermod -aG uccelli pinguino`".
  - Eseguire "`sudo adduser pinguino uccelli`" (solo in sistemi Debian tipici).
  - Eseguire "`sudo vigr`" per `/etc/group` e "`sudo vigr -s`" per `/etc/gshadow` per aggiungere `pinguino` alla fine della riga per `uccelli`.
- Applicare la configurazione usando uno dei due modi seguenti:
  - Spegnimento competo e riaccesso (opzione migliore).
  - Fare il logout dal menu della GUI e fare il login. (Questo potrebbe non funzionare in ambienti desktop moderni.

Si può rimuovere l'utente `pinguino` dal gruppo `uccelli` in due passi:

- Cambiare la configurazione del gruppo usando uno dei modi seguenti:
  - Eseguire "`sudo usermod -rG uccelli pinguino`".
  - Eseguire "`sudo deluser pinguino uccelli`" (solo in sistemi Debian tipici).
  - Eseguire "`sudo vigr`" per `/etc/group` e "`sudo vigr -s`" per `/etc/gshadow` per rimuovere `pinguino` dalla riga per `uccelli`.
- Applicare la configurazione usando uno dei due modi seguenti:
  - Spegnimento competo e riaccesso (opzione migliore).
  - Eseguire "`kill -TERM -1`" e fare alcune azioni di correzione, come "`systemctl restart NetworkManager.ser`
  - Fare il logout dal menu della GUI non è un'opzione per il desktop GNOME.

Qualsiasi tentativo di riavvio a caldo è un fragile sostituto di un vero riavvio completo nei sistemi desktop moderni.

Nota
In alternativa, si possono aggiungere dinamicamente gli utenti ai gruppi durante il processo di autenticazione aggiungendo la riga "`auth optional pam_group.so`" al file "`/etc/pam.d/common-auth`" e configurando "`/etc/security/group.conf`". (Vedere Capitolo 4.)

I dispositivi hardware sono, in un sistema Debian, semplicemente un altro tipo di file. Se si hanno problemi ad accedere a dispositivi quali unità flash USB e CD-ROM da un account utente, si dovrebbe inserire quell'utente nel gruppo appropriato.

Alcuni importanti gruppi pre-impostati dal sistema permettono ai loro membri l'accesso a file e device particolari senza i privilegi di `root`.

gruppo	**descrizione dei file e device accessibili**
`dialout`	accesso diretto e completo alle porte seriali ("`/dev/ttyS[0-3]`")
`dip`	accesso limitato alle porte seriali per connessione **IP dialup** a peer fidati
`cdrom`	unità CD-ROM, DVD+/-RW
`audio`	device audio
`video`	device video
`scanner`	scanner
`adm`	registri di monitoraggio del sistema
`staff`	alcune directory per compiti minori di amministrazione: ""`/usr/local`", "`/home`"

Tabella 1.7: Elenco dei principali gruppi forniti dal sistema per accesso ai file

Suggerimento
È necessario far parte del gruppo `dialout` per riconfigurare il modem, comporre numeri, ecc. Se però `root` crea file di configurazione predefiniti per i peer fidati in "`/etc/ppp/peers/`", è necessario solamente far parte del grupo `dip` per creare una connessione **IP dialup** a tali peer fidati usando i comandi pppd(8), pon(1) e poff(1).

Alcuni importanti gruppi pre-impostati dal sistema permettono ai loro membri di eseguire particolari comandi senza i privilegi di `root`.

Per l'elenco completo degli utenti e gruppi forniti dal sistema, vedere la recente versione del documento "Utenti e gruppi" in "`/usr/share/doc/base-passwd/users-and-groups.html`" fornito dal pacchetto `base-passwd`.

Vedere passwd(5), group(5), shadow(5), newgrp(1), vipw(8), vigr(8) e pam_group(8) per informazioni sui comandi di gestione di utenti e gruppi sul sistema.

gruppo	comandi accessibili
`sudo`	eseguire qualsiasi comando con i privilegi del superutente
`lpadmin`	eseguire comandi per aggiungere, modificare e rimuovere stampanti dal database delle stampanti

Tabella 1.8: Elenco dei principali gruppi forniti dal sistema per l'esecuzione di particolari comandi

1.2.6 Orari

Ci sono tre tipi di orari per un file GNU/Linux.

tipo	significato (definizione storica Unix)
mtime	orario di modifica del file (`ls -l`)
ctime	orario di cambiamento di stato del file (`ls -lc`)
atime	orario dell'ultimo accesso al file (`ls -lu`)

Tabella 1.9: Elenco dei tipi di data

Nota
ctime non è l'orario di creazione del file.

Nota
L'attuale significato di **atime** sui sistemi GNU/Linux potrebbe essere diverso dalla sua definizione storica in Unix.

- La sovrascrittura di un file cambia tutti gli attributi **mtime**, **ctime** e **atime** del file.
- Il cambiamento del proprietario o dei permessi di un file cambia gli attributi **ctime** e **atime** del file.
- La lettura di un file cambia l'attributo **atime** di un file nei sistemi Unix storici.
- La lettura di un file cambia l'attributo **atime** di un file in un sistema GNU/Linux se il file system è montato con "`strictatime`".
- Leggere un file per la prima volta o dopo un giorno modifica l'attributo **atime** del file stesso su sistemi GNU/Linux, se il filesystem è stato montato con "`relatime`". (comportamento di default da Linux 2.6.30)
- Leggere un file non modifica l'attributo **atime** del file stesso su sistemi GNU/Linux se il filesystem è stato montato con "`noatime`".

Nota
Le opzioni di mount "`noatime`" e "`relatime`" sono introdotte per migliorare le prestazioni in lettura del file system nei normali casi d'uso. Semplici operazioni di lettura dei file con l'opzione "`strictatime`" comportano lunghe operazioni di scrittura per aggiornare l'attributo **atime**. L'attributo **atime** però è raramente utilizzato, a parte per il file mbox(5). Vedere mount(8).

Usare il comando touch(1) per cambiare l'orario di file esistenti.

Per gli orari, il comando `ls` produce in output stringhe localizzate con la localizzazione non inglese ("`it_IT.UTF-8`").

```
$ LANG=C  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 Oct 16 21:35 foo
$ LANG=en_US.UTF-8  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 Oct 16 21:35 foo
$ LANG=fr_FR.UTF-8  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 oct. 16 21:35 foo
```

Suggerimento
Vedere Sezione 9.3.4 per personalizzare l'output di "`ls -l`".

1.2.7 Collegamenti

Ci sono due metodi per associale un file "`pippo`" ad un diverso nome file "`pluto`".

- Hard link
  - Nome duplicato per un file esistente
  - "`ln foo bar`"
- Collegamento simbolico o symlink
  - File speciale che punta ad un altro file in base al nome
  - "`ln -s foo bar`"

Vedere l'esempio seguente per notare i cambiamenti nel conteggio dei collegamenti e le sottili differenze tra i risultati del comando `rm`.

```
$ umask 002
$ echo "Original Content" > foo
$ ls -li foo
1449840 -rw-rw-r-- 1 penguin penguin 17 Oct 16 21:42 foo
$ ln foo bar     # hard link
$ ln -s foo baz  # symlink
$ ls -li foo bar baz
1449840 -rw-rw-r-- 2 penguin penguin 17 Oct 16 21:42 bar
1450180 lrwxrwxrwx 1 penguin penguin  3 Oct 16 21:47 baz -> foo
1449840 -rw-rw-r-- 2 penguin penguin 17 Oct 16 21:42 foo
$ rm foo
$ echo "New Content" > foo
$ ls -li foo bar baz
1449840 -rw-rw-r-- 1 penguin penguin 17 Oct 16 21:42 bar
1450180 lrwxrwxrwx 1 penguin penguin  3 Oct 16 21:47 baz -> foo
1450183 -rw-rw-r-- 1 penguin penguin 12 Oct 16 21:48 foo
$ cat bar
Original Content
$ cat baz
New Content
```

Il collegamento fisico può essere creato all'interno dello stesso file system e condivide lo stesso numero di inode, come rivela l'opzione "`-i`" di ls(1).

Il collegamento simbolico ha sempre permessi di accesso nominali "`rwxrwxrwx`, come mostrato nell'esempio precedente, ma con gli effettivi permessi di accesso stabiliti dai permessi del file a cui punta.



Attenzione
In generale è una buona idea non creare collegamenti simbolici complicati o non creare collegamenti fisici per nulla, a meno di non avere una ragione molto buona per farlo. Possono diventare degli incubi quando la combinazione logica dei collegamenti simbolici crea cicli ricorsivi nel file system.

Nota
È in generale preferibile usare collegamenti simbolici piuttosto che fisici, a meno che non sia abbia una buona ragione per usare un collegamento fisico.

La directory "`.`" punta alla directory in cui appare, perciò il conteggio dei collegamenti per ogni nuova directory inizia da 2. La directory "`..`" punta alla directory genitore, perciò il conteggio dei collegamenti di una directory aumenta con l'aggiunta di nuove sottodirectory.

Se si è appena passati da Windows a Linux, appare presto chiaro come sia ben progettato il collegamento dei nomi di file in Unix se paragonato con i collegamenti in ambiente Windows che sono l'equivalente più prossimo in quel sistema. Dato che sono implementati nel file system, le applicazioni non possono vedere alcuna differenza tra un file collegamento ed un originale. Nel caso di collegamenti fisici, non c'è realmente nessuna differenza.

1.2.8 Pipe con nome (FIFO)

Una pipe con nome è un file che si comporta da pipe. Si mette qualcosa dentro il file e questa cosa esce dall'altra parte. Questa è chiamata una FIFO (First-In-First_Out, primo ad entrare-primo ad uscire): la prima cosa che viene immessa nella pipe è la prima ad uscire dall'altra estremità.

Se si scrive su una pipe con nome, il processo che sta eseguendo la scrittura nella pipe non termina fino a che l'informazione scritta nella pipe non viene letta. Se si legge da una pipe con nome, il processo di lettura attende che non ci sia più nulla da leggere prima di terminare. La dimensione della pipe è sempre zero: non archivia dati, ma semplicemente collega due processi come fa la funzionalità fornita dalla sintassi per pipe "`|`" della shell. Tuttavia, dato che questa pipe ha un nome, i due processi non devono essere nella stessa riga di comando e nemmeno essere eseguiti dallo stesso utente. Le pipe sono un'innovazione di Unix di grandissimo impatto.

Per esempio, provare a fare quanto segue.

```
$ cd; mkfifo mypipe
$ echo "hello" >mypipe & # put into background
[1] 8022
$ ls -l mypipe
prw-rw-r-- 1 penguin penguin 0 Oct 16 21:49 mypipe
$ cat mypipe
hello
[1]+  Done                    echo "hello" >mypipe
$ ls mypipe
mypipe
$ rm mypipe
```

1.2.9 Socket

I socket sono usati moltissimo da tutte le comunicazioni Internet, da database e dal sistema operativo stesso. Sono simili alle pipe con nome (FIFO) e permettono ai processi di scambiare informazioni anche tra computer diversi. Per un socket, non è necessario che questi processi siano in esecuzione contemporaneamente, né di essere eseguiti come processi figli di uno stesso processo antenato. Questo è il punto culminante della comunicazione tra processi (IPC). Lo scambio di informazioni può avvenire sulla rete tra host diversi. I due più comuni sono il socket Internet e gli Unix domain socket.

Suggerimento
"`ss -t -u -a`" (dal pacchetto `iproute2`) fornisce una vista d'insieme molto utile dei socket aperti su un dato sistema.

1.2.10 File di device

I file di device fanno riferimento a dispositivi fisici o virtuali sul sistema, come i dischi fissi, la scheda video, lo schermo o la tastiera. Un esempio di dispositivo virtuale è la console, rappresentata da "`/dev/console`".

Ci sono 2 tipi di file di device.

- **Device a caratteri**
  - Vi si accede un carattere alla volta.
  - 1 carattere = 1 byte
  - Es., device della tastiera, porta seriale, ...
- **Device a blocchi**
  - Vi si accede in base a unità più grandi chiamate blocchi.
  - 1 blocco > 1 byte
  - Es., dischi fissi, ...

I file di device possono essere letti e scritti, anche se è probabile che i file contengano dati binari che appaiono come farfuglii incomprensibili per le persone. La scrittura diretta di dati in questi file è utile a volte per trovare la soluzione a problemi con le connessioni hardware. Si può, per esempio, fare il dump di un file di testo in un device di stampa "`/dev/lp0`" o inviare comandi del modem alla porta seriale appropriata "`/dev/ttyS0`". Ma, a meno che ciò non venga fatto con cautela, può causare grandissimi disastri. Perciò, essere prudenti.

Nota
Per il normale accesso ad una stampante, usare lp(1).

I numeri di nodo dei device sono visualizzati se si esegue ls(1) come nell'esempio seguente.

```
$ ls -l /dev/sda /dev/sr0 /dev/ttyS0 /dev/zero
brw-rw---T  1 root disk     8,  0 Oct 16 20:57 /dev/sda
brw-rw---T+ 1 root cdrom   11,  0 Oct 16 21:53 /dev/sr0
crw-rw---T  1 root dialout  4, 64 Oct 16 20:57 /dev/ttyS0
crw-rw-rw-  1 root root     1,  5 Oct 16 20:57 /dev/zero
```

- "`/dev/sda`" ha major number del device 8 e minor number del device 0. È accessibile in lettura e scrittura dagli utenti che appartengono al gruppo `disk`.
- "`/dev/sr0`" ha major number del device 11 e minor number del device 0. È accessibile in lettura e scrittura dagli utenti che appartengono al gruppo `cdrom`.
- "`/dev/ttyS0`" ha major number del device 4 e minor number del device 64. È accessibile in lettura e scrittura dagli utenti che appartengono al gruppo `dialout`.
- "`/dev/zero`" ha major number del device 1 e minor number del device 5. È accessibile in lettura e scrittura da chiunque.

Nei sistemi Linux moderni il file system sotto "`/dev`" viene popolato automaticamente dal meccanismo udev(7).

file di device	azione	descrizione della risposta
`/dev/null`	lettura	restituisce il "carattere EOF (End of File, fine del file)"
`/dev/null`	scrittura	non ritorna nulla (un pozzo senza fondo in cui buttare via dati)
`/dev/zero`	lettura	ritorna il "carattere `\0` (NUL)", diverso dal numero zero in ASCII
`/dev/random`	lettura	ritorna caratteri a caso da un vero generatore di numeri casuali che dà reale entropia (lento)
`/dev/urandom`	lettura	ritorna caratteri casuali da un generatore di numeri pseudocasuali crittograficamente sicuro
`/dev/full`	scrittura	ritorna l'errore di disco pieno (ENOSPC)

Tabella 1.10: Elenco di file dei device speciali

1.2.11 File di device speciali

Ci sono alcuni file di device speciali.

Sono spesso usati insieme alla redirezione di shell (vedere Sezione 1.5.8).

1.2.12 procfs e sysfs

procfs e sysfs, montati in "`/proc`" e "`/sys`" sono pseudo-file system ed espongono strutture interne di dati del kernel nello spazio utente. In altre parole, queste voci sono virtuali, funzionano cioè come una comoda finestra sul funzionamento del sistema operativo.

La directory "`/proc`" contiene (tra le altre cose) una sottodirectory per ciascun processo in esecuzione sul sistema che prende nome dall'ID del processo (PID). Le utilità di sistema, come ps(1), che accedono alle informazioni sui processi, ottengono le loro informazioni da questa struttura di directory.

Le directory in "`/proc/sys/`" contengono interfacce per cambiare certi parametri del kernel durante l'esecuzione. (Si può fare la stessa cosa tramite comandi sysctl(8) specifici o tramite il suo file di precaricamento/configurazione "`/etc/sysctl.d/*.conf`".)

Le persone spesso vanno in panico quando si accorgono di un file in particolare, "`/proc/kcore`" che è particolarmente enorme. È (più o meno) una copia del contenuto della memoria del computer ed è usato per fare il debug del kernel. È un file virtuale che punta alla memoria del computer perciò non ci si preoccupi della sua dimensione.

La directory "`/sys`" contiene strutture dati del kernel esportate, i loro attributi e i collegamenti tra di esse. Contiene anche interfacce per cambiare alcuni parametri del kernel durante l'esecuzione.

Vedere "`proc.txt(.gz)`", "`sysfs.txt(.gz)`" e altri documenti correlati nella documentazione del kernel Linux ("`/usr/share/doc/linux-doc-*/Documentation/filesystems/*`") fornita dai pacchetti `linux-doc-*`.

1.2.13 tmpfs

tmpfs è un file system temporaneo che contiene tutti i file nella memoria virtuale. I dati del tmpfs nella page cache in memoria possono essere spostati nello spazio di swap su disco quando necessario.

La directory "`/run`" viene montata come il tmpfs nelle prime fasi del processo di avvio. Ciò vi permette la scrittura anche quando la directory "`/`" è montata in sola lettura. Questa è la nuova posizione per la memorizzazione dei file transitori e sostituisce diverse posizioni descritte nella versione 2.3 del Filesystem Hierarchy Standard(Standard per la gerarchia del file system):

- "`/var/run`" → "`/run`"
- "`/var/lock`" → "`/run/lock`"

- "/dev/shm" → "/run/shm"

Vedere "tmpfs.txt(.gz)" nella documentazione del kernel Linux ("/usr/share/doc/linux-doc-*/Documentation fornita dai pacchetti linux-doc-*.

1.3 Midnight Commander (MC)

Midnight Commander (MC) è un "coltellino svizzero" GNU per la console Linux ed altri ambienti in terminale. Dà ai principianti la possibilità di usare una console comandata da menu che è molto più facile da imparare dei comandi Unix standard.

Potrebbe essere necessario installare il pacchetto di Midnight Commander che si chiama "mc" con il seguente comando.

```
$ sudo apt-get install mc
```

Usare il comando mc(1) per esplorare il sistema Debian. È il modo migliore di imparare. Esplorare qualche posizione interessante usando semplicemente i tasti freccia e il tasto Invio.

- "/etc" e le sue sottodirectory
- "/var/log" e le sue sottodirectory
- "/usr/share/doc" e le sue sottodirectory
- "/usr/sbin" e "/usr/bin"

1.3.1 Personalizzazione di MC

Per far sì che MC cambi la directory di lavoro in uscita e si sposti con cd nella directory, suggerisco di modificare "~/.bashrc" per includere uno scipt fornito nel pacchetto mc.

```
. /usr/lib/mc/mc.sh
```

Vedere mc(1) (sotto l'opzione "-P") per capirne la ragione. (Se non è chiaro esattamento quello di cui sto parlando, si può tornare a questo più tardi.)

1.3.2 Avvio di MC

MC può essere avviato nel modo seguente.

```
$ mc
```

MC si prende cura di tutte le operazioni sui file attraverso i suoi menu, richiedendo perciò solo un minimo sforzo da parte dell'utente. Premere semplicemente F1 per ottenere la schermata di aiuto. Si può giocherellare con MC premendo semplicemente i tasti freccia e i tasti funzione.

Nota
In alcune console, come gnome-terminal(1), le pressioni dei tasti funzione potrebbero essere rubate dal programma di console. Si possono disabilitare queste funzionalità per gnome-terminal nei menu "Preferenze" → "Generale" e "Scorciatoie".

Se si incontrano problemi di codifica dei caratteri con visualizzazione di caratteri spazzatura, l'aggiunta dell'opzione "-a" alla riga di comando di MC può aiutare a prevenirli.

Se ciò non risolve i problemi di visualizzazione in MC, vedere Sezione 8.3.1.

1.3.3 Gestore dei file in MC

L'impostazione predefinita è con due pannelli di directory contenenti gli elenchi dei file. Un'altra utile modalità è l'impostazione della finestra destra per contenere le "informazioni" per vedere informazioni sui privilegi di accesso dei file, ecc. Di seguito sono elencati alcuni tasti essenziali. Con il demone gpm(8) in esecuzione, si può anche usare il mouse in una console a caratteri Linux. (Assicurarsi di premeri il tasto Maiusc per ottenere il comportamento normale di taglia e incolla in MC.)

tasto	**azione associata**
`F1`	menu di aiuto
`F3`	visualizzatore interno di file
`F4`	editor interno
`F9`	attiva il menu a tendina
`F10`	esce da Midnight Commander
`Tab`	sposta tra le due finestre
`Insert` o `Ctrl-T`	segna il file per una operazione su più file, come la copia
`Del`	cancella il file (essere prudenti: impostare MC per la modalità di cancellazione sicura)
tasti freccia	intuitivi

Tabella 1.11: Le associazioni dei tasti di MC

1.3.4 Trucchetti per la riga di comando di MC

- Il comando `cd` cambia la directory mostrata nel pannello selezionato.
- `Ctrl-Invio` o `Alt-Invio` copia un nome di file nella riga di comando. Usarlo con i comandi cp(1) e mv(1) assieme alla modifica della riga di comando.
- `Alt-Tab` mostra le scelte date dall'espansione di shell dei nomi di file.
- Si possono specificare le directory iniziali per entrambe le finestre come argomenti per MC; per esempio, "`mc /etc /root`".
- `Esc` + `tasto n` → `Fn` (cioè, `Esc` + `1` → `F1`, ecc.; `Esc` + `0` → `F10`)
- Premere `Esc` prima di un tasto ha lo stesso effetto di premere `Alt` ed il tasto contemporaneamente; premere cioè `Esc` + `c` per `Alt-C`. `Esc` è chiamato meta-tasto ed è a volte indicato con "`M-`".

1.3.5 L'editor interno di MC

L'editor interno ha un interessante schema per il taglia e incolla. La pressione di `F3` marca l'inizio di una selezione, una seconda pressione di `F3` marca la fine della selezione e la evidenzia. Si può poi muovere il cursore. Se si preme F6, l'area selezionata viene spostata nella posizione del cursore. Se si preme F5 l'area selezionata viene copiata ed inserita alla posizione del cursore. `F2` salva il file. `F10` fa uscire. La maggior parte dei tasti cursore funziona in modo intuitivo.

Si può avviare direttamente questo editor su di un file usando uno dei comandi seguenti.

```
$ mc -e filename_to_edit
```

```
$ mcedit filename_to_edit
```

Questo non è un editor multi-finestra, ma si possono usare più console Linux per ottenere lo stesso effetto. Per copiare tra finestre, usare i tasti Alt-F*n* per cambiare console virtuale e usare "File→Insert file" o "File→Copy to file" per spostare una porzione di un file in un altro file.

Questo editor interno può essere rimpiazzato da qualsiasi editor esterno a propria scelta.

Inoltre molti programmi usano la variabile d'ambiente "`$EDITOR`" o "`$VISUAL`" per decidere quale editor usare. Se inizialmente non ci si trova a proprio agio con vim(1) o nano(1), si può impostare queste variabili a "`mcedit`" aggiungendo le righe seguenti al file "`~/.bashrc`".

```
export EDITOR=mcedit
export VISUAL=mcedit
```

Io raccomando di impostarle a "`vim`", se possibile.

Se non ci si trova a proprio agio con vim(1), si può continuare ad usare mcedit(1) per la maggior parte dei compiti di manutenzione del sistema.

1.3.6 Il visualizzatore interno di MC

MC è un visualizzatore molto intelligente. È un grande strumento per cercare parole in documenti. Lo uso sempre per i file nella directory "`/usr/share/doc`". È il metodo più veloce per navigare tra una massa di informazioni Linux. Questo visualizzatore può essere avviato direttamente usando uno dei comandi seguenti.

```
$ mc -v path/to/filename_to_view
```

```
$ mcview path/to/filename_to_view
```

1.3.7 Funzionalità di avvio automatico di MC

Se si preme Invio su di un file, un programma appropriato gestirà il contenuto del file (vedere Sezione 9.4.11). Questa è una funzionalità molto comoda di MC.

tipo di file	**reazione al tasto Invio**
file eseguibile	esegue comando
file man	invia i contenuti tramite pipe al software di visualizzazione
file html	invia i contenuti tramite pipe al browser web
file "`*.tar.gz`" e "`*.deb`"	naviga i suoi contenuti come se fosse una sottodirectory

Tabella 1.12: La reazione al tasto Invio in MC

Per far sì che queste funzionalità di visualizzazione dei file e di file virtuali funzionino, i file visualizzabili non dovrebbero essere impostati come eseguibili. Cambiare il loro stato usando chmod(1) o attraverso il menu file di MC.

1.3.8 File system virtuale di MC

MC può essere usato per accedere a file in Internet. Premere F9 per andare nel menu, usare "`Invio`" e "h" per attivare un file system Shell. Inserire un URL nella forma "`sh://[utente@]macchina[:opzioni]/[dir-remota]`", che recupera la directory remota che apparirà come una directory locale usando `ssh`.

1.4 Ambiente di lavoro di base in stile Unix

Anche se MC permette di fare quasi tutto, è bene imparare come usare gli strumenti a riga di comando invocati dal prompt di shell e prendere familiarità con l'ambiente di lavoro in stile Unix.

1.4.1 La shell di login

Dato che la shell di login può essere utilizzata da alcuni programmi di inizializzazione del sistema, è prudente mantenerla come bash(1) e evitare di cambiare la shell di login in chsh(1).

Se si desidera usare un prompt di shell interattiva diverso, impostarlo dalla configurazione dell'emulatore di terminale in GUI o avviarlo da `~/.bashrc`, ad esempio mettendoci "`exec /usr/bin/zsh -i -l`" o "`exec /usr/bin/fish -i -l`".

pacchetto	popcon	dimensione	Shell POSIX	descrizione
`bash`	V:886, I:999	7308	Sì	Bash: GNU Bourne Again SHell (standard de facto)
`bash-completion`	V:37, I:962	1952	N/D	completamento programmabile per la shell bash
`dash`	V:929, I:999	207	Sì	Debian Almquist Shell, buona per script di shell
`zsh`	V:39, I:68	2571	Sì	Z shell: la shell standard con molti miglioramenti
`tcsh`	V:4, I:14	1366	No	TENEX C Shell: una versione migliorata di Berkeley csh
`mksh`	V:4.7, I:7.5	7713	Sì	Una versione della Korn shell
`csh`	V:1.0, I:5.2	348	No	C Shell OpenBSD, una versione di Berkeley csh
`sash`	V:0.3, I:4.7	1245	Sì	Stand-alone shell con comandi interni (non adatta per essere la "`/usr/bin/sh`" standard)
`ksh`	V:0.3, I:7.9	65	Sì	la versione reale di AT&T della Korn shell
`rc`	V:0.07, I:0.70	182	No	implementazione della rc shell di AT&T Plan 9
`posh`	V:0.02, I:0.22	195	Sì	Policy-compliant Ordinary SHell (derivata da `pdksh`)

Tabella 1.13: Elenco di programmi shell

Suggerimento
Sebbene le shell in stile POSIX condividano la stessa sintassi di base, esse possono differire nel comportamento relativo a cose anche basilari, come le variabili della shell e l'espansione dei modelli glob. Per i dettagli controllare la loro documentazione.

In questo capitolo del tutorial la shell interattiva considerata è sempre `bash`.

1.4.2 Personalizzare bash

Si può personalizzare il comportamento di bash(1) con "`~/.bashrc`".

Per esempio provare quanto segue.

```
# enable bash-completion
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

# CD upon exiting MC
```

```
. /usr/lib/mc/mc.sh

# set CDPATH to a good one
CDPATH=.:/usr/share/doc:~:~/Desktop:~
export CDPATH

PATH="${PATH+$PATH:}/usr/sbin:/sbin"
# set PATH so it includes user's private bin if it exists
if [ -d ~/bin ] ; then
  PATH="~/bin${PATH+:$PATH}"
fi
export PATH

EDITOR=vim
export EDITOR
```

Suggerimento
Si possono trovare altri suggerimenti sulla personalizzazione di `bash`, come Sezione 9.3.6, in Capitolo 9.

Suggerimento
Il pacchetto `bash-completion` permette il completamento automatico per `bash`.

1.4.3 Associazioni di tasti speciali

Nell'ambiente *nix, ci sono alcune associazioni di tasti che hanno un significato speciale. Notare che in una console a caratteri Linux normale solo i tasti `Ctrl` e `Alt`sinistri funzionano come atteso. Ecco alcune combinazioni di tasti che vale la pena ricordare.

tasto	descrizione dell'associazione di tasti
`Ctrl-U`	cancella il contenuto della riga prima del cursore
`Ctrl-H`	cancella il carattere prima del cursore
`Ctrl-D`	termina l'input (se si sta usando la shell, esce dalla shell)
`Ctrl-C`	termina un programma in esecuzione
`Ctrl-Z`	arresta temporaneamente un programma spostandolo come compito sullo sfondo
`Ctrl-S`	ferma l'output a schermo.
`Ctrl-Q`	riattiva l'output a schermo.
`Ctrl-Alt-Del`	riavvia/ferma il sistema, vedere inittab(5)
`Left-Alt-key` (opzionalmente, `tasto Windows`)	meta-tasto per Emacs e interfacce utente simili
`Up-arrow`	avvia la ricerca nello storico dei comandi in `bash`
`Ctrl-R`	inizia una ricerca incrementale nello storico dei comandi in `bash`
`Tab`	completa l'input di un nome di file nella riga di comando in `bash`
`Ctrl-V Tab`	inserisce in input un carattere `Tab` senza espansione nella riga di comando in `bash`

Tabella 1.14: Elenco di associazioni di tasti per bash

Suggerimento
La funzionalità di `Ctrl-S` nel terminale può essere disabilitata usando stty(1).

1.4.4 Operazioni del mouse

Il funzionamento del mouse per il testo nel sistema Debian mescola 2 stile con alcune peculiarità:

- Funzioni del mouse in stile Unix tradizionali:
  - uso di 3 pulsanti (clic)
  - uso di PRIMARY (principale)
  - usato da applicazioni X come `xterm` e applicazioni testuali nella console Linux
- Funzionamento del mouse moderno in stile GUI:
  - uso di 2 pulsanti (trascina e clic)
  - uso di PRIMARY (principale) e CLIPBOARD (appunti)
  - usato in applicazioni GUI moderne come `gnome-terminal`

azione	**risposta**
Clic sinistro e trascinamento	marca l'intervallo della selezione PRINCIPALE
Clic sinistro	marca l'inizio dell'intervallo per la selezione PRINCIPALE
Clic destro (tradizionale)	marca la fine dell'intervallo per la selezione PRINCIPALE
Clic destro (moderno)	menu dipendente dal contesto (taglia/copia/incolla)
Clic di mezzo o Maiusc-`Ins`	inserisce la selezione PRINCIPALE alla posizione del cursore
`Ctrl-X`	taglia la selezione PRINCIPALE negli APPUNTI
`Ctrl-C` (`Shift-Ctrl-C` nel terminale)	copia la selezione PRINCIPALE negli APPUNTI
`Ctrl-V`	incolla gli APPUNTI alla posizione del cursore

Tabella 1.15: Elenco di funzioni del mouse e relative azioni dei tasti in Debian

Qui, la selezione PRINCIPALE è l'intervallo di testo evidenziato. All'interno del programma di terminale viene usato invece `Shift-Ctrl-C` per evitare di terminare un programma in esecuzione.

La rotella centrale presente nei mouse moderni viene considerata come il pulsante di mezzo e può essere usata per fare il corrispondente clic. Cliccare contemporaneamente con i pulsanti destro e sinistro funziona da clic di mezzo nelle situazioni in cui si hanno mouse a 2 pulsanti.

Per poter usare il mouse in una console a caratteri Linux, è necessario che gpm(8) sia in esecuzione come demone.

1.4.5 Il paginatore

Il comando less(1) è il paginatore (navigatore del contenuto dei file) migliorato. Legge il file specificato nell'argomento del comando o il suo standard input. Premere "h" se si ha bisogno di aiuto nella navigazione con il comando `less`. Può fare molto di più di more(1) e può essere fornito di superpoteri eseguendo "`eval $(lesspipe)`" o "`eval $(lessfile)`" nello script di avvio della shell. Vedere ulteriori informazioni in "`/usr/share/doc/less/LESSOPEN`". L'opzione "`-R`" permette output raw e abilita le sequenze di escape ANSI per i colori. Vedere less(1).

Suggerimento
Nel comando `less` digitare "h" per vedere la schermata di aiuto, digitare "`/`" o "?" per cercare una stringa e digitare "`-i`" per cambiare la sensibilità a maiuscole/minuscole.

1.4.6 L’editor di testo

Si dovrebbe diventare competenti in una delle varianti dei programmi Vim o Emacs che sono popolari sui sistemi *nix.

Penso che abituarsi ai comandi Vim sia la cosa giusta da fare, dato che un editor Vi è sempre presente nel mondo Linux/Unix. (In realtà, il `vi` originale o il nuovo `nvi` sono programmi che si trovano ovunque. Io ho scelto invece Vim per i principianti dato che offre l’aiuto attraverso il tasto `F1` pur essendo abbastanza simile e più potente.)

Se si sceglie invece Emacs o XEmacs come editor preferito, si sta facendo comunque davvero un’ottima scelta, specialmente per la programmazione. Emacs ha una vastità di altre funzionalità, incluso il funzionamento come newsreader, editor di directory, programma di posta, ecc. Quando usato per la programmazione o la scrittura di script, riconosce intelligentemente il formato che si sta usando e cerca di fornire assistenza. Alcune persone sostengono che l’unico programma di cui hanno bisogno su Linux è Emacs. Dieci minuti spesi ad imparare Emacs ora possono far risparmiare ore in futuro. È caldamente raccomandato avere a portata di mano il manuale di GNU Emacs da consultare quando si sta imparando Emacs.

Tutti questi programmi sono di solito forniti con un programma tutor che aiuta ad impararli facendo pratica. Avviare Vim digitando ”`vim`” e premere il tasto F1. Si dovrebbero leggere come minimo le prime 35 righe. Poi seguire il corso di apprendimento in linea muovendo il cursore su ”`|tutor|`” e premendo `Ctrl-]`.

Nota
I buoni editor, come Vim ed Emacs, possono gestire correttamente testi UTF-8 e testi in altre codifiche esotiche. È opportuno usare l’ambiente GUI con la localizzazione UTF-8 e installare i programmi e i tipi di carattere necessari. Gli editor hanno opzioni per impostare la codifica dei file indipendentemente dall’ambiente GUI. Fare riferimento alla loro documentazione sui testi multibyte.

1.4.7 Impostare un editor di testi predefinito

Debian è fornita con svariati editor differenti. Si raccomanda di installare, come detto in precedenza, il pacchetto `vim`.

Debian fornisce un accesso unificato all’editor predefinito di sistema attraverso il comando ”`/usr/bin/editor`” in modo che altri programmi (ad esempio reportbug(1)) possano richiamarlo. Lo si può modificare con il comando seguente.

```
$ sudo update-alternatives --config editor
```

Io raccomando ai principianti la scelta di ”`/usr/bin/vim.basic`” invece di”`/usr/bin/vim.tiny`” dato che supporta l’evidenziazione della sintassi.

Suggerimento
Molti programmi usano la variabile d’ambiente ”`$EDITOR`” o ”`$VISUAL`” per decidere quale editor usare (vedereSezione 1.3.5 e Sezione 9.4.11. Per coerenza, in un sistema Debian, impostare queste variabili a ”`/usr/bin/editor`”. (Storicamente ”`$EDITOR`” era ”ed” e ”`$VISUAL`” era ”`vi`”.)

1.4.8 Usare vim

vim(1) recente si avvia con la sana opzione ”`nocompatible`” e entra nella modalità `NORMAL`.[1]

Usare il programma ”`vimtutor`” per imparare ad usare `vim` attraverso un corso tutorial interattivo.

Il programma `vim` cambia il suo comportamento in risposta ai tasti premuti in base alla **modalità (mode)**. La digitazione con i tasti nel buffer è fatta soprattutto in modalità `INSERT` e in modalità `REPLACE`. Spostare il cursore viene fatto soprattutto in modalità `NORMAL`. La selezione interattiva viene fatta in modalità `VISUAL`. Se si digita ”`:`” in modalità `NORMAL` si cambia la sua **modalità** in `Ex`-mode. `Ex`-mode accetta comandi.

[1] Anche il vecchio `vim` può avviarsi nella modalità ”`nocompatible`” sensata se lo si avvia con l’opzione ”`-N`”.

modalità	Combinazione di tasti	azione
NORMAL	:help|only	visualizza il file di aiuto
NORMAL	:e nomefile.ext	apre un nuovo buffer per modificare nomefile.ext
NORMAL	:w	sovrascrive il buffer corrente sul file originale
NORMAL	:w nomefile.ext	scrive il buffer corrente su nomefile.ext
NORMAL	:q	esce da vim
NORMAL	:q!	forza l'uscita da vim
NORMAL	:only	chiude tutte le altre finestre staccate aperte
NORMAL	:set nocompatible?	controlla se vim è nella sana modalità nocompatible
NORMAL	:set nocompatible	imposta vim nella sana modalità nocompatible
NORMAL	i	entra nella modalità INSERT
NORMAL	R	entra nella modalità REPLACE
NORMAL	v	entra nella modalità VISUAL
NORMAL	V	entra nella modalità VISUAL a righe
NORMAL	Ctrl-V	entra nella modalità VISUAL a blocchi
tranne TERMINAL-JOB	ESC-key	entra nella modalità NORMAL
NORMAL	:term	entra nella modalità TERMINAL-JOB
TERMINAL-NORMAL	i	entra nella modalità TERMINAL-JOB
TERMINAL-JOB	Ctrl-W N (o Ctrl-\ Ctrl-N)	entra nella modalità TERMINAL-NORMAL
TERMINAL-JOB	Ctrl-W :	entra nella Ex-mode nella modalità TERMINAL-NORMAL

Tabella 1.16: Elenco di combinazioni di tasti base di Vim

Suggerimento
Vim viene distribuito con il pacchetto **Netrw**. Netrw supporta lettura e scrittura di file, navigazione di directory in una rete e navigazione locale! Provare Netrw con "`vim .`" (un punto come argomento) e leggere il suo manuale con "`:help netrw`".

Per la configurazione avanzata di `vim`, vedere Sezione 9.2.

1.4.9 Registrare le attività della shell

L'output dei comandi di shell può scorrere fuori dallo schermo e potrebbe essere perduto per sempre. È buona norma registrare le attività di shell in un file in modo da poterle riguardare in seguito. Questo tipo di registro è essenziale quando si fa una qualsiasi attività di amministrazione del sistema.

Suggerimento
Il nuovo Vim (versione>=8.2) può essere utilizzato per registrare in modo pulito le attività di shell usando la modalità `TERMINAL-JOB`. Vedere Sezione 1.4.8.

Il metodo base per registrare l'attività di shell è di eseguirla sotto script(1).

Per esempio, provare a fare quanto segue.

```
$ script
Script started, file is typescript
```

Eseguire qualsiasi comando di shell sotto `script`.

Premere `Ctrl-D` per uscire da `script`.

```
$ vim typescript
```

Vedere Sezione 9.1.1.

1.4.10 Comandi Unix di base

È bene imparare i comandi Unix di base. Il termine "Unix" è qui usato in senso lato; ogni SO clone di Unix offre di solito comandi equivalenti. Il sistema Debian non fa eccezione. Non ci si preoccupi se, al momento, alcuni comandi non funzionano come si vorrebbe. Se si usa `alias` nella shell, i corrispondenti output dei comandi sono diversi. Questi esempi non sono pensati per essere eseguiti necessariamente in questo ordine.

Provare tutti i comandi seguenti da un account utente non privilegiato.

Nota
Unix tradizionalmente nasconde i nomi di file che iniziano con "`.`". Sono tradizionalmente file contenenti informazioni di configurazione e preferenze dell'utente.
Per il comando `cd` vedere builtins(7).
Il paginatore predefinito del sistema di base di Debian è more(1) che non ha lo scorrimento all'indietro. Installando il pacchetto `less`, con la riga di comando "`apt-get install less`", less(1) diventa il paginatore predefinito e si può così scorrere il testo all'indietro usando i tasti freccia.
I caratteri "`[`" e "`]`" nella espressione regolare del comando "`ps aux | grep -e "[e]xim4*"`" sopra citato, permettono a `grep` di evitare di trovare corrispondenza con sé stesso. La parte "`4*`" nella espressione regolare significa 0 o più ripetizioni del carattere "4" e perciò permette a `grep` di trovare corrispondenza sia con "`exim`" sia con "`exim4`". Sebbene il carattere "`*`" sia usato nei nomi di file con metacaratteri della shell e nelle espressioni regolari, il suo significato è diverso nei due casi. Si possono imparare le espressioni regolari da grep(1).

comando	**descrizione**
`pwd`	mostra il nome della directory attuale/di lavoro
`whoami`	mostra il nome dell'utente attuale
`id`	mostra l'identità dell'utente attuale (nome, uid, gid e gruppi associati)
`file` *pippo*	mostra che tipo di file sia il file "*pippo*"
`type -p` *nomecomando*	mostra la posizione del file del comando "*nomecomando*"
`which` *nomecomando*	" "
`type` *nomecomando*	mostra informazioni sul comando "*nomecomando*"
`apropos` *parola-chiave*	trova comandi riguardanti "*parola-chiave*"
`man -k` *parola-chiave*	" "
`whatis` *nomecomando*	mostra una spiegazione di una riga sul comando "*nomecomando*"
`man -a` *nomecomando*	mostra una spiegazione del comando "*nomecomando*" (in stile Unix)
`info` *nomecomando*	mostra una spiegazione piuttosto lunga del comando "*nomecomando*" (in stile GNU)
`ls`	elenca il contenuto di directory (non i file punto e le directory)
`ls -a`	elenca il contenuto di directory (tutti i file e directory)
`ls -A`	elenca il contenuto di directory (quasi tutti i file e directory, cioè salta "`..`" e "`.`")
`ls -la`	elenca tutto il contenuto di directory con informazioni dettagliate
`ls -lai`	elenca tutto il contenuto di directory con numeri di inode e informazioni dettagliate
`ls -d`	elenca tutte le directory dentro la directory attuale
`tree`	mostra il contenuto in forma di albero
`lsof` *pippo*	mostra lo stato aperto per il file "*pippo*"
`lsof -p` *pid*	mostra i file aperti dal processo con ID "*pid*"
`mkdir` *pippo*	crea una nuova directory "*pippo*" nella directory attuale
`rmdir` *pippo*	rimuove la directory "*pippo*" nella directory attuale
`cd` *pippo*	cambia directory spostandosi nella directory "*pippo*" nella directory attuale o in una directory elencata nella variabile "`$CDPATH`"
`cd /`	cambia directory spostandosi nella directory radice
`cd`	cambia directory spostandosi nella directory home dell'utente
`cd` */pippo*	cambia directory spostandosi nella directory con percorso assoluto "*/pippo*"
`cd ..`	cambia directory spostandosi nella directory genitore
`cd ~`*pippo*	cambia directory spostandosi nella directory home dell'utente "*pippo*"
`cd -`	cambia directory spostandosi nella directory precedente
`</etc/motd pager`	mostra il contenuto di "`/etc/motd`" usando il paginatore predefinito
`touch` *filediprova*	crea un file "*filediprova*" vuoto
`cp` *pippo pluto*	copia un file "*pippo*" esistente in un nuovo file "*pluto*"
`rm` *filediprova*	rimuove il file "*filediprova*"
`mv` *pippo pluto*	rinomina un file "*pippo*" esistente con il nuovo nome "*pluto*" ("*pluto*" non deve esistere)
`mv` *pippo pluto*	muove un file "*pippo*" esistente nella nuova posizione "*pluto/pippo*" (la directory "*pluto*" deve esistere)
`mv` *pippo pluto/paperino*	muove un file "*pippo*" esistente in una nuova posizione con il nuovo nome "*pluto/paperino*" (la directory "*pluto*" deve esistere ma non deve esistere "*bar/baz*")
`chmod 600` *pippo*	fa sì che il file "*pippo*" esistente sia non leggibile e non scrivibile da altri (e non eseguibile per nessuno)
`chmod 644` *pippo*	fa sì che il file "*pippo*" esistente sia leggibile ma non scrivibile da altri (e non eseguibile per nessuno)
`chmod 755` *pippo*	fa sì che il file "*pippo*" esistente sia leggibile ma non scrivibile da altri (ed eseguibile per tutti)
`find . -name` *modello*	trova nomi di file corrispondenti al "*modello*" di shell (lento)
`locate -d . `*modello*	trova nomi di file corrispondenti al "*pattern*" di shell (più veloce, usa un database generato regolarmente)
`grep -e "`*modello*`" *.html`	trova un "*modello*" in tutti i file nella directory attuale che terminano con "`.html`" e mostra tutte le corrispondenze

Come esercizio, esplorare le directory e dare un'occhiata al sistema usando i comandi citati sopra. Se si hanno domande su uno qualsiasi dei comandi in console, assicurarsi di leggere la pagina di manuale.

Per esempio, provare a fare quanto segue.

```
$ man man
$ man bash
$ man builtins
$ man grep
$ man ls
```

Può essere un po' difficile abituarsi allo stile delle pagine man perché sono piuttosto concise, in particolar modo le più vecchie e tradizionali. Una volta che ci si fa la mano, però, si apprezza la loro concisione.

Notare che molti comandi in stile Unix, inclusi quelli da GNU e BSD, mostrano una breve informazione di aiuto se li si invoca in uno dei modi seguenti (o in alcuni casi se lo si fa senza usare opzioni).

```
$ commandname --help
$ commandname -h
```

1.5 Il semplice comando di shell

Ora si ha un'idea di come usare il sistema Debian; è tempo di dare uno sguardo più approfondito al meccanismo di esecuzione dei comandi nel sistema Debian. Qui la realtà è stata semplificata ad uso del principiante. Vedere bash(1) per una spiegazione esatta.

Un semplice comando è formato da una sequenza di componenti.

1. Assegnazioni di variabili (opzionale)
2. Nome del comando
3. Opzioni (opzionali)
4. Ridirezioni (opzionali, > , >> , < , << , ecc.)
5. Operatori di controllo (opzionale, && , | | , *a-capo* , ; , & , (,))

1.5.1 Esecuzione dei comandi e variabili d'ambiente

I valori di alcune variabili d'ambiente cambiano il comportamento di alcuni comandi Unix.

I valori predefiniti delle variabili d'ambiente sono impostati inizialmente dal sistema PAM e poi alcuni di essi possono essere reimpostati da alcuni programmi applicativi.

- Il sistema PAM come `pam_env` può impostare variabili d'ambiente con `/etc/pam.conf`", "`/etc/environment`" e "`/etc/default/locale`".
- Un ambiente GUI moderno eseguito da un'unità systemd in modalità utente (vedere systemd.unit(5)) imposta le sue variabili d'ambiente tramite un fille nella directory "`~/.config/environment.d/`".
- L'inizializzazione dei programmi specifica per l'utente può reimpostare variabili d'ambiente con "`~/.profile`", "`~/.bash_profile`" e "`~/.bashrc`".

1.5.2 La variabile ”$LANG”

La localizzazione predefinita è definita nella variabile d’ambiente ”$LANG” ed è configurata nella forma ”LANG=xx_YY.UTF-8” dall’installatore o dalla successiva configurazione della GUI, es. ”Impostazioni” → ”Regione e lingua” → ”Lingua” / ”Formati” per GNOME.

Nota
Io raccomando di configurare l’ambiente di sistema usando per il momento solo la variabile ”$LANG” e di tenersi alla larga dalle variabili ”$LC_*” a meno che non sia strettamente necessario.

Il valore completo di localizzazione impostato nella variabile ”$LANG” consiste di 3 parti: ”xx_YY.ZZZZ”.

valore di localizzazione	significato
xx	codice ISO 639 della lingua (in minuscole), come ”en”
YY	codice ISO 3166 del paese (in maiuscole) come ”US”
ZZZZ	codeset, impostato sempre ad ”UTF-8”

Tabella 1.18: Le 3 parti del valore di localizzazione

localizzazione raccomandata	lingua (zona)
en_US.UTF-8	inglese (USA)
en_GB.UTF-8	inglese (Gran Bretagna)
fr_FR.UTF-8	francese (Francia)
de_DE.UTF-8	tedesco (Germania)
it_IT.UTF-8	italiano (Italia)
es_ES.UTF-8	spagnolo (Spagna)
ca_ES.UTF-8	catalano (spagna)
sv_SE.UTF-8	svedese (Svezia)
pt_BR.UTF-8	portoghese (Brasile)
ru_RU.UTF-8	russo (Russia)
zh_CN.UTF-8	cinese (Repubblica Popolare Cinese)
zh_TW.UTF-8	cinese (Taiwan Repubblica di Cina)
ja_JP.UTF-8	giapponese (Giappone)
ko_KR.UTF-8	coreano (Corea del Sud)
vi_VN.UTF-8	vietnamita (Vietnam)

Tabella 1.19: Elenco di localizzazioni raccomandate

Una tipica esecuzione di un comando usa una sequenza sulla riga della shell come la seguente.

```
$ echo $LANG
en_US.UTF-8
$ date -u
Wed 19 May 2021 03:18:43 PM UTC
$ LANG=fr_FR.UTF-8 date -u
mer. 19 mai 2021 15:19:02 UTC
```

In questo caso il programma date(1) viene eseguito con differenti valori di localizzazione.

- Per il primo comando ”$LANG” è impostata al valore di localizzazione predefinito di sistema: ”en_US.UTF-8”.
- Per il secondo comando ”$LANG” è impostata al valore di localizzazione UTF-8 francese: ”fr_FR.UTF-8”.

La maggior parte delle invocazioni di comandi non è solitamente preceduta da definizioni di variabili d’ambiente. In alternativa all’esempio precedente, si può eseguire quanto segue.

```
$ LANG=fr_FR.UTF-8
$ date -u
mer. 19 mai 2021 15:19:24 UTC
```

Suggerimento
Quando si segnala un bug, è una buona idea, se si usa un ambiente non inglese, eseguire e controllare il comando nella localizzazione "`en_US.UTF-8`".

Per dettagli più specifici sulla configurazione della localizzazione vedere Sezione 8.1.

1.5.3 La variabile "$PATH"

Quando si digita un comando nella shell, questa cerca il comando nelle directory nell'elenco contenuto nella variabile d'ambiente "`$PATH`". Il valore della variabile d'ambiente "`$PATH`" è anche chiamato percorso di ricerca della shell.

In una installazione Debian standard, la variabile d'ambiente "`$PATH`" degli account utente può non includere "`/usr/sbin`" e "`/usr/sbin`". Il comando `ifconfig`, per esempio, deve essere eseguito con il percorso completo, come in "`/usr/sbin/ifconfig`". (Il comando simile `ip` si trova in "`/usr/bin`".)

Si può modificare la variabile d'ambiente "`$PATH`" della shell Bash tramite il file "`~/.bash_profile`" o "`~/.bashrc`".

1.5.4 La variabile "$HOME"

Molti comandi memorizzano una configurazione specifica per l'utente nella directory home e cambiano il loro comportamento in base al suo contenuto. La directory home è identificata dalla variabile d'ambiente "`$HOME`".

valore di "$HOME"	**situazione di esecuzione del programma**
`/`	programma eseguito da processo init (demone)
`/root`	programma eseguito dalla normale shell di root
`/home/`*`utente_normale`*	programma eseguito dalla shell di un utente normale
`/home/`*`utente_normale`*	programma eseguito dal menu del desktop grafico di un utente normale
`/home/`*`utente_normale`*	programma eseguito come root con "`sudo programma`"
`/root`	programma eseguito come root con "`sudo -H programma`"

Tabella 1.20: Elenco di valori di "`$HOME`"

Suggerimento
La shell espande "`~/` nella directory home dell'utente attuale, cioè "`$HOME/`". La shell espande "`~pippo/`" nella directory home di `pippo`, cioè "`/home/pippo/`".

Vedere Sezione 12.1.5 se `$HOME` non è disponibile per il proprio programma.

1.5.5 Opzioni della riga di comando

Alcuni comandi accettano argomenti. Gli argomenti che iniziano con "`-`" o "`--`" sono chiamati opzioni e controllano il comportamento del comando.

```
$ date
Thu 20 May 2021 01:08:08 AM JST
$ date -R
Thu, 20 May 2021 01:08:12 +0900
```

In questo caso l'opzione "`-R`" sulla riga di comando cambia il comportamento di date(1) producendo in output la data in una stringa conforme alla RFC2822.

1.5.6 Glob della shell

Spesso si desidera che un comando abbia effetto su un gruppo di file senza dover digitarli tutti. I modelli di espansione dei nomi di file che usano i **glob** della shell, a volte detti **metacaratteri** o **caratteri jolly**, facilitano questo compito.

modello di glob della shell	descrizione della regola di corrispondenza
`*`	nome file che non inizia con "`.`"
`.*`	I nomi di file che iniziano con "`.`" (in Bash 5.2 e successive, "`.`" e "`..`" sono esclusi in maniera predefinita)
?	esattamente un carattere
`[…]`	esattamente un carattere tra quelli racchiusi tra le parentesi quadre
`[a-z]`	esattamente un carattere che sia un carattere compreso tra "a" e "z"
`[!…]`	esattamente un carattere che non sia uno di quelli racchiusi tra parentesi quadre (non contando "`!`")

Tabella 1.21: Modelli di glob della shell

Nota
A "`/`" in a pathname cannot be matched by the shell glob pattern listed in the above table.

Per esempio, provare a fare quanto segue.

```
$ mkdir foo; cd foo; touch b1.txt b2.txt b3.c b4.h .b5.txt ..b6.txt
$ mkdir bar; cd bar; touch b1.txt b2.txt b3.c b4.h .b5.txt ..b6.txt ; cd -
$ echo b*.txt
b1.txt b2.txt
$ echo b*
b1.txt b2.txt b3.c b4.h bar
$ echo */*.txt
bar/b1.txt bar/b2.txt
$ echo *.[hc]
b3.c b4.h
$ echo .*
.b5.txt ..b6.txt
$ echo b[!1-3]*
b4.h bar
$ echo b*/b[!1-3]*
bar/b4.h
$ cd ..; rm -rf junk
```

Vedere glob(7).

Nota
A differenza della normale espansione dei nomi della shell, il modello shell "`*`" cercato dal comando find(1) con la ricerca "`-name`" ecc., trova corrispondenza anche con nomi di file che iniziano con "`.`". (Nuova funzionalità POSIX.)

Nota
Si può manipolare il comportamento di BASH in relazione ai modelli di glob usando le sue opzioni shopt incorporate come "`dotglob`", "`noglob`", "`nocaseglob`", "`nullglob`", "`extglob`", ecc. Vedere bash(1).

1.5.7 Valore restituito dal comando

Ogni comando restituisce il suo stato di uscita (variabile "$?") come valore restituito.

stato di uscita del comando	valore numerico restituito	valore logico restituito
successo	zero, 0	**VERO**
errore	non-zero, -1	**FALSO**

Tabella 1.22: Codici di uscita dei comandi

Per esempio provare quanto segue.

```
$ [ 1 = 1 ] ; echo $?
0
$ [ 1 = 2 ] ; echo $?
1
```

Nota
Notare che, nel contesto logico della shell, il **successo** è trattato come il valore logico **VERO** che ha valore numerico 0 (zero). Questo è in qualche modo non intuitivo ed è bene rimarcarlo.

1.5.8 Sequenze tipiche di comandi e ridirezione della shell

È bene cercare di ricordare i seguenti idiomi per comandi di shell che vengono digitati sulla stessa riga come parte del comando di shell.

Il sistema Debian è un sistema multi-tasking. I compiti sullo sfondo permettono agli utenti di eseguire più programmi in una singola shell. La gestione dei processi sullo sfondo coinvolge i comandi interni della shell `jobs`, `fg`, `bg` e `kill`. Leggere le sezioni "SIGNALS" e "JOB CONTROL" di bash(1) e builtins(1).

Per esempio, provare a fare quanto segue.

```
$ </etc/motd pager
```

```
$ pager </etc/motd
```

```
$ pager /etc/motd
```

```
$ cat /etc/motd | pager
```

idioma per comando	descrizione
`comando &`	esegue `comando` sullo **sfondo** in una sotto-shell
`comando1 | comando2`	invia tramite **pipe** lo standard output di `comando1` allo standard input di `comando2` (**esecuzione concorrente**)
`comando1 2>&1 | comando2`	invia tramite **pipe** sia lo standard output sia lo standard error di `comando1` allo standard input di `comando2` (**esecuzione** concorrente)
`comando1 ; comando2`	esegue `comando1` e `comando2` in **sequenza**
`comando1 && comando2`	esegue `comando1`; se ha successo, esegue `comando2` in **sequenza** (restituisce lo stato di successo se sia `comando1` **sia** `comando2` hanno successo)
`comando1 || comando2`	esegue `comando1`; se non ha successo, esegue `comando2` in **sequenza** (restituisce lo stato di successo se `comando1` **o** `comando2` ha successo)
`comando > pippo`	ridirige lo standard output di `comando` al file ”`pippo`” (sovrascrivendolo)
`comando 2> pippo`	ridirige lo standard error di `comando` al file ”`pippo`” (sovrascrivendolo)
`comando >> pippo`	ridirige lo standard output di `comando` al file ”`pippo`” (aggiungendo in coda)
`comando 2>> pippo`	ridirige lo standard error di `comando` al file ”`pippo`” (aggiungendo in coda)
`comando > pippo 2>&1`	ridirige sia lo standard output sia lo standard error di `comando` al file `pippo`
`comando < pippo`	usa come standard input di `comando` il file ”`pippo`”
`comando << delimitatore`	usa come standard input di `comando` le righe che seguono fino a che non viene incontrato il testo ”`delimitatore`” (inserire un documento)
`comando <<- delimitatore`	usa come standard input di `comando` le righe che seguono fino a che non viene incontrato il testo ”`delimitatore`”; i caratteri di tabulazione iniziali vengono eliminate dal testo (inserire un documento)

Tabella 1.23: Idiomi per comandi di shell

Benché tutti i 4 esempi di ridirezione della shell mostrino la stessa cosa, l'ultimo esempio esegue un comando `cat` aggiuntivo e spreca risorse senza motivo.

La shell permette di aprire file usando il comando interno `exec`con un descrittore di file arbitrario.

```
$ echo Hello >foo
$ exec 3<foo 4>bar  # open files
$ cat <&3 >&4       # redirect stdin to 3, stdout to 4
$ exec 3<&- 4>&-    # close files
$ cat bar
Hello
```

I descrittori di file 0-2 sono predefiniti.

device	descrizione	descrittore del file
`stdin`	standard input	0
`stdout`	standard output	1
`stderr`	standard error	2

Tabella 1.24: Descrittori di file predefiniti

1.5.9 Alias di comandi

Si possono impostare alias per i comandi usati di frequente.

Per esempio, provare a fare quanto segue.

```
$ alias la='ls -la'
```

Ora "`la`" funziona da scorciatoia per "`ls -la`" che elenca tutti i file nel formato di elencazione esteso.

Si possono elencare tutti gli alias esistenti con a`lias` (vedere la sezione "SHELL BUILTIN COMMANDS" in bash(1)).

```
$ alias
...
alias la='ls -la'
```

Si può identificare il percorso esatto o l'identità di un comando con `type` (vedere la sezione "SHELL BUILTIN COMMANDS") di bash(1)).

Per esempio, provare a fare quanto segue.

```
$ type ls
ls is hashed (/bin/ls)
$ type la
la is aliased to ls -la
$ type echo
echo is a shell builtin
$ type file
file is /usr/bin/file
```

In questo caso `ls` era stato cercato di recente mentre "`file`" no, percio "`ls`" risulta "hashed", cioè la shell ha un record interno per un accesso veloce alla posizione del comando "`ls`".

Suggerimento
Vedere Sezione 9.3.6.

1.6 Elaborazione di testo stile Unix

Nell'ambiente di lavoro Unix, l'elaborazione del testo viene fatta instradando il testo tramite pipe attraverso catene di strumenti standard per l'elaborazione del testo. Questa è stata un'altra fondamentale innovazione di Unix.

1.6.1 Strumenti Unix per il testo

Ci sono alcuni strumenti di elaborazione del testo che sono usati molto spesso in un sistema *nix.

- Senza l'uso di alcuna espressione regolare:
  - cat(1) concatena file e produce in output il contenuto totale.
  - tac(1) concatena file e produce l'output in senso inverso.
  - cut(1) seleziona parti di righe e le produce in output.
  - head(1) produce in output la prima parte di file.
  - tail(1) produce in output l'ultima parte di file.
  - sort(1) ordina righe in file di testo.
  - uniq(1) rimuove righe duplicate da un file ordinato.
  - tr(1) trasforma o elimina caratteri.
  - diff(1) paragona file riga per riga.
- Con espressioni regolari di base (**BRE**) viene usato in modo predefinito:
  - ed(1) è un primitivo editor di riga.
  - sed(1) è un editor di flussi.
  - grep(1) trova corrispondenze di testo con modelli.
  - vim(1) è un editor a schermo.
  - emacs(1) è un editor a schermo (con **BRE** un po' estese)
- Con l'uso di espressioni regolari estese (**ERE**):
  - awk(1) fa semplici elaborazioni di testo.
  - egrep(1) trova corrispondenze di testo con modelli.
  - `tcl`(3tcl) può fare ogni elaborazione di testo immaginabile: vedere re_syntax(3). Spesso usato con `tk`(3tk).
  - perl(1) può fare ogni elaborazione di testo immaginabile. Vedere perlre(1).
  - pcre2grep(1) dal pacchetto `pcre2-util` trova corrispondenze di testo con modelli PCRE (Perl Compatible Regular Expressions).
  - python(1) con il modulo `re` può fare ogni elaborazione di testo immaginabile. Vedere "`/usr/share/doc/python/htm`

Se non si è sicuri di cosa facciano esattamente questi comandi, usare "`man comando`" per scoprirlo.

Nota
Il criterio di ordinamento e le espressioni di intervalli dipendono dalla localizzazione. Se si desidera ottenere il comportamento tradizionale di un comando, usare la localizzazione **C** o **C.UTF-8** invece delle normali **UTF-8** (vedere Sezione 8.1).

Nota
Le espressioni regolari Perl (perlre(1)), le PCRE (Perl Compatible Regular Expressions) e le espressioni regolari Python fornite dal modulo `re` hanno molte estensioni comuni delle normali **ERE**.

1.6.2 Espressioni regolari

Le espressioni regolari sono usate in molti strumenti di elaborazione del testo. Sono analoghe ai modelli di glob della shell, ma sono più complesse e potenti.

Una espressione regolare descrive il modello a cui trovare corrispondenza ed è composta da caratteri di testo e **metacaratteri**.

Un **metacarattere** è semplicemente un carattere con un significato speciale. Ci sono 2 stili principali, **BRE** e **ERE**, a seconda degli strumenti di testo descritti in precedenza.

BRE	**ERE**	**descrizione della espressione regolare**
`\ . [ ] ^ $ *`	`\ . [ ] ^ $ *`	**metacaratteri** comuni
`\+ \? \( \) \{ \} |`		**metacaratteri** protetti dal carattere "\" di escape solo di BRE
	`+ ? ( ) { } |`	**metacaratteri** non protetti dal carattere "\" di escape solo di ERE
c	c	corrisponde al **non metacarattere** "c"
`\c`	`\c`	corrisponde al carattere letterale "c" anche se "c" è in sé stesso un **metacarattere**
.	.	corrisponde a qualsiasi carattere incluso il carattere di a capo
^	^	posizione all'inizio di una stringa
$	$	posizione alla fine di una stringa
`\<`	`\<`	posizione all'inizio di una parola
`\>`	`\>`	posizione alla fine di una parola
`[abc…]`	`[abc…]`	corrisponde ad ogni carattere in "`abc...`"
`[^abc…]`	`[^abc…]`	corrisponde ad ogni carattere eccetto a quelli in "`abc...`"
`r*`	`r*`	corrisponde a zero o più occorrenze della espressione regolare identificata da "`r`"
`r\+`	`r+`	corrisponde ad una o più occorrenza della espressione regolare identificata da "`r`"
`r\?`	`r?`	corrisponde a zero o una occorrenza dell'espressione regolare identificata da "`r`"
`r1|r2`	`r1|r2`	corrisponde ad una occorrenza della espressione regolare identificata da "`r1`" o "`r2`"
`\(r1|r2\)`	`(r1|r2)`	corrisponde ad una occorrenza dell'espressione regolare identificata da "`r1`" o "`r2`" e la tratta come una espressione regolare **tra parentesi**

Tabella 1.25: Metacaratteri per BRE e ERE

Le espressioni regolari di **`emacs`** sono fondamentalmente **BRE**, ma sono state estese per trattare "+" e "?" come i **metacaratteri** nelle **ERE**. Perciò, nelle espressioni regolari di `emacs`, non è necessario proteggerli con il carattere di escape "\".

Per effettuare ricerche di testo usando espressioni regolari si può usare grep(1).

Per esempio, provare a fare quanto segue.

```
$ egrep 'GNU.*LICENSE|Yoyodyne' /usr/share/common-licenses/GPL
GNU GENERAL PUBLIC LICENSE
GNU GENERAL PUBLIC LICENSE
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
```

Suggerimento
Vedere Sezione 9.3.6.

1.6.3 Espressioni di sostituzione

Per le espressioni di sostituzione alcuni caratteri hanno un significato particolare.

espressione di sostituzione	descrizione del testo che sostituirà l'espressione di sostituzione
`&`	ciò che ha ha avuto corrispondenza con l'espressione regolare (usare `\&` in `emacs`)
`\n`	ciò che ha avuto corrispondenza con la n-esima espressione regolare **tra parentesi** ("n" è un numero)

Tabella 1.26: L'espressione di sostituzione

Per le stringhe di sostituzione Perl viene usato «$&» invece di «&» e «$n» invece di «\n».

Per esempio, provare a fare quanto segue.

```
$ echo zzz1abc2efg3hij4 | \
sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/=&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
sed -E -e 's/(1[a-z]*)[0-9]*(.*)$/=&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
perl -pe 's/(1[a-z]*)[0-9]*(.*)$/=$&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/\2===\1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
sed -E -e 's/(1[a-z]*)[0-9]*(.*)$/\2===\1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
perl -pe 's/(1[a-z]*)[0-9]*(.*)$/$2===$1/'
zzzefg3hij4===1abc
```

In questo esempio si faccia particolare attenzione allo stile delle espressioni regolari **tra parentesi** e come le stringhe corrispondenti siano usate nel processo di sostituzione del testo dai diversi strumenti.

Queste espressioni regolari possono essere usate per spostare il cursore e per fare sostituzioni di testo anche in alcuni editor.

La barra inversa, "\" alla fine della prima riga nella riga di comando di shell protegge il carattere di a capo rendendolo uno spazio e continuando l'input della riga di comando di shell nella riga successiva.

Leggere tutte le pagine man relative per imparare l'uso di questi comandi.

1.6.4 Sostituzione globale con espressioni regolari

Il comando ed(1) può sostituire tutte le occorrenze di "`DA_REGEX`" con "`A_TESTO`" in "`file`".

```
$ ed file <<EOF
,s/FROM_REGEX/TO_TEXT/g
w
q
EOF
```

Il comando sed(1) può sostituire tutte le occorrenze di ”DA_REGEX” con ”A_TESTO” in ”file”.

```
$ sed -i -e 's/FROM_REGEX/TO_TEXT/g' file
```

Il comando vim(1) può sostituire tutte le occorrenze di ”DA_REGEX” with ”A_TESTO” in ”file” usando il comando ex(1).

```
$ vim '+%s/FROM_REGEX/TO_TEXT/gc' '+update' '+q' file
```

Suggerimento
Il modificatore ”c” nell’esempio soprastante assicura che venga chiesta la conferma interattiva per ciascuna sostituzione.

Si possono elaborare file multipli (”file1”, ”file2” e ”file3”) in modo simile con vim(1) o perl(1).

```
$ vim '+argdo %s/FROM_REGEX/TO_TEXT/gce|update' '+q' file1 file2 file3
```

Suggerimento
Il modificatore ”e” nell’esempio soprastante evita che un errore di nessuna corrispondenza trovata (”No match”) rompa il processo.

```
$ perl -i -p -e 's/FROM_REGEX/TO_TEXT/g;' file1 file2 file3
```

Nell’esempio perl(1), ”-i” è per la modificain-situ di ciascun file interessato, e ”-p” è perreiterare implicitamente il comando su tutti i file specificati.

Suggerimento
L’uso dell’opzione ”-i.bak” invece di ”-i” preserva una copia del file originale aggiungendo il suffisso ”.bak” al suo nome file. Questo permette di ripristinare le cose più facilmente in caso di errori in sostituzioni complesse.

Nota
ed(1) e vim(1) usano **BRE**; perl(1) usa **ERE**.

1.6.5 Estrarre dati da file con tabelle di testo

Considerare per esempio un file ti testo chiamato ”DPL” in cui sono elencati, in un formato separato da spazi, alcuni nomi di leader del progetto Debian pre-2004 e la data della loro installazione.

```
Ian     Murdock   August  1993
Bruce   Perens    April   1996
Ian     Jackson   January 1998
Wichert Akkerman  January 1999
Ben     Collins   April   2001
Bdale   Garbee    April   2002
Martin  Michlmayr March   2003
```

Suggerimento
Vedere ”Una breve storia di Debian” per la storia della Guida del progetto Debian aggiornata.

Awk viene usato spesso per estrarre dati da questo tipo di file.

Per esempio, provare a fare quanto segue.

```
$ awk '{ print $3 }' <DPL                   # month started
August
April
January
January
April
April
March
$ awk '($1=="Ian") { print }' <DPL          # DPL called Ian
Ian     Murdock   August  1993
Ian     Jackson   January 1998
$ awk '($2=="Perens") { print $3,$4 }' <DPL # When Perens started
April 1996
```

Questo tipo di file può anche essere analizzato con le shell tipo Bash.

Per esempio, provare a fare quanto segue.

```
$ while read first last month year; do
    echo $month
  done <DPL
... same output as the first Awk example
```

In questo caso, il comando incorporato `read` usa i caratteri in "`$IFS`" (internal field separator, separatore di campi interno) per suddividere le righe in parole.

Se si modifica "`$IFS`" in "`:`", si possono analizzare facilmente con la shell i file "`/etc/passwd`".

```
$ oldIFS="$IFS"   # save old value
$ IFS=':'
$ while read user password uid gid rest_of_line; do
    if [ "$user" = "bozo" ]; then
      echo "$user's ID is $uid"
    fi
  done < /etc/passwd
bozo's ID is 1000
$ IFS="$oldIFS"   # restore old value
```

(Se si usa Awk per fare la stessa cosa, usare "`FS=':'`" per impostare il separatore di campi.)

IFS viene anche usato dalla shell per suddividere i risultati dell'espansione di parametri, sostituzione di comandi ed espansioni aritmetiche. Queste non avvengono all'interno delle virgolette singole o doppie. Il valore predefinito di IFS è la combinazione di *spazio*, *tab* e *a capo*.

Essere prudenti nell'uso di questi trucchetti di shell con IFS. Possono accadere strane cose quando la shell interpreta alcune parti dello script come suo **input**.

```
$ IFS=":,"                        # use ":" and "," as IFS
$ echo IFS=$IFS,   IFS="$IFS"     # echo is a Bash builtin
IFS=  , IFS=:,
$ date -R                         # just a command output
Sat, 23 Aug 2003 08:30:15 +0200
$ echo $(date -R)                 # sub shell --> input to main shell
Sat  23 Aug 2003 08 30 36 +0200
$ unset IFS                       # reset IFS to the default
$ echo $(date -R)
Sat, 23 Aug 2003 08:30:50 +0200
```

1.6.6 Frammenti di script per comandi con pipe

Gli script seguenti fanno alcune cose carine se inseriti in una pipe.

frammento di script (digitarlo in un'unica riga)	**effetto del comando**
`find /usr -print`	trova tutti i file in "`/usr`"
`seq 1 100`	stampa da 1 a 100
`| xargs -n 1 comando`	esegue ripetutamente il comando con ogni elemento dalla pipe come argomento
`| xargs -n 1 echo`	suddivide elementi separati da spazio nella pipe su righe distinte
`| xargs echo`	unisce tutte le righe dalla pipe in una riga
`| grep -e modello_regex`	estrae dalla pipe le righe che contengono *modello_regex*
`| grep -v -e modello_regex`	estrae dalla pipe le righe che non contengono *modello_regex*
`| cut -d: -f3 -`	estrae dalla il terzo campo, separato da "`:`" (file passwd, ecc.)
`| awk '{ print $3 }'`	estrae dalla pipe il terzo campo separato da spazi
`| awk -F'\t' '{ print $3 }'`	estrae dalla pipe il terzo campo separato da tabulazioni
`| col -bx`	rimuove i caratteri backspace ed espande le tabulazioni in sequenze di spazi
`| expand -`	espande le tabulazioni
`| sort| uniq`	ordina e rimuove i duplicati
`| tr 'A-Z' 'a-z'`	converte le maiuscole in minuscole
`| tr -d '\n'`	concatena le righe in un'unica riga
`| tr -d '\r'`	rimuove i caratteri di a capo
`| sed 's/^/# /'`	aggiunge "#" all'inizio di ogni riga
`| sed 's/\.ext//g'`	rimuove "`.ext`"
`| sed -n -e 2p`	stampa la seconda riga
`| head -n 2 -`	stampa le prime due righe
`| tail -n 2 -`	stampa le ultime due righe

Tabella 1.27: Elenco di frammenti di script per comandi con pipe

Uno script di shell di una riga può operare reiteratamente su più file usando find(1) e xargs(1) per fare compiti piuttosto complicati. Vedere Sezione 10.1.5 e Sezione 9.4.9.

Quando usare la shell in modalità interattiva diventa troppo complicato, considerare la scrittura di uno script di shell (vedere Sezione 12.1).

Capitolo 2

Gestione dei pacchetti in Debian

Nota
Questo capitolo è stato scritto assumendo che il rilascio stabile più recente abbia nome in codice: `trixie`.
Ci si riferisce alle fonti di dati per il sistema APT collettivamente con il nome **elenco delle fonti** in questo documento . Possono essere definiti in qualsiasi tra il file "`/etc/apt/sources.list`", i file "`/etc/apt/sources.list.d/*.list`" o i file "`/etc/apt/sources.list.d/*.sources`".

2.1 Prerequisiti per la gestione dei pacchetti Debian

2.1.1 Sistema di gestione dei pacchetti in Debian

Debian è un'organizzazione di volontari che crea distribuzioni **coerenti** di pacchetti binari precompilati di software libero e li distribuisce dal suo archivio.

L'archivio Debian viene fornito da molti siti mirror remoti per l'accesso con i metodi HTTP e FTP. È anche disponibile come CD-ROM/DVD.

L'attuale sistema di gestione dei pacchetti di Debian che può utilizzare tutte queste risorse è APT (Advanced Packaging Tool).

Il sistema di gestione dei pacchetti di Debian, **se usato in modo corretto**, permette all'utente di installare dall'archivio **insiemi coerenti di pacchetti binari** sul sistema. Attualmente ci sono 78072 pacchetti disponibili per l'architettura amd64.

Il sistema di gestione dei pacchetti di Debian ha una storia ricca e fornisce molti programmi con interfacce per l'utente finale e molti metodi di backend per l'accesso agli archivi tra cui scegliere. Attualmente le scelte raccomandate sono le seguenti.

- apt(8) per tutte le operazioni interattive dalla riga di comando, incluse le installazioni e le rimozioni di pacchetti nonché gli aggiornamenti dist-upgrade.
- apt-get(8) per chiamare il sistema di gestione dei pacchetti Debian da script. È anche un'opzione di ripiego quando non è disponibile `apt` (spesso in sistemi Debian più vecchi).
- aptitude(8) per un'interfaccia testuale interattiva per gestire i pacchetti installati e per cercare i pacchetti disponibili.

pacchetto	popcon	dimensione	descrizione
dpkg	V:906, I:1000	4944	sistema di gestione dei pacchetti a basso livello per Debian (basato su file)
apt	V:895, I:1000	4956	front-end APT per gestire i pacchetti con la CLI: apt/apt-get/apt-cache
aptitude	V:31, I:160	4621	front-end APT per gestire i pacchetti interattivamente con console a tutto schermo: aptitude(8)
tasksel	V:38, I:985	351	front-end APT per installare attività (task) selezionate: tasksel(8)
unattended-upgrades	V:130, I:182	320	pacchetto di estensione per APT per abilitare l'installazione automatica degli aggiornamenti di sicurezza
gnome-software	V:158, I:263	4732	Centro del Software per GNOME (interfaccia con GUI per APT)
synaptic	V:31, I:269	7264	gestore grafico dei pacchetti (front-end GTK per APT)
apt-utils	V:420, I:998	1181	programmi di utilità APT: apt-extracttemplates(1), apt-ftparchive(1) e apt-sortpkgs(1)
apt-listchanges	V:403, I:894	562	strumento di notifica dello storico dei cambiamenti dei pacchetti
apt-listbugs	V:4.7, I:6.5	512	elenca i bug critici prima di ogni installazione di APT
apt-file	V:14, I:53	76	utilità APT per ricerca di pacchetti, interfaccia a riga di comando
apt-rdepends	V:0.4, I:4.2	39	elenca le dipendenze dei pacchetti in modo ricorsivo

Tabella 2.1: Elenco degli strumenti Debian di gestione dei pacchetti

2.1.2 Configurazione dei pacchetti

Questi sono alcuni punti fondamentali per la configurazione dei pacchetti in un sistema Debian.

- Per un ambiente desktop moderno, è una buona idea riavviare il sistema dopo una modifica alla configurazione di un pacchetto e l'aggiornamento di un pacchetto, per assicurare il corretto funzionamento del sistema.
- La configurazione manuale fatta dall'amministratore di sistema viene rispettata. In altre parole, il sistema di configurazione dei pacchetti non fa configurazioni invadenti per ragioni di comodità.
- Ogni pacchetto viene fornito con un proprio script di configurazione per l'interfaccia utente standard debconf(7), per aiutare nel processo di configurazione iniziale del pacchetto.
- I Debian Developer cercano di fare del loro meglio per rendere l'aggiornamento un'esperienza senza problemi grazie agli script di configurazione dei pacchetti.
- Sono disponibili per l'amministratore di sistema le complete funzionalità di un pacchetto software, ma quelle con rischi per la sicurezza sono disabilitate nella installazione predefinita.
- Se si attiva manualmente un servizio con rischi per la sicurezza, si è responsabili del contenimento del rischio.
- L'amministratore di sistema può abilitare manualmente configurazioni esotiche; questo può creare interferenze con popolari programmi generici di supporto per la configurazione del sistema.

2.1.3 Precauzioni base



avvertimento
Non installare pacchetti da una miscela casuale di suite. Probabilmente si disgrega la coerenza tra i pacchetti che richiede una conoscenza approfondita dalla gestione del sistema, come dell'ABI del compilatore, versione delle librerie, funzionalità dell'interprete, ecc.

L'amministratore di sistema Debian novizio dovrebbe rimanere con il rilascio **`stable`** di Debian ed applicare solo gli aggiornamenti di sicurezza. Fino a che non si capisce il sistema Debian molto bene, si dovrebbero seguire le precauzioni descritte in seguito.

- Non includere **`testing`** o **`unstable`** nell'"**elenco delle fonti**".
- Non mescolare gli archivi standard Debian con altri archivi non Debian, come quelli di Ubuntu, nell'"**elenco delle fonti** .
- Non creare il file "`/etc/apt/preferences`".
- Non cambiare il comportamento degli strumenti di gestione dei pacchetti attraverso i loro file di configurazione senza capire a pieno il loro effetto.
- Non installare pacchetti presi qua e là con "`dpkg -i` *`pacchetto_a_caso`*".
- Non installare mai pacchetti presi qua e là con "`dpkg --force-all -i` *`pacchetto_a_caso`*".
- Non cancellare o alterare i file in "`/var/lib/dpkg/`".
- Non sovrascrivere file di sistema, installando direttamente programmi software compilati dai sorgenti.
  - Se necessario, installarli in "`/usr/local`" o "`/opt`".

Gli effetti di incompatibilità, causati dalla violazione delle precauzioni elencate sopra, sul sistema di gestione dei pacchetti di Debian possono lasciare il sistema in uno stato inutilizzabile.

Gli amministratori di sistema Debian seri, che gestiscono server di importanza critica, dovrebbero usare particolari precauzioni.

- Non installare alcun pacchetto, compresi quelli di aggiornamenti di sicurezza di Debian, senza averli attentamente testati con la propria particolare configurazione in condizioni sicure.
  - Alla fin fine è l'amministratore di sistema ad essere responsabile del proprio sistema.
  - La lunga tradizione di stabilità dei sistemi Debian non è di per sé una garanzia.

2.1.4 Una vita di aggiornamenti senza fine



Attenzione
Per i **server di produzione**, è raccomandata la suite `stable` con gli aggiornamenti di sicurezza. Lo stesso vale per i PC desktop su cui si investono sforzi di amministrazione limitati.

Nonostante gli avvertimenti descritti prima, si sa che molti lettori di questo documento possono volere eseguire le più nuove suite `testing` o `unstable`.

L'illuminazione data dalla lettura di ciò che segue salva l'utente dall'eterna lotta karmica con l'inferno degli aggiornamenti e gli fa raggiungere il nirvana Debian.

Questa lista è pensata per l'ambiente Desktop **amministrato da soli**.

- Usare la suite `testing` dato che è in pratica il rilascio in progresso gestito automagicamente dall'infrastruttura di QA dell'archivio Debian come la integrazione continua di Debian, le pratiche di caricamento dei soli sorgenti e il tracciamento delle transizioni delle librerie. I pacchetti nella suite `testing` vengono aggiornati sufficientemente di frequente da fornire tutte le funzionalità più recenti.
- Impostare il nome in codice corrispondente alla suite `testing` ("`forky`" durante il ciclo di rilascio `trixie-as-stable`) nell'**elenco delle fonti**.

- Aggiornare manualmente questo nome in codice nell'"**elenco delle fonti** al nuovo solo dopo aver valutato autonomamente la situazione per circa un mese dopo un rilascio maggiore della suite. Anche le mailing list degli utenti e degli sviluppatori Debian sono una buona fonte di informazioni su questi aspetti.

L'uso della suite `unstable` non è raccomandato. La suite `unstable` è **buona per fare il debug dei pacchetti** come sviluppatore ma tende ad esporre l'utente a rischi non necessari per il normale uso per desktop. Anche se la suite `unstable` del sistema Debian sembra molto stabile per la maggior parte del tempo, ci sono stati alcuni problemi con i pacchetti e alcuni di questi non sono stati affatto banali da risolvere.

Ecco alcune idee per misure precauzionali di base per assicurare un recupero facile e veloce da bug nei pacchetti Debian.

- Rendere il sistema di ripristino/recupero disponibile seguendo Sezione 3.2.
- Impostare il sistema per avere **due partizioni avviabili** installando la suite `stable` di Debian in un'altra partizione
- Considerare l'installazione di `apt-listbugs` per controllare le informazioni del Sistema Debian di tracciamento dei Bug (BTS) prima degli aggiornamenti
- Imparare l'infrastruttura del sistema dei pacchetti abbastanza bene da poter aggirare il problema.



Attenzione
Se non si è in grado di intraprendere nessuna di queste azioni precauzionali, probabilmente non si è pronti per le suite `testing` e `unstable`.

2.1.5 Nozioni di base sugli archivi Debian

Suggerimento
La politica ufficiale per gli archivi Debian è definita nel Capitolo 2 - "L'archivo Debian", del manuale Debian Policy.

Guardiamo l'archivio Debian dalla prospettiva dell'utente.

Per un utente di sistema, l'accesso all'archivio Debian avviene usando il sistema APT.

Il sistema APT specifica le proprie fonti di dati come **elenco delle fonti** descritto in sources.list(5).

Per il sistema `trixie` con il tipico accesso HTTP, l'**elenco delle fonti** è fornito nel moderno stile deb822 in "`/etc/apt/sour` come il seguente:

```
Types: deb deb-src
URIs: http://deb.debian.org/debian/
Suites: trixie trixie-updates
Components: main non-free-firmware contrib non-free

Types: deb deb-src
URIs: http://security.debian.org/debian-security/
Suites: trixie-security
Components: main non-free-firmware contrib non-free
```

Suggerimento
Se **l'elenco delle fonti** è fornito nel vecchio stile a singola riga in file "`/etc/apt/sources.list`" o "`/etc/apt/sources.list.d/*.list`", aggiornarli con:

```
$ sudo apt modernize-sources
```

I punti chiave dell'**elenco delle fonti** in stile deb822 sono i seguenti.

- I suoi file di definizione sono in "`/etc/apt/sources.list.d/*.sources`".
- Ogni blocco di righe separato da una riga vuota definisce una fonte di dati per il sistema APT.
- La sezione "`Types:`" definisce la lista dei tipi, come "`deb`" e "`deb-src`".
- La sezione "`URIs:`" definisce la lista degli URI radice dell'archivio Debian.
- La sezione "`Suites:`" definisce la lista dei nomi delle distribuzioni usando il nome della suite o il nome in codice.
- La sezione "`Components:`" definisce la lista dei nomi delle aree valide dell'archivio Debian.

La definizione per "`deb-src`" può essere omessa senza problemi se è solo per aptitude `aptitude` che non accede ai metadati relativi ai sorgenti. Ciò velocizza l'aggiornamento dei metadati dell'archivio.

L'URL può essere "`https://`", "`http://`", "`ftp://`", "`file://`", ...

Le righe che iniziano con "#" sono commenti e vengono ignorate.

Viene qui usato il nome in codice "`trixie` o "`forky`" invece del nome della suite "`stable`" o "`testing`" per evitare sorprese una volta che viene rilasciata la nuova `stable`.

Suggerimento
Se nell'esempio precedente si usa "`sid`" invece di "`trixie`, la riga "`deb: http://security.debian.org/` ..., o il suo equivalente deb822 per gli aggiornamenti di sicurezza, nell'**elenco delle fonti** non è richiesta. Ciò è dovuto al fatto che non esiste un archivio per gli aggiornamenti di sicurezza per "`sid`" (`unstable`).

Ecco l'elenco degli URL dei siti degli archivi Debian e dei nomi di suite e nomi in codice usati nel file di configurazione dopo il rilascio di `trixie`.

URL dell'archivio	nome della suite	nome in codice	scopo del repository
http://deb.debian.org/debian/	`stable`	`trixie`	Rilascio `stabile` quasi statico dopo controlli esaustivi
http://deb.debian.org/debian/	`stable-updates`	`trixie-updates`	Updates for the `stable` suite (important)
http://security.debian.org/debian-security/	`stable-security`	`trixie-security`	Security updates for the `stable` suite (important)
http://deb.debian.org/debian/	`testing`	`forky`	Rilascio `di test` dinamico dopo controlli ragionevoli e brevi tempi di attesa
http://deb.debian.org/debian/	`unstable`	`sid`	Rilascio non `stabile` dinamico dopo controlli minimi e nessun tempo di attesa
http://deb.debian.org/debian/	`experimental`	N/D	Pre-rilasci sperimentali degli sviluppatori (opzionale, solo per sviluppatori)
http://deb.debian.org/debian/	`stable-proposed-updates`	`trixie-proposed-updates`	Aggiornamenti per il prossimo rilascio `stabile` minore (opzionale)
http://deb.debian.org/debian/	`stable-backports`	`trixie-backports`	Raccolta varia di pacchetti ricompilati per lo più dal rilascio `di test` (opzionale)
http://security.debian.org/debian-security/	`testing-security`	`forky-security`	Questo non è attivamente supportato, né usato dal Team per la sicurezza

Tabella 2.2: Elenco dei siti con l'archivo Debian



Attenzione
Solo il rilascio **`stable`** puro con gli aggiornamenti di sicurezza fornisce la massima stabilità. Usare un rilascio **`stable`** con mescolati alcuni pacchetti dai rilasci **`testing`** o **`unstable`** è più rischioso che avere un sistema basato su un rilascio **`unstable`** puro per ciò che riguarda conflitti tra le versioni di libreria, ecc. Se si ha veramente bisogno, nel rilascio **`stable`**, dell'ultima versione di qualche programma usare pacchetti dai servizi stable-updates e backports (vedere Sezione 2.7.4). Questi servizi devono essere usati con estrema cautela.



Attenzione
Fondamentalmente si dovrebbe mettere una sola tra le suite `stable`, `testing` o `unstable` nelle righe "`deb`". Se si elenca una combinazione delle suite `stable`, `testing` e `unstable` nelle righe "`deb`", il programma APT verrà rallentato e solo l'archivio più recente avrà effetto. Elencare più di una voce ha senso quando viene usato il file "`/etc/apt/preferences`" con scopi ben precisi (vedere Sezione 2.7.7).

Suggerimento
Per i sistemi Debian con la suite `stable` è buona norma includere nell'**elenco delle fonti** i contenuti di "`http://security.debian.org/`" per abilitare gli aggiornamenti di sicurezza, come nell'esempio precedente.

Nota
I bug che riguardano la sicurezza per l'archivio `stable` vengono risolti dal Debian Security Team, la cui attività è piuttosto rigorosa ed affidabile. I bug per l'archivio `testing` possono essere risolti dal Debian Security Team. Per svariate ragioni questa attività non è così rigorosa come quella per `stable` e potrebbe essere necessario attendere la migrazione di pacchetti `unstable` corretti nell'archivio `testing`. I bug per l'archivio `unstable` sono corretti dal manutentore del pacchetto; i pacchetti `unstable` attivamente mantenuti sono solitamente in una forma piuttosto buona dato che sfruttano le più recenti correzioni a monte relative alla sicurezza. Per informazioni sul modo in cui Debian gestisce i bug relativi alla sicurezza, vedere le FAQ Debian sulla sicurezza.

area	numero di pacchetti	criterio dei componenti del pacchetto
`main`	76587	aderenti alle DFSG e senza dipendenze da `non-free`
`non-free-firmware`	78	firmware non conforme alle DFSG, necessario per una esperienza di installazione del sistema ragionevole
`contrib`	362	aderenti alle DFSG ma con dipendenze da `non-free`
`non-free`	1045	non aderenti alle DFSG e non in `non-free-firmware`

Tabella 2.3: Elenco delle aree dell'archivio Debian

Nella tabella soprastante il numero dei pacchetti è per l'architettura amd64. L'area `main` fornisce il sistema Debian (vedere Sezione 2.1.6).

L'organizzazione dell'archivio Debian può essere studiata meglio puntando il proprio browser a ciascuno degli URL dell'archivio con aggiunto in coda `dists` o `pool`.

Ci si riferisce alla distribuzione in due modi, con la suite o con il nome in codice. In alternativa la parola distribuzione viene usata come sinonimo di suite in molta documentazione. La relazione tra la suite e il nome in codice può essere riassunta nel modo seguente.

La storia dei nomi in codice è descritta nelle FAQ Debian: 6.2.1 Quali altri nomi in codice sono stati usati in passato?

Nella terminologia per gli archivi Debian più precisa, la parola "sezione" è usata specificatamente per la categorizzazione dei pacchetti in base all'area di applicazione. (Anche se l'espressione "sezione main" può essere a volte usata per descrivere l'area dell'archivio Debian chiamata "main".)

periodo	suite = stable	suite = testing	suite = unstable
dopo il rilascio trixie	nome in codice = trixie	nome in codice = forky	nome in codice = sid
dopo il rilascio forky	nome in codice = forky	nome in codice = duke	nome in codice = sid

Tabella 2.4: Relazione tra suite e nome in codice

Ogni volta che uno sviluppatore Debian (DD) carica un nuovo pacchetto nell'archivio unstable (passando per incoming), gli viene richiesto di assicurare che i pacchetti caricati siano compatibili con l'insieme più recente di pacchetti nell'archivio unstable più recente.

Se un DD rompe questa compatibilità intenzionalmente per importanti aggiornamenti di librerie, ecc. di solito viene fatto un annuncio nella mailing list debian-devel, ecc.

Prima di muovere un insieme di pacchetti dall'archivio unstable all'archivio testing, lo script di gestione degli archivio Debian non solo controlla la maturità (circa 2-10 giorni di età) e lo stato delle segnalazioni di bug RC per i pacchetti, ma cerca anche di assicurare che siano compatibili con il più recente insieme di pacchetti nell'archivio testing. Questo processo rende l'archivio testing molto aggiornato e usabile.

Attraverso il graduale processo di freeze dell'archivio, giudato dal team di rilascio, l'archivio testing viene fatto maturare con un po' di intervento manuale per renderlo completamente coerente e libero da bug. Quindi viene creato il nuovo rilascio stable assegnando il nome in codice per il vecchio archivio testing al nuovo archivio stable e creando il nuovo nome in codice per il nuovo archivio testing. Il contenuto iniziale del nuovo archivio testing è esattamente lo stesso dell'archivio stable appena rilasciato.

Sia l'archivio unstable sia l'archivio testing possono soffrire di problemi temporanei a causa di diversi fattori.

- Pacchetti difettosi caricati nell'archivio (principalmente per unstable)
- Ritardo nell'accettazione di nuovi pacchetti nell'archivio (principalmente per unstable)
- Problemi nei tempi di sincronizzazione degli archivi (sia per testing sia per unstable)
- Interventi manuali all'archivio, come rimozione di pacchetti (più per testing) ecc.

Perciò se si decide di usare questi archivi, si dovrebbe essere in grado di risolvere o aggirare questo tipo di problemi.



Attenzione
Per qualche mese circa dopo un nuovo rilascio di stable, la maggior parte degli utenti desktop dovrebbe usare l'archivio stable con i suoi aggiornamenti di sicurezza, anche se di solito usano gli archivi unstable o testing. Durante questo periodo di transizione, entrambi gli archivi unstable e testing non sono adatti alla maggior parte degli utenti. È difficile mantenere il proprio sistema in condizioni buone di funzionamento con l'archivio unstable dato che è affetto da ondate di aggiornamenti importanti di pacchetti fondamentali. Anche l'archivio testing non è molto utile perché contiene per lo più le stesse cose dell'archivio stable senza il suo supporto per la sicurezza (Debian testing-security-announce 2008-12). Dopo circa un mese, gli archivio unstable o testing potrebbero essere utili se si è cauti.

Suggerimento
Quando si usa l'archivio testing, i problemi causati da un pacchetto rimosso vengono solitamente aggirati installando il corrispondente pacchetto dall'archivio unstable che è caricato per risolvere un bug.

Vedere il manuale Debian Policy per le definizioni degli archivi.

- "Sezioni"
- "Priorità"
- "Sistema base"
- "Pacchetti essenziali"

2.1.6 Debian è al 100% software libero

Debian è al 100% software libero perché:

- Debian installa in modo predefinito solo software libero per rispettare le libertà dell'utente.
- Debian fornisce in `main` solo software libero.
- Debian raccomanda l'esecuzione del solo software libero contenuto in `main`.
- Nessun pacchetto in `main` dipende o raccomanda pacchetti in `non-free`, `non-free-firmware` o `contrib`.

Alcune persone si chiedono se i seguenti due fatti siano in contraddizione o meno tra loro.

- «Debian rimarrà libera al 100%». (Prima voce del Contratto sociale Debian)
- I server Debian ospitano alcuni pacchetti `non-free-firmware`, `non-free` e `contrib`.

Queste due cose non sono in contraddizione, perché:

- Il sistema Debian è libero al 100% e i suoi pacchetti sono ospitati dai server Debian nell'area `main`.
- I pacchetti esterni al sistema Debian sono ospitati dai server Debian nelle aree `non-free`, `non-free-firmware` e `contrib`.

Queste sono spiegate in dettaglio nella quarta e nella quinta voce del Contratto sociale Debian:

- Le nostre priorità sono gli utenti ed il software libero
  - Ci faremo guidare dai bisogni dei nostri utenti e della comunità del software libero. Metteremo al primo posto i loro interessi. Supporteremo le necessità dei nostri utenti di operare in molti diversi tipi di ambienti di calcolo. Non ci opporremo alle opere non libere che siano state pensate per l'uso in sistemi Debian e non richiederemo compensi a chi crea o usa queste opere. Permetteremo ad altri di creare distribuzioni contenenti sia il sistema Debian che altre opere, senza richiedere compensi. Per raggiungere questi scopi, forniremo un sistema integrato di materiali di alta qualità senza alcuna restrizione legale che limiti qualsiasi uso del sistema.
- Opere che non rispettano i nostri standard free software
  - Ci rendiamo conto che alcuni dei nostri utenti richiedono di usare opere non conformi alle Debian Free Software Guidelines. Abbiamo creato le aree "`non-free`", "`non-free-firmware`" e "`contrib`" nel nostro archivio per queste opere. I pacchetti in queste aree non fanno parte del sistema Debian, sebbene siano stati configurati per l'uso con Debian. Invitiamo i realizzatori di CD a leggere le licenze dei pacchetti in queste aree per determinare se possono distribuire i pacchetti sui loro CD. Inoltre, anche se le opere non libere non fanno parte di Debian, supporteremo il loro uso e forniremo infrastrutture per i pacchetti non liberi (come il nostro sistema di tracciamento dei bug e le mailing list). I supporti Debian ufficiali possono includere firmware che non fa parte del sistema Debian per permettere l'uso di Debian con hardware che richiede tale firmware.

Nota
Il testo effitivo della quinta voce dell'attuale Contratto sociale di Debian 1.2 è leggermente diverso dal testo soprastante. Questa deviazione editoriale è intenzionale e volta a rendere questo documento coerente senza cambiare l'effettivo contenuto del Contratto Sociale.

Gli utenti dovrebbero essere a conoscenza dei rischi correlati all'uso di pacchetti nelle aree `non-free`, `non-free-firmwar` e `contrib`:

- mancanza di libertà legata a tali pacchetti software

- mancanza di supporto da parte di Debian per tali pacchetti software (Debian non può supportare del software in maniera adeguata senza avere accesso al suo codice sorgente)
- contaminazione del sistema Debian libero al 100%

Il documento Debian Free Software Guidelines è lo standard di Debian per il software libero. Debian interpreta la parola ”software” nel suo senso più ampio includendo documentazione, firmware, logo e dati artistici nel pacchetto. Questo fa sì che gli standard Debian per il software libero siano molto severi.

I pacchetti `non-free`, `non-free-firmware` e `contrib` tipici includono pacchetti liberamente distribuibili dei seguenti tipi:

- Pacchetti di documentazione distribuiti sotto la GNU Free Documentation License con sezioni invarianti come quelli per GCC e Make (per lo più presenti nella sezione `non-free/doc`).
- Pacchetti di firmware contenenti dati binari senza sorgenti come quelli elencati in Sezione 9.10.5 come `non-free-firmwa` (per lo più trovati nella sezione `non-free-firmware/kernel`).
- Pacchetti di giochi e tipi di carattere con restrizioni sull’uso a scopo commerciale o sulla modifica dei contenuti.

Notare che il numero dei pacchetti `non-free`, `non-free-firmware` e `contrib` è meno del 2% di quello dei pacchetti in `main`. Permettere l’accesso alle aree `non-free`, `non-free-firmware` e `contrib` non oscura la fonte dei pacchetti. L’uso interattivo a schermo intero di aptitude(8) fornisce piena visibilità e controllo su quali pacchetti vengono installati e da quali aree, per permettere di mantenere il proprio sistema libero quanto lo si desidera.

2.1.7 Dipendenze dei pacchetti

Il sistema Debian offre un insieme coerente di pacchetti binari grazie al suo meccanismo di dichiarazione, nei campi di controllo dei file, delle dipendenze binarie basate su versioni . Ecco una definizione molto semplificata delle dipendenze.

- ”Depends” (Dipende)
  - Dichiara una dipendenza assoluta e tutti i pacchetti elencati in questo campo devono essere installati insieme a quello scelto o prima.
- ”Pre-Depends” (Pre-dipende)
  - È come ”Depends”, tranne che richiede l’installazione completa dei pacchetti elencati in anticipo.
- ”Recommends” (Raccomanda)
  - Dichiara una dipendenza forte, ma non assoluta. La maggior parte degli utenti non vorrà il pacchetto in esame a meno che tutti i pacchetti elencati non siano installati.
- ”Suggests” (Consiglia)
  - Dichiara una dipendenza debole. La maggior parte degli utenti del pacchetto potrebbe trarre vantaggio dall’installazione dei pacchetti elencati in questo campo, ma può ottenere una funzionalità adeguata senza di essi.
- ”Enhances” (Migliora)
  - Dichiara una dipendenza debole come ”Suggests” ma in verso opposto.
- ”Breaks” (Rompe)
  - Dichiara un’incompatibilità del pacchetto di solito con qualche specifica versione. Generalmente la soluzione è di aggiornare tutti i pacchetti elencati in questo campo.
- ”Conflicts” (Va in conflitto)

  - Dichiara una incompatibilità assoluta. Tutti i pacchetti elencati in questo campo devono essere rimossi per installare il pacchetto in esame.

- ”Replaces” (Sostituisce)
  - Viene dichiarata quando i file installati dal pacchetto in esame sostituiscono i file nei pacchetti elencati.
- ”Provides” (Fornisce)
  - Viene dichiarata quando il pacchetto fornisce tutti i file e le funzionalità nei pacchetti elencati.

Nota
Notare che una buona configurazione per un pacchetto virtuale deve avere ”Provides”, ”Conflicts” e ”Replaces” simultaneamente definiti. Questo assicura che in un dato momento possa essere installato un solo pacchetto reale che fornisce il pacchetto virtuale.

La definizione ufficiale, compresa quella di dipendenza dei sorgenti, può essere trovata nel manuale Debian Policy, Capitolo 7, Dichiarare le relazioni tra i pacchetti.

2.1.8 Il flusso di eventi nella gestione dei pacchetti

Ecco un riassunto semplificato del flusso di eventi nella gestione dei pacchetti con APT.

- **Aggiornamento informazioni** (”`apt update`”, ”`aptitude update`” o ”`apt-get update`”):
  1. Recupera i metadati dell’archivio dall’archivio remoto
  2. Ricostruisce e aggiorna i metadati locali usati da APT
- **Aggiornamento** («`apt upgrade`» e «`apt full-upgrade`» o «`aptitude safe-upgrade`» e «`aptitude full-upg`
  o «`apt-get upgrade`» e «`apt-get dist-upgrade`»):
  1. Sceglie la versione candidata, che è solitamente la più recente disponibile, per tutti i pacchetti installati (per le eccezioni vedere Sezione 2.7.7)
  2. Risolve le dipendenze dei pacchetti
  3. Se la versione candidata è diversa da quella installata, scarica i pacchetti binari selezionati dall’archivio remoto
  4. Spacchetta i pacchetti binari scaricati
  5. Esegue gli script **preinst**
  6. Installa i file binari
  7. Esegue gli script **postinst**
- **Installazione** («`apt install` …», «`aptitude install` …» o «`apt-get install` …»):
  1. Sceglie i pacchetti elencati nella riga di comando
  2. Risolve le dipendenze dei pacchetti
  3. Scarica i pacchetti binari selezionati dall’archivio remoto
  4. Spacchetta i pacchetti binari scaricati
  5. Esegue gli script **preinst**
  6. Installa i file binari
  7. Esegue gli script **postinst**
- **Rimozione** («`apt remove` …», «`aptitude remove` …» o «`apt-get remove` …»):
  1. Sceglie i pacchetti elencati nella riga di comando

2. Risolve le dipendenze dei pacchetti
3. Esegue gli script **prerm**
4. Rimuove i file installati **tranne** i file di configurazione
5. Esegue gli script **postrm**

- **Rimozione completa** («`apt purge`», «`aptitude purge` …» o «`apt-get purge` …»):

1. Sceglie i pacchetti elencati nella riga di comando
2. Risolve le dipendenze dei pacchetti
3. Esegue gli script **prerm**
4. Rimuove i file installati **inclusi** i file di configurazione
5. Esegue gli script **postrm**

Sono stati intenzionalmente omessi dettagli tecnici a favore di una più chiara vista d'insieme.

2.1.9 Prima risposta a problemi di gestione dei pacchetti

Si dovrebbe leggere la bella documentazione ufficiale. Il primo documento da leggere è il file specifico per Debian, "`/usr/share/doc/`*`nome_pacchetto`*`/README.Debian`". Andrebbe consultata anche l'altra documentazione in "`/usr/share/doc/`*`nome_pacchetto`*`/`. Se la shell è impostata come in Sezione 1.4.2, digitare quanto segue.

```
$ cd package_name
$ pager README.Debian
$ mc
```

Per informazioni dettagliate potrebbe essere necessario installare il pacchetto di documentazione corrispondente il cui nome ha il suffisso "`-doc`".

Se si stanno avendo problemi con un pacchetto specifico, ricordarsi come prima cosa di controllare il sito del Sistema Debian di tracciamento dei bug (BTS).

sito web	**comando**
Pagina web del Sistema Debian di tracciamento dei bug (BTS)	`sensible-browser "https://bugs.debian.org/"`
Segnalazioni di bug di un pacchetto specifico	`sensible-browser "https://bugs.debian.org/`*`nome_pacchetto`*`"`
Segnalazione di bug corrispondente ad un numero di bug	`sensible-browser "https://bugs.debian.org/`*`numero_bug`*`"`

Tabella 2.5: Elenco dei siti web importanti per la risoluzione di problemi con un pacchetto specifico

Cercare con Google con parole chiave e aggiungendo "`site:debian.org`", "`site:wiki.debian.org`", "`site:lists.d`" ecc.

Quando si invia una segnalazione di bug, usare il comando reportbug(1).

2.1.10 Come scegliere i pacchetti Debian

Quando si incontrano più di due pacchetti simili e non si sa quale installare senza doverli installare per prova, si dovrebbe usare un po' di **buon senso**. Considero i punti seguenti un buon consiglio per i paccchetti da preferire.

- Essenziale: sì > no

- Area: main > contrib > non-free
- Priorità: richiesto > importante > standard > opzionale > extra
- Task: pacchetti elencati in task come ”Ambiente desktop”
- Pacchetti selezionati da pacchetti di dipendenze (es., `gcc-16` selezionato da `gcc`)
- Popcon: maggior numero di voti e di installazioni
- Changelog: aggiornamenti regolari da parte del manutentore
- BTS: nessun bug RC (nessun bug critico, grave o serio)
- BTS: manutentore sensibile alle segnalazioni di bug
- BTS: maggior numero di bug risolti di recente
- BTS: minor numero di bug rimanenti non wishlist

Essendo Debian un progetto di volontari con modello di sviluppo distribuito, i suoi archivi contengono molti pacchetti con scopi e qualità molto differenti. Si deve scegliere con la propria testa che uso farne.

2.1.11 Come affrontare requisiti in conflitto

Qualsiasi sia la suite del sistema Debian che si decide di usare, si può comunque desiderare eseguire versioni di programmi che non sono disponibili in tale suite. Anche se si trovano pacchetti binari di tali programmi in altre suite Debian o in altre risorse Debian, i loro requisiti potrebbero essere in conflitto con il proprio sistema Debian attuale.

Sebbene si possa ritoccare il sistema di gestione dei pacchetti con la tecnica di **apt-pinning**, ecc., come descritta in Sezione 2.7.7 per installare tali pacchetti binari non in sincrono, questo approccio di ritocco ha casi d’uso limitati dato che può rendere non funzionanti tali programmi e il sistema.

Prima di installare brutalmente simili pacchetti non in sincrono, si dovrebbero cercare tutte le soluzioni alternative più sicure disponibili, compatibili con il proprio sistema Debian attuale.

- Installare tali programmi usando un pacchetto binario originale in una sandbox (vedere Sezione 7.7)
  - Molti programmi, per lo più con GUI, come applicazioni LibreOffice e GNOME sono disponibili come pacchetti Flatpak, Snap o AppImage.
- Creare un ambiente chroot o simile ed eseguire tali programmi in esso (vedere Sezione 9.11).
  - I comandi CLI possono essere eseguiti facilmente in chroot compatibili (vedere Sezione 9.11.4).
  - Ambienti desktop completi diversi possono essere provati facilmente senza riavviare (vedere Sezione 9.11.5).
- Compilare da soli le versioni desiderate di pacchetti binari che sono compatibili con il proprio sistema Debian attuale.
  - Questa è un’attività non banale (vedere Sezione 2.7.13).

2.2 Operazioni base per la gestione dei pacchetti

Le operazioni di gestione dei pacchetti basate sui repository sul sistema Debian possono essere eseguite da molti strumenti di gestione dei pacchetti basati su APT disponibili sul sistema Debian. Qui, verranno spiegati 3 strumenti base di gestione dei pacchetti: `apt`, `apt-get` / `apt-cache` e `aptitude`.

Per operazioni di gestione dei pacchetti che comportano l’installazione o l’aggiornamento dei metadati dei pacchetti, è necessario avere privilegi di root.

2.2.1 Confronti `apt` - `apt-get` / `apt-cache` - `aptitude`

Anche se `aptitude` è uno strumento interattivo molto bello ed è quello principalmente usato dall'autore, si dovrebbero conoscere alcuni avvertimenti:

- Il comando `aptitude` non è raccomandato per l'aggiornamento del sistema da un rilascio ad un altro sul sistema Debian `stable` dopo un nuovo rilascio.
  - Per questo è raccomandato l'uso di «`apt full-upgrade`» o «`apt-get dist-upgrade`». Vedere il Bug nr. 411280.
- Il comando `aptitude` a volte suggerisce rimozioni in massa di pacchetti per l'aggiornamento del sistema sui sistemi Debian `testing` o `unstable`.
  - Questa situazione ha spaventato molti amministratori di sistema. Niente panico.
  - Ciò sembra essere dovuto soprattutto ad una non corrispondenza tra le versioni di pacchetti da cui dipende un metapacchetto o che sono raccomandate da un metapacchetto, come `gnome-core`.
  - Ciò può essere risolto selezionando «Annulla azioni in attesa» nel menu dei comandi di `aptitude`, uscendo da `aptitude` e usando «`apt full-upgrade`».

I comandi `apt-get` e `apt-cache` sono gli strumenti di gestione dei pacchetti basati su APT più a livello **base**.

- `apt-get` e `apt-cache` offrono solo l'interfaccia utente a riga di comando.
- `apt-get` è più adatto per gli **aggiornamenti principali del sistema** tra rilasci diversi, ecc.
- `apt-get` offre un sistema **robusto** di risoluzione delle dipendenze dei pacchetti.
- `apt-get` è meno avido di risorse di sistema. Consuma meno memoria ed è più veloce durante l'esecuzione.
- `apt-cache` offre una ricerca **standard**, basata su espressioni regolari, nei nomi e nelle descrizioni dei pacchetti.
- `apt-get` e `apt-cache` possono gestire più versioni dei pacchetti usando `/etc/apt/preferences`, ma è piuttosto elaborato farlo.

Il comando `apt` è un'interfaccia a riga di comando di alto livello per la gestione dei pacchetti. Fondamentalmente è un wrapper di `apt-get`, `apt-cache` e comandi simili, originariamente pensata come interfaccia per l'utente finale e abilita in modo predefinito alcune opzioni più adatte per l'uso interattivo.

- `apt` fornisce una comoda barra di avanzamento quando si installano pacchetti usando `apt install`.
- `apt` **elimina** in modo predefinito i pacchetti `.deb` nella cache dopo l'installazione con successo dei pacchetti scaricati.

Suggerimento
È raccomandato agli utenti di usare il nuovo comando apt(8) per l'uso **interactive** e di usare i comandi apt-get(8) e apt-cache(8) negli script di shell.

Il comando `aptitude` è lo strumento di gestione dei pacchetti basato su APT più **versatile**.

- `aptitude` offre un'interfaccia utente testuale interattiva a schermo intero.
- `aptitude` offre anche un'interfaccia utente a riga di comando.
- `aptitude` è più adatto per la **gestione interattiva quotidiana dei pacchetti**, come l'ispezione dei pacchetti installati e la ricerca dei pacchetti disponibili.
- `aptitude` è più avido di risorse di sistema. Consuma più memoria ed è più lento durante l'esecuzione.
- `apt-cache` offre una ricerca **migliorata**, basata su espressioni regolari, su tutti i metadati dei pacchetti.
- `aptitude` può gestire più versioni dei pacchetti senza usare `/etc/apt/preferences` e farlo è piuttosto intuitivo.

2.2.2 Operazioni base per la gestione dei pacchetti dalla riga di comando

Ecco alcune operazioni base di gestione dei pacchetti dalla riga di comando usando apt(8), aptitude(8) e apt-get(8) /apt-cache(8).

sintassi per `apt`	**sintassi per `aptitude`**	**sintassi per `apt-get`/`apt-cache`**	**descrizione**
`apt update`	`aptitude update`	`apt-get update`	aggiorna i metadati dell'archivio dei pacchetti
`apt install foo`	`aptitude install foo`	`apt-get install foo`	installa la versione candidata del pacchetto "`pippo`" con le sue dipendenze
`apt upgrade`	`aptitude safe-upgrade`	`apt-get upgrade`	installa le versioni candidate dei pacchetti installati senza rimuovere altri pacchetti
`apt full-upgrade`	`aptitude full-upgrade`	`apt-get dist-upgrade`	installa le versioni candidate dei pacchetti installati e rimuove, se necessario, altri pacchetti
`apt remove foo`	`aptitude remove foo`	`apt-get remove foo`	rimuove il pacchetto "`pippo`" lasciando i suoi file di configurazione
`apt autoremove`	N/D	`apt-get autoremove`	rimuove i pacchetti installati automaticamente che non sono più necessari
`apt purge foo`	`aptitude purge foo`	`apt-get purge foo`	elimina completamente il pacchetto "`pippo`" con i suoi file di configurazione
`apt clean`	`aptitude clean`	`apt-get clean`	pulisce completamente il repository locale dei file dei pacchetti scaricati
`apt autoclean`	`aptitude autoclean`	`apt-get autoclean`	pulisce il repository locale dei file dei pacchetti scaricati che sono obsoleti
`apt show foo`	`aptitude show foo`	`apt-cache show foo`	mostra informazioni dettagliate sul pacchetto "`pippo`"
`apt search` *`regex`*	`aptitude search` *`regex`*	`apt-cache search` *`regex`*	cerca i pacchetti che corrispondono all'espressione regolare *regex*
N/D	`aptitude why` *`regex`*	N/D	spiega le ragioni per cui i pacchetti che corrispondono a *regex* dovrebbero essere installati
N/D	`aptitude why-not` *`regex`*	N/D	spiega le ragioni per cui i pacchetti che corrispondono a *regex* non dovrebbero essere installati
`apt list --manual-installed`	`aptitude search '~i!~M'`	`apt-mark showmanual`	elenca i pacchetti installati manualmente

Tabella 2.6: Operazioni base di gestione dei pacchetti dalla riga di comando usando aptitude(8) e apt-get(8) /apt-cache(8)

`apt` / `apt-get` e `aptitude` possono essere mescolati senza grandi problemi.

«`aptitude why` *`regex`*» può mostrare ancor più informazioni se usato come «`aptitude -v why` *`regex`*». Informazioni simili possono essere ottenute con «`apt rdepends` *`package`*» o «`apt-cache rdepends` *`package`*».

Quando il comando `aptitude` è avviato in modalità a riga di comando e si trova davanti a problemi come conflitti tra pacchetti, si può passare alla modalità interattiva a tutto schermo premendo successivamente il tasto "e" al prompt.

Nota
Sebbene il comando `aptitude` sia dotato di moltissime funzionalità come il suo risolutore avanzato di conflitti tra pacchetti, questa complessità ha causato (o può ancora causare) alcune regressioni quali i Bug nr. 411123, Bug nr. 514930 e Bug nr. 570377. In caso di dubbio, preferire i comandi `apt`, `apt-get` e `apt-cache` al comando `aptitude`.

Si possono usare opzioni per il comando immediatamente dopo "`aptitude`".

opzione per il comando	descrizione
`-s`	simula il risultato del comando
`-d`	scarica solamente ma non installa/aggiorna
`-D`	mostra delle brevi spiegazioni prima di installazioni e rimozioni automatiche

Tabella 2.7: Opzioni degne di nota per il comando aptitude(8)

Per maggiori informazioni vedere aptitude(8) e il "manuale utente di aptitude" in "`/usr/share/doc/aptitude/README`".

2.2.3 Uso interattivo di aptitude

Per la gestione interattiva dei pacchetti, si avvia `aptitude` in modalità interattiva dal prompt di shell della console nel modo seguente.

```
$ sudo aptitude -u
Password:
```

In questo modo si aggiorna la copia locale delle informazioni sull'archivio e viene mostrato l'elenco dei pacchetti in una schermata a pieno schermo. Aptitude posiziona la sua configurazione in "`~/.aptitude/config`".

Suggerimento
Se si desidera usare la configurazione di root, invece di quella dell'utente, usare "`sudo -H aptitude …`" invece di "`sudo aptitude …`", come nell'esempio precedente.

Suggerimento
`Aptitude` imposta automaticamente le **azioni pendenti** quando viene avviato in modalità interattiva. Se ciò non piace, si può reimpostarlo usando il menu: "Azioni" → "Annulla azioni in attesa".

2.2.4 Associazioni dei tasti per aptitude

Quelle che seguono sono le associazioni di tasti degne di nota usate, nella modalità a schermo intero, per sfogliare lo stato dei pacchetti e per impostare "azioni pianificate" su di essi.

Le espressioni regolari usate per specificare il nome di pacchetto nella riga di comando e nel prompt del menu dopo aver premuto "`l`" o "`//`" sono descritte in seguito. Le espressioni regolari di aptitude possono esplicitamente trovare corrispondenze con nomi di pacchetto usando una stringa che inizia con "~n seguita dal nome del pacchetto.

Suggerimento
È necessario premere "U" per aggiornare tutti i pacchetti installati alla **versione candidata** nell'interfaccia interattiva. Altrimenti vengono aggiornati alla **versione candidata** solo i pacchetti selezionati e quelli da una cui particolare versione dipendono altri.

tasto	azione associata
F10 o Ctrl-t	menu
?	mostra l'**aiuto** per i tasti (elenco più completo)
F10 → Aiuto → Manuale utente	mostra il Manuale utente
u	aggiorna le informazioni sull'archivio dei pacchetti
+	segna il pacchetto per l'**aggiornamento** o l'**installazione**
-	segna il pacchetto per la **rimozione** (mantiene i file di configurazione)
_	segna il pacchetto per l'**eliminazione completa** (rimuove i file di configurazione)
=	**blocca** il pacchetto
U	segna tutti i pacchetti aggiornabili (funziona come **full-upgrade**)
g	inizia a **scaricare** e **installare** i pacchetti selezionati
q	esce dalla schermata attuale e salva i cambiamenti
x	esce dalla schermata attuale e scarta i cambiamenti
Enter	visualizza le informazioni su un pacchetto
C	visualizza il registro dei cambiamenti di un pacchetto
l	cambia la limitazione dei pacchetti visualizzati
/	cerca la prima corrispondenza
\	ripete l'ultima ricerca

Tabella 2.8: Elenco delle associazioni di tasti per aptitude

2.2.5 Viste dei pacchetti in aptitude

Nella modalità interattiva a tutto schermo di aptitude(8), i pacchetti nell'elenco vengono mostrati come nell'esempio seguente.

```
idA   libsmbclient                              -2220kB 3.0.25a-1  3.0.25a-2
```

Il significato degli elementi di questa riga, partendo da sinistra, è il seguente.

- Il flag di "stato attuale" (la prima lettera)
- Il flag di "azione pianificata" (la seconda lettera)
- Il flag "automatico" (la terza lettera)
- Il nome del pacchetto
- Il cambiamento in termini di uso dello spazio su disco collegato dall'"azione pianificata"
- La versione attuale del pacchetto
- La versione candidata del pacchetto

Suggerimento
L'elenco completo dei flag è fornito alla fine della schermata mostrata di **Aiuto** mostrata se si preme "?".

La **versione candidata** è scelta in base alle preferenze locali attuali (vedere apt_preferences(5) e Sezione 2.7.7).

Diversi tipi di viste per i pacchetti sono disponibili nel menu "Viste".

Nota
È estremamente benvenuto ogni aiuto per migliorare l'assegnazione di etichette Debtags ai pacchetti!

vista	descrizione della vista
`Package View`	vedere Tabella 2.10 (vista standard)
`Audit Recommendations`	elenca i pacchetti che sono raccomandati da alcuni dei pacchetti installati, ma non sono ancora installati
`Flat Package List`	elenca i pacchetti senza organizzarli in gruppi (per l'uso con espressioni regolari)
`Debtags Browser`	elenca i pacchetti organizzati in base alle loro voci Debtags
`Source Package View`	elenca i pacchetti raggruppati per pacchetti sorgente

Tabella 2.9: Elenco delle viste di aptitude

categoria	descrizione della vista
`Upgradable Packages`	elenca i pacchetti organizzati in `sezione` → `area` → `pacchetto`
`New Packages`	" "
`Installed Packages`	" "
`Not Installed Packages`	" "
`Obsolete and Locally Created Packages`	" "
`Virtual Packages`	elenca i pacchetti con la stessa funzione
`Tasks`	elenca i pacchetti con funzioni diverse solitamente necessarie per una stessa attività

Tabella 2.10: Organizzazione delle viste standard dei pacchetti

La "`Vista dei pacchetti`" standard organizza i pacchetti un po' nello stile di `dselect` con alcune funzionalità extra.

Suggerimento
La vista dei `Task` può essere usata per scegliere a uno a uno i pacchetti per le proprie attività.

2.2.6 Opzioni per i metodi di ricerca in aptitude

Aptitude offre diverse opzioni per cercare pacchetti usando il suo schema di espressioni regolari.

- Nella riga di comando della shell:
  - "`aptitude search '`*`regex_aptitude`*`'`" per elencare lo stato di installazione, il nome del pacchetto ed una descrizione breve dei pacchetti che corrispondono.
  - "`aptitude show '`*`nome_pacchetto`*`'`" per elencare una descrizione dettagliata del pacchetto
- Nella modalità interattiva a tutto schermo:
  - "`l`" per limitare la vista dei pacchetti ai pacchetti che corrispondono
  - "`/`" per cercare un pacchetto che corrisponda
  - "`\`" per cercare all'indietro un pacchetto che corrisponda
  - "`n`" per trovare il successivo
  - "`N`" per trovare il successivo (all'indietro)

Suggerimento
La stringa *package_name* viene usata per trovare una corrispondenza esatta con il nome di pacchetto a meno che non venga esplicitamente fatta iniziare con "~" per indicare una espressione regolare.

2.2.7 La struttura delle espressioni regolari di aptitude

La struttura delle espressioni regolari di aptitude è quella di **ERE** (vedere Sezione 1.6.2) estese in stile mutt ed il significato delle speciali regole estese di corrispondenza, specifiche di `aptitude`, è il seguente.

- Il formato dell'espressione regolare è la stessa di quello **ERE** usato nei tipici strumenti per il testo in stile Unix "^", "`.*`", "$", ecc. come in egrep(1), awk(1) e perl(1).
- Il *tipo* di relazione è uno tra depends, predepends, recommends, suggests, conflicts, replaces, provides, e definisce la relazione tra i pacchetti.
- Il *tipo* di dipendenza predefinito è "depends".

Suggerimento
Quando *modello_regex* è una stringa vuota, mettere "`~T`" immediatamente dopo il comando.

Ecco alcune scorciatoie.

- "`~P`*`termine`*" == "`~Dprovides:`*`termine`*"
- "`~C`*`termine`*" == "`~Dconflicts:`*`termine`*"
- "…`~W termine`" == "`(…|termine)`"

Gli utenti che hanno familiarità con `mutt` imparano presto, dato che la sintassi per le espressioni si è ispirata a mutt. Vedere "SEARCHING, LIMITING, AND EXPRESSIONS" nel "Manuale utente", "`/usr/share/doc/aptitude/README`".

Nota
Con la versione `lenny` di aptitude(8), può essere usata la nuova sintassi in **forma lunga** come "`?broken`" invece del vecchio equivalente in **forma breve "** "`~b`". Il carattere di spazio " " è ora considerato uno dei caratteri che terminano l'espressione regolare, oltre al carattere tilde "~". Vedere il "Manuale utente" per la sintassi della nuova **forma lunga**.

2.2.8 Risoluzione delle dipendenze di aptitude

Se la voce di menu`F10` → Opzioni → Preferenze → Gestione delle dipendenze" è impostata in modo appropriato, selezionando un pacchetto in `aptitude` non si richiamano solamente i pacchetti definiti nel suo elenco "`Depends:`", ma anche quelli nell'elenco "`Recommends:`". Questi pacchetti installati automaticamente vengo rimossi automaticamente se non sono più necessari secondo `aptitude`.

L'opzione che controlla il comportamento di "installazione automatica" del comando `aptitude` può anche essere manipolata usando il comando apt-mark(8) dal pacchetto `apt`.

2.2.9 Registri delle attività sui pacchetti

Si può controllare lo storico delle attività sui pacchetti nei file di registro.

In realtà non è facile ottenere delle informazioni immediatamente comprensibili da questi file di registro. Vedere Sezione 9.3.9 per un metodo più facile.

2.3 Esempi di operazioni con aptitude

Ecco alcuni esempi di operazioni con aptitude(8).

descrizione della regola estesa di corrispondenza	**struttura della espressione regolare**
corrispondenza con il nome di pacchetto	~n*regex_nome*
corrispondenza con la descrizione	~d*regex_descrizione*
corrispondenza con il nome del task	~t*regex_task*
corrispondenza con il debtag	~G*regex_debtag*
corrispondenza con il manutentore	~m*regex_mantainer*
corrispondenza con la sezione del pacchetto	~s*regex_sezione*
corrispondenza con la versione del pacchetto	~V*regex_versione*
corrispondenza con l’archivio	~A{trixie,forky,sid}
corrispondenza con l’origine	~O{debian,…}
corrispondenza con la priorità	~p{extra,important,optional,required,standard}
corrispondenza con pacchetti essenziali	~E
corrispondenza con pacchetti virtuali	~v
corrispondenza con pacchetti nuovi	~N
corrispondenza con azioni pendenti	~a{install,upgrade,downgrade,remove,purge,hold,keep}
corrispondenza con pacchetti installati	~i
corrispondenza con pacchetti installati e con l’indicatore **A** (pacchetti installati automaticamente)	~M
corrispondenza con i pacchetti installati e senza l’indicatore **A** (pacchetti selezionati dall’amministratore)	~i!~M
corrispondenza con pacchetti installati che sono aggiornabili	~U
corrispondenza con pacchetti rimossi ma non eliminati completamente	~c
corrispondenza con pacchetti rimossi, eliminati o che possono essere rimossi	~g
corrispondenza con pacchetti che dichiarano problemi di dipendenze	~b
corrispondenza con pacchetti con problemi di dipendenze di tipo *tipo*	~B*tipo*
corrispondenza con pacchetti *modello* con dipendenze di tipo *tipo*	~D[*tipo*:]*modello*
corrispondenza con pacchetti *modello* con dipendenze non soddisfatte di tipo *tipo*	~DB[*tipo*:]*modello*
corrispondenza con pacchetti verso i quali il pacchetto che corrisponde a *modello* dichiara una dipendenza di tipo *tipo*	~R[*tipo*:]*modello*
corrispondenza con pacchetti verso i quali il pacchetto *termine* dichiara una dipendenza non soddisfatta di tipo *tipo*	~RB[*tipo*:]*modello*
corrispondenza con pacchetti da cui dipende un qualche altro pacchetto installato	~R~i
corrispondenza con pacchetti da cui non dipende alcun pacchetto installato	!~R~i
corrispondenza con pacchetti da cui dipende o che vengono raccomandati da qualche altro pacchetto	~R~i|~Rrecommends:~i
corrispondenza con pacchetto *modello* con filtro sulle versioni	~S filtro *modello*
corrispondenza con tutti i pacchetti (vero)	~T
corrispondenza con nessun pacchetto (falso)	~F

Tabella 2.11: Elenco delle regole per espressioni regolari di aptitude

file	contenuto
/var/log/dpkg.log	Registro delle attività a livello dpkg per tutte le attività sui pacchetti
/var/log/apt/term.log	Registro delle attività APT generiche
/var/log/aptitude	Registro delle attività di comandi aptitude

Tabella 2.12: File di registro per le attività sui pacchetti

2.3.1 Cercare pacchetti interessanti

Si possono cercare i pacchetti che soddisfano i propri bisogni con aptitude in base alla descrizione del pacchetto o dall'elenco in "Task".

2.3.2 Elencare pacchetti in base alla corrispondenza del nome con espressioni regolari

Il seguente comando elenca i pacchetti i cui nomi corrispondono alle espressioni regolari specificate.

```
$ aptitude search '~n(pam|nss).*ldap'
p libnss-ldap - NSS module for using LDAP as a naming service
p libpam-ldap - Pluggable Authentication Module allowing LDAP interfaces
```

È piuttosto utile se si desidera trovare il nome esatto di un pacchetto.

2.3.3 Sfogliare le corrispondenze ad una espressione regolare

L'espressione regolare "~dipv6" nel prompt di "l" nella vista "Nuovo elenco unico dei pacchetti", limita la visualizzazione dei pacchetti a quelli la cui descrizione corrisponde all'espressione "ipv6" e permette di sfogliare interattivamente le loro informazioni.

2.3.4 Eliminare completamente i pacchetti rimosssi

Si possono eliminare tutti i file di configurazione rimasti di pacchetti rimossi.

Controllare il risultato del comando seguente.

```
# aptitude search '~c'
```

Se si pensa che sia giusto rimuover i pacchetti elencati, eseguire il comando seguente.

```
# aptitude purge '~c'
```

Si può fare la stessa cosa in modalità interattiva per avere un controllo più dettagliato.

Fornire l'espressione regolare "~c" al prompt di "l" nella vista "Nuova vista dei pacchetti". In questo modo si limita la visualizzazione a quei pacchetti che corrispondono all'espressione regolare, cioè "rimossi ma non eliminati completamente". Tutte le corrispondenze trovate possono essere mostrate premendo "[" nelle intestazioni di primo livello.

Premere poi "_" nell'intestazione di primo livello, come "Pacchetti non installati". In questo modo vengono selezionati per l'eliminazione completa solo quei pacchetti contenuti sotto l'intestazione che hanno fatto corrispondenza con l'espressione regolare. Si possono escludere alcuni pacchetti in modo interattivo dall'eliminazione premendo "=" per ciascuno di essi.

Questa tecnica è piuttosto comoda e funziona per molti altri tasti di comando.

2.3.5 Mettere ordine nello stato di installazione automatico/non automatico

Ecco come faccio per mettere ordine nello stato di installazione automatico/non automatico per i pacchetti (dopo aver usato un installatore che non sia aptitude, ecc.).

1. Avviare da root `aptitude` in modalità interattiva.
2. Digitare "u", "U", "f" e "g" per aggiornare l'elenco dei pacchetti e i pacchetti stessi.
3. Digitare "`l`" per impostare la limitazione della visualizzazione con "`~i(~R~i|~Rrecommends:~i)`" e digitare "`M`" su "`Pacchetti installati`" per marcarli come installati automaticamente.
4. Digitare "`l`" per impostare la limitazione della visualizzazione dei pacchetti con "`~prequired|~pimportant|~pstan` e digitare "`m`" su "`Pacchetti installati`" per marcarli come installati manualmente.
5. Digitare "`l`" per impostare la limitazione della visualizzazione dei pacchetti con "`~i!~M`" e rimuovere i pacchetti inutilizzati digitando "`-`" su ciascuno di essi dopo averli visualizzati digitando "`[`" su "`Pacchetti installati`".
6. Digitare "`l`" per impostare la limitazione della visualizzazione dei pacchetti con "`~i`"; poi digitare "`m`" su "`Task`" per marcare quei pacchetti come installati manualmente.
7. Uscire da `aptitude`.
8. Avviare "`apt-get -s autoremove|less`" da root per controllare quelli che non sono usati.
9. Riavviare `aptitude` in modalità interattiva e segnare i pacchetti necessari con "`m`".
10. Riavviare "`apt-get -s autoremove|less`" da root per controllare che in REMOVED siano contenuti soltanto i pacchetti attesi.
11. Avviare "`apt-get autoremove|less`" da root per rimuovere automaticamente i pacchetti non utilizzati.

L'azione "`m`" sopra "`Task`" è opzionale e server per prevenire situazioni di rimozioni di massa di pacchetti in futuro.

2.3.6 Aggiornamento di tutto il sistema

Nota
Quando si passa ad un nuovo rilascio, ecc. si dovrebbe considerare una installazione pulita di un nuovo sistema anche se è possibile aggiornare Debian nel modo descritto in seguito. Ciò dà la possibilità di rimuovere la spazzatura raccolta nel tempo e fornisce la migliore combinazione dei pacchetti più nuovi. Naturalmente si dovrebbe fare un backup completo del sistema in un posto sicuro (vedere Sezione 10.2) prima di farlo. Raccomando di creare una configurazione con due sistemi avviabili usando due diverse partizioni per avere una transizione il più lineare possibile.

Si può fare l'aggiornamento completo del sistema ad un nuovo rilascio cambiando il contenuto dell'**elenco delle fonti** facendolo puntare ad un nuovo rilascio ed eseguendo il comando «`apt update; apt dist-upgrade`».

Per aggiornare da `stable` a `testing` o `unstable`, durante il ciclo di rilascio `trixie`-as-`stable`, sostituire "`trixie`" nell'**elenco delle fonti** di esempio in Sezione 2.1.5 con "`forky`" o "`sid`".

In realtà si possono dover affrontare alcune complicazioni a causa di problemi di transizione dei pacchetti, per lo più a causa di dipendenze. Più è grande la differenza nell'aggiornamento, più è probabile dover affrontare problemi più grossi. Per la transizione dal vecchio rilascio `stable` al nuovo `stable` si possono leggere le nuove Note di rilascio e seguire esattamente la procedura che vi è descritta per minimizzare i problemi.

Quando si decide di passare da `stable` a `testing` prima del suo rilascio formale non ci sono Note di rilascio che possano aiutare. La differenza tra `stable` e `testing` potrebbe essere diventata piuttosto grande dopo il precendente rilascio `stable` e potrebbe rendere l'aggiornamento complicato.

SI dovrebbero prendere misure precauzionali per l'aggiornamento completo raccogliendo le informazioni più aggiornate da mailing list ed usando il buon senso.

1. Leggere le ”Note di rilascio” precedenti.
2. Fare il backup dell’intero sistema (specialmente dei dati e della configurazione).
3. Tenere a portata di mano supporti avviabili in caso di problemi con il bootloader.
4. Informare con un buon anticipo gli utenti nel sistema.
5. Registrare l’attività di aggiornamento con script(1).
6. Applicare l’opzione ”unmarkauto” ai pacchetti richiesti, ad esempio ”`aptitude unmarkauto vim`, per prevenire la loro rimozione.
7. Minimizzare i pacchetti installati per ridurre le probabilità di conflitti tra i pacchetti, ad esempio rimuovere i pacchetti dell’attività desktop.
8. Rimuovere il file ”`/etc/apt/preferences`” (disabilitare **apt-pinning**).
9. Cercare di aggiornare per gradi: `oldstable` → `stable` → `testing` → `unstable`.
10. Aggiornare l’**elenco delle fonti** per farlo puntare solamente al nuovo archivio ed eseguire ”`aptitude update`”.
11. Installare, opzionalmente i nuovi pacchetti partendo dai **pacchetti fondamentali**, ad esempio ”`aptitude install perl`”.
12. Eseguire il comando ”`apt-get -s dist-upgrade`” per avere un’idea dell’impatto dell’aggiornamento.
13. Eseguire da ultimo il comando ”`apt-get dist-upgrade`”.



Attenzione
Non è saggio saltare un rilascio principale Debian quando si aggiorna da un rilascio `stable` ad un altro.



Attenzione
Nelle ”Note di rilascio” precedenti GCC, il kernel Linux, initrd-tools, Glibc, Perl, APT tool chain, ecc. hanno richiesto un po’ di attenzione particolare per gli aggiornamenti dell’intero sistema.



Attenzione
Le ”Note di rilascio” possono non coprire tutti i casi possibili. Se si cambiano configurazioni a basso livello, l’aggiornamento successivo può fallire in modo grave con ”... segfault after upgrade ...”.

Per gli aggiornamenti quotidiani in `unstable`, vedere Sezione 2.4.3.

2.4 Operazioni avanzate per la gestione dei pacchetti

2.4.1 Operazioni avanzate per la gestione dei pacchetti dalla riga di comando

Ecco un elenco di altre operazioni di gestione dei pacchetti per i quali `aptitude` è di livello troppo altro o manca delle funzionalità richieste.

comando	azione
`COLUMNS=120 dpkg -l` *`modello_nome_pacchetto`*	elenca lo stato di un pacchetto installato per una segnalazione di bug
`dpkg -L` *`nome_pacchetto`*	elenca il contenuto di un pacchetto installato
`dpkg -L` *`nome_pacchetto`* `| egrep '/usr/share/man/man.*/.+'`	elenca le pagine man di un pacchetto installato
`dpkg -S` *`modello_nome_file`*	elenca i pacchetti installati che uno un file con un nome che corrisponde al modello
`apt-file search` *`modello_nome_file`*	elenca i pacchetti nell'archivio che hanno un file con un nome che corrisponde al modello
`apt-file list` *`modello_nome_pacchetto`*	elenca il contenuto dei pacchetti nell'archivio che corrispondono al modello
`dpkg-reconfigure` *`nome_pacchetto`*	riconfigura il pacchetto con esattamente quel nome
`dpkg-reconfigure -plow` *`nome_pacchetto`*	riconfigura il pacchetto con esattamente quel nome ponendo le domande più dettagliate
`configure-debian`	riconfigura i pacchetti da un'interfaccia a tutto schermo a menu
`dpkg --audit`	controlla il sistema alla ricerca di pacchetti parzialmente installati
`dpkg --configure -a`	configura tutti i pacchetti parzialmente installati
`apt-cache policy` *`nome_pacchetto_binario`*	mostra la versione disponibile, la priorità e le informazioni nell'archivio di un pacchetto binario
`apt-cache madison` *`nome pacchetto`*	mostra la versione disponibile e le informazioni nell'archivio di un pacchetto
`apt-cache showsrc` *`nome_pacchetto_binario`*	mostra informazioni sul pacchetto sorgente di un pacchetto binario
`apt-get build-dep` *`nome_pacchetto`*	installa i pacchetti richiesti per compilare un pacchetto
`aptitude build-dep` *`nome_pacchetto`*	installa i pacchetti richiesti per compilare un pacchetto
`apt-get source` *`nome_pacchetto`*	scarica un sorgente (da un archivio standard)
`dget` *`URL del file dsc`*	scarica un pacchetto sorgente (da un altro archivio)
`dpkg-source -x` *`nome_pacchetto_versione-versione_debian`*`.dsc`	crea un albero dei sorgenti a partire da un insieme di pacchetti sorgenti ("`*.orig.tar.gz`" e "`*.debian.tar.gz`"/"`*.diff.gz`")
`debuild binary`	compila pacchetti da un albero dei sorgenti locale
`make-kpkg kernel_image`	compila un pacchetto kernel a partire da un albero dei sorgenti del kernel
`make-kpkg --initrd kernel_image`	compila un pacchetto kernel con initramfs abilitato, a partire da un albero dei sorgenti
`dpkg -i` *`nome_pacchetto_versione-versione_debian_arch`*`.deb`	installa un pacchetto locale nel sistema
`apt install /percorso/di/`*`nomefile_pacchetto`*`.deb`	installa un pacchetto locale nel sistema, cercando al contempo di risolvere automaticamente le dipendenze
`debi` *`nome_pacchetto_versione-versione_debian_arch`*`.dsc`	installa pacchetti locali nel sistema
`dpkg --get-selections '*' >selection.txt`	salva le informazioni di stato sui pacchetti selezionati a livello di `dpkg`
`dpkg --set-selections <selection.txt`	imposta le informazioni di stato sui pacchetti selezionati a livello di `dpkg`
`echo` *`nome_pacchetto`* `hold | dpkg --set-selections`	imposta lo stato di selezione di un pacchetto a livello di `dpkg` a **hold** (equivalente a "`aptitude hold`*`nome_pacchetto`*")

Tabella 2.13: Elenco delle operazioni avanzate per la gestione dei pacchetti

Nota
Per un pacchetto con la funzionalità multi-arch, può essere necessario specificare l'architettura per alcuni comandi. Per esempio, usare «`dpkg -L libglib2.0-0:amd64`» per elencare il contenuto del pacchetto `libglib2.0-0` per l'architettura `amd64`.



Attenzione
Gli strumenti per i pacchetti a più basso livello, come "`dpkg -i …`" e "`debi …`" dovrebbero essere usati con molta cura dall'amministratore di sistema. Non si prendono cura in modo automatico delle dipendenze richieste dai pacchetti. Le opzioni "`--force-all`" e simili, per la riga di comando di dpkg (vederedpkg(1), sono pensate per essere usate solamente da utenti esperti. Usarle senza comprenderne pienamente l'effetto potrebbe portare tutto il sistema in uno stato problematico.

Notare quanto segue.

- Tutti i comandi di configurazione ed installazioni di sistema devono essere eseguiti da root.
- A differenza di `aptitude` che usa espressioni regolari (vedere Sezione 1.6.2), gli altri comandi di gestione dei pacchetti usano espansioni di modelli in stile shell (vedere Sezione 1.5.6).
- Per apt-file(1), fornito dal pacchetto `apt-file`, è necessario eseguire prima "`apt-file update`".
- configure-debian(8), fornito dal pacchetto `configure-debian` esegue dpkg-reconfigure(8) come suo backend.
- dpkg-reconfigure(8) esegue gli script dei pacchetti usando debconf(1) come suo backend.
- I comandi "`apt-get build-dep`", "`apt-get source`" e "`apt-cache showsrc`" richiedono la voce "`deb-src`" nell'**elenco delle fonti**.
- dget(1), debuild(1) e debi(1) richiedono il pacchetto `devscripts`.
- Vedere la procedura per (ri)creare pacchetti usando "`apt-get source`" in Sezione 2.7.13.
- Il comando `make-kpkg` richiede il pacchetto `kernel_package` (vedere Sezione 9.10).
- Vedere Sezione 12.9 per informazioni generali sulla creazione di pacchetti.

2.4.2 Verifica dei pacchetti installati

L'installazione di `debsums` permette la verifica dei file di pacchetto installati in base ai valori delle somme di controllo MD5sum nei file "`/var/lib/dpkg/info/*.md5sums`" con debsums(1). Per il funzionamento di MD5sum vedere Sezione 10.3.5.

Nota
Dato che il database MD5sum può essere manomesso da un intruso, `debsums` ha un'utilità limitata come strumento di sicurezza. È adatto solo per controllare modifiche fatte dall'amministratore o danni dovuti ad errori dei supporti.

2.4.3 Salvaguardarsi da problemi coi pacchetti

Molti utenti preferiscono usare il rilascio **testing** (o **unstable**) del sistema Debian per avere le nuove funzionalità ed i nuovi pacchetti. Questo rende il sistema più esposto a bug critici sui pacchetti.

L'installazione del pacchetto `apt-listbugs` salvaguarda il sistema dai bug critici controllando automaticamente, al momento degli aggiornamenti fatti con il sistema APT, il BTS Debian alla ricerca dei bug critici.

L'installazione del pacchetto `apt-listchanges` fornisce importanti notizie contenute in `NEWS.Debian` quando si fanno aggiornamenti con il sistema APT.

2.4.4 Cercare tra i metadati dei pacchetti

Sebbene visitare il sito Debian https://packages.debian.org/ faciliti oggi giorno la ricerca nei metadati dei pacchetti, ci sono metodi più tradizionali per farlo.

I comandi grep-dctrl(1), grep-status(1) e grep-available(1) possono essere usati per cercare in qualsiasi file che sia nel formato generico dei file di controllo dei pacchetti di Debian.

"`dpkg -S` *`modello_nome_file`*" può essere usato per cercare i pacchetti che contengono al loro interno file installati da `dpkg` il cui nome corrisponde al modello. Ma non trova i file creati dagli script dei manutentori.

Se è necessario fare ricerche più elaborate sui metadati dei pacchetti, è necessario eseguire il comando "`grep -e modello_regex *`" nella directory "`/var/lib/dpkg/info/`". In questo modo si cercano le parole indicate negli script e nei testi con le domande di installazione dei pacchetti.

Se si desiderano vedere in modo ricorsivo le dipendenze dei pacchetti, si dovrebbe usare apt-rdepends(8).

2.5 Aspetti tecnici della gestione dei pacchetti in Debian

Verranno ora trattati alcuni aspetti tecnici interni del funzionamento del sistema di gestione dei pacchetti Debian. Queste informazioni dovrebbero aiutare a trovare soluzioni proprie ai problemi con i pacchetti.

2.5.1 Metadati degli archivi

I file con i metadati per ciascuna distribuzione sono archiviati in "`dists/`*`codename`*" su ciascun sito mirror di Debian, ad esempio "`http://deb.debian.org/debian/`". La struttura del suo archivio può essere navigata con il browser web. Ci sono 6 tipi di metadati.

file	**posizione**	**contenuto**
`Release`	directory della distribuzione di più alto livello	descrizione dell'archivio e informazioni sull'integrità
`Release.gpg`	directory della distribuzione di più alto livello	file di firma per il file "`Release`" firmato con la chiave dell'archivio
`Contents-`*`architettura`*	directory della distribuzione di più alto livello	elenca tutti i file per tutti i pacchetti nell'archivio relativo
`Release`	directory più elevata per ciascuna combinazione distribuzione/area/architettura	descrizione dell'archivio usata per le regole di apt_preferences(5)
`Packages`	directory più elevata per ciascuna combinazione distribuzione/area/architettura-binaria	dati `debian/control` concatenati dei pacchetti binari
`Sources`	directory più elevata per ciascuna combinazione distribuzione/area/source	dati `debian/control` concatenati dei pacchetti sorgenti

Tabella 2.14: Il contenuto dei metadati dell'archivio Debian

Negli archivi recenti questi metadati sono memorizzati come file compressi e file di differenze per ridurre il traffico di rete.

2.5.2 File ”Release” nella directory principale ed autenticità

Suggerimento
Il file ”Release” nella directory di livello più alto è usato per firmare gli archivi nel sistema **secure APT**.

Ogni suite dell’archivio Debian ha un file ”Release” nella sua directory di livello più alto, ad esempio ”http://deb.debian.
di questo tipo.

```
Origin: Debian
Label: Debian
Suite: unstable
Codename: sid
Date: Sat, 14 May 2011 08:20:50 UTC
Valid-Until: Sat, 21 May 2011 08:20:50 UTC
Architectures: alpha amd64 armel hppa hurd-i386 i386 ia64 kfreebsd-amd64 kfreebsd-i386 mips ↩
    mipsel powerpc s390 sparc
Components: main contrib non-free
Description: Debian x.y Unstable - Not Released
MD5Sum:
 bdc8fa4b3f5e4a715dd0d56d176fc789 18876880 Contents-alpha.gz
 9469a03c94b85e010d116aeeab9614c0 19441880 Contents-amd64.gz
 3d68e206d7faa3aded660dc0996054fe 19203165 Contents-armel.gz
...
```

Nota
Qui si può scoprire il motivo per cui in questa guida si è scelto di usare ”suite” e ”codename” (nome in codice) in Sezione 2.1.5. Il termine ”distribuzione” è usato per riferirsi sia a ”suite” sia a ”codename”. Tutti i nomi delle ”aree” dell’archivio offerte da un archivio sono elencate in ”Componente”.

L’integrità del file ”Release” nella directory base viene verificata usando l’infrastruttura crittografica chiamata secure apt, come descritto in apt-secure(8).

- Il file ”Release.gpg” con la firma crittografica viene creato a partire dal file ”Release” nella directory base e dalla chiave segreta dell’archivio Debian.
- Le chiavi pubbliche dell’archivio Debian sono installate localmente dal pacchetto debian-archive-keyring più recente.
- Il sistema **secure APT** verifica crittograficamente, in modo automatico, l’integrità del file ”Release” di più alto livello scaricato con questo file ”Release.gpg” e le chiavi pubbliche degli archivi Debian installate localmente.
- L’integrità di tutti i file ”Packages” e ”Sources” viene verificata usando i valori MD5sum nel file ”Release” nella ripsettiva directory base. L’integrità di tutti i file di pacchetto viene verificata usando i valori MD5sum nei file ”Packages” e ”Sources”. Vedere debsums(1) e Sezione 2.4.2.
- Dato che la verifica crittografica della firma è un processo molto più dispendioso in termini di uso della CPU rispetto al calcolo di valori MD5sum, l’uso di valori MD5sum per ciascun pacchetto mentre viene usata la firma crittografica per il file ”Release” principale, fornisce una buona sicurezza con buone prestazioni (vedere Sezione 10.3).

Se la voce nell’**elenco delle fonti** specifica l’opzione ”signed-by”, l’integrità del suo file ”Release” di livello più alto scaricato viene verificata usando la chiave pubblica specifica. Ciò è utile quando l’**elenco delle fonti** contiene archivi non Debian.

Suggerimento
L’uso del comando apt-key(8) per la gestione delle chiavi di APT è deprecato.

Si può anche verificare manualmente l’integrità del file ”Release” con il file ”Release.gpg” e la chiave pubblica dell’archivio Debian pubblicata su ftp-master.debian.org usando gpg.

2.5.3 File ”Release” a livello di archivio

Suggerimento
I file ”`Release`” a livello di archivio sono usati per le regole di apt_preferences(5).

Ci sono file ”`Release`” a livello di archivio per tutte le posizioni nell’archivio specificate dall’**elenco delle fonti**, come ”`http://deb.debian.org/debian/dists/unstable/main/binary-amd64/Release`” o ”`http://deb.debian.`
come nel seguente esempio.

```
Archive: unstable
Origin: Debian
Label: Debian
Component: main
Architecture: amd64
```



Attenzione
Nell’archivio Debian, per la voce «`Archive:`» vengono usati i nomi di suite («`stable`», «`testing`», «`unstable`», ...), mentre vengono usati i nomi in codice («`trusty`», «`xenial`», «`artful`», ...) nell’archivio Ubuntu.

Per alcuni archivi, come `experimental` e `trixie-backports`, che contengono pacchetti che non dovrebbero essere installati automaticamente, c’è una riga aggiuntiva, come ”`http://deb.debian.org/debian/dists/experiment`
nell’esempio seguente.

```
Archive: experimental
Origin: Debian
Label: Debian
NotAutomatic: yes
Component: main
Architecture: amd64
```

Notare che per gli archivi normali senza la riga ”`NotAutomatic: yes`”, il valore predefinito per la priorità di Pin è 500, mentre per gli archivi speciali con ”`NotAutomatic: yes`”, il valore predefinito per la priorità di Pin è 1 (vedere apt_preferences(5) e Sezione 2.7.7).

2.5.4 Recuperare i metadati per un pacchetto

Quando si usano strumenti APT, quali `aptitude`, `apt-get`, `synaptic`, `apt-file`, `auto-apt`, ... è necessario aggiornare le copie locali dei metadati contenenti le informazioni sull’archivio Debian. Queste copie locali hanno i seguenti nomi di file che corrispondono ai nomi di `distribuzione`, `area` e `architettura` specificati nell’**elenco delle fonti** (vedere Sezione 2.1.5).

- ”`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribuzione`*`_Release`”
- ”`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribuzione`*`_Release.gpg`”
- ”`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribuzione_area`*`_binary-`*`architettura`*`_Pa`
- ”`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribuzione_area`*`_source_Sources`”
- ”`/var/cache/apt/apt-file/deb.debian.org_debian_dists_`*`distribuzione`*`_Contents-`*`architettura`*`.`
  (per `apt-file`)

I primi quattro tipi di file sono condivisi da tutti i comandi APT pertinenti e vengono aggiornati dalla riga di comando con "`apt-get update`" e "`aptitude update`". I metadati "`Packages`" sono aggiornati se esiste la riga "`deb`" nell'**elenco delle fonti**. I metadati "`Sources`" sono aggiornati se esiste la riga "`deb-src`" nell'**elenco delle fonti**.

I metadati "`Packages`" e "`Sources`" contengono la voce "`Filename:`" che punta alla posizione dei file dei pacchetti binari e sorgenti. Attualmente questi pacchetti sono posizionati nell'albero di directory "`pool/`" per una transizione migliore tra i rilasci.

Le copie locali dei metadati "`Packages`" possono essere ricercate interattivamente con l'aiuto di `aptitude`. Il comando specializzato di ricerca grep-dctrl(1) può cercare nelle copie locai dei metadati "`Packages`" e "`Sources`".

La copia locale dei metadati "`Contents-`*`architettura`*" può essere aggiornata con "`apt-file update`" e la sua posizione è diversa da quella degli altri quattro file. Vedere apt-file(1). (`auto-apt` usa in modo predefinito una posizione differente per la copia locale di "`Contents-`*`architettura`*`.gz`".)

2.5.5 Lo stato dei pacchetti per APT

In aggiunta ai metadati recuperati da remoto, lo strumento APT, a partire da `lenny`, memorizza le informazioni generate localmente sullo stato delle installazioni in "`/var/lib/apt/extended_states`" che è usato da tutti gli strumenti APT per tenere traccia dei pacchetti installati automaticamente.

2.5.6 Lo stato dei pacchetti per aptitude

In aggiunta ai metadati recuperati da remoto, il comando `aptitude` memorizza le informazioni generate localmente sullo stato delle installazioni in "`/var/lib/aptitude/pkgstates`" che è usato solo da esso.

2.5.7 Copie locali dei pacchetti scaricati

Tutti i pacchetti scaricati da remoto attraverso il meccanismo APT vengono salvati nella directory "`/var/cache/apt/archi`" fino a che non vengono cancellati.

Questa politica di pulizia dei file della cache per `aptitude` può essere impostata in "`Opzioni`" → "`Preferenze`" e può essere forzata usando la sua voce di menu "`Pulisci la cache dei pacchetti`" o "`Cancella i file vecchi`" in "`Azioni`".

2.5.8 Nomi dei file di pacchetto di Debian

I file di pacchetto Debian hanno nomi con una particolare struttura.

tipo di pacchetto	**struttura del nome**
Il pacchetto binario (alias `deb`)	*`nome-pacchetto_versione-originale-versione-debian_archi`*
Il pacchetto binario per l'installatore Debian (alias `udeb`)	*`nome-pacchetto_versione-originale-versione-debian_archi`*
Il pacchetto sorgente (sorgenti originali)	*`nome-pacchetto_versione-originale-versione-debian`*`.orig`
Il pacchetto sorgente `1.0` (modifiche Debian)	*`nome-pacchetto_versione-originale-versione-debian`*`.diff`
Il pacchetto sorgente `3.0 (quilt)` (modifiche Debian)	*`nome-pacchetto_versione-originale-versione-debian`*`.debia`
Il pacchetto sorgente (descrizione)	*`nome-pacchetto_versione-originale-versione-debian`*`.dsc`

Tabella 2.15: La struttura dei nomi dei pacchetti Debian

Suggerimento
Sono qui descritti solamente i formati base dei pacchetti sorgenti. Per ulteriori informazioni vedere dpkg-source(1).

nome porzione	caratteri utilizzabili (espressione regolare)	presenza
nome-pacchetto	`[a-z0-9][-a-z0-9.+]+`	richiesta
epoca`:`	`[0-9]+:`	opzionale
versione-originale	`[-a-zA-Z0-9.+:]+`	richiesta
versione.debian	`[a-zA-Z0-9.+~]+`	opzionale

Tabella 2.16: I caratteri utilizzabili all'interno di ciascuna porzione del nome dei pacchetti Debian

Nota
Si può controllare l'ordine delle versioni dei pacchetti con dpkg(1), per esempio con "`dpkg --compare-versions 7.0 gt 7.~pre1 ; echo $?`".

Nota
L'installatore Debian (d-i) usa come estensione per i suoi pacchetti binari `udeb` invece della normale `deb`. Un pacchetto `udeb` è un pacchetto `deb` ridotto al minimo, con rimossi alcuni contenuti non essenziali come la documentazione, per risparmiare spazio seguendo meno rigorosamente i requisiti richiesti dalle norme per i pacchetti. Entrambi i pacchetti `deb` e `udeb` condividono la stessa struttura di pacchetto. La "u" sta per micro.

2.5.9 Il comando dpkg

dpkg(1) è lo strumento di più basso livello per la gestione dei pacchetti Debian. È molto potente e deve essere usato con cautela.

Quando installa un pacchetto chiamato "*nome_pacchetto*", `dpkg` fa nell'ordine elencato le seguenti operazioni.

1. Spacchetta il file deb (equivalente di "`ar -x`")
2. Esegue "*nome_pacchetto*`.preinst`" usando debconf(1)
3. Installa il contenuto del pacchetto nel sistema (equivalente di "`tar -x`")
4. Esegue "*nome_pacchetto*`.postinst`" usando debconf(1)

Il sistema `debconf` fornisce un'interazione standard con l'utente con supporto per internazionalizzazione e localizzazione (Capitolo 8).

Il file "`status`" è anche usato da altri strumenti come dpkg(1): "`dselect update`" e "`apt-get -u dselect-upgrade`".

Il comando di ricerca specializzato grep-dctrl(1) può ricercare nelle copie locali dei metadati "`status`" e "`available`".

Suggerimento
Nell'ambiente dell'installatore Debian, viene usato il comando `udpkg` per aprire i pacchetti `udeb`. Il comando `udpkg` è una versione ridotta al minimo del comando `dpkg`.

file	descrizione del contenuto
`/var/lib/dpkg/info/`*nome_pacchetto*`.conffiles`	elenco dei file di configurazione (modificabile dall'utente)
`/var/lib/dpkg/info/`*nome_pacchetto*`.list`	elenco dei file e delle directory installate dal pacchetto
`/var/lib/dpkg/info/`*nome_pacchetto*`.md5sums`	elenco dei valori degli hash MD5 per i file installati dal pacchetto
`/var/lib/dpkg/info/`*nome_pacchetto*`.preinst`	script del pacchetto da eseguire prima dell'installazione del pacchetto
`/var/lib/dpkg/info/`*nome_pacchetto*`.postinst`	script del pacchetto da eseguire dopo l'installazione del pacchetto
`/var/lib/dpkg/info/`*nome_pacchetto*`.prerm`	script del pacchetto da eseguire prima della rimozione del pacchetto
`/var/lib/dpkg/info/`*nome_pacchetto*`.postrm`	script del pacchetto da eseguire dopo la rimozione del pacchetto
`/var/lib/dpkg/info/`*nome_pacchetto*`.config`	script per il sistema `debconf`
`/var/lib/dpkg/alternatives/`*nome_pacchetto*	informazioni sulle alternative usate dal comando `update-alternatives`
`/var/lib/dpkg/available`	informazioni sulla disponibilità per tutti i pacchetti
`/var/lib/dpkg/diversions`	le informazioni sulle deviazioni usate da dpkg(1) e impostate da dpkg-divert(8)
`/var/lib/dpkg/statoverride`	le informazioni per la sovrascrittura dello stato usate da dpkg(1) e impostate da dpkg-statoverride(8)
`/var/lib/dpkg/status`	le informazioni sullo stato per tutti i pacchetti
`/var/lib/dpkg/status-old`	la prima generazione di backup del file "`/var/lib/dpkg/status`"
`/var/backups/dpkg.status*`	la seconda generazione e le generazioni precedenti di backup del file "`/var/lib/dpkg/status`"

Tabella 2.17: I file degni di nota creati da `dpkg`

2.5.10 Il comando update-alternatives

Il sistema Debian ha un meccanismo per installare senza problemi programmi che si sovrappongono nelle loro funzioni usando update-alternatives(1). Per esempio, si può far sì che il comando `vi` esegua `vim` quando si sono installati entrambi i pacchetti `vim` e `nvi`.

```
$ ls -l $(type -p vi)
lrwxrwxrwx 1 root root 20 2007-03-24 19:05 /usr/bin/vi -> /etc/alternatives/vi
$ sudo update-alternatives --display vi
...
$ sudo update-alternatives --config vi
  Selection    Command
 -----------------------------------------------
      1        /usr/bin/vim
*+    2        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 1
```

Il sistema delle alternative Debian mantiene le proprie selezioni in forma di collegamenti simbolici in "`/etc/alternatives/`".
Il procedimento di selezione usa i file corrispondenti in "`/var/lib/dpkg/alternatives/`".

2.5.11 Il comando dpkg-statoverride

La **sovrascrittura dello stato** fornita dal comando dpkg-statoverride(8) è un modo per dire a dpkg(1) di usare un proprietario o una modalità diversa per un **file** quando viene installato un pacchetto. Se viene specificato "`--update`" e il file esiste già, viene immediatamente impostato con il prorpietario e la modalità nuovi.



Attenzione
La modifica diretta del proprietario o della modalità, usando `chmod` o `chown`, di un **file** che appartiene ad un pacchetto da parte dell'amministratore del sistema viene vanificata dal successivo aggiornamento del pacchetto.

Nota
È stata usata la parola **file**, ma in realtà si può trattare di un qualsiasi oggetto nel file system gestito da `dpkg`, inclusi directory, device, ecc.

2.5.12 Il comando dpkg-divert

Le **deviazioni** per i file fornite dal comando dpkg-divert(8) sono un modo per forzare dpkg(1) a non installare un file nella sua posizione predefinita, ma in una posizione **deviata**. L'uso di `dpkg-divert` è pensato per gli script di manutenzione dei pacchetti. Il suo uso in maniera superficiale da parte dell'amministratore di sistema è deprecato.

2.6 Ripristino da un sistema con problemi

Quando si usa il sistema `testing` o `unstable`, l'amministratore deve saper ripristinare le cose da una situazione problematica di gestione dei pacchetti.



Attenzione
Alcuni dei metodi descritti comportano azioni ad alto rischio. Ci si consideri avvertiti!

2.6.1 Incompatibilità con vecchie configurazioni utente

Se un programma con interfaccia grafica per il desktop presenta problemi di instabilità dopo un importante aggiornamento della versione originale, si dovrebbero sospettare interferenze con vecchi file locali di configurazione creati da esso. Se è stabile per account utente creati da zero, questa ipotesi è confermata. (Questo è un bug nella creazione del pacchetto e viene solitamente evitato dal curatore del pacchetto.)

Per ripristinare la stabilità, si dovrebbero spostare i corrispondenti file locali di configurazione e riavviare il programma GUI. Potrebbe essere necessario leggere il contenuto dei vecchi file di configurazione per ripristinare successivamente le informazioni di configurazione. (Non cancellarli troppo di fretta.)

2.6.2 Errori di cache dei dati dei pacchetti

Errori nella cache dei dati dei pacchetti causano errori interessanti come "GPG error: ... invalid: BADSIG ..." con APT.

Si dovrebbero rimuovere tutti i dati in cache con "`sudo rm -rf /var/lib/apt/* `" e riprovare. (Se viene usato `apt-cacher-ng` si dovrebbe anche eseguire "`sudo rm -rf /var/cache/apt-cacher-ng/* `".)

2.6.3 Ripristinare con il comando dpkg

Dato che `dpkg` è un pacchetto di strumenti a livello molto basso, può funzionare senza connessione di rete.

Diciamo che il pacchetto `pippo` è difettoso e deve essere corretto.

Si possono trovare copie di versioni più vecchie del pacchetto `pippo`, libere dal bug, nella directory della cache dei pacchetti: «`/var/cache/apt/archives/`». (Se non ci sono, se ne può scaricare una dall'archivio di https://snapshot.debi o copiarla dalla cache dei pacchetti di una macchina funzionante.)

Se si può avviare il sistema, lo si può installare con il comando seguente.

```
# dpkg -i /path/to/foo_old_version_arch.deb
```

Se il tentativo di installare un pacchetto in questo modo fallisce a causa di qualche violazione di dipendenze e si deve necessariamente farlo come ultima spiaggia, si possono sovrascrivere le dipendenze usando "`--ignore-depends`", "`--force-depends`" e altre opzioni di `dpkg`. Se lo si fa, è necessario in seguito fare tutti gli sforzi possibili per ripristinare le dipendenze corrette. Per i dettagli vedere dpkg(8).

Nota
Se il sistema è seriamente danneggiato, se ne dovrebbe fare un backup completo in un posto sicuro (vedere Sezione 10.2) e fare un'installazione pulita. Ciò comporta meno perdita di tempo e alla fine produce un risultato migliore.

Suggerimento
Se il danno al sistema è piccolo, si può in alternativa retrocedere tutto il sistema ad una versione precedente come in Sezione 2.7.11, usando il sistema di più alto livello APT.

2.6.4 Installazione fallita a causa di dipendenze mancanti

Se si forza l'installazione di un pacchetto con "`sudo dpkg -i ...`" in un sistema che non ha tutti i pacchetti nelle dipendenze installati, allora l'installazione del pacchetto fallirà come parzialmente installato.

Si dovrebbero installare tutti i pacchetti di dipendenza usando ripetutamente "`sudo dpkg -i ...`" o usando quanto segue:

```
# apt --fix-broken install
```

Poi, configurare tutti i pacchetti parzialmente installati con il comando seguente.

```
# dpkg --configure -a
```

2.6.5 Pacchetti diversi con file sovrapposti

I sistemi di gestione dei pacchetti a livello dell'archivio, come aptitude(8) o apt-get(1) non provano nemmeno ad installare pacchetti con file sovrapposti, basandosi sulle dipendenze dei pacchetti (vedere Sezione 2.1.7).

Errori dei manutentori dei pacchetti o l'uso di fonti degli archivi mescolate in modo incoerente (vedere Sezione 2.7.6) dall'amministratore di sistema possono creare una situazione con dipendenze dei pacchetti non correttamente definite. Quando, in una situazione di questo tipo, si installa un pacchetto con file sovrapposti usando aptitude(8) o apt-get(1), dpkg(1) che spacchetta i pacchetti si assicura di restituire un errore al programma che lo ha invocato senza sovrascrivere file esistenti.



Attenzione
L'uso di pacchetti di terze parti introduce significativi rischi per il sistema attraverso gli script dei manutentori che sono eseguiti con privilegi di root e che possono fare qualsiasi cosa nel sistema. Il comando dpkg(1) protegge solo contro la sovrascrittura in fase di spacchettamento.

Si possono risolvere installazioni problematiche di questo tipo rimuovendo prima il vecchio pacchetto che crea problemi: *`vecchio-pacchetto`*.

```
$ sudo dpkg -P old-package
```

2.6.6 Risolvere problemi negli script di pacchetto

Quando un comando nello script di un pacchetto restituisce un errore per una qualche ragione e lo script finisce con un errore, il sistema di gestione dei pacchetti cancella l'azione e finisce con pacchetti parzialmente installati. Quando un pacchetto contiene bug nei suoi script di rimozione, il pacchetto può diventare impossibile da rimuovere e fonte di problemi.

Per i problemi di script di pacchetto di "*`nome_pacchetto`*", si dovrebbe guardare negli script di pacchetto seguenti.

- "`/var/lib/dpkg/info/`*`nome_pacchetto`*`.preinst`"
- "`/var/lib/dpkg/info/`*`nome_pacchetto`*`.postinst`"
- "`/var/lib/dpkg/info/`*`nome_pacchetto`*`.prerm`"
- "`/var/lib/dpkg/info/`*`nome_pacchetto`*`.postrm`"

Modificare dall'account root lo script di pacchetto che causa problemi usando le tecniche seguenti.

- disabilitare la riga che causa problemi facendola precedere da "#"
- forzare la restituzione di esecuzione con successo aggiungendo in fondo alla riga che causa problemi "`|| true`"

Poi, configurare tutti i pacchetti parzialmente installati con il comando seguente.

```
# dpkg --configure -a
```

2.6.7 Ripristinare i dati sui pacchetti selezionati

Se "`/var/lib/dpkg/status`" diventa corrotto per una qualsiasi ragione, il sistema Debian perde i dati sulla selezione dei pacchetti e ne ha grave danno. Cercare il vecchio file "`/var/lib/dpkg/status`" in "`/var/lib/dpkg/status-old`" o "`/var/backups/dpkg.status.*`".

Mantenere "`/var/backups/`" in una partizione separata può essere una buona idea dato che questa directory contiene molti dati importanti per il sistema.

Per danni molto seri, raccomando di fare una re-installazione pulita dopo aver fatto il backup del sistema. Anche se tutto ciò che era in "`/var/`" è andato perduto, è sempre possibile recuperare alcune informazioni dalle directory in "`/usr/share/doc/`" per guidare le nuove installazioni.

Reinstallare un sistema (desktop) minimale

```
# mkdir -p /path/to/old/system
```

Montare il vecchio sistema in "`percorso/del/vecchio/sistema/`".

```
# cd /path/to/old/system/usr/share/doc
# ls -1 >~/ls1.txt
# cd /usr/share/doc
# ls -1 >>~/ls1.txt
# cd
# sort ls1.txt | uniq | less
```

Verranno mostrati i nomi dei pacchetti da installare. (Ci potrebbero essere alcuni nomi non di pacchetto, come "`texmf`".)

2.7 Suggerimenti per la gestione dei pacchetti

Per semplicità, gli esempi di **elenco delle fonti** in questa sezione sono presentati come "`/etc/apt/sources.list`" in stile a una riga dopo il rilascio `trixie`.

2.7.1 Chi ha caricato il pacchetto?

Anche se i nomi dei manutentori elencati in "`/var/lib/dpkg/available`" e"`/usr/share/doc/package_name/chang` forniscono alcune informazioni su "chi stia dietro la pacchettizzazione", l'identità di chi ha effettivamente caricato il pacchetto è un po' oscura. who-uploads(1) nel pacchetto `devscripts` identifica l'effettivo autore del caricamento dei pacchetti sorgenti Debian.

2.7.2 Limitare l'uso di banda di APT per gli scaricamenti

Se si desidera limitare l'uso della banda da parte di APT per gli scaricamenti, per esempio a 800Kib/sec (=100kiB/sec), si devono usare i parametri di configurazione di APT nel modo seguente.

```
APT::Acquire::http::Dl-Limit "800";
```

2.7.3 Scaricare ed aggiornare automaticamente i pacchetti

Il pacchetto `apt` viene fornito con il proprio script di cron, "`/etc/cron.daily/apt`" per permettere di scaricare automaticamente i pacchetti. Questo script può essere migliorato, installando il pacchetto `unattended-upgrades`, per effettuare l'aggiornamento automatico dei pacchetti. Può essere personalizzato dai parametri in "`/etc/apt/apt.conf.d/` e "`/etc/apt/apt.conf.d/50unattended-upgrades`" come descritto in "`/usr/share/doc/unattended-upgrade`

Il pacchetto "`unattended-upgrades` è pensato principalmente per gli aggiornamenti di sicurezza del sistema `stable`. Se il rischio di danneggiare un sistema `stable` esistente con gli aggiornamenti automatici è minore di quello che il sistema venga danneggiato da un intruso, usando una falla nella sicurezza chiusa dall'aggiornamento di sicurezza, si dovrebbe considerare l'uso di questi aggiornamenti automatici con parametri simili ai seguenti.

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "1";
```

Se si sta usando un sistema `testing` o `unstable`, non è bene usare gli aggiornamenti automatici perché certamente prima o poi danneggeranno il sistema. Anche per `testing` o `unstable`, per risparmiare tempo nell'aggiornamento interattivo, si può volere scaricare i pacchetti in anticipo usando parametri di configurazione come i seguenti.

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "0";
```

2.7.4 Updates e Backports

Esistono archivi stable-updates ("`trixie`-updates" durante il ciclo di rilascio `trixie`-as-`stable`) e backports.debian.org che forniscono pacchetti di aggiornamento per `stable`.

Per usare questi archivi, si devono elencare tutti quelli richiesti nel file "`/etc/apt/sources.list`" nel modo seguente.

```
deb http://deb.debian.org/debian/ trixie main non-free-firmware contrib non-free
deb http://security.debian.org/debian-security trixie-security main non-free-firmware ↩
    contrib non-free
deb http://deb.debian.org/debian/ trixie-updates main non-free-firmware contrib non-free
deb http://deb.debian.org/debian/ trixie-backports main non-free-firmware contrib non-free
```

Non è necessario impostare esplicitamente un valore di priorità di pin nel file "`/etc/apt/preferences`". Quando nuovi pacchetti diventano disponibili, la configurazione predefinita fornisce gli aggiornamenti più ragionevoli (vedere Sezione 2.5.3).

- Tutti i vecchi pacchetti installati vengono aggiornati con quelli più nuovi da `trixie-updates`.
- Solo i vecchi pacchetti installati manualmente da `trixie-backports` vengono aggiornati a quelli più nuovi da `trixie-backports`.

Ogni volta che si desidera installare manualmente un pacchetto chiamato "*`nome-pacchetto`*" con le sue dipendenze dall'archivio `trixie-backports`, usare questo comando che sposta il rilascio stabilito con l'opzione "`-t`".

```
$ sudo apt-get install -t trixie-backports package-name
```



avvertimento
Non installare tropppi pacchetti da archivi backports.debian.org. Ciò può causare complicazioni nelle dipendenze dei pacchetti. Vedere Sezione 2.1.11 per soluzioni alternative.

2.7.5 Archivi di pacchetti esterni



avvertimento
Si dovrebbe tenere a mente che pacchetti esterni ottengono privilegi di root nel sistema. Si dovrebbero usare solamente archivi fidati per i pacchetti esterni. Vedere Sezione 2.1.11 per soluzioni alternative.

Si può usare secure APT con un archivio di pacchetti esterni compatibile con Debian, aggiungendolo nell'**elenco delle fonti** e mettendo il suo file della chiave dell'archivio nella directory "`/etc/apt/trusted.gpg.d/`". Vedere sources.list(5), apt-secure(8) e apt-key(8).

2.7.6 Pacchetti da fonti miste di archivi senza l'uso dei pin di APT.



Attenzione
Installare pacchetti da fonti mescolate di archivi non è supportato dalla distribuzione Debian ufficiale, tranne che per particolari combinazioni supportate ufficialmente, come `stable` con security updates e stable-updates.

Ecco un esempio di operazioni da fare per includere, sporadicamente, versioni originali più recenti di specifici pacchetti presenti in `unstable` pur mantenendo `testing`.

1. Cambiare temporaneamente il file "`/etc/apt/sources.list`" ad una singola voce "`unstable`".
2. Eseguire "`aptitude update`".
3. Eseguire "`aptitude install` *`nome-pacchetto`*".
4. Ripristinare il file "`/etc/apt/sources.list`" originale per `testing`.
5. Eseguire "`aptitude update`".

Con questo approccio manuale non si deve creare il file "`/etc/apt/preferences`" né ci si deve preoccupare dei **pin di apt**. Ma è piuttosto laborioso.



Attenzione
Quando si usano fonti di archivi mescolate, si deve assicurare la compatibilità tra i pacchetti da soli, dato che Debian non la garantisce. Se esistono incompatibilità tra i pacchetti, si può danneggiare il sistema. Si deve essere in grado di giudicare queste esigenze tecniche. L'uso di fonti mescolate di archivi non correlati è una modalità di funzionamento del tutto facoltativa e il suo uso non è per nulla raccomandato.

Le regole generali per installare pacchetti da archivi diversi sono le seguenti.

- L'installazione di pacchetti non binari ("`Architecture: all`") è **più sicura**.
  - pacchetti di documentazione: nessun requisito particolare
  - pacchetti programma per interpreti: deve essere disponibile un interprete compatibile
- L'installazione di pacchetti binari (non "`Architecture: all`") di solito comporta molte difficoltà e **non è sicura**.
  - compatibilità di versione con librerie (inclusa "`libc`")
  - compatibilità di versione con programmi applicativi correlati
  - compatibilità con l'ABI del kernel
  - compatibilità con l'ABI C++
  - …

Nota
Per rendere l'installazione di un pacchetto **più sicura**, possono essere forniti alcuni pacchetti binari applicativi commerciali non liberi con librerie completamente statiche. Si dovrebbe anche per essi controllare in ogni caso la compatibilità con ABI, ecc.

Nota
Tranne nel caso di evitare per un breve periodo pacchetti non funzionanti, l'installazione di pacchetti binari da archivi non Debian è generalmente una cattiva idea Si dovrebbero valutare tutte le soluzioni tecniche alternative più sicure che sono compatibili con il proprio sistema Debian attuale (vedere Sezione 2.1.11).

2.7.7 Modificare la versione candidata con l'uso dei pin di APT.



avvertimento
L'uso dei **pin di apt** da parte di un utente inesperto è una fonte sicura di grossi problemi. Si dovrebbe evitare l'uso di questa tecnica, tranne nei casi in cui sia strettamente necessario. Vedere Sezione 2.1.11 per soluzioni alternative.

Senza il file "`/etc/apt/preferences`", il sistema APT sceglie la versione più recente disponibile come **versione candidata** usando la stringa di versione. Questa è la situazione normale e l'uso raccomandato del sistema APT. Tutte le combinazioni di archivi supportate ufficialmente non necessitano del file "`/etc/apt/preferences`", dato che gli archivi che non dovrebbero essere usati come fonti automatica per gli aggiornamenti sono marcati come **NotAutomatic** e trattati in modo appropriato.

Suggerimento
La regola di comparazione della stringa di versione può essere verificata con, ad esempio, "`dpkg --compare-versions ver1.1 gt ver1.1~1; echo $?`" (vedere dpkg(1)).

Quando si installano regolarmente pacchetti da fonti di archivi mescolate (vedere Sezione 2.7.6), si possono automatizzare queste operazioni complicate creando il file "`/etc/apt/preferences`" con le voci appropriate e modificando la regola di selezione dei pacchetti per le **versioni candidate** come descritto in apt_preferences(5). Questo è chiamato **usare i pin di APT**.

Quando si usano i **pin di apt** si deve assicurare la compatibilità dei pacchetti da soli dato che Debian non la garantisce. L'uso dei **pin di apt** è una modalità di funzionamento del tutto facoltativa e il suo uso non è per nulla consigliato.

Per le regole di apt_preferences(5) sono usati i file Release a livello di archivio (vedere Sezione 2.5.3). I **pin di apt** perciò funzionano solo con il nome della "suite" per i normali archivi Debian e per gli archivi di sicurezza Debian. (Ciò è diverso per gli archivi Ubuntu.) Per esempio nel file "`/etc/apt/preferences`" si può usare "`Pin: release a=unstable`", ma non "`Pin: release a=sid`".

Quando si usano archivi non Debian con i **pin di apt** si dovrebbe controllare quale sia il loro scopo e la loro credibilità. Ubuntu e Debian, per esempio, non sono fatti per essere mescolati.

Nota
Anche se non si crea il file "`/etc/apt/preferences`" si possono fare operazioni piuttosto complesse (vedere Sezione 2.6.3 e Sezione 2.7.6) senza l'uso dei **pin di apt**.

Quella che segue è una spiegazione semplificata dell**uso dei pin di APT**.

Il sistema APT sceglie, dalle fonti per i pacchetti disponibili definite nel file "`/etc/apt/sources.list`", il pacchetto di **aggiornamento** con la priorità di pin più elevata come **versione candidata** del pacchetto. Se la priorità di pin del pacchetto è maggiore di 1000, la restrizione a versioni di **aggiornamento** non viene considerata per permettere la retrocessione ad una versione precedente (vedere Sezione 2.7.11).

Il valore della priorità di pin per ciascun pacchetto è definito dalle voci "Pin-Priority" nel file "`/etc/apt/preferences`" o viene usato il valore predefinito.

L'archivio del **rilascio obiettivo** può essere impostato con l'opzione per la riga di comando, ad esempio, "`apt-get install -t testing qualche-pacchetto`"

Gli attributi di archivio **NonAutomatico** e **MaConAggiornamentiAutomatici** vengono impostati dal server dell'archivio che ha il file Release a livello di archivio (vedere Sezione 2.5.3) che contiene sia "`NotAutomatic: yes`" sia "`ButAutomaticUpgrades: yes`". L'archivio **NonAutomatico** viene impostato dal server dell'archivio che ha il file Release a livello di archivio che contiene solamente "`NotAutomatic: yes`".

La **situazione dei pin di APT** di *pacchetto* da diverse fonti di archivi viene mostrata con "`apt-cache policy pacchetto`".

Priorità di pin	Effetti del pin di apt sul pacchetto
1001	installa il pacchetto anche se ciò costituisce il ritorno ad una versione precedente
990	usato come valore predefinito per l'archivio del **rilascio prescelto**
500	usato come valore predefinito per l'archivio **normale**
100	usato come valore predefinito per l'archivio **NonAutomatico** e **MaConAggiornamentiAutomatici**
100	usato per i pacchetti **installati**
1	usato come valore predefinito per l'archivio **NonAutomatico**
-1	**non installa mai** il pacchetto nemmeno se raccomandato

Tabella 2.18: Elenco di valori di priorità di pin che esemplificano la tecnica d'**uso dei pin di APT**.

- Una riga che inizia con "`pacchetto pin:`" elenca la versione del pacchetto di **pin** se è definita un'associazione specifica per *pacchetto*, ad esempio "`Package pin: 0.190`".
- Non esiste alcuna riga con "`Package pin:`" se non è definita alcuna associazione solamente con *pacchetto*.
- Il valore di priorità di pin associato solamente con *pacchetto* viene mostrato alla destra di tutte le sringhe di versione, ad esempio "`0.181 700`".
- Se non è definita un'associazione specifica con *pacchetto* viene mostrato "`0`" alla destra di tutte le stringhe di versione, come ad esempio in "`0.181 0`".
- I valori di priorità di pin degli archivi (definiti come "`Package: *`" nel file "`/etc/apt/preferences`") sono elencati alla sinistra di tutti i percorsi di archivio, ad esempio "`100 http://deb.debian.org/debian/trixie-backports/ma Packages`".

2.7.8 Bloccare i pacchetti da installare perché "Raccomandati"



avvertimento
L'uso dei **pin di apt** da parte di un utente inesperto è una fonte sicura di grossi problemi. Si dovrebbe evitare l'uso di questa tecnica, tranne nei casi in cui sia strettamente necessario. Vedere Sezione 2.1.11 per soluzioni alternative.

Se non si desidera installare pacchetti particolari richiamati automaticamente perché "Raccomandati", è necessario creare il file "`/etc/apt/preferences`" ed elencarvi esplicitamente all'inizio tutti quei pacchetti nel modo seguente.

```
Package: package-1
Pin: version *
Pin-Priority: -1

Package: package-2
Pin: version *
Pin-Priority: -1
```

2.7.9 Usare e aggiornare `testing` con alcuni pacchetti da `unstable`



avvertimento
L'uso dei **pin di apt** da parte di un utente inesperto è una fonte sicura di grossi problemi. Si dovrebbe evitare l'uso di questa tecnica, tranne nei casi in cui sia strettamente necessario. Vedere Sezione 2.1.11 per soluzioni alternative.

Quello che segue è un esempio di **uso dei pin di APT** per includere pacchetti specifici con versioni originali più recenti presenti in `unstable` aggiornandoli regolarmente pur mantenendo `testing`. Si devono elencare nel file "`/etc/apt/sources.list`" tutti gli archivi necessari nel modo seguente.

```
deb http://deb.debian.org/debian/ testing main contrib non-free
deb http://deb.debian.org/debian/ unstable main contrib non-free
deb http://security.debian.org/debian-security testing-security main contrib
```

Impostare il file "`/etc/apt/preferences`" come segue.

```
Package: *
Pin: release a=unstable
Pin-Priority: 100
```

Quando, con questa configurazione, si desidera installare un pacchetto chiamato "*`nome-pacchetto`*" con le sue dipendenze dall'archivio `unstable`, usare il comando seguente che cambia il rilascio definito con l'opzione "`-t`" (la priorità di pin di `unstable` diventa 990).

```
$ sudo apt-get install -t unstable package-name
```

Con quessta configurazione, la normale esecuzione di "`apt-get upgrade`" e "`apt-get dist-upgrade`" (o "`aptitude safe-upgrade`" e "`aptitude full-upgrade`") aggiorna i pacchetti che erano stati installati dall'archivio `testing` usando l'attuale archivio `testing` e i pacchetti che erano stati installati dall'archivio `unstable` usando l'attuale archivio `unstable`.



Attenzione
Fare attenzione a non rimuovere la voce "`testing`" dal file "`/etc/apt/sources.list`". Senza la voce "`testing`" in tale file, il sistema APT aggiorna i pacchetti usando l'archivio `unstable` più recente.

Suggerimento
Io di solito modifico il file "`/etc/apt/sources.list` per commentare le voci relative all'archivio "`unstable` immediatamente dopo l'azione descritta sopra. Questo evita il rallentamento del processo di aggiornamento dovuto a troppe voci nel file "`/etc/apt/sources.list`" anche se impedisce l'aggiornamento, usando l'attuale archivio `unstable` dei pacchetti che erano stati installati dall'archivio `unstable`.

Suggerimento
Se viene usato "`Pin-Priority: 1`" invece di "`Pin-Priority: 100`" nel file "`/etc/apt/preferences`", i pacchetti già installati che hanno 100 come valore della priorità di pin, non vengono aggiornati dall'archivio `unstable` nemmeno se viene rimossa la voce "`testing`" dal file "`/etc/apt/sources.list`".

Se si desidera tenere traccia automaticamente di particolari pacchetti in `unstable` senza l'iniziale installazione con "`-t unstable`", si deve creare il file "`/etc/apt/preferences`" ed elencarvi esplicitamente tutti i pacchetti voluti nel modo seguente.

```
Package: package-1
Pin: release a=unstable
Pin-Priority: 700

Package: package-2
Pin: release a=unstable
Pin-Priority: 700
```

Questo imposta il valore della priorità di pin per ciascun pacchetto specifico. Per esempio, per tenere traccia della versione più recente in `unstable` di questo manuale "Debian Reference" in inglese, si dovrebbero avere, nel file "`/etc/apt/preferences`" le voci seguenti.

```
Package: debian-reference-en
Pin: release a=unstable
Pin-Priority: 700

Package: debian-reference-common
Pin: release a=unstable
Pin-Priority: 700
```

Suggerimento
Questo modo di usare i **pin di apt** è valido anche quando si segue l'archivio `stable`. Per l'esperienza dell'autore, installare i pacchetti di documentazione dall'archivio `unstable` è sempre stato sicuro, fino ad ora.

2.7.10 Usare e aggiornare `unstable` con alcuni pacchetti da `experimental`



avvertimento
L'uso dei **pin di apt** da parte di un utente inesperto è una fonte sicura di grossi problemi. Si dovrebbe evitare l'uso di questa tecnica, tranne nei casi in cui sia strettamente necessario. Vedere Sezione 2.1.11 per soluzioni alternative.

Quello che segue è un esempio di **uso dei pin di APT** per includere pacchetti specifici con versioni originali più recenti presenti in `experimental` pur mantenendo `unstable`. Si devono elencare nel file "`/etc/apt/sources.list`" tutti gli archivi necessari nel modo seguente.

```
deb http://deb.debian.org/debian/ unstable main contrib non-free
deb http://deb.debian.org/debian/ experimental main contrib non-free
deb http://security.debian.org/ testing-security main contrib
```

Il valore predefinito della priorità di pin per l'archivio `experimental` è sempre 1 (<<100) dato che è un archivio **Non automatico** (vedere Sezione 2.5.3). Per usare semplicemente l'archivio `experimental` non c'è alcun bisogno di impostare il valore di priorità di pin esplicitamente nel file "`/etc/apt/preferences`", a meno che non si desideri aggiornare automaticamente da esso in futuro particolari pacchetti.

2.7.11 Retrocessione di emergenza alla versione precedente



avvertimento
L'uso dei **pin di apt** da parte di un utente inesperto è una fonte sicura di grossi problemi. Si dovrebbe evitare l'uso di questa tecnica, tranne nei casi in cui sia strettamente necessario. Vedere Sezione 2.1.11 per soluzioni alternative.



Attenzione
La retrocessione ad una versione precedente non è ufficialmente supportata da Debian. Dovrebbe essere fatta solamente come parte di un processo di ripristino di emergenza. Nonostante questo è noto che funziona bene in caso di molti incidenti. Per i sistemi critici, si dovrebbe fare un backup di tutti i dati importanti sul sistema dopo l'operazione di ripristino e reinstallare un nuovo sistema da zero.

Si potrebbe essere fortunati e riuscire a retrocedere da un archivio più recente ad uno più vecchio per ripristinare un aggiornamento di sistema andato male, manipolando le **versioni candidate** (vedere Sezione 2.7.7). Questa è l'alternativa pigra al lavoro tedioso di dover usare molti comandi "`dpkg -i pacchetto-danneggiato_vecchia-versione.deb`" (vedereSezione 2.6.3).

Cercare le righe nel file "`/etc/apt/sources.list`" che rimandano a `unstable`, come la seguente.

```
deb http://deb.debian.org/debian/ sid main contrib non-free
```

Sostituirle con la seguente per puntare a `testing`.

```
deb http://deb.debian.org/debian/ forky main contrib non-free
```

Impostare il file "`/etc/apt/preferences`" come segue.

```
Package: *
Pin: release a=testing
Pin-Priority: 1010
```

Eseguire "`apt-get update; apt-get dist-upgrade`" per forzare la retrocessione dei pacchetti in tutto il sistema.

Rimuovere questo file speciale "`/etc/apt/preferences`" dopo la retrocessione di emergenza.

Suggerimento
È una buona idea rimuovere (non eliminare completamente!) il maggior numero di pacchetti per minimizzare i problemi di dipendenza. Potrebbe essere necessario rimuovere o installare manualmente alcuni pacchetti per retrocedere il sistema. Il kernel Linux, il bootloader, udev, PAM, APT e i pacchetti relativi alla rete ed i loro file di configurazione richiedono particolare attenzione.

2.7.12 Il pacchetto equivs

Se si ha intenzione di compilare un programma dai sorgenti per rimpiazzare un pacchetto Debian, la cosa migliore è creare un vero pacchetto debianizzato locale (`*.deb`) e usare un archivio privato.

Se si sceglie di compilare un programma dai sorgenti e di installarlo invece in "`/usr/local`", si può usare `equivs` come ultima spiaggia per soddisfare le dipendenze mancanti del pacchetto.

```
Package: equivs
Priority: optional
Section: admin
Description: Circumventing Debian package dependencies
 This package provides a tool to create trivial Debian packages.
 Typically these packages contain only dependency information, but they
 can also include normal installed files like other packages do.
 .
 One use for this is to create a metapackage: a package whose sole
 purpose is to declare dependencies and conflicts on other packages so
 that these will be automatically installed, upgraded, or removed.
 .
 Another use is to circumvent dependency checking: by letting dpkg
 think a particular package name and version is installed when it
 isn't, you can work around bugs in other packages' dependencies.
 (Please do still file such bugs, though.)
```

2.7.13 Fare il port di un pacchetto nel sistema stabile



Attenzione
Non esistono garanzie che la procedura descritta qui funzioni senza dover fare altro lavoro manuale date le differenze dei sistemi.

Per aggiornamenti parziali del sistema `stable`, è una buona idea ricompilare un pacchetto, usando il pacchetto sorgente, all'interno del suo ambiente. Questo evita massicci aggiornamenti di pacchetti a causa delle dipendenze.

Aggiungere le voci seguenti al file "`/etc/apt/sources.list`" di un sistema `stable`.

```
deb-src http://deb.debian.org/debian unstable  main contrib non-free
```

Installare i pacchetti richiesti per la compilazione e scaricare il pacchetto sorgente, come nell'esempio seguente.

```
# apt-get update
# apt-get dist-upgrade
# apt-get install fakeroot devscripts build-essential
# apt-get build-dep foo
$ apt-get source foo
$ cd foo*
```

Aggiornare alcuni pacchetti con catene di strumenti come `dpkg` e `debhelper` dai pacchetti backport se sono necessari per fare il backport.

Eseguire quanto segue.

```
$ dch -i
```

Incrementare la versione del pacchetto, ad esempio aggiungendo "`+bp1`" in "`debian/changelog`"

Compilare i pacchetti ed installarli nel sistema nel modo seguente.

```
$ debuild
$ cd ..
# debi foo*.changes
```

2.7.14 Server proxy per APT

Dato che fare il mirror di intere sottosezioni dell'archivio Debian spreca spazio su disco e banda, l'utilizzo di un server proxy locale per APT è una buona idea quando si amministrano molti sistemi in una LAN. APT può essere configurato per usare server proxy web generici (http), quale `squid` (vedere Sezione 6.5) come descritto in apt.confo(5) ed in "`/usr/share/doc/apt/examples/configure-index.gz`". La variabile d'ambiente "`$http_proxy`" può essere usata per scavalcare l'impostazione del server proxy nel file "`/etc/apt/apt.conf`".

Ci sono strumenti proxy specifici per l'archivio Debian. Prima di usarli si dovrebbe contollare il BTS.

pacchetto	popcon	dimensione	descrizione
`apt-cacher`	V:0.28, I:0.33	265	proxy con cache per file di pacchetto e sorgenti Debian (programma Perl)
`apt-cacher-ng`	V:3.9, I:4.1	2063	proxy con cache per la distribuzione di pacchetti software (programma in C++ compilato)

Tabella 2.19: Elenco degli strumenti proxy specifici per l'archivio Debian



Attenzione
Quando Debian riorganizza la struttura del suo archivio, questi strumenti proxy specializzati tendono ad aver bisogno di ritocchi al codice da parte del manutentore del pacchetto e possono non essere funzionanti per un po' di tempo. D'altra parte i server proxy web generici (http) sono più robusti e gestiscono questi cambiamenti più facilmente.

2.7.15 Ulteriori letture sulla gestione dei pacchetti

Si possono ottenere molte altre informazioni sulla gestione dei pacchetti dalla documentazione seguente.

- Documentazione principale sulla gestione dei pacchetti:
  - aptitude(8), dpkg(1), tasksel(8), apt(8), apt-get(8), apt-config(8), apt-secure(8), sources.list(5), apt.conf(5), and apt_preferences(5);
  - "`/usr/share/doc/apt-doc/guide.html/index.html`" e "`/usr/share/doc/apt-doc/offline.html/inde`" dal pacchetto `apt-doc` e
  - "`/usr/share/doc/aptitude/html/en/index.html`" dal pacchetto `aptitude-doc-en`.
- Documentazione ufficiale dettagliata sull'archivio Debian:
  - "Manuale Debian Policy, capitolo 2 - L'archivio Debian",
  - "Debian Developer's Reference, capitolo 4 - Risorse per gli sviluppatori Debian, 4.6 L'archivio Debian" e
  - "FAQ Debian GNU/Linux, capitolo 6 - Gli archivi FTP Debian".
- Tutorial per gli utenti Debian sulla creazione di un pacchetto Debian:
  - «Guida per i manutentori Debian».

Capitolo 3

Inizializzazione del sistema

È bene che l'amministratore di sistema conosca almeno a grandi linee come viene avviato e configurato il sistema Debian. Anche se i dettagli precisi sono nei file sorgenti dei pacchetti installati e nella loro documentazione, essi sono un po' troppo per la maggior parte degli utenti.

Ecco una panoramica di base dei punti principali dell'inizializzazione di un sistema Debian. Dato che un sistema Debian è in costante evoluzione, si dovrebbe far riferimento alla documentazione più recente.

- Debian Linux Kernel Handbook è la fonte principale di informazioni sul kernel Debian.
- bootup(7) descrive il processo di avvio del sistema basato su `systemd` (Debian recenti).
- boot(7) descrive il processo di avvio del sistema basato su UNIX System V Release 4 (Debian più vecchie).

3.1 Panoramica del processo di avvio

Il sistema del computer passa attraverso varie fasi del processo di avvio, dall'accensione a quando offre all'utente il sistema operativo (SO) pienamente funzionante.

Per semplicità la spiegazione è limitata alla piattaforma PC tipica con l'installazione standard.

Il normale processo di avvio è come un razzo a quattro stadi. Ogni stadio del razzo passa il controllo del sistema allo stadio successivo.

- Sezione 3.1.1
- Sezione 3.1.2
- Sezione 3.1.3
- Sezione 3.1.4

Naturalmente questo può essere configurato in modo diverso. Per esempio, se è stato compilato un kernel personalizzato, si potrebbe saltare la fase con il mini-sistema Debian. Non dare per scontato che quanto detto valga per il proprio sistema fino a che non si abbia controllato direttamente.

3.1.1 Stadio 1: l'UEFI

La Unified Extensible Firmware Interface (UEFI) definisce un gestore di avvio come parte della specifica UEFI. Quando un computer viene acceso, il gestore dell'avvio è il primo stadio del processo di avvio che controlla la configurazione di avvio e, in base alle sue impostazioni, esegue poi il boot loader del sistema operativo specificato o il kernel del

sistema operativo (solitamente il boot loader). La configurazione di avvio è definita dalle variabili memorizzate nella NVRAM, incluse variabili che indicano i percorsi del file system dei caricatori dei SO o dei kernel dei SO.

Una partizione di sistema EFI (ESP, EFI System Partition) è una partizione di dispositivo per archiviazione di dati che è utilizzata nei computer che sono conformi alla specifica UEFI. Il firmware UEFI accede ad essa quando un computer viene acceso, essa memorizza le applicazioni UEFI e i file che tali applicazioni devono eseguire, inclusi i boot loader dei sistemi operativi. (Nei sistemi PC vecchi può essere invece utilizzato il BIOS memorizzato nella MBR.)

3.1.2 Stadio 2: il bootloader

Il bootloader è il secondo stadio del processo di avvio che è iniziato dall'UEFI. Carica l'immagine kernel del sistema e l'immagine initrd in memoria e passa il controllo ad esse. L'immagine initrd è l'immagine del file system radice ed il suo supporto dipende dal bootloader usato.

Il sistema Debian normalmente usa il kernel Linux xome kernel di sistema predefinito. L'immagine initrd per l'attuale kernel Linux 5.x è tecnicamente l'immagine initramfs (Initial RAM filesystem, file system RAM iniziale).

Sono disponibili molti bootloader e opzioni di configurazione.

pacchetto	popcon	dimensione	initrd	bootloader	descrizione
grub-efi-amd64	I:466	142	Supportato	GRUB UEFI	Intelligente abbastanza da capire partizioni su disco e file system come vfat, ext4, ... (UEFI)
grub-pc	V:16, I:512	479	Supportato	GRUB 2	Intelligente abbastanza da capire partizioni su disco e file system come vfat, ext4, ... (BIOS)
grub-rescue-pc	V:0.05, I:0.58	7269	Supportato	GRUB 2	È l'immagine di ripristino avviabile di GRUB 2 (CD e floppy) (versione PC/BIOS)
grml-rescueboot	V:0.1, I:1.0	39	N/D	Plug-in GRUB 2	grml-rescueboot aggiunge immagini ISO Linux Live al menu di avvio di grub2
syslinux	V:3, I:31	325	Supportato	Isolinux	Capisce il filesystem ISO 9660. Usato dal CD di avvio.
syslinux	V:3, I:31	325	Supportato	Syslinux	Capisce il filesystem MSDOS (FAT). Usato dai dischetti floppy di avvio.
loadlin	V:0.03, I:0.39	87	Supportato	Loadlin	Il nuovo sistema viene avviato dal sistema FreeDOS/MSDOS.
mbr	V:0.2, I:2.7	47	Non supportato	MBR di Neil Turton	Software libero che sostituisce MBR MSDOS. Capisce solo le partizioni su disco.

Tabella 3.1: Elenco di bootloader

Per i sistemi UEFI, GRUB2 prima legge la partizione ESP e usa l'UUID specificato per `search.fs_uuid` in "`/boot/efi/EF` per determinare la partizione del file di configurazione del menu di GRUB2 "`/boot/grub/grub.cfg`".

La parte della chiave del file di configurazione del menu di GRUB2 è simile a:

```
menuentry 'Debian GNU/Linux' ... {
        load_video
        insmod gzio
        insmod part_gpt
        insmod ext2
        search --no-floppy --fs-uuid --set=root fe3e1db5-6454-46d6-a14c-071208ebe4b1
        echo    'Loading Linux 5.10.0-6-amd64 ...'
        linux   /boot/vmlinuz-5.10.0-6-amd64 root=UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1 ↩
            ro quiet
```

```
        echo    'Loading initial ramdisk ...'
        initrd  /boot/initrd.img-5.10.0-6-amd64
}
```

Per questa porzione di `/boot/grub/grub.cfg`, questa voce di menu significa quanto segue.

impostazione	valore
moduli GRUB2 caricati	`gzio`, `part_gpt`, `ext2`
partizione per file system radice utilizzata	partizione identificata come `UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1`
percorso dell'immagine del kernel nel file system radice	`/boot/vmlinuz-5.10.0-6-amd64`
parametri di avvio del kernel utilizzati	"`root=UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1 ro quiet`"
percorso dell'immagine initrd nel file system radice	`/boot/initrd.img-5.10.0-6-amd64`

Tabella 3.2: Il significato della voce di menu della porzione soprastante di `/boot/grub/grub.cfg`

Nei sistemi Debian, "`/boot/grub/grub.cfg`" viene gestito dal pacchetto GRUB (es. `grub-efi-amd64`) e modifiche dirette a questo file da parte dell'utente sono deprecate. Si dovrebbero invece personalizzare i file di configurazione in "`/etc/grub.d/`" e "`/etc/default/grub`". Poi configurare i file delle impostazioni di GRUB e aggiornare le variabili NVRAM per avviare automaticamente in Debian con:

```
# dpkg-reconfigure grub-efi-amd64
```

Suggerimento
si può abilitare la visualizzazione dei messaggi del log di avvio del kernel rimuovendo "`quiet`" dal valore "`GRUB_CMDLINE_LINUX_DEFAULT`" in "`/etc/default/grub`".

Suggerimento
Si può aggiungere uno sfondo di avvio per GRUB mettendo il suo file immagine in "`/boot/grub/`".

Vedere "`info grub`", grub-install(8), grub-mkconfig(8).

3.1.3 Stadio 3: il mini-sistema Debian

Il mini-sistema Debian è il terzo stadio del processo di avvio che viene iniziato dal bootloader. Esegue il kernel del sistema con il suo filesystem root in memoria. Questo è uno stadio opzionale preparatorio del processo di avvio.

Nota
L'espressione "sistema Debian mini" è stata coniata per descrivere il terzo stadio del processo di avvio in questo documento. Normalmente ci si riferisce a questo sistema come sistema initrd o initramfs. Un sistema simile in memoria è usato dall'installatore Debian.

Il programma "`/init`" viene eseguito come primo programma in questo file system root in memoria. È un programma che inizializza il kernel in spazio utente e passa il controllo allo stadio successivo. Questo mini-sistema Debian offre flessibilità al processo di avvio, come la possibilità di aggiungere moduli del kernel prima del processo di avvio principale o di montare il file system root come cifrato.

- Il programma "`/init`" è uno script di shell se initramfs è stato creato da `initramfs-tools`.
  - Si può interrompere questa parte del processo di avvio per ottenere una shell di root fornendo il parametro di avvio per il kernel "`break=init`" etc. Vedere lo script "`/init`" per ulteriori condizioni di interruzione. Questo ambiente shell è abbastanza sofisticato da fare una buona ispezione dell'hardware della macchina.
  - I comandi disponibili in questo mini-sistema Debian sono versioni minimali e vengono principalmente forniti da uno strumento GNU chiamato busybox(1).
- Il programma "`/init`" è un programma binario di `systemd` se initramfs è stato creato da `dracut`.
  - I comandi disponibili in questo mini-sistema Debian sono un ambiente systemd(1) ridotto al minimo.



Attenzione
È necessario usare l'opzione "`-n`" per il comando `mount` quando si è nel filesystem root in sola lettura.

3.1.4 Stadio 4: il normale sistema Debian

Il sistema Debian normale è il quarto stadio del processo di avvio che viene iniziato dal mini-sistema Debian. Il kernel di sistema del mini-sistema Debian continua ad essere in esecuzione anche in questo ambiente. Il file system root viene cambiato da quello in memoria a quello sul dispositivo fisico di archiviazione.

Il programma init viene eseguito come primo programma con PID=1 per effettuare il processo principale di avvio di far partire molti programmi. Il percorso di file predefinito per il programma init è «`/usr/sbin/init`», ma può essere cambiato con un parametro di avvio del kernel come in «`init=/percorso/del/programma_init`».

"`/usr/sbin/init`" è un collegamento simbolico a "`/lib/systemd/systemd`" a partire da Debian 8 Jessie (rilasciata nel 2015).

Suggerimento
Si può verificare quale è l'effettivo comando init nel proprio sistema con il comando «`ps --pid 1 -f`».

Suggerimento
Vedere la pagina del Wiki Debian sulla velocizzazione del processo di avvio per i più recenti suggerimenti su come velocizzare il processo di avvio.

3.2 Sistema di ripristino/recupero



avvertimento
Non effettuare compiti di amministrazione del sistema relativi al processo di avvio senza avere un sistema di ripristino/recupero.

La disponibilità di un sistema di recupero permette di effettuare compiti delicati come:

- Avviare un sistema da un'installazione difettosa del boot loader
- Correggere un'installazione difettosa del boot loader

pacchetto	popcon	dimensione	descrizione
systemd	V:920, I:980	11386	demone init(8) basato su eventi per concorrenza (alternativa a sysvinit)
cloud-init	V:4.1, I:6.3	3267	sistema di inizializzazione per istanze di infrastrutture cloud
systemd-sysv	V:915, I:983	76	le pagine di manuale e i collegamenti necessari affinché systemd sostituisca sysvinit
init-system-helpers	V:723, I:988	132	strumenti ausiliari per commutare tra sysvinit e systemd
initscripts	V:15, I:54	197	script per inizializzare ed arrestare il sistema
sysvinit-core	V:3.1, I:3.6	365	utilità init(8) in stile System-V
sysv-rc	V:29, I:57	85	meccanismo di cambiamento del runlevel in stile System-V
sysvinit-utils	V:725, I:1000	100	utilità in stile System-V (startpar(8), bootlogd(8), …)
lsb-base	V:205, I:300	12	funzionalità di script init Linux Standard Base 3.2
insserv	V:31, I:57	132	strumento per organizzare la sequenza di avvio usando dipendenze LSB negli script init.d
kexec-tools	V:1.4, I:5.3	320	strumento kexec per riavvii kexec(8) (riavvio a caldo)
systemd-bootchart	V:0.17, I:0.52	131	analizzatore delle prestazioni del processo di avvio
mingetty	V:0.2, I:2.3	36	getty(8) solo console
mgetty	V:0.13, I:0.47	318	rimpiazzio di getty(8) per smart modem

Tabella 3.3: Elenco di utilità di avvio per il sistema Debian

- Estrarre dati da un sistema difettoso non avviabile
- Modificare file system, partizioni sull disco e volumi LVM che riguardano il file system radice

Tipicamente un sistema di recupero è fornito come un file di immagine ISO e scritto su un supporto di archiviazione rimovibile come:

- un'unità flash USB preparata come in Sezione 9.7.2
- un CD / DVD preparato come in Sezione 9.7.7

Per semplicità sono usati come esempio in seguito i casi con unità flash USB, ma possono essere usati anche CD o DVD.

Suggerimento
Può essere necessario modificare alcune variabili UEFI NVRAM per avviare boot loader arbitrari sul supporto di archiviazione rimovibile.

3.2.1 Sistema di recupero GRUB UEFI su USB

Il sistema di recupero GRUB UEFI con menu può essere avviato accendendo il sistema con inserito il "Sistema di ripristino GRUB UEFI su USB".

Questo "Sistema di ripristino GRUB UEFI su USB" viene preparato scrivendo in anticipo l'immagine ISO Super Grub2 Disk su un'unità flash USB.

Nel caso di una configurazione difettosa del boot loader di un altro sistema operativo, ecc., si può risolvere il problema nel seguente modo:

- Avviare il sistema di ripristino GRUB UEFI per scoprire automaticamente i sistemi avviabili installati.

- Avviare il sistema Debian installato dal menu di GRUB.
- Al prompt di shell Linux di root:

```
# dpkg-reconfigure grub-efi-amd64
```

Nota
L'immagine ISO di ripristino avviabile di GRUB può essere generata anche seguendo "`info grub-mkrescue`". Offre un prompt di shell a riga di comando per GRUB, ma non offre il rilevamento automatico dei sistema installati avviabili.

3.2.2 Sistema di ripristino Linux Live su USB

Il sistema di ripristino Linux Live può essere avviato accendendo il sistema con inserito il "Sistema di ripristino Linux Live su USB".

Questo "Sistema di ripristino Linux Live su USB" può essere preparato scrivendo in anticipo una delle immagini ISO di Linux Live basate su Debian su un'unità flash USB. Ecco di seguito alcuni esempi di tali immagini Linux Live.

- Immagii Debian Live
- Kali Linux Live
- Grml Live Linux

Alcuni casi d'uso di questo sistema di ripristino Linux Live sono:

- Correggere la configurazione difettosa del boot loader causata dall'installazione di un altro sistema operativo, ecc.:
  - Avviare il sistema di ripristino Linux Live.
  - Montare la partizione contenente il file system radice del sistema Debian non avviabile in "`/mnt`".
  - Al prompt di shell Linux di root:

```
# chroot /mnt; dpkg-reconfigure grub-efi-amd64
```

- Corrreggere il pacchetto `dpkg` difettoso:
  - Avviare il sistema di ripristino Linux Live.
  - Montare la partizione contenente il file system radice del sistema Debian installato con il pacchetto `dpkg` difettoso in "`/mnt`".
  - Al prompt di shell Linux di root:

```
# dpkg --root /mnt -i /mnt/var/cache/apt/archives/dpkg_old_version_amd64.deb
```

- Effettuare modifiche normalmente non permesse, come operazioni sul file system del sistema installato (vedere Sezione 9.6).



avvertimento
La schermata della GUI del sistema Linux Live può essere bloccata dopo un periodo di inattività.

Suggerimento

- È una buona idea impostare la propria password non appena si avvia un sistema Linux Live.
- Si può impostare la propria password, ad esempio, usando "`sudo passwd utente`" dal prompt di shell dell'utente sulle console virtuali di Linux (vedere Sezione 1.1.6).
- Alcuni sistemi Linux Live possono impostare i loro "*utente*/*password*" predefiniti: "Debian Live" = "user/live", "Kali Linux Live" = "kali/kali"

3.2.3 Sistema di ripristino Linux Live da GRUB

Il sistema di ripristino Linux Live può essere avviato dalla voce di menu di GRUB. La configurazione di GRUB per farlo è preparata nel modo seguente:

- Installare il pacchetto `grml-rescueboot`
- Trovare un'immagine ISO di Linux Live che ha "`/boot/grub/loopback.cfg`" nella sua immagine.
  - Le immagini ISO elencate in Sezione 3.2.2 hanno "`/boot/grub/loopback.cfg`" nella loro immagine.
- Copiare alcune di queste immagini ISO di Linux Live nella directory "`/boot/grml/`".
- Aggiornare il menu di GRUB con:

```
# dpkg-reconfigure grub-efi-amd64
```

All'accensione del sistema, il menu di GRUB visualizza le opzioni per i sistemi di ripristino Linux Live.

3.3 Systemd

3.3.1 Init systemd

Quando il sistema si avvia, `/usr/sbin/init` che è un collegamento simbolico a `/usr/lib/systemd` viene avviato come processo init del sistema (`PID=1`) con proprietario root (`UID=0`). Vedere systemd(1).

Il processo di init `systemd` lancia processi in parallelo sulla base dei file di configurazione delle unità (vedere systemd.unit(5)) che sono scritti in stile dichiarativo invece che in stile procedurale come per SysV.

I processi avviati sono messi in singoli gruppi di controllo (control group) Linux che prendono il nome dall'unità a cui appartengono nella gerarchia privata di systemd (vedere cgroups e Sezione 4.7.5).

Le unità per le modalità di sistema sono caricate dal "System Unit Search Path" come descritto in systemd.unit(5). Quelle principali sono, in ordine di importanza, le seguenti:

- "`/etc/systemd/system/*`": unità di sistema create dall'amministratore
- "`/run/systemd/system/*`": unità runtime
- "`/lib/systemd/system`": unità di sistema installate dal gestore dei pacchetti della distribuzione

Le loro inter-dipendenze sono specificate dalle direttive "`Wants=`", "`Requires=`", "`Before=`", "`After=`", ... (vedere "MAPPING OF UNIT PROPERTIES TO THEIR INVERSES" in systemd.unit(5)). Sono definiti anche i controlli delle risorse (vedere systemd.resource-control(5)).

Il suffisso dei file di configurazione delle unità codifica il loro tipo in questo modo:

- ***.service** descrive un processo controllato e supervisionato da `systemd`. Vedere systemd.service(5).
- ***.device** descrive un device esposto in sysfs(5) come albero di device udev(7). Vedere systemd.device(5).
- ***.mount** descrive un punto di mount del file system controllato e supervisionato da `systemd`. Vedere systemd.mount(5).
- ***.automount** descrive un punto di mount automatico del file system controllato e supervisionato da `systemd`. Vedere systemd.automount(5).
- ***.swap** descrive un device o file di swap controllato e supervisionato da `systemd`. Vedere systemd.swap(5).
- ***.path** descrive un percorso monitorato da `systemd` per l'attivazione basata su percorso. Vedere systemd.path(5).
- ***.socket** descrive un socket controllato e supervisionato da `systemd` per l'attivazione basata su socket. Vedere systemd.socket(5).
- ***.timer** descrive un timer controllato e supervisionato da `systemd` per l'attivazione basata su timer. Vedere systemd.timer(5).
- ***.slice** gestisce risorse con cgroups(7). Vedere systemd.slice(5).
- ***.scope** viene creato programmaticamente usando le interfacce di bus di `systemd` per gestire un insieme di processi di sistema. Vedere systemd.scope(5).
- ***.target** raggruppa altri file di configurazione di unità per creare punti di sincronizzazione durante l'avvio. Vedere systemd.target(5).

All'avvio del sistema (cioè init) il processo `systemd` cerca di avviare "`/lib/systemd/system/default.target`" (normalmente un collegamento simbolico a "`graphical.target`"). Come prima cosa alcune speciali unità target (vedere systemd.special(7)) come "`local-fs.target`", "`swap.target`" e "`cryptsetup.target`" sono richiamate per montare i file system. Poi altre unità target vengono anch'esse richiamate dalle dipendenze delle unità target. Per i dettagli leggere bootup(7).

`systemd` offre funzionalità di compatibilità all'indietro. Gli script di avvio in stile SysV in "`/etc/init.d/rc[0123456S].d/`" sono comunque analizzati e telinit(8) viene tradotto in richieste di attivazione di unità systemd.



Attenzione
Run level emulati da 2 a 4 hanno tutti collegamenti simbolici al corrispondente "`multi-user.target`".

3.3.2 Login di systemd

Quando un utente fa il login nel sistema Debian con gdm3(8), sshd(8), ecc., `/lib/systemd/system --user` viene avviato come processo di gestione dei servizi utente con proprietario l'utente corrispondente. Vedere systemd(1).

Il processo di di gestione dei servizi utente di `systemd` lancia processi in parallelo sulla base dei file di configurazione delle unità dichiarate (vedere systemd.unit(5) e `user@.service`(5)).

Le unità per la modalità utente sono caricate dal "User Unit Search Path" come descritto in systemd.unit(5). Quelle principali, in ordine di priorità, sono le seguenti:

- "`~/.config/systemd/user/*`": unità utente di configurazione
- "`/etc/systemd/user/*`": unità utente create dall'amministratore
- "`/run/systemd/user/*`": unità runtime
- "`/lib/systemd/system`": unità utente installate dal gestore dei pacchetti della distribuzione

Sono gestite nello stesso modo di Sezione 3.3.1.

3.4 I messaggi del kernel

I messaggi di errore del kernel visualizzati nella console possono essere configurati impostando il loro livello di soglia.

```
# dmesg -n3
```

valore del livello di errore	nome del livello di errore	significato
0	KERN_EMERG	il sistema è inutilizzabile
1	KERN_ALERT	bisogna agire immediatamente
2	KERN_CRIT	condizione critica
3	KERN_ERR	condizione di errore
4	KERN_WARNING	condizione di avvertimento
5	KERN_NOTICE	condizione normale ma significativa
6	KERN_INFO	messaggio informativo
7	KERN_DEBUG	messaggio a livello di debug

Tabella 3.4: Elenco dei livelli di errore del kernel

3.5 I messaggi di sistema

Sotto `systemd` sia i messaggi del kernel sia quelli di sistema sono registrati nei log dal servizio journal `systemd-journald.` (alias `journald`) o in dati binari persistenti dentro "`/var/log/journal`" o in dati binari volatili dentro "`/run/log/journal`". A questi dati binari di log si può accedere tramite il comando journalctl(1). Ad esempio si può visualizzare il log dall'ultimo avvio con:

```
$ journalctl -b
```

Operazione	Esempio di comando
Visualizzare il log per i servizi di sistema e il kernel dall'ultimo avvio	"`journalctl -b --system`"
Visualizzare il log per i servizi dell'utente attuale dall'ultimo avvio	"`journalctl -b --user`"
Visualizzare il log di lavoro di "`$unit`" dall'ultimo avvio	"`journalctl -b -u $unit`"
Visualizzare il log di lavoro di "`$unit`" (in stile "`tail -f`") dall'ultimo avvio	"`journalctl -b -u $unit -f`"

Tabella 3.5: Elenco di tipici esempi di comandi per `systemd`

In `systemd` l'utilità per il registro di log di sistema, rsyslogd(8), può essere disinstallato. Se è installato, cambia il comportamento per leggere i dati del log binario volatile (invece del predefinito pre-systemd "`/dev/log`") e per creare dati log di sistema ASCII tradizionali permanenti. Questo può essere personalizzato da "`/etc/default/rsyslog`" e "`/etc/rsyslog.conf`" sia per il file di log sia per la visualizzazione a schermo. Vedere rsyslogd(8) e rsyslog.conf(5). Vedere anche Sezione 9.3.2.

3.6 Gestione del sistema

`systemd` offre non solo un sistema init, ma anche funzionalità generiche di gestione del sistema, con il comando systemctl(1).

Operazione	**Esempio di comando**
Elencare tutti i tipi di unità disponibili	"`systemctl list-units --type=help`"
Elencare tutte le unità target in memoria	"`systemctl list-units --type=target`"
Elencare tutte le unità di servizi in memoria	"`systemctl list-units --type=service`"
Elencare tutte le unità di device in memoria	"`systemctl list-units --type=device`"
Elencare tutte le unità di mount in memoria	"`systemctl list-units --type=mount`"
Elencare tutte le unità socket in memoria	"`systemctl list-sockets`"
Elencare tutte le unità timer in memoria	"`systemctl list-timers`"
Avviare "`$unit`"	"`systemctl start $unit`"
Fermare "`$unit`"	"`systemctl stop $unit`"
Ricaricare la configurazione specifica di un servizio	"`systemctl reload $unit`"
Fermare e riavviare tutte le "`$unit`"	"`systemctl restart $unit`"
Avviare "`$unit`" e fermare tutte le altre	"`systemctl isolate $unit`"
Passare alla modalità "`graphical`" (sistema GUI)	"`systemctl isolate graphical`"
Passare alla modalità "`multi-user`" (sistema CLI)	"`systemctl isolate multi-user`"
Passare alla modalità "`rescue`" (sistema CLI a singolo utente)	"`systemctl isolate rescue`"
Inviare il segnale kill a "`$unit`"	"`systemctl kill $unit`"
Controllare se il servizio "`$unit`" è attivo	"`systemctl is-active $unit`"
Controllare se il servizio "`$unit`" è fallito	"`systemctl is-failed $unit`"
Controllare lo stato di "`$unit\|$PID|device`"	"`systemctl status $unit\|$PID|$device`"
Mostrare le proprietà di "`$unit\|$job`"	"`systemctl show $unit\|$job`"
Ripristinare "`$unit`" fallita	"`systemctl reset-failed $unit`"
Elencare le dipendenze di tutti i servizi unità	"`systemctl list-dependencies --all`"
Elencare i file di unità installati sul sistema	"`systemctl list-unit-files`"
Abilitare "`$unit`" (aggiungere collegamento simbolico)	"`systemctl enable $unit`"
Disabilitare "`$unit`" (rimuovere collegamento simbolico)	"`systemctl disable $unit`"
Togliere maschera a "`$unit`" (rimuovere collegamento simbolico a"`/dev/null`")	"`systemctl unmask $unit`"
Mascherare "`$unit`" (aggiungere collegamento simbolico a"`/dev/null`")	"`systemctl mask $unit`"
Ottenere l'impostazione del target predefinito	"`systemctl get-default`"
Impostare default-target a "`graphical`" (sistema GUI)	"`systemctl set-default graphical`"
Impostare default-target a "`multi-user`" (sistema CLI)	"`systemctl set-default multi-user`"
Mostrare l'ambiente del lavoro	"`systemctl show-environment`"
Impostare la variabile "`variable`" dell'ambiente di lavoro al valore "`value`"	"`systemctl set-environment variable=value`"
Disimpostare la variabile "`variable`" dell'ambiente di lavoro	"`systemctl unset-environment variable`"
Ricaricare tutti i file di unità e i demoni	"`systemctl daemon-reload`"
Spegnere il sistema	"`systemctl poweroff`"
Spegnere e riavviare il sistema	"`systemctl reboot`"
Sospendere il sistema	"`systemctl suspend`"
Ibernare il sistema	"`systemctl hibernate`"

Tabella 3.6: Elenco di tipici esempi di comandi `systemctl`

Negli esempi soprastanti "`$unit`" può essere un singolo nome di unità (suffissi come `.service` e `.target` sono opzionali) o, in molti casi, la specifica di più unità (con glob in stile shell "`*`", "`?`", "`[]`" usando fnmatch(3) con corrispondenze con i nomi primari di tutte le unità attualmente in memoria).

I comandi che cambiano lo stato del sistema negli esempi soprastanti sono tipicamente preceduti da "`sudo`" per ottenere i privilegi amministrativi necessari.

L'output di "`systemctl status $unit|$PID|$device`" usa il colore del puntino ("●") per riassumere lo stato dell'unità a prima vista.

- Un "●" bianco indica uno stato "inattivo" o "deattivato".
- Un "●" rosso indica uno stato di "fallimento" o "errore".
- Un "●" verde indica uno stato "attivo", "in ricaricamento" o "in attivazione".

3.7 Altri strumenti di monitoraggio del sistema

Ecco un elenco di altri esempi di comandi per il monitoraggio in `systemd`. Leggere le relative pagine di manuale, inclusa cgroups(7).

Operazione	Esempio di comando
Mostrare il tempo utilizzato per ogni passo di inizializzazione	"`systemd-analyze time`"
Elencare tutte le unità col tempo di inizializzazione	"`systemd-analyze blame`"
Carica e rileva gli errori nel file "`$unit`"	"`systemd-analyze verify $unit`"
Mostra sintetiche informazioni di stato a runtime dell'utente della sessione chiamante	"`loginctl user-status`"
Mostra sintetiche informazioni di stato a runtime sulla sessione chiamante	"`loginctl session-status`"
Traccia il processo di avvio con cgroups	"`systemd-cgls`"
Traccia il processo di avvio con cgroups	"`ps xawf -eo pid,user,cgroup,args`"
Traccia il processo di avvio con cgroups	Leggere sysfs in "`/sys/fs/cgroup/`"

Tabella 3.7: Elenco di altri esempi di comandi per il monitoraggio in `systemd`

3.8 Configurazione del sistema

3.8.1 Il nome host

Il kernel gestisce il **nome host** del sistema. L'unità systemd avviata da `systemd-hostnamed.service` imposta il nome host del sistema all'avvio al nome memorizzato in "`/etc/hostname`". Questo file dovrebbe contenere **solamente** il nome host del sistema, non un nome di dominio pienamente qualificato.

Per visualizzare il nome host attuale eseguire hostname(1) senza alcun argomento.

3.8.2 Il filesystem

Le opzioni usate per montare i file system normali dei dischi e di rete sono impostate in "`/etc/fstab`". Vedere fstab(5) e Sezione 9.6.7.

La configurazione dei file system cifrati è impostata in "`/etc/crypttab`". Vedere crypttab(5)

La configuraziione del software RAID con mdadm(8) è configurata in "`/etc/mdadm/mdadm.conf`". Vedere mdadm.conf(5).



avvertimento
Dopo aver montato tutti i filesystem, i file temporanei in "`/tmp`", "`/var/lock`" e "`/var/run`" vengono ripuliti ad ogni avvio.

3.8.3 Inizializzazione delle interfacce di rete

Le interfacce di rete sono tipicamente inizializzate in "`networking.service`" per l'interfaccia `lo` e "`NetworkManager.ser`
per le altre interfacce nei moderni sistemi desktop Debian che usano `systemd`.

Vedere Capitolo 5 per come configurarle.

3.8.4 Inizializzazione del sistema cloud

L'istanza del sistema cloud può essere lanciata come clone di "immagini cloud ufficiali Debian" o immagini simili. Per tali istanze di sistema, entità come nome host, file system, rete, localizzazione, chiavi SSH, utenti e gruppi, possono essere configurate usando funzionalità fornite dai pacchetti `cloud-init` e `netplan.io` con diverse fonti di dati, come i file posizionati nell'immagine del sistema originale e dati esterni forniti durante il suo avvio. Questi pacchetti permettono la configurazione dichiarativa del sistema usando dati YAML.

Vedere ulteriori informazioni in "Cloud Computing with Debian and its descendants", "Documentazione di Cloud-init" e Sezione 5.4.

3.8.5 Esempio di personalizzazione per modificare il servizio sshd

Con l'installazione predefinita molti servizi di rete (vedere Capitolo 6) vengono avviati come processi demone dopo `network.target` al momento dell'avvio di sistema da `systemd`. "`sshd` non fa eccezione. Come esempio di personalizzazione cambiamo questo comportamento nell'avvio on-demand di "`sshd`".

Come prima cosa disabilitare l'unità di servizio installata dal sistema.

```
$ sudo systemctl stop sshd.service
$ sudo systemctl mask sshd.service
```

Il sistema di attivazione on-demand dei socket dei servizi classici Unix avveniva attraverso il superserver `inetd` (o `xinetd`). Con `systemd` il comportamento equivalente può essere abilitato aggiungendo file di configurazione di unità ***.socket** e ***.service**.

`sshd.socket` per specificare un socket su cui restare in ascolto

```
[Unit]
Description=SSH Socket for Per-Connection Servers

[Socket]
ListenStream=22
Accept=yes

[Install]
WantedBy=sockets.target
```

`sshd@.service` come file di servizio corrispondente di `sshd.socket`

```
[Unit]
Description=SSH Per-Connection Server

[Service]
ExecStart=-/usr/sbin/sshd -i
StandardInput=socket
```

Poi ricaricare.

```
$ sudo systemctl daemon-reload
```

3.9 Il sistema udev

Il sistema udev fornisce un meccanismo di rilevazione e inizializzazione automatiche dell'hardware (vedere udev(7)) a partire dal kernel Linux 2.6. Per ogni dispositivo rilevato dal kernel, il sistema udev avvia un processo utente che usa le informazioni del file system sysfs (vedere Sezione 1.2.12), carica, usando il programma modprobe(8) (vedere Sezione 3.10), i moduli del kernel necessari per il supporto del dispositivo e crea i nodi di device corrispondenti.

Suggerimento
Se, per una qualche ragione "`/lib/modules/`*`versione-kernel`*`/modules.dep` non viene generato in modo appropriato dal depmod(8), i moduli non possono essere caricati come dovuto dal sistema udev. Per risolvere il problema eseguire "`depmod -a`".
Per le regole di montaggio in "`/etc/fstab`", non è necessario che i nodi di device siano statici. Si possono usare gli UUID per montare i dispositivi, al posto dei nomi di device come "`/dev/sda`". Vedere Sezione 9.6.3.

Dato che il sistema udev è in qualche modo in costante evoluzione, in questo documento sono fornite informazioni base, lasciando i dettagli ad altra documentazione.



avvertimento
Non cercare di eseguire programmi con tempi di esecuzione lunghi, come script di backup con `RUN` in regole udev, come detto in udev(7). Creare invece un file systemd.service(5) corretto e attivarlo. Vedere Sezione 10.2.3.2.

3.10 L'inizializzazione dei moduli del kernel

Il programma modprobe(8) permette di configurare il kernel Linux in esecuzione da processi utente, aggiungendo e rimuovendo moduli del kernel. Il sistema udev (vedere Sezione 3.9) automatizza la sua invocazione per facilitare l'inizializzazione dei moduli del kernel.

Ci sono moduli non-hardware e speciali moduli con driver hardware, come quelli elencati in seguito, che devono essere precaricati elencandoli nel file "`/etc/modules`" (vedere modules(5)).

- moduli TUN/TAP che forniscono device virtuali di rete Point-to-Point (TUN) e device virtuali di rete Ethernet (TAP),
- moduli netfilter che forniscono funzionalità di firewall netfilter (iptables(8), Sezione 5.7 e
- moduli driver watchdog timer.

I file di configurazione per il programma modprobe(8) sono contenuti nella directory "`/etc/modprobes.d/`", come spiegato in modprobe.conf(5). (Se si desidera evitare l'autocaricamento di alcuni moduli del kernel, considerare la loro aggiunta nella lista nera nel file "`/etc/modprobes.d/blacklist`".)

Il file "`/lib/modules/`*`versione`*`/modules.dep`" generato dal programma depmod(8) descrive le dipendenze dei moduli usate dal programma modprobe(8).

Nota
Se si hanno problemi di caricamento dei moduli all'avvio o con modprobe(8), "`depmod -a`" potrebbe risolverli rigenerando il file "`modules.dep`".

Il programma modinfo(8) mostra informazioni su un modulo del kernel Linux.

Il programma lsmod(8) formatta in un bel modo i contenuti di "`/proc/modules`", mostrando quali moduli del kernel siano attualmente caricati.

Suggerimento
Si può identificare l'esatto hardware sul proprio sistema. Vedere Sezione 9.5.3.
Si può configurare l'hardware all'avvio per attivare le funzionalità dell'hardware desiderate. Vedere Sezione 9.5.4.
Si può probabilmente aggiungere il supporto per il proprio speciale dispositivo ricompilando il kernel. Vedere Sezione 9.10.

Capitolo 4

Autenticazione e controllo degli accessi

Quando una persona (o un programma) richiede l'accesso al sistema, l'autenticazione verifica che l'identità sia fidata.



avvertimento
Errori di configurazione di PAM possono lasciare l'utente fuori dal proprio sistema. Si deve avere un CD di ripristino a portata di mano o impostare una partizione di avvio alternativa. Per fare il ripristino, avviare il sistema con uno di essi e correggere le cose da lì.

4.1 Autenticazione Unix normale

L'autenticazione Unix normale è fornita dal modulo pam_unix(8) sotto PAM (Pluggable Authentication Modules). I suoi 3 importanti file di configurazione, con voci separate da "`:`", sono i seguenti.

file	permessi	utente	gruppo	descrizione
`/etc/passwd`	`-rw-r--r--`	`root`	`root`	informazioni sugli account utente (ripulite)
`/etc/shadow`	`-rw-r-----`	`root`	`shadow`	informazioni sicure sugli account utente
`/etc/group`	`-rw-r--r--`	`root`	`root`	informazioni sui gruppi

Tabella 4.1: I 3 importanti file di configurazione per pam_unix(8)

"`/etc/passwd`" contiene righe come le seguenti.

```
 ...
user1:x:1000:1000:User1 Name,,,:/home/user1:/bin/bash
user2:x:1001:1001:User2 Name,,,:/home/user2:/bin/bash
 ...
```

Come spiegato in "passwd(5), le voci separate da "`:` in questo file hanno il significato seguente.

- Nome di login
- Voce di specificazione della password
- ID numerico dell'utente
- ID numerico del gruppo

- Nome dell'utente o campo di commento
- Directory home dell'utente
- Voce opzionale per l'interprete di comandi utente

La seconda voce del file "`/etc/passwd`" era usata per la password cifrata. Dopo l'introduzione di "`/etc/shadow`", questa voce è usata per la specificazione della password.

contenuto	**significato**
(vuoto)	account senza password
x	la password cifrata è in "`/etc/shadow`"

Tabella 4.2: Il contenuto della seconda voce di "`/etc/passwd`"

"`/etc/shadow`" contiene righe come le seguenti.

```
 ...
user1:$1$Xop0FYH9$IfxyQwBe9b8tiyIkt2P4F/:13262:0:99999:7:::
user2:$1$vXGZLVbS$ElyErNf/agUDsm1DehJMS/:13261:0:99999:7:::
 ...
```

Come spiegato in "shadow(5), le voci separate da "`:` in questo file hanno il significato seguente.

- Nome di login
- Password cifrata (I caratteri "`$1$`" iniziali indicano l'uso della cifratura MD5. "*" indica nessun login.)
- Data dell'ultimo cambiamento di password, espressa come numero di giorni trascorsi dall'1 gennaio 1970
- Numero di giorni che l'utente deve aspettare prima di essere autorizzato a cambiare nuovamente la password
- Numero di giorni trascorsi i quali l'utente dovrà cambiare la propria password
- Numero di giorni di preavviso per avvisare l'utente che la password è in scadenza
- Numero di giorni dopo che una password è scaduta in cui la password viene sempre accettata
- Data di scadenza dell'account, espressa come numero di giorni trascorsi dall'1 gennaio 1970.
- …

"`/etc/group`" contiene righe come le seguenti.

```
group1:x:20:user1,user2
```

Come spiegato in "group(5), le voci separate da "`:` in questo file hanno il significato seguente.

- Nome del gruppo
- Password cifrata (non realmente usato)
- ID numerico del gruppo
- Elenco di nomi utente separati da ","

Nota
"`/etc/gshadow`" fornisce funzioni simili a "`/etc/shadow`" per "`/etc/group`", ma non è realmente usato.

Nota
Può essere dinamicamente aggiunta la reale appartenenza di un utente ad un gruppo se viene aggiunta la riga "`auth optional pam_group.so` al file "`/etc/pam.d/common-auth`" e se viene impostata in "`/etc/security/group.conf`". Vedere pam_group(8).

Nota
Il pacchetto `base-passwd` contiene una lista autorevole di utenti e gruppi: "`/usr/share/doc/base-passwd/users-and-groups.html`".

4.2 Gestire le informazioni su account e password

Quelli che seguono sono alcuni comandi degni di nota per gestire le informazioni sugli account

comando	funzione
`getent passwd` *`nome_utente`*	sfoglia le informazioni sull'account di "*`nome_utente`*"
`getent shadow` *`nome_utente`*	sfoglia le informazioni shadow sull'account di "*`nome_utente`*"
`getent group` *`nome_gruppo`*	sfoglia le informazioni sul gruppo "*`nome_gruppo`*"
`passwd`	gestisce la password per l'account
`passwd -e`	imposta una password usa e getta per l'attivazione dell'account
`chage`	gestisce le informazioni sulla scadenza della password

Tabella 4.3: Elenco di comandi per gestire informazioni su account

Può essere necessario avere i privilegi di root per far funzionare alcune di queste funzioni. Vedere crypt(3) per la cifratura di password e dati.

Nota
Nei sistemi configurati con PAM e NSS, come la macchina Debian salsa Debian, il contenuto dei file locali "`/etc/passwd`", "`/etc/group`" ed "`/etc/shadow`" può non essere attivamente usato dal sistema. I comandi sopra descritti sono validi anche in un ambiente di questo tipo.

4.3 Password buone

Quando si crea un account, durante l'installazione del sistema o con il comando passwd(1), si dovrebbe scegliere una buona password che consiste, secondo passwd(1), di un numero di caratteri da almeno 6 a 8, incluso uno o più caratteri da ciascuno dei seguenti insiemi.

- Lettere dell'alfabeto minuscole
- Cifre da 0 a 9
- Segni di punteggiatura



avvertimento
Non scegliere parole indovinabili come password. Il nome dell'account, il codice fiscale, il numero di telefono, l'indirizzo, il giorno del compleanno, il nome di membri della propria famiglia o di animali, parole del dizionario, semplici sequenze di caratteri come "12345" o "qwerty", … sono tutte scelte pessime come password.

4.4 Creare password cifrate

Ci sono strumenti indipendenti per generare password cifrate con salt.

pacchetto	popcon	dimensione	comando	funzione
whois	V:23, I:203	384	mkpasswd	frontend ricco di funzionalità per la libreria crypt(3)
openssl	V:846, I:996	2564	openssl passwd	calcola hash di password (OpenSSL). passwd(1ssl)

Tabella 4.4: Elenco di strumenti per generare password

4.5 PAM e NSS

I moderni sistemi *nix come il sistema Debian, forniscono all'amministratore di sistema i meccanismi PAM (Pluggable Authentication Modules) e NSS (Name Service Switch) per configurare il sistema. Il ruolo di questi meccanismi può essere riassunto nel modo seguente.

- PAM offre un meccanismo di autenticazione flessibile usato dal software applicativo e comporta pertanto scambio di dati sulle password
- NSS offre un meccanismo di servizio dei nomi flessibile che è usato di frequente dalla libreria standard C per ottenere i nomi di utenti e gruppi per programmi come ls(1) e id(1).

Questi sistemi PAM e NSS devono essere configurati in modo coerente.

I pacchetti degni di nota relativi ai sistemi PAM e NSS sono i seguenti.

pacchetto	popcon	dimensione	descrizione
libpam-modules	V:942, I:1000	892	Pluggable Authentication Modules, moduli di autenticazione inseribili (servizio base)
libpam-ldapd	V:7, I:13	79	Pluggable Authentication Module, modulo di autenticazione inseribile, che permette interfacce LDAP
libpam-systemd	V:764, I:969	848	Pluggable Authentication Module, modulo di autenticazione inseribile, per registrare sessioni utente per logind
libpam-doc	I:7.5	1492	Pluggable Authentication Modules, moduli di autenticazione inseribili (documentazione in html e testo semplice)
libc6	V:936, I:1000	5483	Libreria GNU C: librerie condivise che forniscono anche il servizio "Name Service Switch"
glibc-doc	I:5.1	3927	Libreria GNU C: pagine man
glibc-doc-reference	I:3.1	14764	Libreria GNU C: manuale di riferimento nei formati info, pdf e html (non-free)
libnss-mdns	V:299, I:507	142	Modulo NSS per la risoluzione dei nomi DNS Multicast
libnss-ldapd	V:8, I:16	131	Modulo NSS per l'uso di LDAP come servizio per i nomi

Tabella 4.5: Elenco dei pacchetti degni di nota per i sistemi PAM e NSS

- La guida per l'amministratore di sistema di Linux-PAM, "The Linux-PAM System Administrators' Guide", in libpam-doc è essenziale per imparare la configurazione di PAM.
- La sezione "System Databases and Name Service Switch" in glibc-doc-reference è essenziale per imparare la configurazione di NSS.

Nota
Si può ottenere un elenco più esteso ed aggiornato usando il comando "`aptitude search 'libpam-|libnss-'`". L'acronimo NSS può anche significare "Network Security Service" che è una cosa diversa da "Name Service Switch".

Nota
PAM è il metodo più basilare per inizializzare le variabili d'ambiente per ciascun programma con valori predefiniti a livello di sistema.

Con systemd, il pacchetto `libpam-systemd` viene installato per gestire il login degli utenti registrando le sessioni utente nella gerarchia dei gruppi di controllo di `systemd` per logind. Vedere systemd-logind(8), logind.conf(5) e pam_systemd(8).

4.5.1 File di configurazione letti da PAM e NSS

Quelli che seguono sono alcuni file degni di nota letti da PAM e NSS.

file di configurazione	funzione
`/etc/pam.d/`*`nome_programma`*	imposta la configurazione PAM per il programma "*`nome_programma`*"; vedere pam(7) e pam.d(5)
`/etc/nsswitch.conf`	imposta la configurazione NSS con la voce per ciascun servizio. Vedere nsswitch.conf(5)
`/etc/nologin`	limita il login utente con il modulo pam_nologin(8)
`/etc/securetty`	limita la tty per l'accesso root con il modulo pam_securetty(8)
`/etc/security/access.conf`	imposta il limite di accesso con il modulo pam_access(8)
`/etc/security/group.conf`	imposta restrizioni basate su gruppi con il modulo pam_group(8)
`/etc/security/pam_env.conf`	imposta le variabili d'ambiente con il modulo pam_env(8)
`/etc/environment`	imposta variabili d'ambiente aggiuntive con il modulo pam_env(8) con l'argomento "`readenv=1`"
`/etc/default/locale`	imposta la localizzazione con il modulo pam_env(8) con l'argomento "`readenv=1 envfile=/etc/default/locale`" (Debian)
`/etc/security/limits.conf`	imposta limitazioni alle risorse (ulimit, core, …) con il modulo pam_limits(8)
`/etc/security/time.conf`	imposta limitazioni temporali con il modulo pam_time(8)
`/etc/systemd/logind.conf`	imposta la configurazione del gestore di login di `systemd` (vedere logind.conf(5) e systemd-logind.service(8))

Tabella 4.6: Elenco di file di configurazione letti da PAM e NSS

Le restrizioni sulla scelta delle password sono implementate dai moduli PAM pam_unix(8) e pam_cracklib(8). Possono essere configurati tramite i loro argomenti.

Suggerimento
I moduli PAM usano il suffisso "`.so`" nei loro nomi file.

4.5.2 La moderna gestione centralizzata del sistema

La moderna gestione centralizzata del sistema può essere messa in atto usando il server del Protocollo LDAP (Lightweight Directory Access Protocol) per amministrare molti sistemi *nix e non *nix in rete. L'implementazione open source del protocollo LDAP è il software OpenLDAP.

Il server LDAP fornisce le informazioni sugli account attraverso l'uso di PAM e NSS con i pacchetti per il sistema Debian `libpam-ldapd` e `libnss-ldapd`. Per abilitare ciò sono necessarie diverse azioni. (Non ho mai usato questa configurazione e le informazioni che seguono sono di seconda mano. Tenerlo a mente quando si legge quanto segue.)

- Si configura un server LDAP centralizzato eseguendo un programma come il demone LDAP autonomo slapd(8).
- Si modificano i file di configurazione di PAM nella directory "`/etc/pam.d/`" per usare "`pam_ldap.so`" invece del predefinito "`pam_unix.so`".
- Si modifica la configurazione di NSS nel file "`/etc/nsswitch.conf`" per usare "`ldap`" invece della scelta predefinita ("`compat`" o "`file`").
- Per la sicurezza delle password è necessario far sì che `libpam-ldapd` usi una connessione SSL (o TLS).
- Per assicurare l'integrità dei dati, si può far sì che `libnss-ldapd` usi una connessione SSL (o TLS) a prezzo di un maggiore carico sulla rete LDAP.
- Si dovrebbe eseguire nscd(8) localmente per mettere nella cache ogni risultato di ricerche LDAP in modo da ridurre il traffico di rete LDAP.

Vedere la documentazione in nsswitch.conf(5), pam.conf(5), ldap.conf(5) e "`/usr/share/doc/libpam-doc/html/`" fornita dal pacchetto `libpam-doc` e "`info libc 'Name Service Switch'`" fornita dal pacchetto `glibc-doc`.

In modo simile si possono impostare sistemi centralizzati alternativi con altri metodi.

- Integrazione di utenti e gruppi con il sistema Windows.
  - Accesso ai servizi di dominio Windows con i pacchetti `winbind` e `libpam_winbind`.
  - Vedere winbindd(8) e Integrare reti MS Windows con Samba.
- Integrazione di utenti e gruppi con sistemi simil-Unix datati
  - Accesso a NIS (chiamato in origine YP) o NIS+ con il pacchetto `nis`.
  - Vedere il Linux NIS(YP)/NYS/NIS+ HOWTO.

4.5.3 "Perché GNU su non supporta il gruppo wheel"

Questa è la famosa sezione scritta da Richard M. Stallman, alla fine della vecchia pagina "`info su`". Non c'è da preoccuparsi: l'attuale comando `su` in Debian usa PAM, perciò questo può limitare la possibilità di usare `su` verso il gruppo `root` abilitando la riga con "`pam_wheel.so`" in "`/etc/pam.d/su`".

4.5.4 Regole più stringenti per le password

L'installazione del pacchetto `libpam-cracklib` permette di imporre regole più stringenti sulle password.

In un sistema GNOME tipico che installa automaticamente `libpam-gnome-keyring`, "`/etc/pam.d/common-password`" è simile a:

```
# here are the per-package modules (the "Primary" block)
password  requisite     pam_cracklib.so retry=3 minlen=8 difok=3
password  [success=1 default=ignore]  pam_unix.so obscure use_authtok try_first_pass  ↩
    yescrypt
# here's the fallback if no module succeeds
password  requisite     pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password  required      pam_permit.so
# and here are more per-package modules (the "Additional" block)
password  optional  pam_gnome_keyring.so
# end of pam-auth-update config
```

4.6 Sicurezza dell'autenticazione

Nota
Le informazioni fornite in questo documento **potrebbero non essere sufficienti** per le proprie necessità di sicurezza ma dovrebbero essere un **buon punto di partenza**.

4.6.1 Password sicure in Internet

Molti servizi per livello di trasporto popolari comunicano i loro messaggi, compresa l'autenticazione con password, in puro testo. È una pessima idea trasmettere password in puro testo attraverso l'Internet selvaggia dove possono essere intercettate. Si possono eseguire questi servizi attraverso "TLS" (Transport Layer Security, sicurezza del livello di trasporto), o il suo predecessore "SSL" (Secure Sockets Layer, livello per socket sicuri), per rendere sicura tramite cifratura tutta la comunicazione, compresa la password.

nome del servizio non sicuro	porta	nome del servizio sicuro	porta
`www (http)`	80	`https`	443
`smtp (mail)`	25	`ssmtp (smtps)`	465
`ftp-data`	20	`ftps-data`	989
`ftp`	21	`ftps`	990
`telnet`	23	`telnets`	992
`imap2`	143	`imaps`	993
`pop3`	110	`pop3s`	995
`ldap`	389	`ldaps`	636

Tabella 4.7: Elenco di servizi e porte sicuri e non sicuri

La cifratura ha un costo in termini di tempo CPU. Come alternativa leggera per la CPU, si può mantenere la comunicazione in testo semplice, rendendo allo stesso tempo sicura la sola password con un protocollo di autenticazione sicura come "APOP" (Authenticated Post Office Protocol) per POP e "CRAMD-MD5" (Challenge-Response Authentication Mechanism MD5) per SMTP e IMAP. (Per inviare messaggi di posta elettronica via Internet dal proprio programma di posta al proprio server di posta è diventato popolare recentemente l'uso per SMTP della porta 587 invece della porta tradizionale 25, per evitare il blocco da parte del fornitore del servizio Internet della porta 25 autenticandosi allo stesso tempo con CRAM-MD5.)

La tradizionale autenticazione basata su password soffre ancora di problemi intrinseci di sicurezza„ come il riutilizzo di password su siti web multipli. Vedere altrimetodi di autenticazione per una soluzione sicura: TOTP (Time-based one-time password, password usa e getta a tempo), U2F (Universal 2nd Factor, universale a 2 fattori), Autenticazione Web (WebAuthn, comunemente chiamata "passkey")

4.6.2 Secure Shell, shell sicura

Il programma Secure Shell (SSH fornisce comunicazioni sicure cifrate tra due host non fidati attraverso una rete non sicura, grazie ad un'autenticazione sicura. Consiste del client OpenSSH, ssh(1) e del demone OpenSSH, sshd(8). SSH può essere usato per fare da tunnel sicuro attraverso Internet per le comunicazioni con protocollo non sicuro come POP ed X, con la funzionalità di inoltro delle porte.

Il client cerca di autenticarsi usando un'autenticazione basata sull'host, su una chiave pubblica, challenge-response o con password. L'uso di un'autenticazione con chiave pubblica permette il login remoto senza password. Vedere Sezione 6.3.

4.6.3 Misure aggiuntive di sicurezza per Internet

Anche quando si eseguono servizi sicuri, come server SSH (Secure shell) e PPTP (Point-to-Point Tunneling Protocol), esiste sempre la possibilità di un'intrusione da Internet con attacchi basati sull'indovinare la password usando metodi con forza bruta, ecc. L'uso di una politica di firewall (vedere Sezione 5.7) insieme agli strumenti di sicurezza elencati in seguito, può migliorare la sicurezza generale.

pacchetto	popcon	dimensione	descrizione
`knockd`	V:0.7, I:1.6	110	demone, knockd(1), e client, knock(1) piccoli per bussare alle porte
`fail2ban`	V:101, I:111	2275	strumento per interdire IP che causano errori di autenticazione multipli
`libpam-shield`	V:0.04, I:0.05	115	blocca all'esterno gli attacchi remoti che cercano di indovinare le password

Tabella 4.8: Elenco di strumenti per fornire misure aggiuntive di sicurezza

4.6.4 Rendere sicura la password di root

Per impedire che qualcuno possa accedere alla propria macchina con privilegi di root, è necessario compiere le azioni seguenti.

- Impedire l'accesso fisico ai dispositivi di archiviazione del sistema (HDD / SSD / ...)
- Bloccare UEFI/BIOS ed impedire l'avvio da supporti removibili
- Impostare una password per la sessione interattiva di GRUB
- Bloccare il menu di GRUB impedendo i cambiamenti

Avendo accesso fisico al dispositivo di archiviazione del sistema, reimpostare la password è relativamente semplice seguendo i passi seguenti.

1. Spostare il dispositivo di archiviazione del sistema in un PC con UEFI/BIOS avviabile da USB.
2. Avviare il sistema con un supporto di ripristino (vedere Sezione 3.2.2).
3. Montare la partizione root con accesso in lettura e scrittura
4. Modificare il file "`/etc/passwd`" nella partizione root e rendere la seconda voce per l'account di `root` vuota.

Se si ha l'accesso in modifica alle voci di menu di GRUB (vedere Sezione 3.1.2) per `grub-rescue-pc` all'avvio, è ancora più semplice, seguendo i passi seguenti.

1. Avviare il sistema con i parametri del kernel modificati in qualcosa del tipo "`root=/dev/sda6 rw init=/bin/sh`".
2. Modificare il file "`/etc/passwd`" e rendere la seconda voce per l'account di `root` vuota.
3. Riavviare il sistema.

Si può ora accedere alla shell di root del sistema senza password.

Nota
Una volta ottenuto l'accesso alla shell di root, si ha l'accesso a qualsiasi cosa sul sistema e si può reimpostare qualsiasi password. Inoltre, si possono compromettere le password per tutti gli account utente usando strumenti di violazione delle password con attacchi a forza bruta, come quelli nei pacchetti `john` e `crack` (vedere Sezione 9.5.10). Queste password violate possono portare alla compromissione di altri sistemi.

L'unica soluzione software ragionevole per evitare tutte queste preoccupazioni è l'uso di una partizione root (o partizione "`/etc`") cifrata, usando dm-crypt e initramfs (vedere Sezione 9.9). Tuttavia è sempre necessaria la password per avviare il sistema.

4.7 Altri controlli sugli accessi

Esistono controlli degli accessi al sistema diversi dall'autenticazione basata su password e dei permessi dei file.

Nota
Vedere Sezione 9.4.16 per limitare la funzionalità SAK (Secure Attention Key) del kernel.

4.7.1 ACL (Access Control List, liste di controllo degli accessi)

Le ACL sono un sovrainsieme dei permessi regolari, come spiegato in Sezione 1.2.3.

Si trovano le ACL in funzione negli ambienti desktop moderni. Quando un dispositivo di archiviazione USB formattato viene montato automaticamente, ad esempio come "`/media/penguin/USBSTICK`", un normale utente `penguin` può eseguire:

```
 $ cd /media/penguin
 $ ls -la
total 16
drwxr-x---+ 1 root    root    16 Jan 17 22:55 .
drwxr-xr-x  1 root    root    28 Sep 17 19:03 ..
drwxr-xr-x  1 penguin penguin 18 Jan  6 07:05 USBSTICK
```

"+" in the 11th column indicates ACLs are in action. Without ACLs, a normal user `penguin` shouldn't be able to list like this since `penguin` isn't in `root` group. You can see ACLs as:

```
 $ getfacl .
# file: .
# owner: root
# group: root
user::rwx
user:penguin:r-x
group::---
mask::r-x
other::---
```

Qui:

- "`user::rwx`", "`group::---`", e "`other::---`" corrispondono ai regolari permessi per proprietario, gruppo e altri.
- La ACL "`user:penguin:r-x`" permette a un normale utente `penguin` di avere permessi "`r-x`". Ciò ha permesso a "`ls -la`" di elencare il contenuto della directory.
- La ACL "`mask::r-x`" imposta il limite in alto per i permessi.

Vedere "Liste di controllo degli accessi POSIX in Linux", acl(5), getfacl(1) e setfacl(1) per maggiori informazioni.

4.7.2 sudo

sudo(8) è un programma progettato per permettere ad un amministratore di sistema di dare privilegi di root limitati ad utenti, e di registrare l'attività come root. `sudo` richiede solo la password di un utente regolare. Installare il pacchetto `sudo` e attivarlo impostando le opzioni in "`/etc/sudoers`". Vedere esempi di configurazione in "`/usr/share/doc/sudo/examples/sudoers`" e Sezione 1.1.12.

Il mio uso di `sudo` per un sistema con un singolo utente (vedere Sezione 1.1.12) è mirato a proteggere me stesso dalla mia stupidità. Personalmente condidero l'uso di `sudo` come un'alternativa migliore all'uso costante del sistema dall'account root. Per esempio, il comando seguente cambia il proprietario di "*`un_certo_file`*" in "*`mio_nome`*".

```
$ sudo chown my_name some_file
```

Naturalmente se si conosce la password di root (come accade per ogni utente Debian che ha installato il proprio sistema), qualsiasi comando può essere eseguito come utente root da qualsiasi account utente usando "`su -c`".

4.7.3 PolicyKit

PolicyKit è un componente del sistema operativo per controllare privilegi a livello di sistema in sistemi operativi simil-Unix.

Le applicazioni GUI più recenti non sono pensate per essere eseguite come processi privilegiati. Per effettuare operazioni amministrative comunicano con processi privilegiati attraverso PolicyKit.

PolicyKit limita tali operazioni agli account utente che appartengono al gruppo `sudo`, in sistemi Debian.

Vedere polkit(8).

4.7.4 Limitare l'accesso ad alcuni servizi server

Per la sicurezza del sistema è una buona idea disabilitare il maggior numero di programmi server possibile. Questo aspetto diventa critico per i server di rete. Avere server inutilizzati, attivati direttamente come demoni o attraverso un programma super-server, è considerato un rischio per la sicurezza.

Molti programmi, come sshd(8), usano un controllo degli accessi basato su PAM. Ci sono molti modi per limitare gli accessi ad un qualche servizio server.

- file di configurazione: "`/etc/default/`*`nome_programma`*"
- configurazione dell'unità di servizio di systemd per il demone
- PAM (Pluggable Authentication Modules)
- "`/etc/inetd.conf`" per il super-server
- "`/etc/hosts.deny`" e "`/etc/hosts.allow`" per il wrapper TCP, tcpd(8)
- "`/etc/rpc.conf`" per Sun RPC
- "`/etc/at.allow`" e "`/etc/at.deny`" per atd(8)
- "`/etc/cron.allow`" e "`/etc/cron.deny`" per crontab(1)
- il firewall di rete della infrastruttura netfilter

Vedere Sezione 3.6, Sezione 4.5.1 e Sezione 5.7.

Suggerimento
I servizi Sun RPC devono essere attivi per i programmi NFS ed altri programmi basati su RPC.

Suggerimento
Se si hanno problemi con l'accesso remoto in sistemi Debian recenti, commentare la configurazione responsabile come "`ALL: PARANOID`" in "`/etc/hosts.deny`", se esiste. (Essere però consapevoli dei rischi per la sicurezza che questo tipo di azione comporta.)

4.7.5 Funzionalità di sicurezza di Linux

Il kernel Linux si è evoluto e gestisce funzionalità di sicurezza che non sono presenti nelle implementazioni UNIX tradizionali.

Linux gestisce gli attributi estesi che estendono gli attributi UNIX tradizionali (vedere xattr(7)).

Linux divide i privilegi tradizionalmente associati con il superutente in unità distinte, note come capabilities(7), che possono essere abilitate e disabilitate in modo indipendente. Le capabilities sono un attributo per singolo thread a partire dalla versione 2.2 del kernel.

L'infrastruttura Linux Security Module (LSM) fornisce un meccanismo per vari controlli di sicurezza in modo che nuove estensioni del kernel si aggancino ad essi. Per esempio:

- AppArmor
- Security-Enhanced Linux (SELinux)
- Smack (Simplified Mandatory Access Control Kernel)
- Tomoyo Linux

Dato che queste estensioni possono restringere il modello dei privilegi in modo più stringente delle politiche del normale modello di sicurezza in stile Unix, anche i poteri di root possono essere ristretti. È consigliato leggere la documentazione dell'infrastruttura Linux Security Module (LSM) su kernel.org.

Gli spazi dei nomi Linux creano un involucro (wrap) per una risorsa di sistema globale come astrazione che fa sembrare ai processi all'interno dello spazio dei nomi di avere una propria istanza isolata della risorsa globale. Le modifiche alla risorsa globale sono visibile agli altri processi che sono membri dello spazio dei nomi, ma sono invisibili agli altri processi. A partire dalla versione 5.6 del kernel, ci sono 8 tipi di spazi dei nomi (vedere namespaces(7), unshare(1), nsenter(1)).

In Debian 11 Bullseye (2021) Debian usa la gerarchia cgroup unificata (alias cgroups-v2).

Esempi di uso di spazi dei nomi con cgroups per isolare i loro processi e permettere il controllo delle risorse sono:

- Systemd. Vedere Sezione 3.3.1.
- Ambiente sandbox. Vedere Sezione 7.7.
- Contenitori Linux come Docker, LXC. Vedere Sezione 9.11.

Queste funzionalità non possono essere realizzate con Sezione 4.1. Questi argomenti avanzati sono per lo più al di fuori di questo documento introduttivo.

Capitolo 5

Impostazione della rete

Suggerimento
Per una guida moderna sull'uso della rete specifica per Debian, leggere The Debian Administrator's Handbook — Configuring the Network.

Suggerimento
In systemd, si può usare networkd per gestire il le reti. Vedere systemd-networkd(8).

5.1 L'infrastruttura base di rete

In questa sezione viene descritta in breve l'infrastruttura base di rete in un sistema Debian moderno.

5.1.1 Risoluzione dei nomi di host

La risoluzione del nome host è attualmente supportata dal meccanismo NSS (Name Service Switch). Il flusso di eventi nella risoluzione è il seguente.

1. Il file "`/etc/nsswitch.conf`" con blocchi tipo "`hosts: files dns`" detta l'ordine di risoluzione dei nomi di host. (Questo rimpiazza la vecchia funzionalità del blocco "`order` in "`/etc/host.conf`".)
2. Il metodo `files` è invocato per primo. Se il nome host viene trovato nel file "`/etc/hosts`", vengono restituiti tutti gli indirizzi validi per esso e il programma termina. (Il file "`/etc/host.conf`" contiene "`multi on`".)
3. Viene invocato il metodo `dns`. Se il nome host viene trovato dalla interrogazione al DNS (Internet Domain Name System, sistema dei nomi di dominio Internet), identificato dal file "`/etc/resolv.conf`", vengono restituiti tutti gli indirizzi validi per esso e il programma termina.

Una postazione di lavoro tipica può essere installata con il nome host impostato, ad esempio, come "*host_name*" e il suo nome di dominio opzionale impostato come stringa vuota. Poi, `/etc/hosts`" avrà l'aspetto seguente.

```
127.0.0.1 localhost
127.0.1.1 host_name

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

pacchetto	popcon	dimensione	tipo	descrizione
network-manager	V:427, I:474	7365	config::NM	NetworkManager (demone): gestisce la rete automaticamente
network-manager-gnome	V:36, I:147	19	config::NM	NetworkManager (frontend GNOME)
netplan.io	V:2.1, I:8.4	340	config::NM+networkd	Netplan (generatore): interfaccia unificata, dichiarativa ai backend NetworkManager e systemd-networkd
ifupdown	V:649, I:973	204	config::ifupdown	strumento standard per attivare e disattivare la rete (specifico di Debian)
pppoeconf	V:0.2, I:3.9	176	config::helper	strumento di aiuto di configurazione per la connessione PPPoE
wpasupplicant	V:421, I:521	3896	config::helper	supporto client per WPA e WPA2 (IEEE 802.11i)
wpagui	V:0.2, I:1.2	779	config::helper	client GUI Qt per WPA supplicant
wireless-tools	V:193, I:262	232	config::helper	strumenti per manipolare le estensioni wireless per Linux
iw	V:40, I:478	332	config::helper	strumento per configurare dispositivi wireless per Linux
iproute2	V:781, I:987	4268	config::iproute2	iproute2, IPv6 e altra configurazione di rete avanzata: ip(8), tc(8), ecc.
iptables	V:381, I:635	2408	config::Netfilter	strumenti amministrativi per filtraggio di pacchetti di rete e NAT (Netfilter)
nftables	V:266, I:880	189	config::Netfilter	strumenti amministrativi per filtraggio di pacchetti di rete e NAT (Netfilter) (successore di {ip,ip6,arp,eb}tables)
iputils-ping	V:206, I:998	188	test	testa la raggiungibilità di rete di un host remoto con nome host o indirizzo IP (iproute2)
iputils-arping	V:2, I:17	53	test	testa la raggiungibilità di rete di un host remoto specificato con il suo indirizzo ARP
iputils-tracepath	V:2, I:19	50	test	traccia il percorso di rete verso un host remoto
ethtool	V:98, I:251	1093	test	mostra e modifica le impostazioni dei device Ethernet
mtr-tiny	V:4, I:38	182	test::low-level	traccia il percorso di rete verso un host remoto (curses)
mtr	V:4, I:40	231	test::low-level	traccia il percorso di rete verso un host remoto (curses e GTK)
gnome-nettool	V:0.5, I:8.4	2480	test::low-level	strumenti per operazioni comuni sulle informazioni di rete (GNOME)
nmap	V:25, I:182	4934	test::low-level	strumento per mappatura della rete / esplorazione delle porte (Nmap, console)
tcpdump	V:20, I:177	1346	test::low-level	analizzatore del traffico di rete (Tcpdump, console)
wireshark	V:4, I:39	17068	test::low-level	analizzatore del traffico di rete (Wireshark, GTK)
tshark	V:2, I:23	433	test::low-level	analizzatore del traffico di rete (console)
tcptrace	V:0.2, I:1.5	407	test::low-level	produce un riassunto delle connessioni dall'output di `tcpdump`
dnsutils	V:5, I:146	23	test::low-level	client di rete forniti con BIND: nslookup(8), nsupdate(8), `dig`
dlint	V:0.1, I:2.0	52	test::low-level	controlla l'informazione DNS di zona usando interrogazioni del server di nomi
dnstracer	V:0.1, I:1.1	63	test::low-level	traccia una catena di server DNS fino alla sorgente

Tabella 5.1: Elenco degli strumenti di configurazione della rete

Ogni riga inizia con un indirizzo IP e prosegue con l'hostname associato.

L'indirizzo IP `127.0.1.1` nella seconda riga dell'esempio può non essere presente in altri sistemi simil-Unix. L'Installatore Debian crea questa voce per i sistemi senza un indirizzo IP permanente come soluzione per alcuni software (es., GNOME) come documentato nel bug#719621.

Il *nome_host* corrisponde al nome host definito in "`/etc/hostname`" (vedere Sezione 3.8.1).

Per un sistema con un indirizzo IP permanente, si dovrebbe usare qui tale indirizzo invece di `127.0.1.1`.

Per un sistema con un indirizzo IP permanente e un FQDN (Fully Qualified Domain Name, nome di dominio pienamente qualificato) fornito dal DNS (Domain Name System), dovrebbe essere usato qui il canonico *nome_host*.*nome_dominio*, invece del semplice *nome_host*.

Il file "`/etc/resolv.conf`" è un file statico se non è installato il pacchetto `resolvconf`. Se invece quest'ultimo è installato il file è un collegamento simbolico. In ogni caso contiene le informazioni che inizializzano le routine del risolutore. Se il DNS si trova all'IP="`192.168.11.1`", il file conterrà la riga seguente.

```
nameserver 192.168.11.1
```

Il pacchetto `resolvconf` rendo questo file "`/etc/resolv.conf`" un collegamento simbolico e gestisce il suo contenuto con script eseguiti in automatico.

Per le postazioni PC nei tipici ambienti LAN ad hoc, il nome host può essere risolto usando il Multicast DNS (mDNS) in aggiunta ai metodi di base con `file` e `dns`.

- Avahi fornisce un'infrastruttura per il Multicast DNS Service Discovery in Debian.
- È equivalente a Apple Bonjour / Apple Rendezvous.
- Il pacchetto plugin `libnss-mdns` fornisce la risoluzione dei nomi host attraverso mDNS per la funzionalità GNU Name Service Switch (NSS) della libreria C GNU (glibc).
- Il file "`/etc/nsswitch.conf`" dovrebbe contenere una sezione simile a "`hosts: files mdns4_minimal [NOTFOUN` `dns`" (vedere `/usr/share/doc/libnss-mdns/README.Debian` per altre configurazioni).
- I nomi host che terminano con lo pseudo-dominio di livello più alto ".local" vengono risolti inviando un messaggio di interrogazione mDNS in un pacchetto UDP multicast usando l'indirizzo IPv4 "`224.0.0.251`" o l'indirizzo IPv6 "`FF02::FB`".

Nota
L'espansione dei gTLD (generic Top-Level Domain, domini di più alto livello generici) nel Domain Name System (Sistema di nomi di dominio) è in corso d'opera. Fare attenzione a possibili collisioni di nomi quando si sceglie un nome di dominio usato solamente all'interno di una LAN.

Nota
L'uso di pacchetti come `libnss-resolve` insieme a `systemd-resolved`, o `libnss-myhostname`, o `libnss-mymachine`, con quanto corrispondente elencato nella riga "`hosts`" nel file "`/etc/nsswitch.conf`" può scavalcare la configurazione di rete tradizionale discussa sopra. Vedere nss-resolve(8), systemd-resolved(8), nss-myhostname(8) e nss-mymachines(8) per maggiori informazioni.

5.1.2 Il nome dell'interfaccia di rete

systemd usa "nomi di interfacce di rete prevedibili (Predictable Network Interface Name)", come "`enp0s25`".

5.1.3 L'intervallo degli indirizzi di rete per la LAN

Ecco un ripasso degli intervalli di indirizzi IPv4 a 32 bit in ciascuna classe, riservati per l'uso in reti locali (LAN) dalla rfc1918. È garantito che questi indirizzi non creino conflitti con altri indirizzi della vera e propria rete Internet.

Nota
Indirizzi IP scritti con i due punti sono indirizzi IPv6, es. "`::1`" per `localhost`.

Classe	indirizzi di rete	maschera di rete	maschera rete /bit	numero di sottoreti
A	10.x.x.x	255.0.0.0	/8	1
B	172.16.x.x — 172.31.x.x	255.255.0.0	/16	16
C	192.168.0.x — 192.168.255.x	255.255.255.0	/24	256

Tabella 5.2: Elenco di intervalli di indirizzi di rete

Nota
Se uno di questi indirizzi viene assegnato ad un host, allora l'host non deve accedere ad Internet direttamente, ma deve accedervi attraverso un gateway che agisca da proxy per i singoli servizi o che, in alternativa, faccia da NAT (Network Address Translation, traduzione degli indirizzi di rete). I router a banda larga di solito agiscono da NAT per l'ambiente LAN dell'utente finale.

5.1.4 Il supporto per i dispositivi di rete

Nonostante la maggior parte dei dispositivi hardware sia supportata dal sistema Debian, ci sono alcuni dispositivi di rete che necessitano, per essere supportati, firmware non libero secondo la definizione nelle DFSG. Vedere Sezione 9.10.5.

5.2 La configurazione moderna della rete per il desktop

Le interfacce di rete sono tipicamente inizializzate in "`networking.service`" per l'interfaccia `lo` e "`NetworkManager.ser`
per le altre interfacce nei moderni sistemi desktop Debian che usano `systemd`.

Debian può gestire la connessione di rete attraverso software per demoni di gestione come NetworkManager (NM) (network-manager e pacchetti associati).

- Questi sono forniti con propri programmi per l'interfaccia utente con GUI o a riga di comando.
- Sono forniti con un proprio demone come sistema di backend.
- Permettono al proprio sistema di connettersi facilmente ad Internet.
- Permettono una facile gestione della configurazione delle reti cablate e wireless.
- Permettono di configurare la rete in modo indipendente dal datato pacchetto `ifupdown`.

Nota
Non usare questi strumenti di configurazione automatica della rete per server. Sono pensati principalmente per utenti con desktop mobili su portatili.

Questi strumenti moderni di configurazione della rete devono essere configurati in modo corretto per evitare conflitti con il datato pacchetto `ifupdown` e con il suo file di configurazione "`/etc/network/interfaces`".

5.2.1 Strumenti grafici di configurazione della rete

La documentazione ufficiale per NM in Debian è fornita in "`/usr/share/doc/network-manager/README.Debian`".

Fondamentalmente, la configurazione di rete per il desktop viene fatta nel modo seguente.

1. Aggiungere l'utente desktop, ad esempio `pippo`, al gruppo "`netdev`" con il comando seguente. (In alternativa farlo automaticamente attraverso D-bus nei moderni ambienti desktop come GNOME e KDE).

```
$ sudo usermod -a -G netdev foo
```

2. Mantenere la configurazione di "`/etc/network/interfaces`" semplice come la seguente.

```
auto lo
iface lo inet loopback
```

3. Riavviare NM con il comando seguente.

```
$ sudo systemctl restart NetworkManager
```

4. Configurare la propria rete attraverso l'interfaccia grafica.

Nota
Solo le interfacce che **non** sono elencate in "`/etc/network/interfaces`" sono gestite da NM, per evitare conflitti con `ifupdown`.

Suggerimento
Se si desiderano estendere le capacità di configurazione della rete di NM, cercare i moduli plugin appropriati e i pacchetti supplementari, quali `network-manager-openconnect`, `network-manager-openvpn-gnome`, `network-manager-pptp-gnome`, `mobile-broadband-provider-info`, `gnome-bluetooth`, ecc.

5.3 La configurazione moderna della rete senza GUI

Con systemd, la rete può essere configurata invece in `/etc/systemd/network/`. Vedere systemd-resolved(8), resolved.conf(5) e systemd-networkd(8).

Ciò permette la configurazione moderna della rete senza GUI.

Una configurazione di client DHCP può essere impostata creando "`/etc/systemd/network/dhcp.network`". Es.:

```
[Match]
Name=en*

[Network]
DHCP=yes
```

Una configurazione di rete statica può essere impostata creando "`/etc/systemd/network/static.network`". Es.:

```
[Match]
Name=en*

[Network]
Address=192.168.0.15/24
Gateway=192.168.0.1
```

5.4 La configurazione moderna della rete per il cloud

La configurazione moderna della rete per il cloud può usare i pacchetti `cloud-init` e `netplan.io` (vedere Sezione 3.8.4).

Il pacchetto `netplan.io` supporta `systemd-networkd` e `NetworkManager` come suoi backend di configurazione della rete, e permette la configurazione dichiarativa della rete usando dati YAML. Quando si cambia YAML:

- eseguire il comando "`netplan generate`" per generare tutta la necessaria configurazione del backend da YAML.
- Eseguire il comando "`netplan apply`" per applicare la configurazione generata ai backend.

Vedere la "documentazione di Netplan", netplan(5), netplan-generate(8) e netplan-apply(8).

Vedere anche la "documentazione di Cloud-init" (in particolare relativamente alle "fonti di configurazione" e "Netplan Passthrough") per come `cloud-init` può integrare la configurazione di `netplan.io` con fonti alternative di dati.

5.4.1 La configurazione moderna della rete per il cloud con DHCP

Una configurazione di client DHCP può essere impostata creando un file di sorgente di dati "`/etc/netplan/50-dhcp.yaml`":

```
network:
  version: 2
  ethernets:
    all-en:
      match:
        name: "en*"
      dhcp4: true
      dhcp6: true
```

5.4.2 La configurazione moderna della rete per il cloud con IP statico

Una configurazione di rete statica può essere impostata creando un file di sorgente di dati "`/etc/netplan/50-static.yaml`":

```
network:
  version: 2
  ethernets:
    eth0:
      addresses:
        - 192.168.0.15/24
      routes:
        - to: default
          via: 192.168.0.1
```

5.4.3 La configurazione moderna della rete per il cloud con Network Manager

Una configurazione di client di rete che usa l'infrastruttura Network Manager può essere impostata creando un file di sorgente di dati "`/etc/netplan/00-network-manager.yaml`":

```
network:
  version: 2
  renderer: NetworkManager
```

5.5 Configurazione della rete a basso livello

Per la configurazione di rete a basso livello in Linux usare i programmi iproute2 (ip(8), …) .

5.5.1 Comandi iproute2

I comandi Iproute2 offrono funzionalità complete di configurazione della rete a basso livello. Quella che segue è una tabella di traduzione dai comandi obsoleti di net-tools ai nuovi comandi iproute2 ecc.

net-tools obsoleti	nuovi iproute2, ecc.	manipolazione
ifconfig(8)	`ip addr`	indirizzo di protocollo (IP o IPv6) di un device
route(8)	`ip route`	voce nella tabella di instradamento
arp(8)	`ip neigh`	voce nella cache ARP o NDISC
`ipmaddr`	`ip maddr`	indirizzo multicast
`iptunnel`	`ip tunnel`	tunnel over IP
nameif(8)	ifrename(8)	nomina le interfacce di rete in base all'indirizzo MAC
mii-tool(8)	ethtool(8)	impostazioni del device Ethernet

Tabella 5.3: Tabella di traduzione dai comandi obsoleti `net-tools` ai nuovi comandi `iproute2`

Vedere ip(8) e Linux Advanced Routing & Traffic Control.

5.5.2 Operazioni sicure a basso livello sulla rete

Si possono usare i comandi a basso livello per la rete seguenti in modo sicuro dato che non cambiano la configurazione della rete.

comando	descrizione
`ip addr show`	mostra lo stato del collegamento e l'indirizzo delle interfacce attive
`route -n`	mostra tutta la tabella di instradamento in indirizzi numerici
`ip route show`	mostra tutta la tabella di instradamento in indirizzi numerici
`arp`	mostra l'attuale contenuto delle tabelle cache ARP
`ip neigh`	mostra l'attuale contenuto delle tabelle cache ARP
`plog`	mostra il registro del demone ppp
`ping yahoo.com`	controlla la connessione Internet verso "`yahoo.com`"
`whois yahoo.com`	controlla chi ha registrato "`yahoo.com`" nel database dei domini
`traceroute yahoo.com`	traccia la connessione Internet verso "`yahoo.com`"
`tracepath yahoo.com`	traccia la connessione Internet verso "`yahoo.com`"
`mtr yahoo.com`	traccia la connessione Internet verso "`yahoo.com`" (ripetutamente)
`dig [@dns-server.com] esempio.com [{a|mx|any}]`	controlla i record DNS di "`esempio.com`" con "`dns-server.com`" alla ricerca di un record "a", "mx" o "any"
`iptables -L -n`	controlla il filtraggio dei pacchetti
`ss -a`	trova tutte le porte aperte
`ss -l`	trova le porte in ascolto
`ss -l -t -n`	trova le porte (numeriche) TCP in ascolto
`dlint esempio.com`	controlla le informazioni DNS di zona per "`esempio.com`"

Tabella 5.4: Elenco di comandi di rete a basso livello

Suggerimento
Alcuni di questi strumenti di configurazione di basso livello della rete sono contenuti in "`/usr/sbin/`". Potrebbe essere necessario fornire il percorso completo dei comandi, come "`/usr/sbin/ifconfig`" o aggiungere "`/usr/sbin`" all'elenco in "`$PATH`" nel proprio file "`~/.bashrc`".

5.6 Ottimizzazione della rete

L'ottimizzazione generale della rete va oltre gli scopi che questo documento si prefigge. Vengono trattati solo quegli argomenti che interessano le connessioni dei computer personali.

pacchetto	popcon	dimensione	descrizione
iftop	V:6, I:85	93	mostra informazioni sull'uso della banda su di un'interfaccia di rete
iperf	V:2, I:32	431	strumento di misurazione della banda del protocollo Internet
ifstat	V:0.5, I:5.3	52	Monitoraggio delle STATistiche sulle InterFacce
bmon	V:2, I:19	154	strumento portabile per monitoraggio della banda e stime della velocità
ethstatus	V:0.2, I:2.3	41	script che misura velocemente la quantità di dati passanti per un device di rete
bing	V:0.08, I:0.47	80	tester empirico stocastico della banda
bwm-ng	V:1.0, I:9.1	95	strumento piccolo e semplice basato sulla console per il monitoraggio della banda
ethstats	V:0.05, I:0.36	21	strumento basato su console per il monitoraggio delle statistiche Ethernet
ipfm	V:0.04, I:0.09	78	strumento di analisi della banda

Tabella 5.5: Elenco degli strumenti di ottimizzazione della rete

5.6.1 Trovare l'MTU ottimale

NM di solito imposta automaticamente la Maximum Transmission Unit (MTU) ottimale.

In alcune occasioni si può volere impostare l'MTU manualmente dopo esperimenti con ping(8) con l'opzione "`-M do`" per inviare un pacchetto ICMP con varie dimensioni di pacchetti di dati. MTU è la dimensione massima di pacchetti dati con successo senza frammentazione IP più 28 byte per IPv4 e 48 byte per IPv6. Per esempio, quanto segue scopre che l'MTU per la connessione IPv4 è 1460 e l'MTU per la connessione IPv6 è 1500.

```
$ ping -4 -c 1 -s $((1500-28)) -M do www.debian.org
PING  (149.20.4.15) 1472(1500) bytes of data.
ping: local error: message too long, mtu=1460

---  ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

$ ping -4 -c 1 -s $((1460-28)) -M do www.debian.org
PING  (130.89.148.77) 1432(1460) bytes of data.
1440 bytes from klecker-misc.debian.org (130.89.148.77): icmp_seq=1 ttl=50 time=325 ms

---  ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 325.318/325.318/325.318/0.000 ms
```

```
$ ping -6 -c 1 -s $((1500-48)) -M do www.debian.org
PING www.debian.org(mirror-csail.debian.org (2603:400a:ffff:bb8::801f:3e)) 1452 data bytes
1460 bytes from mirror-csail.debian.org (2603:400a:ffff:bb8::801f:3e): icmp_seq=1 ttl=47 ↩
    time=191 ms

--- www.debian.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 191.332/191.332/191.332/0.000 ms
```

Questo processo si chiama ”Path MTU (PMTU) discovery (RFC1191) e il comado tracepath(8) può automatizzarlo.

ambiente di rete	MTU	motivo
collegamento dial-up (IP: PPP)	576	standard
collegamento Ethernet (IP: DHCP o statico)	1500	standard e predefinito

Tabella 5.6: Linee guida di base per il valore di MTU ottimale

In aggiunta a queste linee guida di base si dovrebbe sapere quanto segue.

- Qualsiasi uso di metodi di tunnelling (VPN, ecc.) può ridurre ulteriormente l’MTU ottimale per il loro carico.
- Il valore di MTU non dovrebbe superare il valore di PMTU determinato sperimentalmente.
- A patto di seguire le indicazioni precedenti un valore maggiore di MTU è meglio.

Il valore di MSS (Maximum Segment Size, dimensione massima di segmento)MSS è usato come misura alternativa della dimensione dei pacchetti. Le relazioni tra MSS e MTU sono le seguenti.

- MSS = MTU - 40 per IPv4
- MSS = MTU - 60 per IPv6

Nota
L’ottimizzazione basata su iptables(8) (vedere Sezione 5.7) può bloccare la dimensione dei pacchetti al valore MSS ed è utile per il router. Vedere ”TCPMSS” in iptables(8).

5.6.2 Ottimizzazione di TCP per la WAN

Il flusso TCP può essere massimizzato regolando i parametri di dimensione del buffer TCP come descritto in ”TCP tuning” per le moderne WAN a larga banda e alta latenza. Per adesso, le attuali impostazioni predefinite di Debian funzionano bene persino per la mia LAN connessa con un servizio FTTP veloce a 1G bps.

5.7 Infrastruttura netfilter

Netfilter fornisce un’infrastruttura per firewall con stati e traduzione di indirizzi di rete (NAT) con moduli del kernel Linux (vedere Sezione 3.10).

Vedere Il progetto ”nftables” di netfilter.org per il più recente comando in spazio utente nft(8) nel pacchetto `nftables`. (Kernel Linux >= 3.13)

Vedere il progetto ”iptables” di netfilter.org per il più vecchio iptables(8) in spazio utente e comandi simili nel pacchetto `iptables` e altri correlati. (Kernel Linux >= 2.4.x)

pacchetto	popcon	dimensione	descrizione
nftables	V:266, I:880	189	strumenti amministrativi per filtraggio di pacchetti di rete e NAT (Netfilter) (successore di {ip,ip6,arp,eb}tables)
iptables	V:381, I:635	2408	strumenti di amministrazione per netfilter (iptables(8) per IPv4, ip6tables(8) per IPv6)
arptables	V:0.1, I:1.6	102	strumenti di amministrazione per netfilter (arptables(8) per ARP)
ebtables	V:14, I:23	276	strumenti di amministrazione per netfilter (ebtables(8) per bridge Ethernet)
iptstate	V:0.1, I:1.6	121	monitora continuamente lo stato di netfilter (in modo simile a top(1))
ufw	V:83, I:109	862	Uncomplicated Firewall (UFW) è un programma per gestire un firewall netfilter
gufw	V:6, I:11	3682	interfaccia utente grafica per Uncomplicated Firewall (UFW)
firewalld	V:15, I:22	2581	firewalld è un programma per firewall gestito dinamicamente con supporto per le zone di rete
firewall-config	V:0.8, I:3.2	1096	interfaccia utente grafica per firewalld
shorewall-init	V:0.17, I:0.33	88	inizializzazione di Shoreline Firewall
shorewall	V:2.1, I:4.7	3090	generatore di file di configurazione per Shoreline Firewall, netfilter
shorewall-lite	V:0.03, I:0.06	71	generatore di file di configurazione per Shoreline Firewall, netfilter (versione leggera)
shorewall6	V:0.6, I:1.2	1334	generatore di file di configurazione per Shoreline Firewall, netfilter (versione IPv6)
shorewall6-lite	V:0.01, I:0.02	71	generatore di file di configurazione per Shoreline Firewall, netfilter (versione IPv6 leggera)

Tabella 5.7: Elenco di strumenti per firewall

Si può configurare manualmente netfilter in modo interattivo dalla shell, salvare il suo stato con "`nft list ruleset > file`" o iptables-save(8) e ripristinarlo attraverso uno script di init con "`nft -f file`" o iptables-restore(8) al riavvio del sistema.

Script di aiuto per la configurazione come shorewall facilitano questo processo.

Vedere la documentazione in Netfilter Documentation (o in "`/usr/share/doc/iptables/html/`").

- Linux Networking-concepts HOWTO
- Linux 2.4 Packet Filtering HOWTO
- Linux 2.4 NAT HOWTO
- nftables HOWTO

Capitolo 6

Applicazioni per la rete

Dopo aver stabilito la connessione di rete (vedere Capitolo 5), si possono eseguire svariate applicazioni per la rete.

Suggerimento
Per una guida moderna sull'infrastruttura di rete specifica per Debian leggere The Debian Administrator's Handbook — Network Infrastructure.

Suggerimento
Se si abilita la "Verifica in 2 passaggi" con alcuni ISP è necessario ottenere una password di applicazione per accedere ai servizi POP e SMTP dal proprio programma. Può essere necessario approvare l'IP del proprio host in anticipo.

6.1 Browser web

Ci sono molti pacchetti per browser web per accedere a contenuti remoti tramite HTTP (Hypertext Transfer Protocol).

pacchetto	popcon	dimensione	tipo	descrizione del browser web
chromium	V:31, I:100	320285	per X	Chromium, (browser open-source da Google)
firefox	V:15, I:21	308351	" "	Firefox, (browser open-source da Mozilla, disponibile solo in Debian Unstable)
firefox-esr	V:186, I:425	299827	" "	Firefox ESR, (Firefox Extended Support Release)
epiphany-browser	V:2, I:10	6735	" "	Epiphany, per GNOME, aderente alle HIG
konqueror	V:26, I:111	8179	" "	Konqueror, per KDE
dillo	V:0.4, I:3.9	2203	" "	Dillo, (browser leggero, basato su FLTK)
w3m	V:10, I:129	2853	testuali	w3m
lynx	V:29, I:455	2031	" "	Lynx
elinks	V:2, I:15	1791	" "	ELinks
links	V:2, I:20	2321	" "	Links (solo testo)
links2	V:0.8, I:9.4	5466	grafici	Links (grafica in console senza X)

Tabella 6.1: Elenco di browser web

6.1.1 Falsificare la stringa User-Agent

Per potere accedere ad alcuni siti web eccessivamente restrittivi si può dover falsificare la stringa User-Agent restituita dal programma del browser web. Vedere:

- MDN Web Docs: userAgent
- Chrome Developers: Override the user agent string
- How to change your user agent
- Come cambiare User-Agent in Chrome, Firefox, Safari e altri
- Come cambiare l'User Agent del proprio browser senza installare alcuna estensione
- Come cambiare l'User Agent in Gnome Web (epiphany)

6.1.2 Estensione per il browser

Tutti i browser con GUI moderni supportano estensioni per il browser basate su codice sorgente e ciò sta diventando standardizzato come estensioni web.

6.2 Il sistema di posta

Questa sezione si focalizza su tipiche postazioni di lavoro mobile con connessioni Internet di largo consumo.



Attenzione
Se si desidera impostare il server di posta per scambiare la posta direttamente con Internet, si dovrebbe leggere una documentazione più dettagliata di questo documentazione base.

6.2.1 Nozioni di base sulla posta elettronica

Un messaggio email è formato da tre componenti: la busta del messaggio, l'intestazione del messaggio e il corpo del messaggio.

- Le informazioni "To" e "From" nella busta del messaggio vengono usate dall'SMTP per consegnare l'email. (L'informazione "From" nella busta del messaggio è anche chiamata indirizzo di bounce, From_, ecc.)
- Le informazioni "To" e "From" nell'intestazione del messaggio vengono visualizzate dal programma di posta. (Benché nella maggior parte dei casi questi sono identici a quelli nella busta del messaggio, può non essere sempre così.)
- Il formato dei messaggi di email che copre i dati nell'intestazione e nel corpo è esteso da MIME (Multipurpose Internet Mail Extensions) dal semplice testo ASCII ad altre codifiche di caratteri, oltre ad allegati per audio, video, immagini e programmi applicativi.

I programmi di posta completi basati su GUI offrono tutte le funzioni seguenti utilizzando una configurazione intuitiva basata su GUI.

- Creano e interpretano i dati nell'intestazione e nel corpo del messaggio usando MIME (Multipurpose Internet Mail Extensions) per gestire il tipo di dati e la codifica del contenuto.

- Si autenticano con i server SMTP e IMAP dell'ISP usando la vecchia autenticazione di accesso di base o la moderna OAuth 2.0. (Per OAuth 2.0, impostarla attraverso le impostazioni dell'ambiente desktop. Ad esempio "Impostazioni" -> "Account online".)
- Inviano i messaggi al server smarthost SMTP dell'ISP in ascolto per l'arrivo dei messaggi sulla porta (587).
- Ricevono i messaggi memorizzati sul server dell'ISP dalla porta TLS/IMAP4 (993).
- Possono filtrare la posta in base ai suoi attributi.
- Possono offrire funzionalità aggiuntive: contatti, calendario, attività, promemoria.

pacchetto	popcon	dimensione	tipo
`evolution`	V:27, I:229	489	programma per X con interfaccia utente grafica (GNOME, suite groupware)
`thunderbird`	V:39, I:102	303329	programma per X con interfaccia utente grafica (GTK, Mozilla Thunderbird)
`kmail`	V:43, I:103	25265	programma per X con interfaccia utente grafica (KDE)
`mutt`	V:10, I:82	7334	programma basato su terminale a caratteri probabilmente usato con `vim`
`mew`	V:0.01, I:0.14	2319	programma basato su terminale a caratteri in `(x)emacs`

Tabella 6.2: Elenco di programmi di posta (MUA)

6.2.2 Limitazioni moderne ai servizi di posta

Ai servizi di posta moderni vengono applicate alcune restrizioni al fine di minimizzare l'esposizione ai problemi di spam (posta non desiderata e non richiesta).

- Eseguire un server SMTP su una rete di un utente privato per inviare posta in modo diretto ad host remoti in modo affidabile non è un'ipotesi realistica.
- Un messaggio di posta può essere rifiutato in modo silenzioso da qualsiasi host sul percorso verso la destinazione, a meno che non appaia il più autentico possibile.
- Non è realistico attendersi che un singolo smarthost invii messaggi di indirizzi di posta di origine non correlati ad host remoti in modo affidabile.

Questo perché:

- Le connessioni alla porta SMTP (25) da host serviti dalla connessione alla rete Internet per normali consumatori sono bloccate.
- Le connessioni alla porta SMTP (25) verso host serviti dalla connessione alla rete Internet per normali consumatori sono bloccate.
- I messaggi in uscita da host serviti dalla connessione alla rete Internet per normali consumatori possono essere inviati solamente attraverso la porta di invio dei messaggi (587).
- Tecniche anti-spam come DKIM (DomainKeys Identified Mail) e SPF (Sender_Policy_Framework) e DMARC (Domain-based Message Authentication, Reporting and Conformance) sono usate di frequente per il filtraggio delle email.
- Il servizio DomainKeys Identified Mail può essere fornito per la propria posta inviata attraverso uno smarthost.
- Lo smarthost può riscrivere l'indirizzo di posta dell'origine nell'intestazione del messaggio con l'account di posta dell'utente sullo smarthost per evitare la falsificazione dell'indirizzo email.

6.2.3 Attese storiche da un servizio di posta

Alcuni programmi in Debian si aspettano, come impostazione predefinita o personalizzata, di accedere al comando `/usr/sbin/sendmail` per inviare email dato che il servizio di posta in un sistema UNIX storicamente funzionava così:

- Viene creato un messaggio email come file di testo.
- L'email viene passata al comando `/usr/sbin/sendmail`.
- Per un indirizzo di destinazione sullo stesso host il comando `/usr/sbin/sendmail` fa una consegna locale dell'email aggiungendola in fondo al file `/var/mail/$username`.
  - Comandi che si aspettano questa funzionalità: `apt-listchanges`, `cron`, `at`, ...
- Per un indirizzo di destinazione su un host remoto, il comando `/usr/sbin/sendmail` fa un trasferimento remoto usando SMTP dell'email all'host di destinazione trovato dal record MX del DNS.
  - Comandi che si aspettano questa funzionalità: `popcon`, `reportbug`, `bts`, ...

6.2.4 Agente di trasporto della posta (MTA)

Le macchine Debian portatili possono essere configurate semplicemente con programmi di posta completi con GUI senza un programma MTA (agente di trasferimento posta) a partire da Debian 12 Bookworm.

Debian tradizionalmente installava un programma MTA per supportare i programmi che si aspettano il comando `/usr/sbin/sendmail`. Un tale MTA in una macchina portatile deve vedersela con Sezione 6.2.2 e Sezione 6.2.3.

Per le postazioni di lavoro mobile, la scelta tipica per un MTA è `exim4-daemon-light` o `postfix` con selezionata la loro opzione di installazione tipo "Mail inviata via smarthost; ricevuta con SMTP o fetchmail". Questi sono MTA leggeri che rispettano "`/etc/aliases`".

Suggerimento
Configurare `exim4` per inviare la posta Internet usando smarthost multipli che corrispondono a indirizzi email di origine multipli non è banale. Se è necessaria questa funzionalità per alcuni programmi, impostarli per utilizzare `msmtp` che è facile da configurare per indirizzi email di origine multipli. Poi lasciare l'MTA principale solo per un singolo indirizzo email.

6.2.4.1 La configurazione di exim4

Per la posta di Internet attraverso uno smarthost, riconfigurare i pacchetti `exim4-*` nel modo seguente.

```
$ sudo systemctl stop exim4
$ sudo dpkg-reconfigure exim4-config
```

Selezionare "posta inviata tramite «uno smarthost», ricevuta via SMTP o fetchmail" per "Tipo di configurazione del sistema di posta".

Impostare "Mail name del sistema:" al suo valore predefinito come FQDN (vedere Sezione 5.1.1).

Impostare "indirizzi IP sui quali attendere connessioni SMTP in ingresso:" al suo valore predefinito "127.0.0.1 ; ::1".

Svuotare il contenuto di "Altre destinazioni per conto delle quali accettare posta:".

Svuotare il contenuto di "Sistemi per i quali fare il "relay":".

Impostare l'"Indirizzo IP o nome host dello smarthost per la posta in uscita:" a "smtp.hostname.dom:587".

Selezionare "No" per "Omettere il mail name locale dai messaggi in uscita?". (Usare invece "`/etc/email-addresses`" come in Sezione 6.2.4.3.)

Rispondere a "Mantenere al minimo il numero di richieste DNS (Dial-on-Demand)?" in uno dei modi seguenti.

pacchetto	popcon	dimensione	descrizione
exim4-daemon-light	V:200, I:209	1776	agente di trasporto della posta Exim4 (MTA predefinito in Debian)
exim4-daemon-heavy	V:4.8, I:4.9	1954	agente di trasferimento della posta Exim4 (MTA alternativo flessibile)
exim4-base	V:205, I:214	1685	documentazione (formato testo) e file comuni per Exim4
exim4-doc-html	I:0.98	3809	documentazione per Exim4 (in formato HTML)
exim4-doc-info	I:0.55	649	documentazione per Exim4 (in formato info)
postfix	V:102, I:107	4274	agente di trasferimento della posta Postfix (MTA, alternativa sicura)
postfix-doc	I:4.3	5028	documentazione per Postfix (formati HTML e testo)
sasl2-bin	V:4.5, I:9.9	381	implementazione dell'API SASL Cyrus (di supporto a Postfix per SMTP AUTH)
cyrus-sasl2-doc	I:0.68	2140	SASL Cyrus - documentazione
msmtp	V:9, I:15	822	MTA leggero
msmtp-mta	V:7.2, I:9.1	139	MTA leggero (estensione per la compatibilità con sendmail per `msmtp`)
nullmailer	V:7.4, I:7.7	483	MTA ridotto, niente posta locale
ssmtp	V:3.9, I:5.9	134	MTA ridotto, niente posta locale
sendmail-bin	V:10, I:10	1954	MTA completo (solo se si ha già familiarità con esso)
git-email	V:0.3, I:9.3	1229	il programma git-send-email(1) per inviare email con serie di patch

Tabella 6.3: Elenco di pacchetti base relativi ai server di trasporto della posta

- "No", se il sistema è connesso ad Internet durante l'avvio.
- "Sì", se il sistema **non** è connesso ad Internet durante l'avvio.

Impostare "Modalità di consegna per la posta locale:" a "Formato mbox in /var/mail/".

Selezionare "Sì" per "Dividere la configurazione in molti piccoli file?:".

Creare voci per la password dello smarthost modificando il file "`/etc/exim4/passwd.client`".

```
$ sudo vim /etc/exim4/passwd.client
 ...
$ cat /etc/exim4/passwd.client
^smtp.*\.hostname\.dom:username@hostname.dom:password
```

Configurare exim4(8) con "`QUEUERUNNER='queueonly'`", "`QUEUERUNNER='nodaemon'` ecc. in "`/etc/default/exim4`" per minimizzare l'uso delle risorse di sistema (opzionale).

Avviare `exim4` con il comando seguente.

```
$ sudo systemctl start exim4
```

Il nome host in "`/etc/exim4/passwd.client`" non dovrebbe essere l'alias. Si può controllare il vero nome host nel modo seguente.

```
$ host smtp.hostname.dom
smtp.hostname.dom is an alias for smtp99.hostname.dom.
smtp99.hostname.dom has address 123.234.123.89
```

Per aggirare il problema degli alias, io uso espressioni regolari nel file "`/etc/exim4/passwd.client`" SMTP AUTH probabilmente funziona anche se il fornitore di servizi Internte sposta l'host a cui punta l'alias.

Si può aggiornare manualmente la configurazione di `exim4` facendo quanto segue:

- Aggiornare i file di configurazione di `exim4` in "`/etc/exim4/`".
  - Creare "`/etc/exim4/exim4.conf.localmacros`" per impostare le MACRO e modificare "`/etc/exim4/exim4.co`" (Configurazione non suddivisa.)
  - Creare nuovi file o modificare quelli esistenti nelle sottodirectory "`/etc/exim4/exim4.conf.d`". (Configurazione suddivisa.)
- Eseguire "`systemctl reload exim4`".



Attenzione
Se si è scelta la ripsosta "No" (risposta predefinita) per la domanda di debconf "Mantenere al minino il numero di richieste DNS (Dial-on-Demand)?" ed il sistema **non** è connesso ad Internet, l'avvio di `exim4` può richiedere un tempo molto lungo.

Leggere la guida ufficiale in "`/usr/share/doc/exim4-base/README.Debian.gz`" e update-exim4.conf(8).



avvertimento
In base a tutte le considerazioni pratiche, usare SMTP con STARTTLS sulla porta 587 o SMTPS (SMTP via SSL) sulla porta 465, invece del semplice SMTP sulla porta 25.

6.2.4.2 La configurazione di Postfix con SASL

Per la posta Internet via smarthost, si dovrebbe come prima cosa leggere la documentazione di Postfix e le pagine man principali.

comando	funzione
postfix(1)	programma di controllo di Postfix
postconf(1)	utilità di configurazione per Postfix
postconf(5)	parametri di configurazione di Postfix
postmap(1)	gestione delle tabelle di consultazione di Postfix
postalias(1)	gestione del database degli alias di Postfix

Tabella 6.4: Elenco delle pagine di manuale di Postfix importanti

Si possono (ri)configurare i pacchetti `postfix` e `sasl2-bin` nel modo seguente.

```
$ sudo systemctl stop postfix
$ sudo dpkg-reconfigure postfix
```

Scegliere "Internet con smarthost".

Impostare "relay host SMTP (vuoto per nessuno):" a "`[smtp.hostname.dom]:587`" e configurarlo nel modo seguente.

```
$ sudo postconf -e 'smtp_sender_dependent_authentication = yes'
$ sudo postconf -e 'smtp_sasl_auth_enable = yes'
$ sudo postconf -e 'smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd'
$ sudo postconf -e 'smtp_sasl_type = cyrus'
$ sudo vim /etc/postfix/sasl_passwd
```

Creare le voci con le password per lo smarthost.

```
$ cat /etc/postfix/sasl_passwd
[smtp.hostname.dom]:587     username:password
$ sudo postmap hush:/etc/postfix/sasl_passwd
```

Avviare `postfix` con il comando seguente.

```
$ sudo systemctl start postfix
```

In questo caso l'uso di "`[`" e "`]`" nel dialogo di `dpkg-reconfigure` e in "`/etc/postfix/sasl_passwd`" assocura che non venga controllato il record MX, ma usato direttamente l'esatto nome host specificato. Vedere "Abilitare l'autenticazione SASL nel client SMTP di Postfix" in "`/usr/share/doc/postfix/html/SASL_README.html`".

6.2.4.3 La configurazione dell'indirizzo di posta

Ci sono diversi file di configurazione dell'indirizzo di posta per gli strumenti di trasporto della posta, di consegna e per i programma di posta.

file	funzione	applicazione
`/etc/mailname`	nome host predefinito per la posta (in uscita)	specifico di Debian, mailname(5)
`/etc/email-addresses`	sostituzione del home host per la posta in uscita	specifico di exim(8), exim4-config_files(5)
`/etc/postfix/generic`	sostituzione del home host per la posta in uscita	specifico di postfix(1), attivato dopo l'esecuzione del comando postmap(1)
`/etc/aliases`	alias dei nomi degli account per la posta in entrata	generale, attivato dopo l'esecuzione del comando newaliases(1)

Tabella 6.5: Elenco dei file di configurazione correlati all'indirizzo di posta

Il **nomeposta** nel file "`/etc/mailname`" è solitamente un nome di dominio pienamente qualificato (FQDN) che è risolto ad uno degli indirizzi IP dell'host. Per le postazioni mobili che non hanno un nome host con un indirizzo IP risolvibile, impostare questo **nomemail** al valore di "`hostname -f`". (Questa è una scelta sicura e funziona sia per `exim4-*` sia per `postfix`.)

Suggerimento
Il contenuto di "`/etc/mailname`" è usato da molti programmi non MTA per determinare il loro comportamento predefinito. Per `mutt`, impostare le variabili "`hostname`" e `from`" nel file `~/muttrc` per sovrascrivere il valore di **mailname**. Per i programmi nel pacchetto `devscripts`, come bts(1) e dch(1), esportare le variabili d'ambiente "`$DEBFULLNAME`" e "`$DEBEMAIL`" per sovrascriverlo.

Suggerimento
Il pacchetto `popularity-contest` normalmente invia posta dall'account di root con FQDN. È necessario impostare `MAILFROM` in `/etc/popularity-contest.conf` come descritto nel file `/usr/share/popularity-contest/default.conf`. In caso contrario la posta verrà rigettata dal server SMTP dello smarthost. Sebbene sia una scocciatura, questo approccio è più sicuro della riscrittura dell'indirizzo sorgente di tutta la posta di root da parte dell'MTA e dovrebbe essere usato per altri demoni e script cron.

Qunado si imposta **mailname** al valore di "`hostname -f`", la modifica dell'indirizzo di posta d'origine con il MTA può essere fatta nel modo seguente.

- Usando il file "`/etc/email-addresses`" per exim4(8), come spiegato in exim4-config_files(5).

- Usando il file "`/etc/postfix/generic`" per postfix(8), come spiegato in generic(5).

Per `postfix` sono necessarie, in aggiunta, le azioni seguenti.

```
# postmap hash:/etc/postfix/generic
# postconf -e 'smtp_generic_maps = hash:/etc/postfix/generic'
# postfix reload
```

Si può testare la configurazione dell'indirizzo di posta nel modo seguente.

- per exim(8) con le opzioni `-brw, -bf, -bF, -bV, …`
- per postmap(1) con l'opzione `-q`.

Suggerimento
Exim viene fornito con svariati programmi di utilità, come exiqgrep(8) e exipick(8). Per conoscere i comandi disponibili vedere "`dpkg -L exim4-base|grep man8/`".

6.2.4.4 Operazioni di base degli MTA

Ci sono svariate operazioni di base degli MTA. Alcune possono essere fatte usando l'interfaccia di compatibilità con sendmail(1).

comando exim	comando postfix	descrizione
`sendmail`	`sendmail`	legge la posta dallo standard input e gestisce la consegna (`-bm`)
`mailq`	`mailq`	elenca la coda della posta con stato e ID dei messaggi in coda (`-bp`)
`newaliases`	`newaliases`	inizializza il database degli alias (`-I`)
`exim4 -q`	`postqueue -f`	invia tutta la posta in attesa (`-q`)
`exim4 -qf`	`postsuper -r ALL deferred; postqueue -f`	invia tutta la posta
`exim4 -qff`	`postsuper -r ALL; postqueue -f`	invia anche la posta congelata
`exim4 -Mg queue_id`	`postsuper -h queue_id`	congela un messaggio in base al suo ID nella coda
`exim4 -Mrm queue_id`	`postsuper -d queue_id`	rimuove un messaggio in base al suo ID nella coda
N/D	`postsuper -d ALL`	rimuove tutti i messaggi

Tabella 6.6: Elenco di operazioni base degli MTA

Suggerimento
Può essere una buona idea far inviare tutta la posta da uno script in "`/etc/ppp/ip-up.d/*`".

6.3 Il server e le utilità per l'accesso remoto (SSH)

SSH (Secure SHell) è il metodo **sicuro** per connettersi in Internet. Una versione libera di SSH chiamata OpenSSH è disponibile nei pacchetti Debian `openssh-client` e `openssh-server`.

pacchetto	popcon	dimensione	strumento	descrizione
openssh-client	V:682, I:997	3611	ssh(1)	client SSH
openssh-server	V:774, I:820	3615	sshd(8)	server SSH
ssh-askpass	I:15	103	ssh-askpass(1)	chiede all'utente una passphrase per ssh-add (X semplice)
ssh-askpass-gnome	V:0.5, I:2.9	46	ssh-askpass-gnome(1)	chiede all'utente una passphrase per ssh-add (GNOME)
ssh-askpass-fullscreen	V:0.09, I:0.40	41	ssh-askpass-fullscreen(1)	chiede all'utente una passphrase per ssh-add (GNOME) con abbellimenti aggiuntivi
shellinabox	V:0.61, I:0.94	528	shellinaboxd(1)	server web per emulatore di terminale VT100 accessibile dal browser

Tabella 6.7: Elenco dei server e delle utilità per l'accesso remoto

Per l'utente ssh(1) funziona come un telnet(1) più intelligente e più sicuro. A differenza del comando `telnet`, `ssh` non si interrompe a seguito del carattere di escape di `telnet` (impostazione predefinita iniziale CTRL-]).

Sebbene `shellinabox` non sia un programma SSH, è elencato qui come alternativa interessante per l'accesso remoto a terminale.

Vedere anche Sezione 7.9 per le connessioni a programmi client X remoti.



Attenzione
Se il proprio SSH è accessibile da Internet, vedere Sezione 4.6.3.

Suggerimento
Per permettere al processo della shell remota di sopravvivere all'interruzione della connessione, usare il programma screen(1) (vedere Sezione 9.1.2).

6.3.1 Nozioni di base su SSH

Il demone SSH OpenSSH ha solamente il supporto per il protocollo SSH.

Leggere "`/usr/share/doc/openssh-client/README.Debian.gz`", ssh(1), sshd(8), ssh-keygen(1), ssh-add(1) e ssh-agent(1).



avvertimento
Se si desidera eseguire il server OpenSSH, non deve esistere il file "`/etc/ssh/sshd_not_to_be_run`". Non abilitare l'autenticazione basata su rhost (`HostbasedAuthentication` in `/etc/ssh/sshd_config`).

I comandi seguenti avviano una connessione ssh(1) da un client.

file di configurazione	descrizione del file di configurazione
`/etc/ssh/ssh_config`	impostazioni predefinite per il client SSH, vedere ssh_config(5)
`/etc/ssh/sshd_config`	impostazioni predefinite per il server SSH, vedere ssh_config(5)
`~/.ssh/authorized_keys`	chiavi SSH pubbliche predefinite che i client usano per connettersi a questo account su questo server SSH
`~/.ssh/id_rsa`	chiave RSA SSH-2 segreta dell'utente
`~/.ssh/id_`*`nome-tipo-chiave`*	chiave SSH-2 segreta *nome-tipo-chiave* come `ecdsa`, `ed25519`, ... dell'utente

Tabella 6.8: Elenco dei file di configurazione per SSH

comando	descrizione
`ssh nomeutente@nomehost.dominio.ext`	connette nella modalità predefinita
`ssh -v nomeutente@nomehost.dominio.ext`	connette nella modalità predefinita con messaggi di debug
`ssh -o PreferredAuthentications=password nomeutente@nomehost.dominio.ext`	forza l'uso di password con SSH versione 2
`ssh -t nomeutente@nomehost.dominio.ext passwd`	esegue il programma `passwd` per aggiornare la password su un host remoto

Tabella 6.9: Elenco di esempi di avvio di client SSH

6.3.2 Nome utente sull'host remoto

Se si usa lo stesso nome utente sull'host locale e su quello remoto, si può evitare di digitare "`nomeutente@`".

Anche se si usa un nome utente diverso nell'host locale rispetto a quello remoto, si può evitare di digitare quella parte usando "`~/.ssh/config`". Per il servizio Debian Salsa con nome account "`pippo-guest`", impostare "`~/.ssh/config`" in modo che contenga quanto segue.

```
Host salsa.debian.org people.debian.org
User foo-guest
```

6.3.3 Connettersi senza password remote

Si può evitare di doversi ricordare le password per sistemi remoti usando "`PubkeyAuthentication`" (protocollo SSH-2).

Sul sistema remoto impostare in "`/etc/ssh/sshd_config`" le rispettive voci "`PubkeyAuthentication yes`".

Generare localmente le chiavi di autenticazione ed installare la chiave pubblica sul sistema remoto con i comandi seguenti.

```
$ ssh-keygen -t rsa
$ cat .ssh/id_rsa.pub | ssh user1@remote "cat - >>.ssh/authorized_keys"
```

Si possono aggiungere opzioni alle voci in "`~/.ssh/authorized_keys`" per limitare gli host e per eseguire comandi specifici. Vedere "AUTHORIZED_KEYS FILE FORMAT" in sshd(8).

6.3.4 Gestire client SSH estranei

Sono disponibili alcuni client SSH liberi per altre piattaforme.

ambiente	programma ssh libero
Windows	puTTY (PuTTY: un client SSH e Telnet libero) (GPL)
Windows (cygwin)	SSH in cygwin (Cygwin: l'esperienza Linux, in Windows) (GPL)
Mac OS X	OpenSSH; usare `ssh` nell'applicazione Terminale (GPL)

Tabella 6.10: Elenco di client SSH per altre piattaforme

6.3.5 Impostare ssh-agent

È meglio per ragioni di sicurezza proteggere la propria chiave segreta di autenticazione SSH con una passphrase. Se non è già stata impostata una passphrase usare "`ssh-keygen -p`" per farlo.

Mettere la propria chiav SSH pubblica (ad esempio "`~/.ssh/id_rsa.pub`") in "`~/.ssh/authorized_keys`" su un host remoto usando una connessione all'host remoto basata su password, come descritto in precedenza.

```
$ ssh-agent bash
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/username/.ssh/id_rsa:
Identity added: /home/username/.ssh/id_rsa (/home/username/.ssh/id_rsa)
```

Per il comando successivo non sarà più necessaria da questo momento la password remota.

```
$ scp foo username@remote.host:foo
```

Premere ~D per terminare la sessione ssh-agent.

Per il server X, il normale script di avvio Debian esegue `ssh-agent` come processo genitore. Perciò è necessario eseguire `ssh-add` una volta sola. Per ulteriori informazioni, leggere ssh-agent(1) e ssh-add(1).

6.3.6 Inviare un messaggio di posta da un host remoto

Se si ha un account di shell SSH su un server con impostazioni DNS appropriate, si può inviare un messaggio di posta generato sulla propria postazione di lavoro come email genuinamente inviata dal server remoto.

```
$ ssh username@example.org /usr/sbin/sendmail -bm -ti -f "username@example.org" < mail_data ↩
    .txt
```

6.3.7 Forwarding della porta per tunnel SMTP/POP3

Per stabilire una pipe per connettersi alla porta 25 del `server-remoto` dalla porta 4025 dell'`host locale` e alla porta 110 del `server-remoto` dalla porta 4110 dell'`host locale` attraverso `ssh`, eseguire sull'host locale il comando seguente.

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 username@remote-server
```

Questo è un metodo sicuro di creare connessioni a server SMTP/POP3 in Internet. Impostare nel file "`/etc/ssh/sshd_con` dell'host remoto la voce "`AllowTcpForwarding`" a "`yes`".

6.3.8 Spegnere il sistema remoto su SSH

È necessario proteggere il processo che esegue "`shutdown -h now`" (vedere Sezione 1.1.8) dalla terminazione di SSH usando il comando at(1) (vedere Sezione 9.4.13) nel modo seguente.

```
# echo "shutdown -h now" | at now
```

Eseguire "`shutdown -h now`" in una sessione screen(1) (vedere Sezione 9.1.2) è un altro modo di ottenere lo stesso risultato.

6.3.9 Risolvere il problemi con SSH

Se si hanno problemi, controllare i permessi dei file di configurazione ed eseguire `ssh` con l'opzione "`-v`".

Se si è root e si hanno problemi con un firewall usare l'opzione "`-p`"; questo evita l'uso delle porte 1 - 1023 del server.

Se le connessioni `ssh` ad un sito remoto smettono di funzionare improvvisamente, potrebbe essere a causa di modifiche fatte dall'amministratore di sistema, molto probabilmente cambiamenti in "`host-key`" durante l'amministrazione del sistema. Dopo essersi accertati che questa sia davvero la causa e che nessuna stia cercando di falsificare l'host remoto con qualche trucchetto, si può riottenere una connessione rimuovendo la voce "`host-key`" dal file "`~/.ssh/known_hosts`" sull'host locale.

6.4 Server ed utilità per la stampa

Nei vecchi sistemi in stile Unix, il demone di stampa in linea (lpd) BSD era lo standard e il formato di stampa standard del software libero classico era PostScript (PS). Un sistema di filtri veniva usato insieme a Ghostscript per permettere la stampa su stampanti non-PostScript. Vedere Sezione 11.4.1.

Nei sistemi Debian moderni, Common UNIX Printing System (CUPS) è lo standard di fatto e il formato standard per la stampa nel software libero moderno è Portable Document Format (PDF).

CUPS usa il protocollo IPP (Internet Printing Protocol). IPP è lo standard di fatto multipiattaforma per la stampa da remoto con capacità di comunicazione bidirezionale.

Grazie alla funzionalità di auto-conversione in base al formato dei file del sistema CUPS, il semplice passaggio di qualsiasi tipo di dati al comando `lpr` dovrebbe generare l'output di stampa atteso. (In CUPS, il comando `lpr` può essere abilitato installando il pacchetto `cups-bsd`.)

Il sistema Debian ha alcuni pacchetti degni di nota per ciò che riguarda i server e le utilità di stampa

pacchetto	popcon	dimensione	porta	descrizione
`lpr`	V:1.9, I:2.2	378	printer (515)	lpr/lpd BSD (Line printer daemon, demone di stampa in linea)
`cups`	V:144, I:447	1088	IPP (631)	server CUPS di stampa Internet
`cups-client`	V:152, I:459	429	" "	comandi System V per la stampa con CUPS: lp(1), lpstat(1), lpoptions(1), cancel(1), lpmove(8), lpinfo(8), lpadmin(8), …
`cups-bsd`	V:34, I:182	131	" "	comandi BSD per la stampa con CUPS: lpr(1), lpq(1), lprm(1), lpc(8)
`printer-driver-gutenprint`	V:11, I:52	1117	Non applicabile	driver di stampa per CUPS

Tabella 6.11: Elenco di server e utilità di stampa

Suggerimento
Si può configurare il sistema CUPS indirizzando il proprio browser web all'indirizzo "http://localhost:631/".

6.5 Altri server di rete

Ecco un elenco di altri server applicativi di rete.

CIFS (Common Internet File System Protocol) è lo stesso protocollo di SMB (Server Message Block) ed è ampiamente usato da Microsoft Windows.

pacchetto	popcon	dimensione	protocollo	descrizione
telnetd	V:0.3, I:1.6	56	TELNET	server TELNET
nfs-kernel-server	V:47, I:54	874	NFS	condivisione di file Unix
samba	V:110, I:123	5182	SMB	condivisione di file e stampanti Windows
netatalk	V:0.62, I:0.85	933	ATP	condivisione di file e stampanti Apple/Mac (AppleTalk)
proftpd-basic	V:3.3, I:8.4	452	FTP	scaricamento generico di file
apache2	V:175, I:211	587	HTTP	server web generico
squid	V:9.0, I:9.4	9583	” ”	server proxy web generico
bind9	V:33, I:35	904	DNS	indirizzo IP per altri host
kea	I:0.51	245	DHCP	indirizzo IP per il client stesso

Tabella 6.12: Elenco di altri server applicativi di rete

Suggerimento
Vedere Sezione 4.5.2 per l’integrazione di sistemi server.

Suggerimento
La risoluzione del nome di host è solitamente fornita dal server DNS. Per l’indirizzo IP dell’host assegnato dinamicamente da DHCP, può essere impostato un DNS dinamico per la risoluzione del nome host, usando `bind9` e `kea` come descritto nella pagina del Wiki Debian sui DDNS.

Suggerimento
L’uso di server proxy come `squid` è molti più efficiente per risparmiare banda rispetto all’uso di server mirror locali con il completo contenuto dell’archivio Debian.

6.6 Altri client di rete

Ecco un elenco di altri client applicativi di rete.

6.7 Diagnosi dei demoni di sistema

Il programma `telnet` permette la connessione manuale ai demoni di sistema e la loro diagnosi.

Per testare il semplice servizio POP3 provare il comando seguente.

```
$ telnet mail.ispname.net pop3
```

Per testare il servizio POP3 con TLS/SSL abilitato di alcuni fornitori di servizi Internet, è necessario un client `telnet` con TLS/SSL abilitato fornito dal pacchdetto `telnet-ssl` o `openssl`.

```
$ telnet -z ssl pop.gmail.com 995
```

```
$ openssl s_client -connect pop.gmail.com:995
```

Le seguenti RFC forniscono le conoscenze necessarie per ciascun demone di sistema.

L’uso delle porte è descritto in ”`/etc/services`”.

pacchetto	popcon	dimensione	protocollo	descrizione
netcat-traditional	V:55, I:906	139	TCP/IP	coltellino svizzero TCP/IP
netcat-openbsd	V:22, I:119	105	TCP/IP	coltellino svizzero TCP/IP con supporto per IPv6, proxy e socket Unix
openssl	V:846, I:996	2564	SSL	eseguibile SSL (Secure Socket Layer) e strumenti crittografici relativi
stunnel4	V:6.6, I:9.9	24	” ”	wrapper SSL universale
telnet	V:10, I:190	56	TELNET	client TELNET
nfs-common	V:147, I:191	1158	NFS	condivisione di file Unix
smbclient	V:25, I:206	2117	SMB	client per condivisione di file e stampanti MS Windows
cifs-utils	V:32, I:118	368	” ”	comandi mount e umount per file MS Windows remoti
wget	V:181, I:982	3784	HTTP e FTP	scaricatore per web
curl	V:260, I:727	520	” ”	” ”
transmission-gtk	V:9, I:143	6259	BitTorrent	client BitTorrent (GTK)
transmission-qt	V:0.6, I:2.6	6213	” ”	client BitTorrent (Qt)
ktorrent	V:1.3, I:5.0	5031	” ”	client BitTorrent (Qt)
qbittorrent	V:8, I:23	15378	” ”	client BitTorrent (Qt)
bind9-host	V:124, I:940	136	DNS	host(1) da bind9, ”Priorità: standard”
dnsutils	V:5, I:146	23	” ”	dig(1) da bind, ”Priorità: standard”
ldap-utils	V:11, I:57	790	LDAP	ottiene dati da server LDAP

Tabella 6.13: Elenco di altri client applicativi di rete

RFC	descrizione
rfc1939 e rfc2449	servzio POP3
rfc3501	servizio IMAP4
rfc2821 (rfc821)	servizio SMTP
rfc2822 (rfc822)	formato file di posta
rfc2045	MIME (Multipurpose Internet Mail Extensions)
rfc819	servizio DNS
rfc2616	servizio HTTP
rfc2396	definizione di URI

Tabella 6.14: Elenco di RFC popolari

Capitolo 7

Sistema GUI

7.1 Ambiente desktop GUI

Ci sono diverse scelte di ambienti desktop GUI completi nel sistema Debian.

pacchetto con task	popcon	dimensione	descrizione
task-gnome-desktop	I:192	9	ambiente desktop GNOME
task-xfce-desktop	I:88	9	ambiente desktop Xfce
task-kde-desktop	I:94	6	ambiente desktop KDE Plasma
task-mate-desktop	I:30	9	ambiente desktop MATE
task-cinnamon-desktop	I:35	9	ambiente desktop Cinnamon
task-lxde-desktop	I:19	9	ambiente desktop LXDE
task-lxqt-desktop	I:16	9	ambiente desktop LXQt
task-gnome-flashback-desktop	I:9.6	6	ambiente desktop GNOME Flashback

Tabella 7.1: Lista di ambienti desktop

Suggerimento
I pacchetti di dipendenza selezionati da un metapacchetto per task possono essere fuori sincrono rispetto al più recente stato di transizione nell'ambiente Debian `unstable`/`testing`. Per `task-gnome-desktop`, può essere necessario aggiustare la selezione dei pacchetti come segue:

- Avviare aptitude(8) con `sudo aptitude -u`.
- Spostare il cursore su "Task" e premere "Invio".
- Spostare il cursore su "Utente finale" e premere "Invio".
- Spostare il cursore su "GNOME" e premere "Invio".
- Spostare il cursore su `task-gnome-desktop` e premere "Invio".
- Spostare il cursore su "Dipende" e premere "m" (selezionato manualmente).
- Spostare il cursore su "Raccomanda" e premere "m" (selezionato manualmente).
- Spostare il cursore su "`task-gnome-desktop`" e premere "-" (eliminare).
- Aggiustare i pacchetti selezionati abbandonando quelli problematici che causano conflitti tra i pacchetti.
- Premere "g" per iniziare l'installazione.

Questo capitolo si concentrerà soprattutto sull'ambiente desktop predefinito in Debian: `task-gnome-desktop` che fornisce GNOME su wayland.

7.2 Il protocollo di comunicazione della GUI

Il protocollo di comunicazione della GUI usato nel desktop GNOME può essere:

- Wayland (protocollo per display server) (nativo)
- protocollo principale del sistema X Window (via `xwayland`)

Controllare sul sito freedesktop.org in cosa l'architettura Wayland è differente dall'architettura X Window.

Dal punto di vista dell'utente le differenze possono essere a grandi linee riassunte in:

- Wayland è un protocollo di comunicazione GUI per unico host: nuovo, più semplice, più veloce, senza binario con setuid root
- X Window è un protocollo di comunicazione GUI con funzionalità di rete: tradizionale, complesso, più lento, con binario con setuid root

Per le applicazioni che usano il protocollo Wayland, l'accesso al contenuto delle loro schermata da un host remoto è supportato da VNC o RDP. Vedere Sezione 7.8.

I server X moderni hanno l'estensione MIT Shared Memory e comunicano con i loro client X locali usando la memoria condivisa locale. Ciò scavalca il canale di comunicazione tra i processi Xlib trasparente alla rete e fa guadagnare in prestazioni. Questa situazione è stata il punto di partenza per la creazione di Wayland come protocollo di comunicazione GUI solo locale.

Usando il programma `xeyes` avviato dal terminale GNOME si può controllare il protocollo di comunicazione GUI usato da ciascuna applicazione GUI.

```
$ xeyes
```

- Se il cursore del mouse è su un'applicazione come "Terminale di GNOME", che usa il protocollo per display server Wayland, gli occhi non si muovono con il movimento del cursore.
- Se il cursore del mouse è su un'applicazione come "xterm", che usa il protocollo base del sistema X Window, gli occhi si muovono insieme al cursore del mouse rendendo evidente la natura non-così-isolata di un'architettura X Window.

Alla data di aprile 2021 molte popolari applicazioni GUI come GNOME e le applicazioni LibreOffice (LO) sono migrate al protocollo per display server Wayland. L'autore nota come `xterm`, `gitk`, `chromium`, `firefox`, `gimp`, `dia` e tutte le applicazioni KDE usino ancora il protocollo di base del sistema X Window.

Nota
Sia per xwayland in Wayland che per il sistema X Window nativo, il vecchio file di configurazione del server X "`/etc/X11/xorg.conf`" non dovrebbe esistere sul sistema. I dispositivi grafici e di input sono ora configurati dal kernel con DRM, KMS e udev. Il server X nativo è stato riscritto per usarli. Vedere "supporto modedb predefinito delle modalità video" nella documentazione del kernel Linux.

7.3 Infrastruttura GUI

Questi sono pacchetti degni di nota dell'infrastruttura GUI per l'ambiente GNOME su Wayland.

pacchetto	popcon	dimensione del pacchetto	descrizione
`mutter`	V:1, I:22	216	Il gestore di finestre mutter di GNOME **[auto]**
`xwayland`	V:256, I:335	2652	Un server X in esecuzione sopra a wayland **[auto]**
`gnome-remote-desktop`	V:147, I:241	2743	Demone per desktop remoto per GNOME che usa PipeWire **[auto]**
`gnome-tweaks`	V:17, I:232	1145	Impostazioni avanzate di configurazione per GNOME
`gnome-shell-extension-prefs`	V:8, I:117	73	Strumenti per abilitare/disabilitare le estensioni per GNOME Shell

Tabella 7.2: Elenco di pacchetti importanti per l'infrastruttura GUI

Qui "**[auto]**" significa che questi pacchetti sono installati automaticamente quando viene installato `task-gnome-desktop`.

Suggerimento
`gnome-tweaks` è l'indispensabile utilità di configurazione. Per esempio:

- Si può forzare la "sovra-amplificazione" del volume dell'audio da "Generale".
- Si può forzare "Maiusc" a diventare "Esc" da "Tastiera e mouse" -> "Tastiera" -> "Opzioni aggiuntive disposizione".

Suggerimento
I dettagli delle funzionalità dell'ambiente desktop GNOME possono essere configurati con utilità avviate digitando "`settings`", "`tweaks`" o "`extensions`" dopo aver premuto il tasto `Super`.

7.4 Applicazioni GUI

In Debian sono ora disponibili molte utili applicazioni GUI. L'installazione di pacchetti software come `scribus` (KDE) nell'ambiente desktop GNOME è piuttosto accettabile dato che le funzionalità corrispondenti non sono disponibili nel desktop GNOME. Tuttavia l'installazione di troppi pacchetti con funzionalità duplicate può rendere il sistema affollato.

Ecco un elenco di applicazioni GUI che hanno attirato la mia attenzione.

7.5 Directory dell'utente

I nomi predefiniti per le directory dell'utente, come "`~/Desktop`", "`~/Documenti`", ..., usate dagli ambienti desktop dipendono dalla localizzazione usata per l'installazione del sistema. Possono essere reimpostate a quelle in inglese usando:

```
$ LANGUAGE=C xdg-user-dirs-update --force
```

Poi si spostano manualmente tutti i dati nelle nuove directory. Vedere xdg-user-dirs-update(1).

È anche possibile impostarle manualmente a qualsiasi nome modificando "`~/.config/user-dirs.dirs`". Vedere user-dirs.dirs(5).

7.6 Tipi di carattere

In Debian sono a disposizione degli utenti molti tipi di carattere utili scalabili. La preoccupazione per l'utente è come evitare ridondanze e come configurare la disabilitazione di parte dei tipi di carattere installati. Altrimenti scelte di tipi di carattere inutili possono ingombrare i menu delle applicazioni GUI.

Il sistema Debia usa la libreria FreeType 2.0 per rasterizzare molti formati di tipi di carattere scalabili per lo schermo e la stampa:

- Tipi di carattere Type 1 (PostScript) che usano curve di Bézier cubiche (formato quasi obsoleto)
- Tipi di carattere TrueType che usano curve di Bézier quadratiche (buona scelta di formato)
- Tipi di carattere OpenType che usano curve di Bézier cubiche (migliora scelta per il formato)

7.6.1 Tipi di carattere base

La tabella seguente è compilata nella speranza che aiuti gli utenti a scegliere i tipi di carattere scalabili appropriati con una chiara idea della compatibilità delle metriche e delle coperture dei glifi. La maggior parte dei tipi di carattere copre tutti i caratteri latini, greci e cirillici. La scelta finale dei tipi di carattere da attivare può anche essere influenzata da gusti estetici personali. Questi tipi di carattere possono essere usati per la visualizzazione a schermo o per la stampa su carta.

Qui:

- "MCM" sta per "Metrica Compatibile con tipi di carattere forniti da Microsoft"
- "MCMATC" sta per "Metrica Compatibile con tipi di carattere forniti da Microsoft: Arial, Times New Roman, Courier New"
- "MCAHTC" sta per "Metrica Compatibile con tipi di carattere forniti da Adobe: Helvetica, Times, Courier"
- I numeri nelle colonne del tipo di carattere rappresentano la larghezza "M" approssimata relativa per il carattere alla stessa dimensione di punto.

pacchetto	popcon	dimensione del pac-chet-to	tipo	descrizione
evolution	V:27, I:229	489	GNOME	gestione di informazioni personali (groupware e posta elettronica)
thunderbird	V:39, I:102	303329	GTK	programma di posta (Mozilla Thunderbird)
kontact	V:1, I:10	2360	KDE	gestione di informazioni personali (groupware e posta elettronica)
libreoffice-writer	V:105, I:422	34232	LO	elaboratore di testi
abiword	V:0.8, I:4.7	3596	GNOME	elaboratore di testi
calligrawords	V:0.2, I:2.7	7144	KDE	elaboratore di testi
scribus	V:1, I:12	32942	KDE	editor per desktop publishing per modificare file PDF
glabels	V:0.3, I:2.5	1283	GNOME	editor di etichette
libreoffice-calc	V:101, I:419	29620	LO	foglio di calcolo
gnumeric	V:3, I:10	9945	GNOME	foglio di calcolo
calligrasheets	V:0.1, I:1.7	14014	KDE	foglio di calcolo
libreoffice-impress	V:88, I:417	2499	LO	presentazioni
calligrastage	V:0.1, I:1.6	6237	KDE	presentazioni
libreoffice-base	V:21, I:67	5130	LO	gestione di database
kexi	V:0.05, I:0.80	7565	KDE	gestione di database
libreoffice-draw	V:89, I:418	11418	LO	editor di grafica vettoriale (draw)
inkscape	V:12, I:76	113353	GNOME	editor di grafica vettoriale (draw)
karbon	V:0.1, I:2.1	4029	KDE	editor di grafica vettoriale (draw)
dia	V:1, I:16	3846	GTK	editor di grafi e diagrammi di flusso
gimp	V:34, I:211	33315	GTK	editor di grafica bitmap (paint)
shotwell	V:14, I:246	6334	GTK	organizzatore di foto digitali
digikam	V:1.5, I:8.3	333	KDE	organizzatore di foto digitali
darktable	V:4, I:11	36092	GTK	tavolo luminoso e camera oscura per fotografi
gnome-video-trimmer	V:0.1, I:1.0	1665	GNOME	taglio senza perdita di file video
kdenlive	V:3, I:21	22134	KDE	editor di video non lineare
shotcut	V:1.0, I:7.5	8270	applicazione Qt	editor di video non lineare
openshot-qt	V:0.9, I:8.8	196007	applicazione Qt	editor di video non lineare
flowblade	V:0.2, I:2.2	42213	GTK	editor di video non lineare
planner	V:0.2, I:4.2	1400	GNOME	gestione progetti
calligraplan	V:0.2, I:2.9	23545	KDE	gestione progetti
gnucash	V:2.0, I:6.7	29434	GNOME	contabilità personale
homebank	V:0.3, I:1.5	3194	GTK	contabilità personale
kmymoney	V:0.4, I:1.9	19297	KDE	contabilità personale
frescobaldi	V:0.3, I:1.5	11102	applicazione Qt	editor di testo per partiture musicali LylyPond
librecad	V:1, I:13	9164	applicazione Qt	sistema CAD (Computer-Aided Design) (2D)
freecad	V:1, I:20	112	applicazione Qt	sistema CAD (Computer-Aided Design) (3D)
kicad	V:3, I:14	224155	GTK	sofware per progettazione di schemi elettronici e PCB
xsane	V:9, I:128	1512	GTK	frontend per lo scanner
libreoffice-math	V:84, I:420	1982	LO	editor di equazioni matematiche e formule
calibre	V:7, I:24	71216	KDE	convertitore e gestione di una biblioteca per libri elettronici
fbreader	V:0.7, I:5.4	3827	GTK	Lettori di ebook

pacchetto	popcon	dimensione	sans	serif	mono	nota sul carattere
fonts-cantarell	V:174, I:288	224	59	-	-	Cantarell (GNOME 3, visualizzazione)
fonts-noto	I:150	30	61	63	40	Tipi di carattere Noto (Google, multi-lingua con CJK)
fonts-dejavu	I:385	34	58	68	40	DejaVu (GNOME 2, MCM:Verdana, Bitstream Vera esteso)
fonts-liberation2	V:40, I:156	15	56	60	40	Tipi di carattere Liberation per LibreOffice (Red Hat, MCMATC)
fonts-croscore	V:20, I:36	5260	56	60	40	Chrome OS: Arimo, Tinos e Cousine (Google, MCMATC)
fonts-crosextra-carlito	V:20, I:90	2696	57	-	-	Chrome OS: Carlito (Google, MCM:Calibri)
fonts-crosextra-caladea	V:12, I:83	347	-	55	-	Chrome OS: Caladea (Google, MCM:Cambria) (solo latini)
fonts-freefont-ttf	V:78, I:203	14460	57	59	40	GNU FreeFont (URW Nimbus esteso)
fonts-quicksand	V:210, I:452	392	56	-	-	Quicksand del task-desktop Debian (visualizzazione, solo latini)
fonts-hack	V:34, I:136	2509	-	-	40 P	Un carattere progettato per il codice sorgente, Hack (Facebook)
fonts-sil-gentiumplus	V:0, I:28	14345	-	54	-	Gentium SIL
fonts-sil-charis	V:1, I:28	6704	-	59	-	Charis SIL
fonts-urw-base35	V:187, I:534	15558	56	60	40	URW Nimbus (Nimbus Sans, Roman No. 9 L, Mono L, MCAHTC)
fonts-ubuntu	V:2.2, I:4.7	4339	58	-	33 P	Tipi di carattere Ubuntu (visualizzazione)
fonts-terminus	V:0.3, I:3.9	451	-	-	33	Cool retro terminal fonts
ttf-mscorefonts-installer	V:1, I:40	85	56?	60	40	Scaricatore per tipi di carattere non liberi di Microsoft (vedere più sotto)

Tabella 7.4: Elenco di tipi di carattere TrueType e OpenType degni di nota

- ”P” nelle colonne dei tipi di carattere mono a spaziatura fissa rappresenta l’usabilità per la programmazione grazie a ”0”/”O” e ”1”/”l”/”I” chiaramente distinguibili.
- Il pacchetto `ttf-mscorefonts-installer` scarica i ”Tipi di carattere principali per il Web” di Microsoft e installa Arial, Times New Roman, Courier New, Verdana, ... Questi tipi di carattere installati sono dati non liberi.

Molti tipi di carattere Latin sono una discendenza dalla famiglia URW Nimbus o Bitstream Vera.

Suggerimento
Se la propria localizzazione necessita di tipi di carattere non coperti bene dai tipi di carattere elencati sopra, usare aptitude per controllare nei pacchetti task elencati in ”Task” -> ”Localizzazione”. I pacchetti di tipi di carattere elencati come ”Dipende:” o ”Raccomanda:” nei pacchetti dei task di localizzazione sono i candidati principali.

7.6.2 Rasterizzazione dei tipi di carattere

Debia usa FreeType per rasterizzare i tipi di carattere. La sua infrastruttura di scelta dei caratteri è fornita dalla libreria per configurazione dei tipi di carattere Fontconfig.

pacchetto	popcon	dimensione	descrizione
`libfreetype6`	V:589, I:997	1050	FreeType, libreria per rasterizzazione di tipi di carattere
`libfontconfig1`	V:577, I:808	344	Fontconfig, libreria per la configurazione dei tipi di carattere
`fontconfig`	V:471, I:693	415	`fc-*`: comandi CLI per Fontconfig
`font-manager`	V:2.3, I:6.6	1116	Font Manager: comando GUI per Fontconfig
`nautilus-font-manager`	V:0.07, I:0.38	38	Estensione di Nautilus per Font Manager

Tabella 7.5: Elenco di ambienti per i tipi di carattere degli di nota e pacchetti correlati

Suggerimento
Alcuni pacchetti di tipi di carattere, come `fonts-noto*` installano troppi tipi di carattere. Si può anche voler mantenere installati alcuni pacchetti di caratteri ma disabilitarli per le situazioni normali d’uso. Per alcuni punti del codice Unicode sono necessari glifi multipli a causa della unificazione Han e glifi non desiderati possono essere scelti dalla libreria Fontconfig non configurata. Uno dei casi più noiosi sono ”U+3001 IDEOGRAPHIC COMMA” e ”U+3002 IDEOGRAPHIC FULL STOP” nei paesi CJK. Si può facilmente evitare questa situazione problematica configurando la disponibilità dei tipi di carattere usando la GUI per la gestione dei tipi di carattere (font-manager).

È possibile anche elencare lo stato di configurazione dei tipi di carattere dalla riga di comando.

- ”`fc-match(1)`” per le impostazioni predefinite per i tipi di carattere di fontconfig
- ”`fc-list(1)`” per i tipi di carattere disponibili per fontconfig

Si può impostare lo stato di configurazione di un tipo di carattere dall’editor di testo, ma non è una cosa banale. Vedere fonts.conf(5).

7.7 Sandbox

Molte applicazioni, soprattutto GUI, in Linux sono disponibili solamente in forma binaria da fonti non-Debian.

- AppImage -- App Linux che funzionano ovunque
- FLATHUB -- App per Linux, proprio qui
- snapcraft -- Negozio di app per Linux



avvertimento
I binari da questi siti possono includere pacchetti software proprietari non liberi.

Esiste una ragione d'essere per queste distribuzioni in formato binario per gli aficionados del Software Libero che usano Debian, dato che queste possono fornire un insieme pulito di librerie usato per ciascuna applicazione dal corrispondente sviluppatore a monte, indipendentemente da quelle fornite da Debian.

Il rischio intrinseco dell'esecuzione di binari esterni può essere ridotto utilizzando un ambiente sandbox che sfrutta le funzionalità di sicurezza moderne di Linux (vedere Sezione 4.7.5.

- Per i binari da AppImage e alcuni siti a monte, eseguirli in firejail con configurazione manuale.
- Per i binari da FLATHUB, eseguirli in Flatpak . (Non è necessaria una configurazione manuale).
- Per i binari da snapcraft, eseguirli in Snap . (Nessuna configurazione manuale richiesta; compatibile con programmi demone.)

Il pacchetto `xdg-desktop-portal` fornisce un'API standardizzata per le funzionalità comuni del desktop. Vedere xdg-desktop-portal (flatpak) e xdg-desktop-portal (snap).

pacchetto	popcon	dimensione	descrizione
`flatpak`	V:104, I:115	9293	Flatpak, infrastruttura di messa in produzione di applicazioni per app per desktop
`gnome-software-plugin-flatpak`	V:29, I:41	291	Supporto Flatpak per il Software GNOME
`snapd`	V:58, I:62	62738	Demone e strumentazione che abilitano i pacchetti snap
`gnome-software-plugin-snap`	V:1.4, I:2.2	144	Supporto Snap per il Software GNOME
`xdg-desktop-portal`	V:373, I:439	2291	Portale di integrazione nel desktop per Flatpak e Snap
`xdg-desktop-portal-gtk`	V:337, I:438	715	backend xdg-desktop-portal per GTK (GNOME)
`xdg-desktop-portal-kde`	V:81, I:112	3333	backend xdg-desktop-portal per Qt (KDE)
`xdg-desktop-portal-wlr`	V:2.4, I:6.6	159	backend xdg-desktop-portal per wlroots (Wayland)
`firejail`	V:1.2, I:4.4	1827	programma per sandbox con sicurezza SUID firejail per l'uso con AppImage

Tabella 7.6: Elenco di ambienti sandbox degni di nota e pacchetti correlati

Questa tecnologia per ambiente sandbox è molto simile a come le app nei SO degli smartphone vengono eseguite con accesso controllato alle risorse.

Anche alcune grandi applicazioni GUI come browser web in Debian usano internamente la tecnologia degli ambienti sandbox per renderle più sicure.

pacchetto	popcon	dimensione	protocolli	descrizione
gnome-remote-desktop	V:14, I:241	2743	RDP	server GNOME Remote Desktop
xrdp	V:28, I:31	4757	RDP	xrdp, server RDP (Remote Desktop Protocol)
x11vnc	V:9, I:46	1863	RFB (VNC)	x11vnc, server Remote Framebuffer Protocol (VNC)
tigervnc-standalone-server	V:5, I:15	3124	RFB (VNC)	TigerVNC, server Remote Framebuffer Protocol (VNC)
gnome-connections	V:6, I:149	1726	RDP, RFB (VNC)	client per desktop remoto GNOME
vinagre	V:1, I:21	4249	RDP, RFB (VNC), SPICE, SSH	Vinagre: client per desktop remoto di GNOME
remmina	V:16, I:59	983	RDP, RFB (VNC), SPICE, SSH, ...	Remmina: client GTK per desktop remoto
krdc	V:2, I:15	4144	RDP, RFB (VNC)	KRDC: client KDE per desktop remoto
virt-viewer	V:4, I:37	1278	RFB (VNC), SPICE	client con display GUI di Virtual Machine Manager del sistema operativo ospite

Tabella 7.7: Elenco di server per l'accesso remoto degni di nota

7.8 Desktop remoto

7.9 Connessione a server X

Ci sono diversi modi per connettersi da un'applicazione su un host remoto al server X, incluso `xwayland` sull'host locale.

pacchetto	popcon	dimensione	comando	descrizione
openssh-server	V:774, I:820	3615	`sshd` con l'opzione `X11-forwarding`	server SSH (sicuro)
openssh-client	V:682, I:997	3611	`ssh -X`	client SSH (sicuro)
xauth	V:192, I:974	81	`xauth`	utilità per file X authority
x11-xserver-utils	V:373, I:533	559	`xhost`	controllo degli accessi al server per X

Tabella 7.8: Elenco di metodi di connessione al server X

7.9.1 Connessione locale al server X

L'accesso al server X locale da parte delle applicazioni locali che usano il protocollo di base X può essere connesso localmente attraverso un socket locale di dominio UNIX. Ciò può essere autorizzato dal file di autorità che contiene i cookie di accesso. Il file authority è identificato dalla variabile d'ambiente "`$XAUTHORITY`" e il display X è identificato

dalla variabile d'ambiente "`$DISPLAY`". Dato che queste vengono normalmente impostate automaticamente, non è necessaria alcuna azione speciale, ad es. "`gitk`" come segue.

```
username $ gitk
```

Nota
Per `xwayland`, `XAUTHORITY` contiene un valore come "`/run/user/1000/.mutter-Xwaylandauth.YVSU30`".

7.9.2 Connessione remota a server X

L'accesso al display del server X locale da parte di applicazioni remote che usano il protocollo principale X è supportato usando la funzionalità di inoltro di X11.

- Aprire un `gnome-terminal` sull'host locale.
- Eseguire ssh(1) con l'opzione `-X` per stabilire una connessione con il sito remoto nel modo seguente.

```
localname @ localhost $ ssh -q -X loginname@remotehost.domain
Password:
```

- Eseguire un comando applicativo X, ad esempio "`gitk`", sul sito remoto nel modo seguente.

```
loginname @ remotehost $ gitk
```

Questo metodo può mostrare l'output da un client X remoto come se fosse connesso localmente attraverso un socket UNIX locale.

Vedere Sezione 6.3 per SSH/SSHD.



avvertimento
La connessione TCP/IP remota al server X è disabilitata in modo predefinito nel sistema Debian per ragioni di sicurezza. Non abilitarla impostando semplicemente "`xhost +`", né abilitando connessioni XDMCP, se lo si può evitare.

7.9.3 Connessione chroot a server X

L'accesso al server X da parte delle applicazioni che usano il protocollo centrale di X e vengono eseguite sullo stesso host ma in un ambiente come una chroot dove il file di autorità non è accessibile può essere autorizzato in maniera sicura con `xhost` usando l'accesso basato su utente, es. "`gitk`" come segue:

```
username $ xhost + si:localuser:root ; sudo chroot /path/to
# cd /src
# gitk
# exit
username $ xhost -
```

7.10 Appunti

Per ritagliare testo negli appunti, vedere Sezione 1.4.4.

Per tagliare grafica negli appunti vedere Sezione 11.6.

Anche alcuni comandi CLI possono manipolare gli appunti a carattere (PRIMARY e CLIPBOARD).

pacchetto	popcon	dimensione del pacchetto	target	descrizione
`xsel`	V:8, I:42	55	per X	interfaccia a riga di comando per la selezione in X (appunti)
`xclip`	V:18, I:78	62	per X	interfaccia a riga di comando per la selezione in X (appunti)
`wl-clipboard`	V:10, I:27	174	Wayland	`wl-copy` `wl-paste`: interfaccia a riga di comando per gli appunti di Wayland
`gpm`	V:8.3, I:9.0	524	Console Linux	un demone che cattura eventi del mouse sulla console Linux

Tabella 7.9: Elenco di programmi correlati con la manipolazione degli appunti a caratteri

Capitolo 8

I18N e L10N

Il supporto per le lingue native o M17N (Multilingualization) per un software applicativo è ottenuto in 2 passi.

- L'internazionalizzazione (I18N): per rendere un software capace di gestire potenzialmente localizzazioni multiple.
- Localizzazione (L10N): per fare gestire dal software una localizzazione specifica.

Suggerimento
Ci sono 17, 18 o 10 lettere tra le lettere "m" e "n", "i" e "n" o "l" e "n" in, rispettivamente, "multilingualization", "internazionalization" e "localization" che sono i termini inglesi corrispondenti a M17N, I18N e L10N. Per i dettagli vedere Internationalizzazione e localizzazione.

8.1 La localizzazione

Il comportamento dei programmi che supportano l'internazionalizzazione è configurato dalla variabile d'ambiente "`$LANG`" per supportare la localizzazione. L'effettivo supporto delle funzionalità dipendenti dalla localizzazione da parte della libreria `libc` richiede l'installazione dei pacchetti `locales` o `locales-all`. Il pacchetto `locales` deve essere inizializzato correttamente.

Se non è installato né il pacchetto `locales` né `locales-all`, il supporto delle funzionalità di localizzazione è perso e il sistema usa i messaggi in inglese US e gestisce i dati come **ASCII**. Questo comportamento è uguale a quando "`$LANG`" è impostato a "`LANG=`", "`LANG=C`" o "`LANG=POSIX`".

Il software moderno, come GNOME e KDE ha il supporto per più lingue. È internazionalizzato rendendolo capace di gestire dati UTF-8 e localizzato fornendo i messaggi tradotti attraverso l'infrastruttura gettext(1). I messaggi tradotti possono essere forniti in pacchetti separati di localizzazione.

L'attuale sistema GUI del desktop Debian normalmente imposta la localizzazione nell'ambiente GUI come "`LANG=xx_YY.UT`
Qui "`xx`" è il codice ISO 639 di lingua e "YY" è il codice ISO 3166 di paese. Questi valori sono impostati dal dialogo GUI di configurazione del desktop e cambiano il comportamento del programma. Vedere Sezione 1.5.2.

8.1.1 Logica alla base dell'uso della localizzazione UTF-8

La più semplice rappresentazione di dati testuali è **ASCII** che è sufficiente per l'inglese e usa meno di 127 caratteri (rappresentabili con 7 bit).

Anche un testo in semplice inglese può contenere caratteri non ASCII; le virgolette singole ricurve destra e sinistra per esempio non sono disponibili in ASCII.

```
b''“b''double quoted textb''”b'' is not "double quoted ASCII"
b''‘b''single quoted textb''’b'' is not 'single quoted ASCII'
```

Per poter gestire più caratteri, molti insiemi di caratteri e sistemi di codifica sono stati usati per supportare molte lingue (vedere Tabella 11.2).

L'insieme di caratteri Unicode può rappresentare praticamente tutti i caratteri conosciuti con un intervallo di codici a 21 bit (cioè da 0 a10FFFF in notazione esadecimale).

Il sistema di codifica UTF-8 fa rientrare i codici Unicode in un flusso di dati ragionevole a 8 bit per la maggior parte compatibile con il sistema di elaborazione dei dati ASCII. Questo fa di **UTF-8** la scelta moderna preferita. **UTF** sta per Unicode Transformation Format (formato di trasformazione di Unicode). Quando i dati in testo semplice ASCII sono convertiti in dati UTF-8, questi hanno esattamente lo stesso contenuto e dimensione di quelli ASCII originali. Perciò non si perde nulla utilizzando la localizzazione UTF-8.

Nella localizzazione UTF-8, se si usa un programma applicativo compatibile, si possono visualizzare e modificare dati di testo in qualsiasi lingua straniera, purché siano installati e abilitati i tipi di carattere e i metodi di input richiesti. Per esempio, nella localizzazione "`LANG=it_IT.UTF-8`", gedit(1) (editor di testo per il desktop GNOME) può visualizzare e modificare dati testuali in caratteri cinesi continuando a presentare i menu in italiano.

Suggerimento
Sia la nuova localizzazione standard "`en_US.UTF-8`", sia la vecchia localizzazione standard "`C`"/"`POSIX`" usano i messaggi in inglese americano standard, ma hanno sottili differenze negli ordinamenti, ecc. Se si desidera gestire in maniera corretta non solo caratteri ASCII, ma anche tutti i caratteri codificati UTF-8 pur mantenendo il vecchio comportamento locale "`C`", usare in Debian la localizzazione non standard "`C.UTF-8`".

Nota
Alcuni programmi usano più memoria dopo l'inclusione del supporto per l'internazionalizzazione. Questo avviene perché il loro codice è programmato per usare internamente UTF-32(UCS4) per supportare Unicode al fine di ottimizzare la velocità e consumano 4 byte per ogni dato di carattere ASCII, indipendentemente dalla localizzazione selezionata. Ancora una volta usando la localizzazione UTF-8 non si perde nulla.

8.1.2 La (ri)configurazione della localizzazione

Per far sì che un sistema abbia accesso ad una localizzazione particolare, è necessario che i dati della localizzazione siano stati compilati a partire dal database della localizzazione.

Il pacchetto `locales` **non** viene fornito con dati di localizzazione pre-compilati. È necessario configurarlo.

```
# dpkg-reconfigure locales
```

Questo processo comprende 2 passi.

1. Selezionare tutti i dati di localizzazione richiesti per la compilazione in formato binario. (Assicurarsi di includere almeno una localizzazione UTF-8.)
2. Impostare il valore della localizzazione predefinita a livello di tutto il sistema creando "`/etc/default/locale`", per l'uso da parte di PAM (vedere Sezione 4.5).

Il valore della localizzazione predefinita a livello di sistema impostato in "`/etc/default/locale`" può essere scavalcato dalla configurazione con GUI delle applicazioni GUI.

Nota
I sistemi di codifica tradizionali possono essere identificati da "`/usr/share/i18n/SUPPORTED`". Perciò "`LANG=en_US`" è "`LANG=en_US.ISO-8859-1`".

Il pacchetto `locales-all` viene fornito con tutti i dati di localizzazione pre-compilati. Dato che non crea "`/etc/default/lo`" può essere sempre necessario installare anche il pacchetto `locales`.

Suggerimento
Il pacchetto `locales` di alcune distribuzioni derivate da Debian viene fornito con dati di localizzazione precompilati per tutte le localizzazioni. Per emulare un tale ambiente di sistema in Debian è necessario installare entrambi i pacchetti `locales` e `locales-all`.

8.1.3 Codifica per i nomi di file

Per lo scambia di dati interpiattaforma (vedere Sezione 10.1.7), può essere necessario montare alcuni file system con codifiche particolari. Per esempio, mount(8), se usato senza opzioni, assume che venga usata la codifica CP437 per il file system vfat. È necessario fornire esplicitamente opzioni di montaggio per usare nomi di file UTF-8 o CP932.

Nota
Quando una unità flash USB inseribile a caldo viene automaticamente montata in un ambiente desktop moderno come GNOME, si può fornire una opzione di montaggio di questo tipo cliccando con il tasto destro sull'icona nel desktop, cliccando sulla scheda "Drive", cliccare per espandere "Impostazioni" ed inserire "utf8" in "Opzioni di mount:". La prossima volta che questa chiavetta di memoria verrà montata, sarà abilitato il montaggio con UTF-8.

Nota
Se si sta facendo l'aggiornamento di un sistema o spostando dischi da un sistema non UTF-8, i nomi di file con caratteri non ASCII potranno essere codificati con codifiche usate una volta e ora deprecate, come ISO-8859-1 o eucJP. Cercare aiuto sugli strumenti di conversione dei testi per convrtirli in UTF-8. Vedere Sezione 11.1.

Samba usa in modo predefinito Unicode per i client più moderni (Windows NT, 200x, XP), ma usa CP850 per client più vecchi (DOS e Windows 9x/Me). Questo comportamento predefinito per i client più vecchi può essere modificato usando "`dos charset`" nel file "`/etc/samba/smb.conf`", per esempio usando "CP932" per il giapponese.

8.1.4 Messaggi localizzati e documentazione tradotta

Esistono le traduzioni di molti dei messaggi di testo e dei documenti che sono mostrati nel sistema Debian, come messaggi di errore, output standard dei programmi, menu e pagine di manuale. L'insieme di strumenti GNU gettext(1) è usato come strumento di backend per la maggior parte delle attività di traduzione.

aptitude(8) fornisce in "Task" → "Localizzazione" un ampio elenco di utili pacchetti binari che aggiungono alle applicazioni messaggi localizzati e che forniscono documentazione nella versione tradotta.

Per esempio, si possono ottenere i messaggi localizzati per le pagine man installando il pacchetto `manpages-`*`LINGUA`*. Per leggere le pagine man di *nomeprogramma* in italiano contenute in "`/usr/share/man/it/`", eseguire il comando seguente.

```
LANG=it_IT.UTF-8 man programname
```

GNU gettext può gestire liste di priorità delle lingue di traduzione con la variabile d'ambiente $LANGUAGE. Per esempio:

```
$ export LANGUAGE="pt:pt_BR:es:it:fr"
```

Per ulteriori informazioni vedere `info gettext` e leggere la sezione "The LANGUAGE variable".

8.1.5 Effetti della localizzazione

Il criterio di ordinamento dei caratteri con sort(1) e ls(1) è influenzato dalla localizzazione. Esportando `LANG=en_US.UTF-8` l'ordinamento avviene nell'ordine `A->a->B->b...->Z->z`, mentre se si esporta `LANG=C.UTF-8` l'ordinamento avviene come in ASCII binario `A->B->...->Z->a->b...`.

Il formato della data di ls(1) è influenzato dalla localizzazione (vedere Sezione 9.3.4).

Il formato della data di date(1) è influenzato dalla localizzazione. Ad esempio:

```
 $ unset LC_ALL
 $ LANG=en_US.UTF-8 date
Thu Dec 24 08:30:00 PM JST 2023
 $ LANG=en_GB.UTF-8 date
Thu 24 Dec 20:30:10 JST 2023
 $ LANG=es_ES.UTF-8 date
jue 24 dic 2023 20:30:20 JST
 $ LC_TIME=en_DK.UTF-8 date
2023-12-24T20:30:30 JST
```

I caratteri di punteggiatura usati per i numeri sono diversi nelle varie localizzazioni. Per esempio, nella localizzazione inglese mille virgola uno è rappresentato come "`1,000.1`", mentre nella localizzazione in italiano è mostrato come "`1.000,1`". Si può vedere questa differenza nei programmi per fogli di calcolo.

Ogni caratteristica specifica della variabile d'ambiente "`$LANG`" può essere scavalcata impostando le variabili "`$LC_*`". Queste variabili d'ambiente a loro volta possono essere scavalcate impostando la variabile "`$LC_ALL`". Vedere la pagina di manuale locale(7) per i dettagli. A meno di non avere forti motivi per creare configurazioni complesse, stare lontani da esse e usare solo la variabile "`$LANG`" impostata ad una delle localizzazioni UTF-8.

8.2 L'input da tastiera

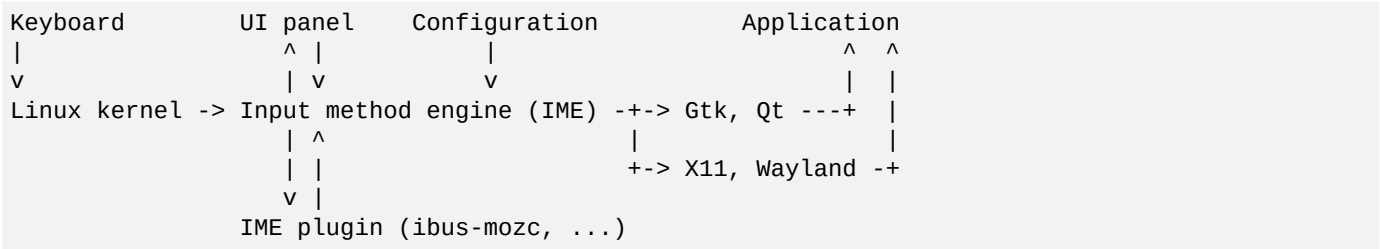
La tastiera di sistema può essere configurata a diversi livelli del sistema

- Kernel Linux: keyboard(5)
- Server X: setxkbmap(1), xkeyboard-config(5), variabile d'ambiente `XMODIFIERS`
- Ambiente desktop grafico: Infrastruttura per metodo di input (IM): `ibus`, `fcitx5`
- Applicazioni: variabili d'ambiente per impostare la fonte di input: `GTK_IM_MODULE`, `QT_IM_MODULE`, `QT_IM_MODULES`, ...

Una infrastruttura per metodo di input è costituita da:

- Motore di metodo di input (IME): l'effettivo metodo di input
- Configurazione: gestisce la configurazione per IBus e altri servizi come i plugin per IME
- Pannello: interfaccia utente come barra della lingua e tabella per selezione lingue candidate

L'input multilingua verso l'applicazione è elaborato grossomodo così:



8.2.1 L'input da tastiera per la console Linux e X Window

Il sistema Debian può essere configurato per funzionare con molte disposizioni di tastiera internazionali usando il pacchetto `keyboard-configuration`.

```
# dpkg-reconfigure keyboard-configuration
```

Per la console Linux e il sistema X Window, questo aggiorna i parametri di configurazione in "`/etc/default/keyboard`". Molti caratteri non ASCII, inclusi quelli accentati usati da molte lingue europee, possono essere resi disponibili con il "tasto morto" (dead key), il tasto AltGr e il tasto di composizione (compose).

Nota
Se ibus è attivo, la configurazione classica della tastiera X tramite `setxkbmap` può essere scavalcata da `ibus` anche neli ambienti desktop classici basati su X. Si può disabilitare un `ibus` installato usando `im-config` per impostare il metodo di input a "None" (Nessuno). Per ulteriori informazioni, vedere le inforzioni del Debian Wiki sulla tastiera.

8.2.2 L'input da tastiera per Wayland

A differenza del protocollo X Window, il protocollo centrale di Wayland non supporta nemmeno l'inserimento di caratteri accentati. I compositori Wayland di comune utilizzo, come Mutter per GNOME o KWin per KDE, implementano protocolli di estensione come il `text-input-unstable-v3` per l'input di testo (vedere "protocolli Wayland attuali e stato del loro supporto").

Il protocollo `text-input-unstable-v3` funziona bene con metodi di input per Wayland (vedere il "progetto per metodo di input per Wayland post-mortem").

- La maggior parte delle applicazioni con GUI sono create con librerie GUI come GTK o Qt che supportano `text-input-uns`
- I motori per metodo di input (IME) di uso comune, come IBus o Fcitx (version 5), possono funzionare con `text-input-uns`
- Gli IME supportano l'input di testo per molte lingue tramite plugin.
- Gli IME recenti integrano funzionalità "X Keyboard Extension (XKB)" come `setxkbmap`, in passato fornite da X Window per supportare l'input di caratteri accentati per le lingue europee per Wayland.

8.2.3 Il supporto per metodo di input con IBus

Per GNOME, "`ibus`" è l'IME predefinito che viene automaticamente installato attraverso le dipendenze dei pacchetti.

- La maggior parte delle funzionalità di input multilingua con la tastiera può essere configurata con le "Impostazioni di GNOME" o "Personalizzazioni di GNOME" (Gnome Tweaks).
- Per alcune funzionalità di input multilingua con la tastiera può essere necessario configurarle con il comando ibus-setup(1).
- L'input di emoji da tastiera è disponibile digitando "SUPER-." (Premere contemporaneamente il tasto Windows e il tasto del punto) e quindi una parola chiave per ogni emoji e poi il tasto Spazio.

Ecco l'elenco dei pacchetti di IBus e dei suoi plugin.

8.2.4 Il supporto per metodo di input con Fcitx

L'infrastruttura per metodo di input Fcitx (versione 5) è popolare tra gli utenti cinesi ed è compatibile con "`ibus`".

Ecco l'elenco dei pacchetti di "`fcitx5`" e dei suoi plugin.

pacchetto	popcon	dimensione	localizzazioni supportate
ibus	V:212, I:253	1884	infrastruttura per metodo di input che usa dbus
ibus-mozc	V:2.1, I:3.8	1054	giapponese
ibus-anthy	V:0.5, I:1.1	8740	giapponese
ibus-skk	V:0.03, I:0.14	242	giapponese
ibus-kkc	V:0.04, I:0.19	206	giapponese
ibus-libpinyin	V:1.1, I:5.1	123	cinese (per zh_CN)
ibus-chewing	V:0.16, I:0.84	298	cinese (per zh_TW)
ibus-libzhuyin	I:0.10	41021	cinese (per zh_TW)
ibus-rime	V:0.29, I:0.51	76	cinese (per zh_CN/zh_TW)
ibus-cangjie	V:0.02, I:0.10	235	cinese (per zh_HK)
ibus-hangul	V:0.3, I:1.9	261	coreano
ibus-libthai	V:0.00, I:0.05	84	thailandese
ibus-table-thai	I:0.04	59	thailandese
ibus-unikey	V:0.22, I:0.40	286	vietnamita
keyman	V:0.01, I:0.10	507	multilingue: plugin Keyman per più di 2000 lingue
ibus-table	V:0.04, I:0.92	2364	plugin per tabelle per IBus
ibus-m17n	V:0.1, I:1.8	373	multilingue: indico, arabico e altri

Tabella 8.1: Elenco di pacchetti di IBus e dei suoi plugin

pacchetto	popcon	dimensione	localizzazioni supportate
fcitx5	V:8, I:13	712	infrastruttura per metodo di input compatibile con "`ibus`"
fcitx5-mozc	V:1.1, I:1.8	1328	giapponese
fcitx5-anthy	V:0.07, I:0.25	853	giapponese
fcitx5-skk	V:0.05, I:0.15	370	giapponese
fcitx5-kkc	V:0.01, I:0.06	418	giapponese
fcitx5-chinese-addons	I:9.2	17	cinese (metapacchetto per zh_*)
fcitx5-pinyin	V:4.0, I:9.7	1000	cinese (per zh_CN)
fcitx5-chewing	V:0.29, I:0.98	228	cinese (per zh_TW)
fcitx5-zhuyin	I:0.08	108535	cinese (per zh_TW)
fcitx5-rime	V:0.46, I:0.93	393	cinese (per zh_CN/zh_TW)
fcitx5-table-cangjie-large	I:0.13	1292	cinese (per zh_HK)
fcitx5-hangul	V:0.12, I:0.27	239	coreano
fcitx5-libthai	V:0.00, I:0.04	119	thailandese
fcitx5-table-thai	I:0.09	34	thailandese
fcitx5-unikey	V:0.13, I:0.23	624	vietnamita
fcitx5-m17n	V:0.16, I:0.61	171	multilingue: indico, arabico e altri
fcitx5-table	V:0.3, I:9.6	428	plugin per tabelle per fcitx5
fcitx5-keyman	V:0.04, I:0.05	241	multilingue: plugin Keyman per più di 2000 lingue

Tabella 8.2: Elenco di pacchetti di Fcitx5 e dei suoi plugin

8.2.5 Un esempio per il giapponese

Trovo che il metodo di input per il giapponese avviato dall'ambiente inglese ("`en_US.UTF-8`") sia molto utile. Ecco come farlo con IBus:

1. Installare il pacchetto con gli strumenti di input per il giapponese `ibus-mozc` (o `ibus-anthy`).
2. • Generico: eseguire ibus-setup(1) → selezionare "Metodo di Input" → cliccare su "`Aggiungi`" → "Giapponese" → "Mozc (o Anthy)" → cliccare "Aggiungi"
   • GNOME: selezionare "Impostazioni" → "Tastiera" → "Sorgenti input" → cliccare su"+" in "Sorgenti input" → "Giapponese" → "Mozc (o Anthy)" → cliccare "Aggiungi"
3. Si possono scegliere più sorgenti di input.
4. Rieseguire il login nell'account utente
5. Impostare ciascuna fonte di input cliccando con il pulsante destro sull'icona della barra degli strumenti GUI.
6. Passare da una all'altra delle fonti di input installate con SUPER-SPAZIO. (SUPER è solitamente il tasto Windows.)

Suggerimento
Se si desidera avere accesso ad un ambiente tastiera con solo alfabeto con la tastiera fisica giapponese in cui Maiusc-2 ha stampato " (virgolette doppie), selezionare "Japanese" nella procedura descritta sopra. Si può inserire giapponese usando "Japanese mozc (o anthy)" con una tastiera fisica "US" in cui Maiusc-2 ha stampato `@` (la chiocciolina).

- Per Wayland:
  - Il pacchetto `im-config` non fa nulla e può essere rimosso senza problemi.
  - Probabilmente non è necessario impostare variabili d'ambiente eccetto che per compatibilità all'indietro, ecc.
  - Se è necessario impostare variabili d'ambiente, creare un file del tipo "`~/.config/environment.d/50-input-meth` per impostarle.
- Per X Window:
  - Installare il pacchetto `im-config`.
  - La voce di menu GUI per im-config(8) è "Input method".
  - In alternativa eseguire "`im-config`" dalla shell dell'utente.
  - im-config(8) si comporta in modo diverso a seconda se è eseguito da root o no.
  - im-config(8) abilita il miglior metodo di input sul sistema come impostazione predefinita senza alcuna azione dell'utente.

8.3 L'output sul display

La console Linux può mostrare solamente un numero limitato di caratteri. (È necessario usare speciali programmi per terminale come jfbterm(1) per mostrare lingue non Europee nella console non-GUI.)

L'ambiente GUI (Capitolo 7) può mostrare qualsiasi carattere in UTF-8 purché siano installati e abilitati i tipi di carattere richiesti. (La codifica dei dati dei tipi di carattere originali viene gestita in modo trasparente per l'utente.)

8.3.1 Configurazione del terminale

Il sistema Debian può essere configurato per funzionare con molte disposizioni per la console udando il pacchetto `console-setup`.

```
# dpkg-reconfigure console-setup
```

Per la console Linux e il sistema X Window, questo aggiorna i parametri di configurazione in "`/etc/default/console-set` per visualizzare molti caratteri non ASCII, inclusi quelli accentati usati da molte lingue europee che vengono resi disponibili.

Ci sono diversi componenti per configurare le funzionalità della console a caratteri e il sistema ncurses(3).

- Il file "`/etc/terminfo/*/*`" (terminfo(5))
- La variabile d'ambiente "`$TERM`" (term(7))
- setterm(1), stty(1), tic(1) e toe(1)

Se, con un `xterm` non Debian, la voce `terminfo` per `xterm` non funziona, cambiare il tipo di terminale "`$TERM`" da "`xterm`" ad una delle versioni con funzionalità limitate come "`xterm-r6`" quando si fa il login ad un sistema Debian da remoto. Per ulteriori informazioni vedere "`/usr/share/doc/libncurses5/FAQ`" . "`dumb`" è il minimo denominatore comune per "`$TERM`".

8.3.2 Caratteri dell'Asia dell'est con larghezza ambigua

Nella localizzazione dell'Asia dell'est i caratteri di disegno di riquadri, i caratteri greci e cirillici possono essere visualizzati più larghi della larghezza desiderata e causare un output su terminale non allineato (vedere Unicode Standard Annex #11, 4.2 Ambiguous Characters).

Questo problema può essere aggirato:

- `gnome-terminal`: Modifica → Preferenze → Profili → *Nome del profilo* → Compatibilità → Caratteri a larghezza ambigua → Stretto
- `ncurses`: impostare l'ambiente `export NCURSES_NO_UTF8_ACS=0`.

Capitolo 9

Suggerimenti per il sistema

In questa sezione vengono descritti suggerimenti base per configurare e gestire il sistema, per lo più dalla console.

9.1 Suggerimenti per la console

Esistono alcuni programmi di utilità per aiutare nelle attività in console.

pacchetto	popcon	dimensione	descrizione
mc	V:40, I:176	1590	Vedere Sezione 1.3
bsdutils	V:421, I:999	338	Il comando script(1) per creare trascrizioni della sessione in terminale
screen	V:50, I:191	1007	multiplexer per terminale con emulazione di terminale VT100/ANSI
tmux	V:106, I:167	1423	alternativa per multiplexer per terminale (usare invece ”Control-B”)
ripgrep	V:17, I:48	5284	ricerca veloce e ricorsiva di stringhe nell’albero del codice sorgente con filtraggio automatico
fzf	V:10, I:36	5677	strumento per ricerca di testo fuzzy
fzy	V:0.07, I:0.34	59	strumento per ricerca di testo fuzzy
rlwrap	V:1, I:11	328	wrapper a riga di comando per la funzionalità readline
ledit	V:0.5, I:7.3	375	wrapper a riga di comando per la funzionalità readline
rlfe	V:0.02, I:0.48	45	wrapper a riga di comando per la funzionalità readline

Tabella 9.1: Elenco di programmi che supportano le attività in console

9.1.1 Registrare le attività della shell in modo pulito

Il semplice uso di script(1) (vedere Sezione 1.4.9) per registrare l’attività della shell produce un file con caratteri di controllo. Ciò può essere evitato usando col(1) nel modo seguente.

```
$ script
Script started, file is typescript
```

Fare tutto quello che si vuole … e poi premere `Ctrl-D` per uscire da `script`.

```
$ col -bx < typescript > cleanedfile
$ vim cleanedfile
```

Ci sono metodi alternativi per registrare le attività nella shell:

- Usare `tee` (utilizzabile durante il processo di avvio nell'initramfs):

```
$ sh -i 2>&1 | tee typescript
```

- Usare `gnome-terminal` con il buffer delle righe esteso per scorrere all'indietro.
- Usare `screen` con "`^A H`" (vedere Sezione 9.1.2) per effettuare la registrazione della console.
- Usare `vim` con "`:terminal`" per entrare nella modalità terminale. Usare "`Ctrl-W N`" per uscire dalla modalità terminale nella modalità normale. Usare "`:w typescript`" per scrivere il buffer in un file.
- Usare `emacs` con "`M-x shell`", "`M-x eshell`" o "`M-x term`" per entrare nella console di registrazione. Usare "`C-x C-w`" per scrivere il buffer in un file.

9.1.2 Il programma screen

screen(1) non permette solamente il funzionamento di una finestra di terminale con processi multipli, ma permette anche ai **processi in shell remote di sopravvivere a connessioni interrotte**. Quello che segue è un tipico scenario di uso di screen(1).

1. Si fa il login in una macchina remota.
2. Si avvia `screen` in una console singola.
3. Si eseguono svariati programmi in finestre `screen` create con `^A c` ("Control-A" seguito da "c").
4. Ci si sposta tra le svariate finestre `screen` create con `^A n` ("Control-A" seguito da "n").
5. All'improvviso si ha la necessità di lasciare il terminale, ma non si vuole perdere il lavoro attivo e si vuole mantenere la connessione.
6. Si può **scollegare** la sessione `screen` in uno qualsiasi dei metodi seguenti.
   - Scollegare brutalmente la connessione di rete
   - Digitare `^A d` ("Control-A" seguito da "d") e fare manualmente il log out dalla connessione remota.
   - Digitare `^A DD` ("Control-A" seguito da "DD") per far si che `screen` si scolleghi e faccia il log out.
7. Ci si ricollega alla stessa macchina remota (anche da un terminale diverso).
8. Si avvia `screen` con "`screen -r`".
9. `screen` magicamente **ricollega** tutte le finestre `screen` precedente con tutti i programmi in esecuzione attivi.

Suggerimento
Con `screen` si può risparmiare sui costi di connessione per connessioni a tempo, come dial-up o conteggiate a pacchetti, perché si può lasciare un processo attivo mentre si è disconnessi e poi ricollegarvisi successivamente quando è possibile connettersi di nuovo.

In una sessione `screen` tutto l'input da tastiera viene inviato alla finestra attuale, tranne per le combinazioni di tasti per i comandi. Tutte le combinazioni di tasti per i comandi di `screen` vengono inserite digitando ^A ("Control-A") più un singolo tasto [più eventuali parametri]. Ecco alcune combinazioni di tasti importanti da ricordare.

Vedere screen(1) per i dettagli.

Vedere tmux(1) per le funzionalità del comando alternativo.

azione associata	significato
^A ?	mostra una schermata di aiuto (visualizza le associazioni di tasti)
^A c	crea una nuova finestra e si sposta in essa
^A n	va alla finestra successiva
^A p	va alla finestra precedente
^A 0	va alla finestra numero 0
^A 1	va alla finestra numero 1
^A w	mostra l'elenco delle finestre
^A a	invia Ctrl-A alla finestra attuale come input da tastiera
^A h	scrive una copia della schermata della finestra attuale in un file
^A H	inizia/termina la registrazione dell'attività della finestra attuale in un file
^A ^X	blocca il terminale (protetto da password)
^A d	scollega la sessione screen dal terminale
^A DD	scollega la sessione di screen e fa il log out

Tabella 9.2: Elenco di associazioni di tasti per screen

9.1.3 Navigare nelle directory

In Sezione 1.4.2 sono descritti 2 suggerimenti per permettere una navigazione veloce nelle directory: `$CDPATH` e `mc`.

Se si usa un programma per filtro di testo fuzzy, si può evitare di digitare il percorso esatto. Per `fzf`, includere quanto segue in `~/.bashrc`.

```
FZF_KEYBINDINGS_PATH=/usr/share/doc/fzf/examples/key-bindings.bash
if [ -f $FZF_KEYBINDINGS_PATH ]; then
  . $FZF_KEYBINDINGS_PATH
fi
```

Per esempio:

- Ci si può spostare in una sottodirectory molto in profondità con uno sforzo minimo. Prima si digita "`cd **`" e si preme il tasto `Tab`. Poi verrano mostrati i percorsi candidati. Digitando stringhe parziali del percorso, ad esempio `s/d/b foo`, si possono restringere i percorsi candidati. Si seleziona il percorso che deve usare `cd` con i tasti per spostare il cursore e il tasto Invio.
- Si può selezionare un comando dalla cronologia dei comandi in modo più efficiente con sforzo minimo. Si preme `Ctrl-R` al prompt dei comandi. Poi verranno mostrati comandi candidati. Digitando stringhe di comando parziali, es. `vim d`, si restringono i candidati. Si seleziona quello da utilizzare con i tasti per il cursore e il tasto Invio.

9.1.4 Wrapper per readline

Alcuni comandi come `/usr/bin/dash` che mancano della funzionalità di modifica della cronologia della riga di comando possono aggiungere tale funzionalità in modo trasparente venendo eseguito sotto `rlwrap` o un suo equivalente.

```
$ rlwrap dash -i
```

Questo fornisce una comoda piattaforma per testare dettagli per `dash` con un ambiente amichevole in stile `bash`.

9.1.5 Strumenti per la fusione di codice sorgente

Il comando rg(1) nel pacchetto `ripgrep` offre un'alternativa più veloce al comando grep(1) per scansionare l'albero del codice sorgente nelle situazioni tipiche. Sfrutta le moderne CPU multi-core e applica automaticamente filtri ragionevoli per saltare alcuni file.

9.2 Personalizzare vim

Dopo aver imparato le basi di vim(1) attraverso Sezione 1.4.8, leggere"Seven habits of effective text editing (2000)" di Bram Moolenaar per capire come `vim` dovrebbe essere usato.

9.2.1 Personalizzare vim con le funzionalità interne

Il comportamento di `vim` può essere modificato in modo significativo abilitando le sue funzionalità interne attraverso i comandi della modalità `Ex` come "`set ...`" per impostare le opzioni di vim.

Questo comandi per la modalità `Ex` possono essere inclusi nel file vimrc dell'utente, tradizionalmente "`~/.vimrc`" o quello adatto a git "`~/.vim/vimrc`". Ecco un esempio molto semplice 1:

```
""" Generic baseline Vim and Neovim configuration (~/.vimrc)
"""   - For NeoVim, use "nvim -u ~/.vimrc [filename]"
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
let mapleader = ' '             " :h mapleader
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
set nocompatible                " :h 'cp -- sensible (n)vim mode
syntax on                       " :h :syn-on
filetype plugin indent on       " :h :filetype-overview
set encoding=utf-8              " :h 'enc (default: latin1) -- sensible encoding
""" current vim option value can be verified by :set encoding?
set backspace=indent,eol,start  " :h 'bs (default: nobs) -- sensible BS
set statusline=%<%f%m%r%h%w%=%y[U+%04B]%2l/%2L=%P,%2c%V
set listchars=eol:¶,tab:b''▫b''\ ,extends:b''▫b'',precedes:b''▫b'',nbsp:b''▫b''
set viminfo=!,'100,<5000,s100,h " :h 'vi -- bigger copy buffer etc.
""" Pick "colorscheme" from blue darkblue default delek desert elflord evening
""" habamax industry koehler lunaperche morning murphy pablo peachpuff quiet ron
""" shine slate torte zellner
colorscheme industry
""" don't pick "colorscheme" as "default" which may kill SpellUnderline settings
set scrolloff=5                 " :h 'scr -- show 5 lines around cursor
set laststatus=2                " :h 'ls (default 1)  k
""" boolean options can be unset by prefixing "no"
set ignorecase                  " :h 'ic
set smartcase                   " :h 'scs
set autoindent                  " :h 'ai
set smartindent                 " :h 'si
set nowrap                      " :h 'wrap
"set list                        " :h 'list (default nolist)
set noerrorbells                " :h 'eb
set novisualbell                " :h 'vb
set t_vb=                       " :h 't_vb -- termcap visual bell
set spell                       " :h 'spell
set spelllang=en_us,cjk         " :h 'spl -- english spell, ignore CJK
set clipboard=unnamedplus       " :h 'cb -- cut/copy/paste with other app
set hidden                      " :h 'hid
set autowrite                   " :h 'aw
set timeoutlen=300              " :h 'tm
```

La mappa dei tasti di `vim` può essere modificata nel file vimrc dell'utente. Es.:



Attenzione
Non cercare di cambiare le associazioni dei tasti predefinite senza ottime ragioni.

1Esempi di personalizzazione più elaborati: "Vim Galore", "sensible.vim", ...

```
""" Popular mappings (imitating LazyVim etc.)
""" Window moves without using CTRL-W which is dangerous in INSERT mode
nnoremap <C-H> <C-W>h
nnoremap <C-J> <C-W>j
nnoremap <C-K> <C-W>k
silent! nnoremap <C-L> <C-W>l
""" Window resize
nnoremap <C-LEFT> <CMD>vertical resize -2<CR>
nnoremap <C-DOWN> <CMD>resize -2<CR>
nnoremap <C-UP> <CMD>resize +2<CR>
nnoremap <C-RIGHT> <CMD>vertical resize +2<CR>
""" Clear hlsearch with <ESC> (<C-L> is mapped as above)
nnoremap <ESC> <CMD>noh<CR><ESC>
inoremap <ESC> <CMD>noh<CR><ESC>
""" center after jump next
nnoremap n nzz
nnoremap N Nzz
""" fast "jk" to get out of INSERT mode (<ESC>)
inoremap  jk <CMD>noh<CR><ESC>
""" fast "<ESC><ESC>" to get out of TERM mode (CTRL-\ CTRL-N)
tnoremap <ESC><ESC> <C-\><C-N>
""" fast "jk" to get out of TERM mode (CTRL-\ CTRL-N)
tnoremap jk <C-\><C-N>
""" previous/next trouble/quickfix item
nnoremap [q <CMD>cprevious<CR>
nnoremap ]q <CMD>cnext<CR>
""" buffers
nnoremap <S-H> <CMD>bprevious<CR>
nnoremap <S-L> <CMD>bnext<CR>
nnoremap [b <CMD>bprevious<CR>
nnoremap ]b <CMD>bnext<CR>
""" Add undo break-points
inoremap  , ,<C-G>u
inoremap  . .<C-G>u
inoremap  ; ;<C-G>u
""" save file
inoremap <C-S> <CMD>w<CR><ESC>
xnoremap <C-S> <CMD>w<CR><ESC>
nnoremap <C-S> <CMD>w<CR><ESC>
snoremap <C-S> <CMD>w<CR><ESC>
""" better indenting
vnoremap < <gv
vnoremap > >gv
""" terminal (Somehow under Linux, <C-/> becomes <C-_> in Vim)
nnoremap <C-_> <CMD>terminal<CR>
"nnoremap <C-/> <CMD>terminal<CR>
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
if ! has('nvim')
""" Toggle paste mode with <SPACE>p for Vim (no need for Nvim)
set pastetoggle=<leader>p
""" nvim default mappings for Vim.  See :h default-mappings in nvim
""" copy to EOL (no delete) like D for d
noremap Y y$
""" sets a new undo point before deleting
inoremap <C-U> <C-G>u<C-U>
inoremap <C-W> <C-G>u<C-W>
""" <C-L> is re-purposed as above
""" execute the previous macro recorded with Q
nnoremap Q @@
""" repeat last substitute and *KEEP* flags
nnoremap & :&&<CR>
```

```
""" search visual selected string for visual mode
xnoremap * y/\V<C-R>"<CR>
xnoremap # y?\V<C-R>"<CR>
endif
```

Per far sì che le associazioni di tasti precedenti funzionino correttamente, il programma di terminale deve essere configurato per generare ”ASCII DEL” per il tasto `Backspace` e ”sequenza Escape” per il tasto `Canc`.

Altre configurazioni varie possono essere modificate nel file vimrc dell’utente. Es.:

```
""" Use faster 'rg' (ripgrep package) for :grep
if executable("rg")
  set grepprg=rg\ --vimgrep\ --smart-case
  set grepformat=%f:%l:%c:%m
endif
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" Retain last cursor position :h '"
augroup RetainLastCursorPosition
  autocmd!
  autocmd BufReadPost *
    \ if line("'\"") > 0 && line ("'\"") <= line("$") |
    \   exe "normal! g'\"" |
    \ endif
augroup END
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" Force to use underline for spell check results
augroup SpellUnderline
  autocmd!
  autocmd ColorScheme * highlight SpellBad term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellCap term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellLocal term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellRare term=Underline gui=Undercurl
augroup END
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" highlight tailing spaces except when typing as red (set after colorscheme)
highlight TailingWhitespaces ctermbg=red guibg=red
""" \s\+     1 or more whitespace character: <Space> and <Tab>
""" \%#\@<!  Matches with zero width if the cursor position does NOT match.
match TailingWhitespaces /\s\+\%#\@<!$/
```

9.2.2 Personalizzare vim con pacchetti esterni

Pacchetti per plugin esterni interessanti possono essere trovati in:

- Vim - the ubiquitous text editor -- Il sito originale a monte ufficiale di Vim e degli script vim
- VimAwsome -- Elenco di plugin per Vim
- vim-scripts -- pacchetto Debian: una raccolta di script per vim

I pacchetti di plugin nel pacchetto vim-scripts possono essere abilitati usando il file vimrc dell’utente. Es:

```
packadd! secure-modelines
packadd! winmanager
" IDE-like UI for files and buffers with <space>w
nnoremap <leader>w         :WMToggle<CR>
```

Il nuovo sistema di pacchetti nativo di Vim funziona bene con ”`git`” e ”`git submodule`”. Una configurazione d’esempio di questo tipo può essere trovata inmy git repository: dot-vim. Essenzialmente fa:

- Usando "`git`" e "`git submodule`", i pacchetti esterni più recenti, come "*nome*", sono messi in `~/.vim/pack/*/opt/nom` e simili.
- Aggiungendo la riga `:packadd! nome` al file vimrc dell'utente, questi pacchetti vengono messi nel `runtimepath`.
- Vim carica questi pacchetti in `runtimepath` durante la sua inizializzazione.
- Al termine della sua inizializzazione, i tag per i documenti installati sono aggiornati con "`helptags ALL`".

Per altro ancora, avviare `vim` con "`vim --startuptime vimstart.log`" per controllare l'effettiva sequenza di esecuzione e il tempo speso in ogni passo.

Può essere fonte di confusione vedere troppi modi[2] per gestire e caricare questi pacchetti esterni in `vim`. Controllare le informazioni originali è la soluzione migliore.

Combinazione di tasti	**informazioni**
`:help pacchetto`	spiegazione sul meccanismo dei pacchetti di vim
`:help runtimepath`	spiegazione del meccanismo `runtimepath`
`:version`	stati interni inclusi i candidati per il file vimrc
`:echo $VIM`	la variabile d'ambiente "`$VIM`" usata per localizzare il file vimrc
`:set runtimepath?`	elenco di directory in cui verranno cercati tutti i file di supporto a runtime
`:echo $VIMRUNTIME`	la variabile d'ambiente "`$VIMRUNTIME`" usata per localizzare vari file di supporto a runtime forniti dal sistema

Tabella 9.3: Informazioni sull'inizializzazione di `vim`

9.3 Registrazione e presentazione di dati

9.3.1 Demoni di registro

Molti programmi registrano le proprie attività in formato di file testuale nella directory "`/var/log/`".

logrotate(8) viene usato per semplificare l'amministrazione dei file di log in un sistema che genera molti file di log.

Molti nuovi programmi registrano le loro attività nel formato di file binario, usando il servizio Journal systemd-journald(8), nella directory "`/var/log/journal`".

Si può fare il registro di dati nel Journal systemd-journald(8) da uno script di shell usando il comando systemd-cat(1).

Vedere Sezione 3.5 e Sezione 3.4.

9.3.2 Analizzatori di registro

Quelli che seguono sono alcuni analizzatori di registro degni di nota ("`~Gsecurity::log-analyzer`" in aptitude(8)).

Nota
CRM114 fornisce un'infrastruttura basata su un linguaggio per scrivere filtri **fuzzy** con la libreria per espressioni regolari TRE. Il suo utilizzo più comune è come filtro per la posta spazzatura, ma può anche essere usato come analizzatore di registro.

[2] vim-pathogen era popolare.

pacchetto	popcon	dimensione	descrizione
fail2ban	V:101, I:111	2275	strumento per interdire IP che causano errori di autenticazione multipli
logwatch	V:9, I:10	2440	analizzatore di registro scritto in Perl con un bell'output
awstats	V:5.1, I:8.0	6934	analizzatore potente e ricco di funzionalità per registro di server web
analog	V:3, I:83	3739	analizzatore del registro di server web
sarg	V:0.65, I:0.71	863	generatore di rapporti sull'analisi di squid
pflogsumm	V:1.5, I:3.5	173	strumento per riassunti delle voci di registro di Postfix
fwlogwatch	V:0.10, I:0.16	487	analizzatore del registro del firewall
squidview	V:0.04, I:0.41	224	monitora e analizza i file access.log di squid
swatch	V:0.07, I:0.30	99	visualizzatore di file di registro con corrispondenze con espressioni regolari, evidenziazione ed eventi
crm114	V:0.06, I:0.32	1371	Controllable Regex Mutilator e filtro per spam (CRM114)
icmpinfo	V:0.01, I:0.35	42	interpreta i messaggi ICMP

Tabella 9.4: Elenco di analizzatori del registro di sistema

9.3.3 Visualizzazione personalizzata di dati di testo

Sebbene gli strumenti di paginazione, come more(1) e less(1) (vedere Sezione 1.4.5) e gli strumenti personalizzati per l'evidenziazione e la formattazione (vedere Sezione 11.1.8) possano mostrare il testo in un modo piacevole, gli editor generici (vedere Sezione 1.4.6) sono più versatili e personalizzabili.

Suggerimento
Per vim(1) e la sua modalità per paginatore, ossia view(1), "`:set hls`" abilita la ricerca con evidenziazione.

9.3.4 Visualizzazione personalizzata di date e orari

Il formato predefinito per la visualizzazione delle date e degli orari per il comando "`ls -l`" dipende dalla **localizzazione** (vedere Sezione 1.2.6 per il valore). Si fa dapprima riferimento alla variabile "`$LANG`" che può essere scavalcata dalle variabili d'ambiente "`$LC_TIME`" o "`$LC_ALL`".

Il formato effettivo di visualizzazione predefinito per ciascuna localizzazione dipende dalla versione della libreria C standard (il pacchetto `libc6`) usata. Differenti rilasci di Debian hanno cioè valori predefiniti diversi. Per i formati ISO, vedere ISO 8601.

Se si desidera veramente personalizzare questo formato di visualizzazione delle date e degli orari, oltre a ciò che è fatto con la **localizzazione**, si deve impostare il **valore dello stile degli orari** con l'opzione "`--time-style`" o con il valore di "`$TIME_STYLE`" (vedere ls(1), date(1), "`info coreutils 'ls invocation'`").

Suggerimento
Si può evitare di digitare lunghe opzioni nella riga di comando usando alias per i comandi (vedere Sezione 1.5.9):

```
alias ls='ls --time-style=+%d.%m.%y %H:%M'
```

valore dello stile per gli orari	localizzazione	visualizzazione di data e ora
iso	qualsiasi	01-19 00:15
long-iso	qualsiasi	2009-01-19 00:15
full-iso	qualsiasi	2009-01-19 00:15:16.000000000 +0900
locale	C	Jan 19 00:15
locale	en_US.UTF-8	Jan 19 00:15
locale	es_ES.UTF-8	ene 19 00:15
+%d.%m.%y %H:%M	qualsiasi	19.01.09 00:15
+%d.%b.%y %H:%M	C o en_US.UTF-8	19.Jan.09 00:15
+%d.%b.%y %H:%M	es_ES.UTF-8	19.ene.09 00:15

Tabella 9.5: Esempi di visualizzazione di date e orari per il comando "`ls -l`" con il **valore dello stile per gli orari**

9.3.5 Output colorato per la shell

L'output inviato a schermo dalla shell nella maggior parte dei terminali moderni può essere colorato usando codici di escape ANSI (vedere "`/usr/share/doc/xterm/ctlseqs.txt.gz`").

Per esempio, provare a fare quanto segue.

```
$ RED=$(printf "\x1b[31m")
$ NORMAL=$(printf "\x1b[0m")
$ REVERSE=$(printf "\x1b[7m")
$ echo "${RED}RED-TEXT${NORMAL} ${REVERSE}REVERSE-TEXT${NORMAL}"
```

9.3.6 Comandi colorati

I comandi colorati sono comodi per ispezionare il loro output in modo interattivo. Nel mio file "`~/.bashrc`" io includo quanto segue.

```
if [ "$TERM" != "dumb" ]; then
    eval "`dircolors -b`"
    alias ls='ls --color=always'
    alias ll='ls --color=always -l'
    alias la='ls --color=always -A'
    alias less='less -R'
    alias ls='ls --color=always'
    alias grep='grep --color=always'
    alias egrep='egrep --color=always'
    alias fgrep='fgrep --color=always'
    alias zgrep='zgrep --color=always'
else
    alias ll='ls -l'
    alias la='ls -A'
fi
```

Questo uso degli alias limita gli effetti colorati all'uso interattivo dei comandi. Ha il vantaggio, rispetto all'esportazione della variabile d'ambiente "`export GREP_OPTIONS='--color=auto'`", che i colori possono essere visti in programmi di paginazione come less(1). Se si desidera eliminare i colori quando si invia l'output con una pipe ad altri comandi, usare nell'esempio precedente per "`~/.bashrc`" "`--color=auto`".

Suggerimento
Questi alias di colorazione possono essere disattivati nell'ambiente interattivo invocando la shell con il comando "`TERM=dumb bash`".

9.3.7 Registrare le attività dell'editor per ripetizioni complesse

È possibile registrare le attività dell'editor per ripetere azioni complesse.

Per vim fare nel modo seguente.

- "qa": avviare la registrazione dei caratteri digitati in un registro chiamato "a".
- … attività nell'editor
- "q": terminare la registrazione dei caratteri digitati.
- "@a": eseguire il contenuto del registro "a".

Per Emacs fare nel modo seguente.

- "`C-x (`": iniziare la definizione di una macro da tastiera.
- … attività nell'editor
- "`C-x )`": terminare la definizione di una macro da tastiera.
- "`C-x e`": eseguire la definizione di una macro da tastiera.

9.3.8 Registrare l'immagine grafica di un'applicazione X

Ci sono alcuni modi per registrare l'immagine grafica di un'applicazione X, incluso il display di un `xterm`.

pacchetto	popcon	dimensione	Schermata
`gnome-screenshot`	V:12, I:94	1115	Wayland
`flameshot`	V:7, I:17	3542	Wayland
`gimp`	V:34, I:211	33315	Wayland + X
`x11-apps`	V:31, I:443	2461	per X
`imagemagick`	V:7, I:272	80	per X
`scrot`	V:4, I:48	144	per X

Tabella 9.6: Elenco di strumenti di manipolazione di immagini grafiche

9.3.9 Registrare i cambiamenti a file di configurazione

Esistono strumenti specializzati per registrare i cambiamenti nei file di configurazione con l'aiuto del sistema DVCS e per creare istantanee del sistema in Btrfs.

pacchetto	popcon	dimensione	descrizione
`etckeeper`	V:24, I:27	157	archivia i file di configurazione ed i loro metadati con Git (predefinito), Mercurial o GNU Bazaar (nuovo)
`timeshift`	V:7, I:13	4481	utilità di ripristino del sistema che usa rsync o istantanee BTRFS
`snapper`	V:6.1, I:8.3	2396	strumento di gestione di istantanee del file system Linux

Tabella 9.7: Elenco di pacchetti che possono registrare la cronologia della configurazione

Si può anche pensare di usare un approccio con script locale Sezione 10.2.3.

9.4 Monitorare, controllare ed avviare l'attività dei programmi

Le attività dei programmi possono essere monitorare e controllate usando strumenti specializzati.

pacchetto	popcon	dimensione	descrizione
coreutils	V:912, I:1000	17994	nice(1): esegue un programma con priorità di schedulazione modificata
bsdutils	V:421, I:999	338	renice(1): modifica la priorità di schedulazione di un processo in esecuzione
procps	V:831, I:997	2691	utilità per il file system "/proc": ps(1), top(1), kill(1), watch(1), …
psmisc	V:400, I:715	950	utilità per il file system "/proc": killall(1), fuser(1), peekfd(1), pstree(1)
time	V:6, I:76	148	time(1): esegue un programma per riportare l'uso delle risorse di sistema in funzione del tempo
sysstat	V:115, I:164	1904	sar(1), iostat(1), mpstat(1), …: strumenti per le prestazioni di sistema per Linux
isag	V:0.1, I:3.1	109	Interactive System Activity Grapher (tracciamento interattivo dell'attivita) per sysstat
lsof	V:456, I:950	492	lsof(8): elenca, usando l'opzione "-p", i file aperti da un processo in esecuzione
strace	V:10, I:100	3209	strace(1): traccia le chiamate e i segnali di sistema
ltrace	V:1, I:11	420	ltrace(1): traccia le chiamate di libreria
xtrace	V:0.03, I:0.69	342	xtrace(1): traccia la comunicazione tra client e server X11
powertop	V:36, I:230	696	powertop(1): informazioni sull'uso dell'alimentazione da parte del sistema
cron	V:923, I:997	248	esegue processi dal demone cron(8) sullo sfondo in base ad una pianificazione
anacron	V:423, I:481	110	pianificatore di comandi in stile cron, per i sistemi che non sono in esecuzione 24 ore al giorno
at	V:69, I:91	158	at(1) o batch(1): esegue un compito ad un orario specificato o quando il carico di sistema scende sotto un certo livello

Tabella 9.8: Elenco di strumenti per monitorare e controllare l'attività dei programmi.

Suggerimento
Il pacchetto procps fornisce strumenti estremamente di base per monitorare, controllare ed avviare le attività dei programmi. È consigliabile imparare ad usarli tutti.

9.4.1 Cronometrare un processo

Mostrare il tempo usato dal processo invocato da un comando.

```
# time some_command >/dev/null
real    0m0.035s        # time on wall clock (elapsed real time)
user    0m0.000s        # time in user mode
sys     0m0.020s        # time in kernel mode
```

9.4.2 Priorità di schedulazione

Per controllare la priorità di schedulazione di un processo è usato il valore di nice.

valore di nice	priorità di schedulazione
19	processo dalla priorità più bassa possibile (nice - gentile)
0	processo a priorità molto alta per l'utente
-20	processo a priorità molto alta per root (not-nice - non gentile)

Tabella 9.9: Elenco di valori di nice per la priorità di schedulazione

```
# nice  -19 top                                      # very nice
# nice --20 wodim -v -eject speed=2 dev=0,0 disk.img # very fast
```

A volte un valore molto alto di nice fa più male che bene al sistema; usare quindi questo comando con molta cautela.

9.4.3 Il comando ps

Il comando ps(1) in un sistema Debian supporta sia le funzionalità BSD sia quelle SystemV ed aiuta ad identificare l'attività dei processi in modo statico.

stile	comando tipico	funzione
BSD	`ps aux`	mostra %CPU %MEM
System V	`ps -efH`	mostra PPID

Tabella 9.10: Elenco degli stili per il comando ps

È possibile uccidere i processi figli zombie (defunti) tramite l'ID di processo del genitore identificato dal campo "`PPID`".

Il comando pstree(1) mostra un albero dei processi.

9.4.4 Il comando top

top(1) in sistemi Debian ha molte funzionalità e aiuta ad identificare in modo dinamico i processi che si stanno comportando in modo strano.

È un programma interattivo a tutto schermo. Si possono ottenere le informazioni di aiuto sul suo uso premendo il tasto «h» e uscire premendo il tasto «q».

9.4.5 Elencare i file aperti da un processo

È possibile elencare tutti i file aperti da un processo attraverso il suo PID (Identificativo di processo), ad esempio 1, con il comando seguente.

```
$ sudo lsof -p 1
```

Di solito il processo con PID=1 è il programma `init`.

9.4.6 Tenere traccia delle attività di un programma

Si può tenere traccia dell'attività di un programma con strace(1), ltrace(1) o xtrace(1) rispettivamente per quello che riguarda chiamate e segnali di sistema, chiamate di libreria o comunicazioni tra client e server X11.

Si può tenere traccia delle chiamate di sistema del comando `ls` nel modo seguente.

```
$ sudo strace ls
```

Suggerimento
Usare lo script **strace-graph** che si trova in **/usr/share/doc/strace/examples/** per creare una bella vista ad albero.

9.4.7 Identificazione di processi in base a file o socket

Usando fuser(1) è anche possibile identificare i processi in base ai file usando, ad esempio per "`/var/log/mail.log`" con il comando seguente.

```
$ sudo fuser -v /var/log/mail.log
                     USER        PID ACCESS COMMAND
/var/log/mail.log:   root       2946 F.... rsyslogd
```

Come si vede il file "`/var/log/mail.log`" è aperto in scrittura dal comando rsyslogd(8).

Usando fuser(1) si può anche identificare i processi in base ai socket, ad esempio per "`smtp/tcp`" con il comando seguente.

```
$ sudo fuser -v smtp/tcp
                     USER        PID ACCESS COMMAND
smtp/tcp:            Debian-exim   3379 F.... exim4
```

Ora si può vedere che sul sistema è in esecuzione exim4(8) per gestire le connessioni TCP alla porta SMTP (25).

9.4.8 Ripetere un comando ad intervalli costanti

watch(1) esegue un programma in modo ripetitivo ad intervalli regolari mostrando il suo output sullo schermo.

```
$ watch w
```

Questo comando mostra chi è attualmente connesso al sistema in modo aggiornato ogni 2 secondi.

9.4.9 Ripetere un comando su diversi file

Ci sono svariati modi di ripetere uno stesso comando su diversi file che rispondono ad una qualche condizione, ad esempio che corrispondono al modello glob "`*.ext`".

- Metodo del ciclo for nella shell (vedere Sezione 12.1.4):

```
for x in *.ext; do if [ -f "$x"]; then command "$x" ; fi; done
```

- Combinazione di find(1) e xargs(1):

```
find . -type f -maxdepth 1 -name '*.ext' -print0 | xargs -0 -n 1 command
```

- find(1) con l'opzione "`-exec`" con un comando:

```
find . -type f -maxdepth 1 -name '*.ext' -exec command '{}' \;
```

- find(1) con l'opzione "`-exec`" con un breve script di shell:

```
find . -type f -maxdepth 1 -name '*.ext' -exec sh -c "command '{}' && echo 'successful'" \;
```

Gli esempi precedenti sono stati scritti per assicurare la gestione appropriata di nomi di file particolari come quelli contenenti spazi. Per usi più avanzati di find(1), vedere Sezione 10.1.5.

9.4.10 Avviare un programma dalla GUI

Perl l'interfaccia a riga di comando (CLI), viene eseguito il primo programma con un nome corrispondente trovato nelle directory specificate nella variabile d'ambiente `$PATH`. Vedere Sezione 1.5.3.

Per l'interfaccia utente grafica (GUI) conforme agli standard di freedesktop.org i file `*.desktop` nella directory `/usr/share/applications/` forniscono gli attributi necessari per la visualizzazione di ogni programma nel menu della GUI. Ogni pacchetto che è conforme al sistema di menu xdg di Freedesktop.org installa i propri dati di menu forniti da "*.desktop" in "/usr/share/applications/". Gli ambienti desktop moderni che sono conformi allo standard Freedesktop.org usano questi dati per generare i loro menu usando il pacchetto xdg-utils. Vedere "/usr/share/doc/xdg-utils/README".

Per esempio, il file `chromium.desktop` definisce gli attributi per il «Browser Web Chromium» come «Name» per il nome di programma, «Exec» per il percorso e gli argomenti di esecuzione del programma, «Icon» per l'icona usata, ecc. (vedere laSpecifica per le voci per desktop), nel modo seguente:

```
[Desktop Entry]
Version=1.0
Name=Chromium Web Browser
GenericName=Web Browser
Comment=Access the Internet
Comment[fr]=Explorer le Web
Exec=/usr/bin/chromium %U
Terminal=false
X-MultipleArgs=false
Type=Application
Icon=chromium
Categories=Network;WebBrowser;
MimeType=text/html;text/xml;application/xhtml_xml;x-scheme-handler/http;x-scheme-handler/ ↩
    https;
StartupWMClass=Chromium
StartupNotify=true
```

Questa è una descrizione molto semplificata. I file `*.desktop` vengono analizzati nel modo seguente.

L'ambiente desktop imposta le variabili d'ambiente `$XDG_DATA_HOME` e `$XDG_DATA_DIR`. Per esempio, in GNOME:

- `$XDG_DATA_HOME` non è impostata. (Viene usato il valore predefinito: `$HOME/.local/share`.)
- `$XDG_DATA_DIRS` è impostata a `/usr/share/gnome:/usr/local/share/:/usr/share/`.

Perciò le directory base (vedere la XDG Base Directory Specification) e le directory `applications` sono le seguenti.

- `$HOME/.local/share/` → `$HOME/.local/share/applications/`
- `/usr/share/gnome/` → `/usr/share/gnome/applications/`
- `/usr/local/share/` → `/usr/local/share/applications/`

- `/usr/share/` → `/usr/share/applications/`

I file `*.desktop` vengono analizzati all'interno di queste directory `applications` in tale ordine.

Suggerimento
Una voce del menu GUI personalizzata dell'utente può essere creata aggiungendo un file `*.desktop` nella directory `$HOME/.local/share/applications/`.

Suggerimento
La riga "`Exec=...`" non viene analizzata dalla shell. Usare il comando env(1) se devono essere impostate variabili d'ambiente.

Suggerimento
In modo analogo, se viene creato un file `*.desktop` nella directory `autostart` all'interno di queste directory di base, il programma specificato nel file `*.desktop` viene eseguito automaticamente all'avvio dell'ambiente desktop. Vedere la specifica Desktop Application Autostart Specification.

Suggerimento
In modo simile, se viene creato un file `*.desktop` nella directory `$HOME/Desktop` e l'ambiente desktop è configurato per supportare la funzionalità di avviatore delle icone, il programma specificato in tale file viene eseguito quando si fa clic sull'icona. Notare che il nome effettivo della directory `$HOME/Desktop` dipende dalla localizzazione. Vedere xdg-user-dirs-update(1).

9.4.11 Personalizzare i programmi da avviare

Alcuni programmi avviano automaticamente altri programmi. Quelli che seguono sono alcuni punti fondamentali per la personalizzazione di questo processo.

- Menu di configurazione delle applicazioni:
  - Desktop GNOME: "Impostazioni" → "Applicazioni" → "Applicazioni predefinite"
  - Desktop KDE: "Avviatore applicazioni" → "Impostazioni di sistema → "Applicazioni predefinite"
  - Browser Iceweasel: "Modifica" → "Preferenze" → "Applicazioni"
  - mc(1): "`/etc/mc/mc.ext`"
- Variabili d'ambiente quali "`$BROWSER`", "`$EDITOR`", "`$VISUAL`" e "`$PAGER`" (vedere environ(7))
- Il sistema update-alternatives(1) per i programmi come "`editor`", "`view`", "`x-www-browser`", "`gnome-www-browser`" e "`www-browser`" (vedere Sezione 1.4.7)
- Il contenuto dei file "`~/.mailcap`" e "`/etc/mailcap`" che associano i tipi MIME con programmi (vedere mailcap(5))
- Il contenuto dei file "`~/.mime.types`" e "`/etc/mime.types`" che associano le estensioni dei nomi di file con tipi MIME (vedere run-mailcap(1))

Suggerimento
update-mime(8) aggiorna il file "`/etc/mailcap`" usando il file "`/etc/mailcap.order`" (vedere mailcap.order(5)).

Suggerimento
Il pacchetto `debianutils` fornisce sensible-browser(1), sensible-editor(1) e sensible-pager(1) che prendono decisioni sensate riguardo, rispettivamente, a quale browser web, editor e paginatore invocare. La lettura di questi script di shell è raccomandata.

Suggerimento
Per eseguire un'applicazione per console, come `mutt`, come applicazione preferita in X si dovrebbe creare un'applicazione X nel modo seguente ed impostare "`/usr/local/bin/mutt-term`" come applicazione preferite da avviare come descritto in precedenza.

```
# cat /usr/local/bin/mutt-term <<EOF
#!/bin/sh
gnome-terminal -e "mutt \$@"
EOF
# chmod 755 /usr/local/bin/mutt-term
```

9.4.12 Uccidere un processo

Per uccidere un processo (o inviare ad esso un segnale) in base al suo ID (identificativo) usare kill(1).

Per fare la stessa cosa ma in base al nome del comando del processo o ad altri attributi, usare killall(1) o pkill(1).

9.4.13 Pianificare compiti da eseguire una volta sola

Per pianificare un compito da eseguire una volta soltanto eseguire il comando at(1) nel modo seguente.

```
$ echo 'command -args'| at 3:40 monday
```

9.4.14 Pianificare compiti in modo regolare

Per pianificare compiti in modo regolare usare cron(8). Vedere crontab(1) e crontab(5).

Si può pianificare l'esecuzione di processi come utente normale, ad esempio l'utente `pippo`, creando un file crontab(5) come "`/var/spool/cron/crontabs/pippo`" con il comando "`crontab -e`".

Quello seguente è un esempio di file crontab(5).

```
# use /usr/bin/sh to run commands, no matter what /etc/passwd says
SHELL=/bin/sh
# mail any output to paul, no matter whose crontab this is
MAILTO=paul
# Min Hour DayOfMonth Month DayOfWeek command (Day... are OR'ed)
# run at 00:05, every day
5  0  *  * *   $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 14:15 on the first of every month -- output mailed to paul
15 14 1  * *   $HOME/bin/monthly
# run at 22:00 on weekdays(1-5), annoy Joe. % for newline, last % for cc:
0 22 *   * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%.%%
23 */2 1 2 *   echo "run 23 minutes after 0am, 2am, 4am ..., on Feb 1"
5  4 *   * sun echo "run at 04:05 every Sunday"
# run at 03:40 on the first Monday of each month
40 3 1-7 * *   [ "$(date +%a)" == "Mon" ] && command -args
```

valore del segnale	nome del segnale	azione	nota
0	---	nessun segnale viene inviato (vedere kill(2))	controlla se il processo è in esecuzione
1	SIGHUP	termina il processo	terminale disconnesso (segnale hang up)
2	SIGINT	termina il processo	interrupt dalla tastiera (CTRL-C)
3	SIGQUIT	termina il processo e fa un core dump	esce dalla tastiera (CTRL-\)
9	SIGKILL	termina il processo	segnale di terminazione non bloccabile
15	SIGTERM	termina il processo	segnale di terminazione bloccabile

Tabella 9.11: Elenco dei segnali usati comunemente con il comando kill

Suggerimento
Per i sistemi non in esecuzione in maniera continuata, installare il pacchetto `anacron` per pianificare l'esecuzione di comandi periodici, in maniera il più possibile vicina agli intervalli specificati, in base a quanto permesso dal tempo di attività della macchina. Vedere anacron(8) e anacrontab(5).

Suggerimento
Gli script con compiti pianificati di amministrazione del sistema possono essere eseguiti periodicamente dall'account di root, ponendoli in "`/etc/cron.hourly/`", "`/etc/cron.daily/`", "`/etc/cron.weekly/`" o "`/etc/cron.monthly/`". L'orario di esecuzione di questi script può essere personalizzato con "`/etc/crontab`" e "`/etc/anacrontab`".

Systemd ha una funzionalità a basso livello per pianificare l'esecuzione di programmi senza il demone `cron`. Per esempio, `/lib/systemd/system/apt-daily.timer` e `/lib/systemd/system/apt-daily.service` impostano le attività giornaliere di scaricamento di apt. Vedere systemd.timer(5) .

9.4.15 Pianificare compiti da eseguire in caso di un evento

Systemd può pianificare programmi non solo in base ad eventi temporizzati, ma anche in caso di eventi di mount. Vedere Sezione 10.2.3.3 e Sezione 10.2.3.2 per alcuni esempi.

9.4.16 Tasto Alt-SysRq

Premendo Alt-RSist (Stamp) seguito da un tasto fa la magia di ripristinare il controllo del sistema.

tasto premuto dopo Alt-R_Sist	descrizione dell'azione
k	uccide (**k**ill) tutti i processi nella **console virtuale attuale** (SAK)
s	**s**incronizza tutti i file system montati per evitare corruzione di dati
u	monta nuovamente tutti i file system in sola letturra (**u**mount)
r	ripristina la tastiera dalla modalità **r**aw dopo un crash di X

Tabella 9.12: Elenco di tasti per il comando SAK degni di nota

Vedere ulteriori informazioni in "Linux kernel user's and administrator's guide" » "Linux Magic System Request Key Hacks"

Suggerimento
Si può usare la funzione Alt-R_Sist da un terminale SSH, ecc. scrivendo su "`/proc/sysrq-trigger`". Per esempio, "`echo s > /proc/sysrq-trigger; echo u > /proc/sysrq-trigger`" dal prompt di shell di root **s**incronizza ed esegue **u**mount per tutti i file system montati.

L'attuale (2021) kernel Linux amd64 di Debian ha `/proc/sys/kernel/sysrq=438=0b110110110`:

- 2 = 0x2 - abilitare il controllo del livello di log della console (ON)
- 4 = 0x4 - abilitare il controllo della tastiera (SAK, unraw) (ON)
- 8 = 0x8 - abilitare dump di debug dei processi, ecc. (OFF)
- 16 = 0x10 - abilitare commando sync (ON)

- 32 = 0x20 - abiitare il ri-montaggio in sola lettura (ON)
- 64 = 0x40 - abilitare il segnali per i processi (term, kill, oom-kill) (OFF)
- 128 = 0x80 - permettere riavvio/spegnimento (ON)
- 256 = 0x100 - permettere l'impostazione valore di nice di tutte le attività RT (ON)

9.5 Suggerimenti per l'amministrazione del sistema

9.5.1 Chi è nel sistema?

Si può controllare chi è connesso al sistema nei modi seguenti.

- who(1) mostra chi ha fatto il login ed è connesso.
- w(1) mostra l'elenco di chi ha fatto il login ed è connesso e cosa sta facendo.
- last(1) mostra l'elenco degli utenti che hanno fatto il login più recentemente.
- lastb(1) mostra l'elenco degli utenti che hanno tentato il login non riuscendovi più recentemente.

Suggerimento
Queste informazioni sugli utenti sono contenute in "`/var/run/utmp`" e "`/var/log/wtmp`". Vedere login(1) e utmp(5).

9.5.2 Avvertire tutti gli utenti

Si può mandare un messaggio a tutti gli utenti che sono connessi al sistema con wall(1) nel modo seguente.

```
$ echo "We are shutting down in 1 hour" | wall
```

9.5.3 Identificazione dell'hardware

Per i dispositivi di tipo simil-PCI (AGP, PCI-Express, CardBus, ExpressCard, ecc.), lspci(8) (probabilmente con l'opzione "`-nn`") è un buon punto di partenza per l'identificazione dell'hardware.

In alternativa, si può identificare l'hardware leggendo il contenuto di "`/proc/bus/pci/devices`" o sfogliando l'albero di directory in "`/sys/bus/pci`" (vedere Sezione 1.2.12).

9.5.4 Configurazione dell'hardware

Sebbene nei moderni sistemi desktop con interfaccia grafica, come GNOME e KDE, la maggior parte della configurazione dell'hardware possa essere gestita attraverso strumenti di configurazione con interfaccia grafica da essi forniti, è bene conoscere alcuni metodi di base di configurazione.

Per la configurazione della tastiera e del terminale, vedere Sezione 8.2 e Sezione 8.3.

ACPI è un'infrastruttura per il sistema di gestione dell'energia più nuovo di APM

Suggerimento
La modifica della frequenza della CPU nei sistemi moderni è controllata da moduli del kernel come `acpi_cpufreq`.

pacchetto	popcon	dimensione	descrizione
pciutils	V:264, I:994	289	utilità PCI Linux: lspci(8)
usbutils	V:90, I:892	322	utilità Linux USB: lsusb(8)
nvme-cli	V:27, I:36	2222	utilità NVMe per Linux: nvme(1)
pcmciautils	V:4.1, I:6.1	92	utilità PCMCIA per Linux: pccardctl(8)
scsitools	V:0.2, I:2.1	261	raccosta di strumenti per la gestione dell'hardware SCSI:lsscsi(8)
procinfo	V:0.4, I:5.8	149	informazioni sul sistema ottenute da "/proc": lsdev(8)
lshw	V:12, I:88	971	informazioni sulla configurazione hardware: lshw(1)
discover	V:24, I:561	81	sistema di identificazione dell'hardware: discover(8)

Tabella 9.13: Elenco di strumenti per l'identificazione dell'hardware

pacchetto	popcon	dimensione	descrizione
console-setup	V:81, I:976	422	utilità Linux per i tipi di carattere e la mappa di tastiera in console
keyd	V:1.5, I:1.7	1437	demone per rimappare i tasti della tastiera usando primitive di input a livello del kernel (evdev, uinput)
keyd-application-mapper	V:0.01, I:0.08	34	demone per rimappare i tasti della tastiera - ri-mappatore specifico per applicazione
x11-xserver-utils	V:373, I:533	559	utilità del server X: xset(1), xmodmap(1)
acpid	V:52, I:78	158	demone per gestire eventi inviati dall'ACPI (Advanced Configuration and Power Interface, interfaccia avanzata per configurazione ed energia)
acpi	V:6, I:73	49	utilità per mostrare informazioni sui device ACPI
sleepd	V:0.07, I:0.10	84	demone per sospendere un portatile durante l'inattività
hdparm	V:110, I:200	246	ottimizzazione degli accessi al disco fisso (vedere Sezione 9.6.9)
smartmontools	V:242, I:270	2455	controllo e monitoraggio dei sistemi di archiviazione usando S.M.A.R.T.
setserial	V:3.0, I:4.9	104	raccolta di strumenti per la gestione delle porte seriali
memtest86+	V:1, I:19	12483	raccolta di strumenti per la gestione di hardware di memoria
scsitools	V:0.2, I:2.1	261	raccolta di strumenti per la gestione di hardware SCSI
setcd	V:0.05, I:0.61	33	ottimizzazione dell'accesso alle unità CD
big-cursor	I:0.67	26	puntatori del mouse più grandi per X

Tabella 9.14: Elenco di strumenti di configurazione dell'hardware

9.5.5 Orario di sistema e hardware

I comandi seguenti impostano l'ora di sistema e hardware a MM/GG hh:mm, AAAA.

```
# date MMDDhhmmCCYY
# hwclock --utc --systohc
# hwclock --show
```

Nei sistemi Debian, gli orari sono mostrati normalmente come ora locale, ma l'ora di sistema e quella hardware usano di solito l'ora UTC(GMT).

Se l'ora hardware è impostata ad UTC, modificare l'impostazione nel file "`/etc/default/rcS`" ad "`UTC=yes`".

Il comando seguente riconfigura il fuso orario utilizzato dal sistema Debian.

```
# dpkg-reconfigure tzdata
```

Se si desidera tenere aggiornata l'ora di sistema attraverso la rete, si consideri l'uso del servizion NTP con pacchetti come `ntp`, `ntpdate` e `chrony`.

Suggerimento
In systemd, usare invece `systemd-timesyncd` per l'ora di rete. Vedere systemd-timesyncd(8).

Vedere la documentazione seguente.

- Managing Accurate Date and Time HOWTO
- Progetto servizi NTP pubblici
- Il pacchetto `ntp-doc`

Suggerimento
ntptrace(8), nel pacchetto `ntp` può tracciare una catena di server NTP all'indietro fino alla fonte originaria.

9.5.6 L'infrastruttura audio

I driver di dispositivo per le schede audio per l'attuale Linux sono forniti da ALSA (Advanced Linux Sound Architecture). ALSA fornisce una modalità di emulazione per la compatibilità con il precedente sistema OSS (Open Sound System).

I software applicativi possono essere configurati non solo per accedere direttamente a device audio, ma anche per accedervi attraverso un qualche sistema server audio standardizzato. Attualmente PulseAudio, JACK e PipeWire vengono usati come sistemi per server audio. Vedere la pagina del wiki Debian sull'Audio per la situazione più recente.

Normalmente esiste un motore audio comune per ciascun ambiente desktop. Ciascun motore audio usato dalle applicazioni può scegliere di connettersi a diversi server audio.

Suggerimento
Per testare l'altoparlante usare "`cat /dev/urandom > /dev/audio`" oppure speaker-test(1) (^C per interrompere).

Suggerimento
Se non si ottiene l'audio, è possibile che l'altroparlante sia connesso ad un output impostato come muto. I moderni sistemi sonori hanno svariati output. alsamixer(1) nel pacchetto `alsa-utils` è utile per configurare le impostazioni del volume e di muto.

pacchetto	popcon	dimensione	descrizione
alsa-utils	V:342, I:460	2700	utilità per configurare ed usare ALSA
oss-compat	V:0.6, I:9.2	21	compatibilità OSS in ALSA per prevenire errori "/dev/dsp not found"
pipewire	V:326, I:370	136	server multimediale per motore di elaborazione audio e video - metapacchetto
pipewire-bin	V:333, I:370	2251	server multimediale per motore di elaborazione audio e video - server e CLI audio
pipewire-alsa	V:172, I:241	194	server multimediale per motore di elaborazione audio e video - server audio per rimpiazzare ALSA
pipewire-pulse	V:298, I:347	53	server multimediale per motore di elaborazione audio e video - server audio per rimpiazzare PulseAudio
pulseaudio	V:147, I:171	6616	server PulseAudio
libpulse0	V:443, I:579	984	libreria client PulseAudio
jackd	V:2, I:14	8	server JACK (JACK Audio Connection Kit) (bassa latenza)
libjack0	V:1.8, I:8.6	328	libreria JACK (JACK Audio Connection Kit) (bassa latenza)
libphonon4qt5-4	V:18, I:42	572	Phonon: motore audio di KDE

Tabella 9.15: Elenco di pacchetti relativi all'audio

9.5.7 Disabilitare il salvaschermo

Per disabilitare il salvaschermo usare i comandi seguenti.

ambiente	comando
Console Linux	setterm -powersave off
X Window (disabilitare il salvaschermo)	xset s off
X Window (disabilitare DPMS)	xset -dpms
X Window (configurazione tramite GUI del salvaschermo)	xscreensaver-command -prefs

Tabella 9.16: Elenco di comandi per disabilitare il salvaschermo

9.5.8 Disabilitare i bip sonori

Per disabilitare i bip sonori è sempre possibile disconnettere l'altoparlante del PC; la rimozione del modulo pcspkr del kernel fa stessa cosa per conto dell'utente.

Il comando seguente evita che il programma readline(3) usato da bash(1) emetta suoni bip quando incontra un carattere di allerta (ASCII=7).

```
$ echo "set bell-style none">> ~/.inputrc
```

9.5.9 Uso della memoria

Ci sono 2 risorse disponibile per l'utente per ottenere la situazione dell'uso della memoria.

- I messaggi di avvio del kernel nel file "/var/log/dmesg" contengono la dimensione esatta della memoria disponibile.
- free(1) e top(1) mostrano informazioni sulle risorse di memoria nel sistema mentre è in funzione.

Ecco un esempio.

```
# grep '\] Memory' /var/log/dmesg
[    0.004000] Memory: 990528k/1016784k available (1975k kernel code, 25868k reserved, 931k ↩
     data, 296k init)
$ free -k
             total       used       free     shared    buffers     cached
Mem:        997184     976928      20256          0     129592     171932
-/+ buffers/cache:     675404     321780
Swap:      4545576          4    4545572
```

Ci si potrebbe chiedere perché dmesg dice che ci sono 990 MB liberi e free -k dice che sono liberi 320 MB. Ne mancano più di 600 MB.

Non ci si deve preoccupare della grande dimensione del valore "`used`" (usata) e del piccolo valore di "`free`" (libera) nella riga "`Mem:`", ma si può invece leggere la riga sottostante (con 675404 e 321780 nell'esempio precedente) e rilassarsi.

Per il mio MacBook con 1GB=1048576k DRAM (il sistema video ne ruba un po'), vedo le informazioni seguenti.

fonte	**dimensione**
Dimensione totale in dmesg	1016784k = 1GB - 31792k
Libera in dmesg	990528k
Dimensione totale nella shell	997184k
Libera nella shell	20256k (ma in effetti 321780k)

Tabella 9.17: Elenco di dimensioni della memoria riportate

9.5.10 Verifica della sicurezza e dell'integrità del sistema

Una cattiva manutenzione del sistema può esporlo ad attacchi esterni.

Per verificare la sicurezza e l'integrità del sistema, si dovrebbe iniziare dai punti seguenti.

- Il pacchetto `debsums`; vedere debsums(1) e Sezione 2.5.2.
- Il pacchetto `chkrootkit`; vedere chkrootkit(1).
- La famiglia di pacchetti `clamav`; vedere clamscan(1) e freshclam(1).
- FAQ Debian Security.
- Manuale Securing Debian.

Con il piccolo script seguente è possibile controllare la presenza di tipici errori con permessi di scrittura per tutti per i file sbagliati.

```
# find / -perm 777 -a \! -type s -a \! -type l -a \! \( -type d -a -perm 1777 \)
```



Attenzione
Data che il pacchetto `debsums` usa le somme di controllo MD5 salvate localmente, non può essere completamente affidabile come strumento di controllo della sicurezza del sistema contro attacchi malevoli.

pacchetto	popcon	dimensione	descrizione
logcheck	V:4.4, I:5.4	120	demone per inviare via posta anomalie nei file di registro di sistema all'amministratore
debsums	V:4, I:29	108	utilità per controllare i file dei pacchetti installati con le somme di controllo MD5
chkrootkit	V:10, I:14	987	rilevatore di rootkit
clamav	V:8, I:38	19877	utilità anti-virus per Unix - interfaccia a riga di comando
tiger	V:1.0, I:1.3	7800	riporta vulnerabilità nella sicurezza di sistema
tripwire	V:1.1, I:1.4	5060	strumento di controllo dell'integrità di file e directory
john	V:1.1, I:7.3	469	strumento di violazione delle password attive
aide	V:2.2, I:2.6	328	AIDE - ambiente avanzato di rilevamento delle intrusioni - binario statico
integrit	V:0.09, I:0.17	2759	programma di controllo dell'integrità di file
crack	V:0.06, I:0.55	151	programma per indovinare password

Tabella 9.18: Elenco di strumenti per verificare la sicurezza e l'integrità del sistema

9.6 Suggerimenti per l'archiviazione dei dati

Avviare il sisteam come sistema Linux Live (vedere Sezione 3.2.2 e Sezione 3.2.3) rende facile riconfigurare l'archiviazione dei dati sul sistema installato.

Nota
Le informazioni date riguardo gli hard disk (HDD) sono applicabili ad altri dispositivi di archiviazione, come SSD / unità flash USB / Schede di memoria / ... . Sostituire negli esempi il nome del dispositivo, ad esempio `/dev/sda` con i nomi di dispositivo effettivi, `/dev/nvme0`, `/dev/mmcblk0`, ...

Può essere necessario smontare manualmente dalla riga di comando con umount(8) alcuni dispositivi prima di operare su di essi se sono stati automaticamente montati dal sistema desktop GUI.

9.6.1 Uso dello spazio su disco

L'uso dello spazio su disco può essere valutato con i programmi forniti dai pacchetti `mount`, `coreutils` e `xdu`:

- mount(8) mostra tutti i file system (= dischi) montati.
- df(1) mostra l'uso dello spazio su disco da parte dei file system.
- du(1) mostra l'uso dello spazio su disco da parte dell'albero di directory.

Suggerimento
Si può indirizzare l'output di du(8) a xdu(1) per far sì che che produca la sua rappresentazione grafica e interattiva, usando "`du -k . |xdu`", "`sudo du -k -x / |xdu`", ecc.

9.6.2 Configurazione del partizionamento dei dischi

Per la configurazione del partizionamento dei dischi, benché fdisk(8) sia considerato lo strumento standard, parted(8) merita un po' di attenzione. "Dati di partizionamento del disco", "tabella delle partizioni", "mappa delle partizioni" e "etichetta del disco" sono tutti sinonimi.

I PC più vecchi usano il classico schema basato su MBR (Master Boot Record) per contenere i dati sul partizionamento del disco nel primo settore, cioè il settore LBA 0 (512 byte).

I PC recenti con Unified Extensible Firmware Interface (UEFI), compresi i Mac basati su Intel, usano lo schema GPT (GUID Partition Table) per contenere i dati sul partizionamento del disco non nel primo settore.

Sebbene fdisk(8) sia stato lo strumento standard per il partizionamento dei dischi, parted(8) lo sta sostituendo.

pacchetto	popcon	dimensione	descrizione
`util-linux`	V:913, I:1000	4826	utilità di sistema varie inclusi fdisk(8) e cfdisk(8)
`parted`	V:449, I:569	126	programma GNU Parted per il ridimensionamento delle partizioni dei dischi
`gparted`	V:12, I:83	2313	editor delle partizioni di GNOME basato su `libparted`
`gdisk`	V:19, I:253	967	editor delle partizioni per i dischi ibridi GPT/MBR
`kpartx`	V:15, I:24	78	programma per creare la mappatura a device per le partizioni

Tabella 9.19: Elenco di pacchetti di gestione delle partizioni dei dischi



Attenzione
Sebbene parted(8) sostenga di creare e ridimensionare anche i file system, è più sicuro fare queste cose usando gli strumenti specializzati meglio mantenuti, come mkfs(8) (mkfs.msdos(8), mkfs.ext2(8), mkfs.ext3(8), mkfs.ext4(8), …) e resize2fs(8).

Nota
Per poter commutare tra GPT e MBR, è necessario cancellare direttamente i primi pochi blocchi del contenuto del disco (vedere Sezione 9.8.6) e usare "`parted /dev/sdx mklabel gpt`" o "`parted /dev/sdx mklabel msdos`", per fare il cambiamento. Notare che in questo contesto è usato "`msdos`" per MBR.

9.6.3 Accedere alle partizioni usando UUID

Anche se la riconfigurazione delle partizioni o l'ordine di attivazione di supporti di archiviazione removibili può portare ad avere nomi diversi per le partizioni, è possibile accedere ad esse in modo coerente. Ciò è utile anche se si hanno più dischi ed il BIOS/UEFI non assegna loro un nome di dispositivo costante.

- mount(8) con l'opzione "`-U`" può montare un device a blocchi usando l'UUID invece di usare il suo nome di file come "`/dev/sda3`".
- Il file "`/etc/fstab`" (vedere fstab(5)) può usare gli UUID.
- Anche i bootloader (Sezione 3.1.2) possono usare gli UUID.

Suggerimento
Si può scoprire l'UUID di un device a blocchi speciale con blkid(8).
Si può anche sondare UUID e altre informazioni con "`lsblk -f`".

9.6.4 LVM2

LVM2 è un gestore di volumi logici per il kernel Linux. Con LVM2 si possono creare partizioni dei dischi in volumi logici invece che sugli hard disk fisici.

LVM richiede quanto segue.

- la gestione di device-mapper nel kernel Linux (predefinita per i kernel Debian)
- la libreria per la gestione di device-mapper in spazio utente (pacchetto `libdevmapper*`)
- gli strumenti LVM2 in spazio utente (pacchetto `lvm2`)

Per iniziare a comprendere LVM2 guardare le pagine di manuale seguenti.

- lvm(8): Principi di base del meccanismo LVM2 (elenco di tutti i comandi di LVM2)
- lvm.conf(5): File di configurazione per LVM2
- lvs(8): Riporta informazioni sui volumi logici
- vgs(8): Riporta informazioni sui gruppi di volumi
- pvs(8): Riporta informazioni sui volumi fisici

9.6.5 Configurazione del file system

Per il file system ext4, il pacchetto `e2fsprogs` fornisce gli strumenti seguenti.

- mkfs.ext3(8) per creare nuovi file system ext4
- fsck.ext4(8) per controllare e riparare file system ext4 esistenti
- tune2fs(8) per configurare i superblocchi di un file system ext4
- debugfs(8) per fare il debug di file system ext4 in modo interattivo. (Ha un comando `undel` per ripristinare file eliminati.)

I comandi mkfs(8) e fsck(8) sono forniti dal pacchetto `e2fsprogs` come front-end per vari programmi dipendenti dal file syste (`mkfs.tipofs` e `fsck.tipofs`). Per il file system ext4 , sono mkfs.ext4(8) e fsck.ext4(8) (sono un collegamento simbolico a mke2fs(8) e e2fsck(8)).

Sono disponibili comandi simili per ciascun file system supportato da Linux.

Suggerimento
Il file system Ext4 è il predefinito per il sistema Linux e il suo uso è caldamente raccomandato a meno che non sia abbiano specifiche ragioni per non farlo.
Lo stato di Btrfs può essere trovato nelle pagine del wiki Debian su btrfs e in quelle del wiki di kernel.org su btrfs.
Ci si aspetta che sarà il prossimo file system predefinito dopo il file system ext4.
Alcuni strumenti permettono l'accesso a file system non supportati del kernel Linux (vedere Sezione 9.8.2).

9.6.6 Creare file system e verificarne l'integrità

In un sistema Linux, il comando mkfs(8) crea i file system ed il comando fsck(8) fornisce funzioni di controllo dell'integrità e di riparazione dei file system.

Debian ora in modo predefinito non fa `fsck` periodici dopo la creazione del file system.



Attenzione
In generale l'esecuzione di `fsck` su **file system montati** non è sicura.

---

pacchetto	popcon	dimensione	descrizione
e2fsprogs	V:821, I:997	1560	utilità per i file system ext2/ext3/ext4
btrfs-progs	V:52, I:79	5255	utilità per i file system Btrfs
reiserfsprogs	V:11, I:21	473	utilità per i file system Reiserfs
zfsutils-linux	V:32, I:33	1905	utilità per i file system OpenZFS
dosfstools	V:265, I:568	310	utilità per i file system FAT (Microsoft: MS-DOS, Windows)
exfatprogs	V:36, I:466	580	utilità per i file system exFAT mantenuti Samsung
exfat-fuse	V:2, I:40	73	driver (Microsoft) in lettura/scrittura per file system exFAT per FUSE
xfsprogs	V:42, I:84	4651	utilità per i file system XFS (SGI: IRIX)
ntfs-3g	V:208, I:514	1507	driver in lettura/scrittura per i file system NTFS (Microsoft: Windows NT, …) per FUSE
jfsutils	V:0.4, I:6.7	1104	utilità per i file system JFS (IBM: AIX, OS/2)
xorriso	V:15, I:65	378	utilità per scrittura di file system ISO-9660 e CD/DVD da libburnia
wodim	V:7, I:83	898	strumento per scrittura di CD/DVD a riga di comando da cdrkit
genisoimage	V:17, I:153	1567	strumento a riga di comando per file system ISO-9660 da cdrkit
reiser4progs	V:0.1, I:1.1	1367	utilità per i file system Reiser4
hfsprogs	V:0.3, I:3.1	394	utilità per i file system HFS e HFS Plus (Apple: Mac OS)
zerofree	V:7, I:116	30	programma per impostare a 0 i blocchi liberi in file system ext2/3/4

Tabella 9.20: Elenco di pacchetti di gestione dei file system

Suggerimento
Si può eseguire in modo sicuro il comando fsck(8) su tutti i file system, incluso il file system radice, al riavvio impostando "`enable_periodic_fsck`" in "`/etc/mke2fs.conf`" e il conteggio massimo di montaggi a 0 usando "`tune2fs -c0 /dev/`*`nome_partizione`*". Vedere mke2fs.conf(5) e tune2fs(8).
Per vedere i risultati del comando fsck(8) avviato dallo script di avvio, controllare i file in "`/var/log/fsck/`".

9.6.7 Ottimizzare il file system con opzioni di mount

La configurazione statica di base del file system è fornita da «`/etc/fstab`». Ad esempio,

```
«file system»                    «mount point» «type» «options»    «dump» «pass»
proc                                      /proc proc   defaults          0 0
UUID=709cbe4c-80c1-56db-8ab1-dbce3146d2f7 /     ext4   errors=remount-ro 0 1
UUID=817bae6b-45d2-5aca-4d2a-1267ab46ac23 none  swap   sw                0 0
/dev/scd0                          /media/cdrom0  udf,iso9660 user,noauto  0 0
```

Suggerimento
Per identificare un dispositivo a blocchi si può usare il suo UUID (vedere Sezione 9.6.3) invece del normale nome di device a blocchi quale «`/dev/sda1`», «`/dev/sda2`», …

A partire da Linux 2.6.30 il kernel ha in modo predefinito il comportamento fornito dalla opzione "`relatime`".

Vedere fstab(5) e mount(8).

9.6.8 Ottimizzare il file system tramite il superblocco

Le caratteristiche di un file system possono essere ottimizzate attraverso il suo superblocco usando il comando tune2fs(8).

- L'esecuzione di "`sudo tune2fs -l /dev/sda1`" mostra il contenuto del superblocco del file system in "`/dev/sda1`".
- L'esecuzione di "`sudo tune2fs -c 50 /dev/sda1`" cambia per "`/dev/sda1`" la frequenza dei controlli dei file system (l'esecuzione di `fsck` all'avvio) a 50 avvii.
- L'esecuzione di "`sudo tune2fs -j /dev/sda1`" aggiunge la funzionalità di journaling al file system in "`/dev/sda1`", cioè converte il file system da ext2 a ext3. (Eseguire questo comando su file system non montati.)
- L'esecuzione di "`sudo tune2fs -O extents,uninit_bg,dir_index /dev/sda1 && fsck -pf /dev/sda1`" converte il filesystem in "`/dev/sda1`" da ext3 a ext4. (Eseguire questo comando su file system non montati.)

Suggerimento
Nonostante il suo nome, tune2fs(8) non funziona soltanto sui file system ext2, ma anche sui file system ext3 e ext4.

9.6.9 Ottimizzare il disco rigido



avvertimento
Prima di giocare con la configurazione dei dischi controllare il proprio hardware e leggere la pagina man di hdparm(8), perché è una cosa piuttosto pericolosa per l'integrità dei dati.

Si può testare la velocità di accesso ai dischi di un disco rigido, ad esempio "`/dev/sda`" con "`hdparm -tT /dev/sda`". È possibile velocizzare alcuni dischi fissi connessi con (E)IDE con "`hdparm -q -c3 -d1 -u1 -m16 /dev/sda`" che abilita il "supporto (E)IDE per I/O a 32 bit", l'uso dell'opzione "using_dma", imposta l'opzione "interrupt-unmask" e imposta l'"I/O di settori multipli a 16" (pericoloso!).

Si possono testare le capacità della cache in scrittura di un disco fisso, ad esempio "`/dev/sda`", con "`hdparm -W /dev/sda`". Si può disabilitare la funzionalità di cache in scrittura con "`hdparm -W 0 /dev/sda`".

Potrebbe essere possibile leggere CDROM masterizzati male in unità CD-ROM moderne ad alta velocità rallentandole con "`setcd -x 2`".

9.6.10 Ottimizzare le unità a stato solido

Le unità SSD (Solid State Drive, a stato solido) sono ora rilevate automaticamente.

Ridurre accessi non necessari al disco per prevenire l'usura del disco montando "`tmpfs`" in percorsi dei dati volatili in `/etc/fstab`.

9.6.11 Usare SMART per prevedere danni ai dischi fissi

Con il demone `smartd` (8) è possibile monitorare e registrare i dischi fissi che sono conformi a SMART.

1. Abilitare la funzionalità SMART nel BIOS.
2. Installare il pacchetto `smartmontools`.
3. Identificare i dispositivi dei dischi fissi usando df(1).
   - Si supponga che uno dei dispositivi dei dischi fissi da monitorare sia "`/dev/sda`".
4. Controllare l'output di "`smartctl -a /dev/sda`" per vedere se la funzionalità SMART è veramente abilitata.

- Se non la è, abilitarla con "`smartctl -s on -a /dev/sda`".

5. Abilitare l'esecuzione del demone smartd(8) nel modo seguente.
   - Rimuovere il carattere iniziale di commento dalla riga "`start_smartd=yes`" nel file "`/etc/default/smartmonto`
   - Riavviare il demone smartd(8) con "`sudo systemctl restart smartmontools`".

Suggerimento
Il demone smartd(8) può essere personalizzato con il file `/etc/smartd.conf`, incluso per ciò che riguarda le notifiche dei messaggi di avvertimento.

9.6.12 Specificare una directory per l'archiviazione di dati temporanei usando $TMPDIR

Le applicazioni creano file temporanei normalmente nella directory di memorizzazione temporanea «`/tmp`». Se «`/tmp`» non contiene abbastanza spazio, si può specificare una directory di memorizzazione temporanea usando la variabile `$TMPDIR` per i programmi che si comportano in modo corretto.

9.6.13 Espandere lo spazio di archiviazione utilizzabile con LVM

Le partizioni create al momento dell'installazione su LVM (Logical Volume Manager (funzionalità di Linux) possono essere facilmente ridimensionate concatenando ad esse delle estensioni o suddividendo le loro estensioni su dispositivi di archiviazione multipli senza riconfigurazioni importanti del sistema.

9.6.14 Espandere lo spazio di archiviazione utilizzabile montando un'altra partizione

Se è disponibile una nuova partizione vuota (per esempio "`/dev/sdx`"), la si può formattare con mkfs.ext4(1) e montarla con mount(8) in una directory in cui è necessario avere più spazio. (È necessario copiare il contenuto originale della directory.)

```
$ sudo mv work-dir old-dir
$ sudo mkfs.ext4 /dev/sdx
$ sudo mount -t ext4 /dev/sdx work-dir
$ sudo cp -a old-dir/* work-dir
$ sudo rm -rf old-dir
```

Suggerimento
In alternativa si può montare un file immagine vuoto del disco (vedere Sezione 9.7.5) come device loop (vedere Sezione 9.7.3). Il reale uso del disco cresce mano a mano che vengono archiviati i dati.

9.6.15 Espandere lo spazio di archiviazione utilizzabile montando un collegamento ad un'altra directory

Se è disponibile una directory vuota (ad esempio "`/percorso/della/dir-vuota`") in un'altra partizione con spazio disponibile, si può, usando mount(8) con l'opzione "`--bind`", montarla in una directory (ad esempio "`dir-di-lavoro`") dove è necessario più spazio.

```
$ sudo mount --bind /path/to/emp-dir work-dir
```

9.6.16 Espandere lo spazio di archiviazione utilizzabile montando un'altra directory da sovrapporre

Se è disponibile dello spazio utilizzabile in un'altra partizione (ad esempio "`/percorso/della/vuota`" e "`/percorso/di/`
si può creare in essa una directory e impilarla in una vecchia directory (es., "`/percorso/della/vecchia`") in cui si ha bisogno di spazio usando OverlayFS con un kernel Linux 3.18 o successivo (Debian Stretch 9.0 o successiva).

```
$ sudo mount -t overlay overlay \
  -olowerdir=/path/to/old-dir,upperdir=/path/to/empty,workdir=/path/to/work
```

Qui, "`/percorso/della/vuota`" e "`/percorso/di/lavoro`" devono essere nella partizione abilitata in lettura e scrittura da scrivere in "`/percorso/della/vecchia`".

9.6.17 Espandere lo spazio di archiviazione utilizzabile usando collegamenti simbolici



Attenzione
Questo è un metodo deprecato. Certo software può non funzionare bene con i "collegamenti simbolici ad una directory". Usare invece gli approcci che usano il "mount" descritti in precedenza.

Se è disponibile una directory vuota (ad esempio "`/percorso/della/dir-vuota`" in un'altra partizione con spazio disponibile, si può creare un collegamento simbolico alla directory con ln(8).

```
$ sudo mv work-dir old-dir
$ sudo mkdir -p /path/to/emp-dir
$ sudo ln -sf /path/to/emp-dir work-dir
$ sudo cp -a old-dir/* work-dir
$ sudo rm -rf old-dir
```



avvertimento
Non usare un "collegamento simbolico ad una directory" per le directory gestite dal sistema, come "`/opt`". Un collegamento simbolico simile potrebbe essere sovrascritto quando il sistema viene aggiornato.

9.7 Immagine del disco

Questa sezione tratta della manipolazione di immagini di dischi.

9.7.1 Creare un file con un'immagine di disco

Il file di immagine di disco, "`disk.img`", di un dispositivo non montato, es. la secondo unità SCSI o ATA "`/dev/sdb`", può essere creata in uno dei modi seguenti.

```
# dd if=/dev/sdb of=disk.img; sync
```

```
# cp /dev/sdb disk.img ; sync
```

```
# cat /dev/sdb > disk.img ; sync
```

Si può creare l'immagine del disco del MBR (master boot record) dei PC tradizionali (vedere Sezione 9.6.2), che risiede nel primo settore del disco IDE primario usando dd(1) nel modo seguente.

```
# dd if=/dev/sda of=mbr.img bs=512 count=1
# dd if=/dev/sda of=mbr-nopart.img bs=446 count=1
# dd if=/dev/sda of=mbr-part.img skip=446 bs=1 count=66
```

- "`mbr.img`": MBR con tabella delle partizioni
- "`mbr-nopart.img`": MBR senza tabella delle partizioni
- "`mbr-part.img`": solamente la tabella delle partizioni nell'MBR

Se si sta creando l'immagine di una partizione del disco originale, sostituire "`/dev/sda`" con "`/dev/sda1`", ecc.

9.7.2 Scrivere direttamente sul disco

Il file immagine del disco, "`disco.img`" può essere scritto in un dispositivo non montato, ad esempio la seconda unità SCSI "`/dev/sdb`" di dimensione corrispondente in uno dei modi seguenti.

```
# dd if=disk.img of=/dev/sdb ; sync
```

```
# cp disk.img /dev/sdb ; sync
```

```
# cat disk.img >/dev/sdb ; sync
```

Analogamente il file immagine di una partizione del disco, "`partizione.img`" può essere scritto in una partizione non montata, ad esempio la prima partizione della seconda unità SCSI "`/dev/sdb1`" di dimensione corrispondente nel modo seguente.

```
# dd if=partition.img of=/dev/sdb1 ; sync
```

9.7.3 Montare un file con un'immagine di disco

Un'immagine di disco "`partizione.img`" contenente l'immagine di un'unica partizione, può essere montata e smontata usando il device loop nel modo seguente.

```
# losetup --show -f partition.img
/dev/loop0
# mkdir -p /mnt/loop0
# mount -t auto /dev/loop0 /mnt/loop0
...hack...hack...hack
# umount /dev/loop0
# losetup -d /dev/loop0
```

Questo può essere semplificato nel modo seguente.

```
# mkdir -p /mnt/loop0
# mount -t auto -o loop partition.img /mnt/loop0
...hack...hack...hack
# umount partition.img
```

Ogni partizione di un'immagine di disco "`disco.img`" contentente più partizioni, può essere montata usando il device loop.

```
# losetup --show -f -P disk.img
/dev/loop0
# ls -l /dev/loop0*
brw-rw---- 1 root disk   7,  0 Apr  2 22:51 /dev/loop0
brw-rw---- 1 root disk 259, 12 Apr  2 22:51 /dev/loop0p1
brw-rw---- 1 root disk 259, 13 Apr  2 22:51 /dev/loop0p14
brw-rw---- 1 root disk 259, 14 Apr  2 22:51 /dev/loop0p15
# fdisk -l /dev/loop0
Disk /dev/loop0: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 6A1D9E28-C48C-2144-91F7-968B3CBC9BD1

Device         Start     End Sectors  Size Type
/dev/loop0p1  262144 4192255 3930112  1.9G Linux root (x86-64)
/dev/loop0p14   2048    8191    6144    3M BIOS boot
/dev/loop0p15   8192  262143  253952  124M EFI System

Partition table entries are not in disk order.
# mkdir -p /mnt/loop0p1
# mkdir -p /mnt/loop0p15
# mount -t auto /dev/loop0p1 /mnt/loop0p1
# mount -t auto /dev/loop0p15 /mnt/loop0p15
# mount |grep loop
/dev/loop0p1 on /mnt/loop0p1 type ext4 (rw,relatime)
/dev/loop0p15 on /mnt/loop0p15 type vfat (rw,relatime,fmask=0002,dmask=0002,allow_utime ↩
    =0020,codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)
...hack...hack...hack
# umount /dev/loop0p1
# umount /dev/loop0p15
# losetup -d /dev/loop0
```

In alternativa, un risultato simile può essere ottenuto utilizzando i device mapper creati da kpartx(8), contenuto nel pacchetto `kpartx`, nel modo seguente.

```
# kpartx -a -v disk.img
add map loop0p1 (253:0): 0 3930112 linear 7:0 262144
add map loop0p14 (253:1): 0 6144 linear 7:0 2048
add map loop0p15 (253:2): 0 253952 linear 7:0 8192
# fdisk -l /dev/loop0
Disk /dev/loop0: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 6A1D9E28-C48C-2144-91F7-968B3CBC9BD1

Device         Start     End Sectors  Size Type
/dev/loop0p1  262144 4192255 3930112  1.9G Linux root (x86-64)
/dev/loop0p14   2048    8191    6144    3M BIOS boot
/dev/loop0p15   8192  262143  253952  124M EFI System

Partition table entries are not in disk order.
# ls -l /dev/mapper/
total 0
crw------- 1 root root 10, 236 Apr  2 22:45 control
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p1 -> ../dm-0
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p14 -> ../dm-1
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p15 -> ../dm-2
```

```
# mkdir -p /mnt/loop0p1
# mkdir -p /mnt/loop0p15
# mount -t auto /dev/mapper/loop0p1 /mnt/loop0p1
# mount -t auto /dev/mapper/loop0p15 /mnt/loop0p15
# mount |grep loop
/dev/loop0p1 on /mnt/loop0p1 type ext4 (rw,relatime)
/dev/loop0p15 on /mnt/loop0p15 type vfat (rw,relatime,fmask=0002,dmask=0002,allow_utime ↩
    =0020,codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)
...hack...hack...hack
# umount /dev/mapper/loop0p1
# umount /dev/mapper/loop0p15
# kpartx -d disk.img
```

9.7.4 Pulire un file con un'immagine di disco

Si può pulire un file con immagine di disco "`disco.img`" da tutti i file cancellati creando un'immagine pulita "`nuova.img`" nel modo seguente.

```
# mkdir old; mkdir new
# mount -t auto -o loop disk.img old
# dd bs=1 count=0 if=/dev/zero of=new.img seek=5G
# mount -t auto -o loop new.img new
# cd old
# cp -a --sparse=always ./ ../new/
# cd ..
# umount new.img
# umount disk.img
```

Se "`disco.img`" è in ext2, ext3 o ext4, si può anche usare zerofree(8), contenuto nel pacchetto `zerofree`, nel modo seguente.

```
# losetup --show -f disk.img
/dev/loop0
# zerofree /dev/loop0
# cp --sparse=always disk.img new.img
# losetup -d /dev/loop0
```

9.7.5 Creare un file con immagine di disco vuoto

Si può creare un'immagine di disco vuota "`disco.img`", che può crescere fino a 5GiB, usando dd(1) nel modo seguente.

```
$ dd bs=1 count=0 if=/dev/zero of=disk.img seek=5G
```

Invece di usare dd(1), qui si può usare lo specializzato fallocate(8).

È possibile creare un file system ext4 in questa immagine di disco, "`disco.img`" usando il device loop nel modo seguente.

```
# losetup --show -f disk.img
/dev/loop0
# mkfs.ext4 /dev/loop0
...hack...hack...hack
# losetup -d /dev/loop0
$ du  --apparent-size -h disk.img
5.0G  disk.img
$ du -h disk.img
83M disk.img
```

La dimensione del file "`disco.img`" è di 5.0GiB e il suo effettivo uso del disco è di soli 83MiB. Questa discrepanza è resa possibile dal fatto che il file system ext4 può contentere file sparsi.

Suggerimento
L'uso effettivo del disco dei file sparsi cresce insieme ai dati che in essi sono scritti.

Usando operazioni simili su device creati dal device loop o dal device mapper, come in Sezione 9.7.3, si può partizionare tale immagine di disco "`disco.img`" usando parted(8) o fdisk(8) e si può creare in essa file system usando mkfs.ext4(8), mkswap(8), ecc.

9.7.6 Creare un file con un'immagine ISO9660

Suggerimento
Sia genisoimage(1) fornito da cdrkit sia xorrisofs(1) fornito da Libburnia condividono la stessa sintassi per i comandi, tranne che il nome del comando.

Si può creare un file immagine ISO9660 "`cd.iso`" dell'albero di directory originale in "`directory_sorgente`" usando genisoimage(1) fornito da cdrkit nel modo seguente.

```
#  genisoimage -r -J -T -V volume_id -o cd.iso source_directory
```

Analogamente, si può creare un file immagine ISO9660 avviabile, "`cdboot.iso`", da un albero di directory simile a quello del `debian-installer` in "`directory_sorgente`" nel modo seguente.

```
#  genisoimage -r -o cdboot.iso -V volume_id \
   -b isolinux/isolinux.bin -c isolinux/boot.cat \
   -no-emul-boot -boot-load-size 4 -boot-info-table source_directory
```

In questo esempio viene usato per l'avvio il bootloader Isolinux (vedere Sezione 3.1.2).

Si può calcolare il valore md5sum e creare l'immagine ISO9660 direttamente dal device CD-ROM nel modo seguente.

```
$ isoinfo -d -i /dev/cdrom
CD-ROM is in ISO 9660 format
...
Logical block size is: 2048
Volume size is: 23150592
...
# dd if=/dev/cdrom bs=2048 count=23150592 conv=notrunc,noerror | md5sum
# dd if=/dev/cdrom bs=2048 count=23150592 conv=notrunc,noerror > cd.iso
```



avvertimento
Per ottenere un risultato corretto, si deve accuratamente evitare il bug di Linux riguardante il read ahead del file system ISO9660, come nell'esempio precedente.

9.7.7 Scrivere direttamente sul CD/DVD-R/RW

Suggerimento
Un DVD è solo un CD molto grande per wodim(1) fornito da cdrkit e xorrecord(1) fornito da Libburnia. Questi comandi condividono la stessa sintassi dei comandi tranne che per il nome del comando.

Si può trovare un device utilizzabile usando il comando seguente.

```
# wodim --devices
```

Poi si inserisce un CD-R vergine nell'unità CD e si scrive il file immagine ISO9660 "`cd.iso`" su questo device, ad esempio "`/dev/sda`", usando wodim(1) nel modo seguente.

```
# wodim -v -eject dev=/dev/sda cd.iso
```

Se viene usato un CD-RW invece di un CD-R, usare quest'altro comando.

```
# wodim -v -eject blank=fast dev=/dev/sda cd.iso
```

Suggerimento
Se il sistema desktop usato monta automaticamente i CD, prima di usare wodim(1) smontarlo usando dalla console "`sudo umount /dev/sda`".

9.7.8 Montare un file con un'immagine ISO9660

Se "`cd.iso`" contiene un'immagine ISO9660, lo si può montare manualmente in "`/cdrom`" usando il comando seguente.

```
# mount -t iso9660 -o ro,loop cd.iso /cdrom
```

Suggerimento
I sistemi desktop moderni possono montare automaticamente i supporti removibili come i CD in formato ISO9660 (vedere Sezione 10.1.7).

9.8 I dati binari

Questa sezione tratta della manipolazione diretta dei dati binari su supporti di archiviazione.

9.8.1 Visualizzare e modificare dati binari

Il metodo di visualizzazione dei dati binai più basilare è l'uso del comando "`od -t x1`".

Suggerimento
HEX è usato come acronimo per il formato esadecimale con base 16. OCTAL è usato per il formato ottale con base 8. ASCII è usato per American Standard Code for Information Interchange, cioè la normale codifica per testi in inglese. EBCDIC è usato per Extended Binary Coded Decimal Interchange Code usato nei sistemi operativi deimainframe IBM.

9.8.2 Manipolare file senza montare i dischi

Esistono strumenti per leggere e scrivere file senza montare i dischi.

pacchetto	popcon	dimensione	descrizione
coreutils	V:912, I:1000	17994	pacchetto base che contiene od(1) per fare il dump di file (HEX, ASCII, OCTAL, …)
bsdmainutils	V:4, I:121	17	pacchetto di utilità che contiene hd(1) per fare il dump di file (HEX, ASCII, OCTAL, …)
hexedit	V:0.8, I:7.5	70	visualizzatore ed editor binario (HEX, ASCII)
bless	V:0.1, I:1.2	924	editor esadecimale completo (GNOME)
okteta	V:1, I:11	1591	editor esadecimale completo (KDE4)
ncurses-hexedit	V:0.1, I:1.1	130	visualizzatore ed editor binario (HEX, ASCII, EBCDIC)
beav	V:0.04, I:0.26	137	visualizzatore ed editor binario (HEX, ASCII, EBCDIC, OCTAL, …)

Tabella 9.21: Elenco di pacchetti che visualizzano e modificano dati binari

pacchetto	popcon	dimensione	descrizione
mtools	V:7, I:52	400	utilità per file MSDOS senza montarli
hfsutils	V:0.1, I:2.5	178	utilità per file HFS e HFS+ senza montarli

Tabella 9.22: Elenco di pacchetti per manipolare file senza montare i dischi

9.8.3 Dati ridondanti

I sistemi software RAID offerti dal kernel Linux forniscono un livello di ridondanza dei dati nel file system a livello del kernel, allo scopo di ottenere una più alta affidabilità dell'archiviazione.

Esistono strumenti per aggiungere dati ridondanti a file a livello di programmi applicativi per ottenere anche in questo modo una più alta affidabilità dell'archiviazione.

pacchetto	popcon	dimensione	descrizione
par2	V:9, I:116	330	insiemi di volumi di archivi di parità, per controllare e riparare file
dvdisaster	V:0.1, I:1.1	1422	protezione contro perdita di dati/graffi/invecchiamento di supporti CD/DVD
dvbackup	V:0.01, I:0.03	413	strumento di backup che usa camcorder MiniDV (fornisce rsbep(1))

Tabella 9.23: Elenco di strumenti per aggiungere dati ridondanti a file

9.8.4 recupero di file dati ed analisi forensi

Esistono strumenti per recuperare file dati e fare analisi forensi.

Suggerimento
Si possono ripristinare i file sui file system ext2 usando i comandi `list_deleted_inodes` e `undel` di debugfs(8) nel pacchetto `e2fsprogs`.

pacchetto	popcon	dimensione	descrizione
testdisk	V:2, I:25	528	utilità per scansione delle partizione e ripristino di dischi
magicrescue	V:0.2, I:1.7	257	utilità per ripristinare file cercando byte magici
scalpel	V:0.2, I:2.2	89	strumento di escavazione di file frugale e ad alte prestazioni
myrescue	V:0.2, I:1.9	81	recupera dati da dischi fissi danneggiati
extundelete	V:0.5, I:7.3	152	utilità per decancellare file da file system ext3/4
ext4magic	V:0.2, I:3.5	235	utilità per decancellare file da file system ext3/4
ext3grep	V:0.2, I:1.8	299	strumento per aiutare a recuperare file cancellati da file system ext3
scrounge-ntfs	V:0.1, I:1.6	49	programma di recupero di dati da file system NTFS
gzrt	V:0.02, I:0.43	34	insieme di strumenti di recupero di gzip
sleuthkit	V:3, I:25	1119	strumenti per analisi forensi (Sleuthkit)
autopsy	V:0.1, I:1.1	1026	interfaccia grafica per SleuthKit
foremost	V:0.3, I:3.8	101	applicazione forense per il recupero dei dati
guymager	V:0.08, I:0.71	1049	strumento forense per immagini basato su Qt
dcfldd	V:0.3, I:2.6	113	versione migliorata di dd per analisi forensi e sicurezza

Tabella 9.24: Elenco di pacchetti per recupero di file dati ed analisi forensi.

9.8.5 Suddividere un file grande in file più piccoli

Quando dei dati sono troppo grandi affinché ne venga fatto il backup in un file singolo, si può fare il backup dei suoi contenuti dopo averlo suddiviso in pezzi di, ad esempio 2000MiB, che saranno successivamente riuniti a formare il file originale.

```
$ split -b 2000m large_file
$ cat x* >large_file
```



Attenzione
Assicurarsi di non avere altri file che iniziano con ”x” per evitare conflitti nei nomi.

9.8.6 Pulire il contenuto di file

Per ripulire i contenuti di un file, come un file di registro, non usare rm(1) per cancellarlo e poi crearne un altro, perché nell’intervallo tra i due comandi potrebbe essere ancora possibile accedere al file. Quello che segue è il metodo sicuro per pulire il contenuto di un file.

```
$ :>file_to_be_cleared
```

9.8.7 File fittizi

I comandi seguenti creano file fittizi o vuoti.

```
$ dd if=/dev/zero    of=5kb.file bs=1k count=5
$ dd if=/dev/urandom of=7mb.file bs=1M count=7
$ touch zero.file
$ : > alwayszero.file
```

Dovrebbero essere ora presenti i seguenti file.

- "`5kb.file`" è costituito da 5KB di zero.
- "`7mb.file`" è costituito da 7MB di dati casuali.
- "`zero.file`" potrebbe essere un file di 0 byte. Se fosse stato preesistente, il suo orario `mtime` sarebbe stato aggiornato mentre sarebbero stati mantenuti i suoi contenuti e la sua grandezza.
- "`semprezero.file`" è sempre un file di 0 byte. Se fosse stato preesistente, il suo orario `mtime` sarebbe stato aggiornato e il suo contenuto azzerato.

9.8.8 Cancellare un intero disco fisso

Esistono diversi modi di cancellare completamente i dati da un intero device simile ad un disco fisso, ad esempio una unità flash USB in "`/dev/sda`".



Attenzione
Prima di eseguire i comandi indicati in seguito controllare la posizione della unità flash USB con mount(8). Il device a cui punta "`/dev/sda`" potrebbe essere il disco fisso SCSI o Serial-ATA che contiene l'intero sistema.

Cancellare tutti i contenuti del disco reimpostando tutti i dati a 0 con il comando seguente.

```
# dd if=/dev/zero of=/dev/sda
```

Cancellare tutto sovrascrivendo dati casuali con il comando seguente.

```
# dd if=/dev/urandom of=/dev/sda
```

Cancellare tutto sovrascrivendo dati casuali in modo molto efficiente con il comando seguente.

```
# shred -v -n 1 /dev/sda
```

In alternativa si può usare badblocks(8) con l'opzione `-t random`.

Dato che dd(1) è disponibile dalla shell di molti CD Linux avviabili, come il CD dell'installatore Debian, si può cancellare completamente il sistema installato su un disco fisso, ad esemio "`/dev/sda`", "`/dev/sda`", ecc., eseguendo un comando di cancellazione da un supporto CD simile.

9.8.9 Cancellare area inutilizzate di un disco fisso

Area inutilizzate di un disco fisso (o una unità flash USB), ad esempio "`/dev/sdb1`", potrebbero contenere ancora i dati cancellati stessi, dato che questi sono semplicemente scollegati dal file system. È possibile pulire queste aree sovrascrivendole.

```
# mount -t auto /dev/sdb1 /mnt/foo
# cd /mnt/foo
# dd if=/dev/zero of=junk
dd: writing to `junk': No space left on device
...
# sync
# umount /dev/sdb1
```



avvertimento
Solitamente questo procedimento è sufficientemente buono per le unità flash USB. Ma non è perfetto. La maggior parte dei nomi di file cancellati e dei loro attributi potrebbe ancora essere nascosta e rimanere nel file system.

9.8.10 De-cancellare file cancellati ma ancora aperti

Anche se un file è stato cancellato per errore, fintanto che è usato da un'applicazione (in lettura o scrittura), è possibile recuperarlo.

Per esempio, provare a fare quanto segue.

```
$ echo foo > bar
$ less bar
$ ps aux | grep ' less[ ]'
bozo    4775  0.0  0.0  92200   884 pts/8    S+   00:18   0:00 less bar
$ rm bar
$ ls -l /proc/4775/fd | grep bar
lr-x------ 1 bozo bozo 64 2008-05-09 00:19 4 -> /home/bozo/bar (deleted)
$ cat /proc/4775/fd/4 >bar
$ ls -l
-rw-r--r-- 1 bozo bozo 4 2008-05-09 00:25 bar
$ cat bar
foo
```

Quando si ha il pacchetto `lsof` installato, eseguire in un altro terminale quanto segue.

```
$ ls -li bar
2228329 -rw-r--r-- 1 bozo bozo 4 2008-05-11 11:02 bar
$ lsof |grep bar|grep less
less 4775 bozo 4r REG 8,3 4 2228329 /home/bozo/bar
$ rm bar
$ lsof |grep bar|grep less
less 4775 bozo 4r REG 8,3 4 2228329 /home/bozo/bar (deleted)
$ cat /proc/4775/fd/4 >bar
$ ls -li bar
2228302 -rw-r--r-- 1 bozo bozo 4 2008-05-11 11:05 bar
$ cat bar
foo
```

9.8.11 Cercare tutti i collegamenti fisici

I file con collegamenti fisici possono essere identificati usando "`ls -li`".

```
$ ls -li
total 0
2738405 -rw-r--r-- 1 root root 0 2008-09-15 20:21 bar
2738404 -rw-r--r-- 2 root root 0 2008-09-15 20:21 baz
2738404 -rw-r--r-- 2 root root 0 2008-09-15 20:21 foo
```

Entrambi "`pippo`" e "`pluto`" hanno il numero di collegamenti uguale a "2" (>1), il che mostra che hanno collegamenti fisici. Hanno il numero di inode, "2738404", in comune. Ciò significa che sono lo stesso file con un collegamento fisico. Nell'eventualità che non si trovino tutti i file con collegamento fisico, li si possono cercare in base all'inode, ad esempio "2738404", nel modo seguente.

```
# find /path/to/mount/point -xdev -inum 2738404
```

9.8.12 Consumo invisibile dello spazio su disco

Tutti i file cancellati ma aperti consumano spazio su disco, anche se non sono visibili con il normale comando du(1). Possono essere elencati, insieme alla loro dimensione, nel modo seguente.

```
# lsof -s -X / |grep deleted
```

9.9 Suggerimenti per la cifratura dei dati

Avendo l'accesso fisico alla macchina, chiunque può facilmente ottenere privilegi di root e accedere a tutti i file nel PC (vedere Sezione 4.6.4). Ciò significa che il sistema delle password di login non può mettere al sicuro i dati privati e sensibili contro un possibile furto del PC. Per farlo deve essere usata una tecnologia di cifratura dei dati. Sebbene GNU Privacy Guard (vedere Sezione 10.3) possa cifrare file, richiede un po' di lavoro da parte dell'utente.

dm-crypt facilita la cifratura automatica dei dati attraverso moduli nativi del kernel Linux con un minimo lavoro da parte dell'utente usando device-mapper.

pacchetto	popcon	dimensione	descrizione
`cryptsetup`	V:34, I:77	469	utilità per device a blocchi cifrati (dm-crypt / LUKS)
`cryptmount`	V:1.8, I:2.3	236	utilità per device a blocchi cifrati (dm-crypt / LUKS) con particolare attenzione al montaggio/smontaggio da parte di utenti normali
`fscrypt`	V:0.4, I:1.1	6844	utilità per file cifratura di file system Linux (fscrypt)
`libpam-fscrypt`	V:0.31, I:0.34	6941	modulo PAM per cifratura di file system Linux (fscrypt)

Tabella 9.25: Elenco di utilità per la cifratura dei dati



Attenzione
La cifratura dei dati consuma tempo della CPU, ecc. I dati cifrati diventano inaccessibili se la password viene persa. Valutare bene i suoi costi e benefici.

Nota
È possibile installare un intero file system Debian in un disco cifrato con l'Installatore Debian (lenny o successivo) usando dm-crypt/LUKS e initramfs.

Suggerimento
Per lo strumento di cifratura in spazio utente GNU Privacy Guard vedere Sezione 10.3.

9.9.1 Cifratura di dischi removibili con dm-crypt/LUKS

Si può cifrare il contenuto di dispositivi di memorizzazione di massa removibili, ad esempio una unità flash USB in "`/dev/sdx`", usando dm-crypt/LUKS. Formattarla semplicemente nel modo seguente.

```
# fdisk /dev/sdx
... "n" "p" "1" "return" "return" "w"
# cryptsetup luksFormat /dev/sdx1
...
# cryptsetup open /dev/sdx1 secret
...
# ls -l /dev/mapper/
total 0
crw-rw---- 1 root root  10, 60 2021-10-04 18:44 control
lrwxrwxrwx 1 root root       7 2021-10-04 23:55 secret -> ../dm-0
# mkfs.vfat /dev/mapper/secret
...
# cryptsetup close secret
```

Si può poi montarla come una chiavetta qualunque in "`/media/nomeutente/etichetta_disco`", tranne per il fatto che verrà chiesta la password (vedere Sezione 10.1.7) nei moderni ambienti desktop che usano il pacchetto `udisks2`. La differenza è che ogni dato scritto in essa è cifrato. L'inserimento della password può essere automatizzato usando un portachiavi (vedere Sezione 10.3.6).

In alternativa si può formattare il supporto con diversi file system, es. ext4 con "`mkfs.ext4 /dev/mapper/sdx1`". Se viene invece usato btrfs, deve essere installato il pacchetto `udisks2-btrfs`. Per questi file system può essere necessario configurare la proprietà e i permessi dei file.

9.9.2 Montare dischi removibili con dm-crypt/LUKS

Ad esempio, una partizione di disco cifrata creata con dm-crypt/LUKS su "`/dev/sdc5`" dall'Installatore Debian può essere montata in "`/mnt`" nel modo seguente:

```
$ sudo cryptsetup open /dev/sdc5 ninja --type luks
Enter passphrase for /dev/sdc5: ****
$ sudo lvm
lvm> lvscan
  inactive          '/dev/ninja-vg/root' [13.52 GiB] inherit
  inactive          '/dev/ninja-vg/swap_1' [640.00 MiB] inherit
  ACTIVE            '/dev/goofy/root' [180.00 GiB] inherit
  ACTIVE            '/dev/goofy/swap' [9.70 GiB] inherit
lvm> lvchange -a y /dev/ninja-vg/root
lvm> exit
  Exiting.
$ sudo mount /dev/ninja-vg/root /mnt
```

9.10 Il kernel

Debian distribuisce, per le architetture supportate, kernel Linux modulari contenuti in pacchetti.

Se si sta leggendo questa documentazione probabilmente non è necessario compilare in proprio il kernel Linux.

9.10.1 Parametri del kernel

Molte caratteristiche di Linux possono essere configurate tramite parametri del kernel, nei modi seguenti.

- Parametri del kernel inizializzati dal bootloader (vedere Sezione 3.1.2)
- Se accessibili attraverso sysfs (vedere Sezione 1.2.12), parametri del kernel modificati da sysctl(8) durante l'esecuzione del sistema.
- Parametri di moduli impostati dagli argomenti di modprobe(8) al momento dell'attivazione di un modulo (vedere Sezione 9.7.3)

Per i dettagli vedere "The Linux kernel user's and administrator's guide » The kernel's command-line parameters".

9.10.2 Header del kernel

La maggior parte dei **normali programmi** non ha bisogno degli header del kernel per essere compilata, anzi di fatto può corrompersi se si usano direttamente gli header. Questi programmi dovrebbero essere compilati nel sistema Debian usando gli header in "`/usr/include/linux`" e "`/usr/include/asm`" forniti dal pacchetto `libc6-dev` (creato dal pacchetto sorgente `glibc`).

Nota
Per compilare alcuni programmi specifici per il kernel, come moduli per il kernel da fonti esterne e il demone automounter (amd), è necessario includere nella propria riga di comando il percorso ai corrispondenti header del kernel, ad esempio "`-I/usr/src/linux-particular-version/include/`".

9.10.3 Compilare il kernel ed i moduli relativi

Debian ha un proprio metodo di compilazione del kernel e dei moduli relativi.

pacchetto	popcon	dimensione	descrizione
`build-essential`	V:20, I:510	17	pacchetti essenziali per compilare pacchetti Debian: `make`, `gcc`, …
`bzip2`	V:163, I:973	113	utilità di compressione e decompressione per file bz2
`libncurses5-dev`	I:31	6	librerie di sviluppo e documentazione per ncurses
`git`	V:422, I:625	52512	git: sistema distribuito di controllo delle revisioni usato dal kernel Linux
`fakeroot`	V:32, I:511	225	fornisce un ambiente fakeroot per compilare pacchetti da utente non root
`initramfs-tools`	V:495, I:989	49	strumento per compilare un initramfs (specifico di Debian)
`dkms`	V:73, I:147	235	DKMS (Dynamic Kernel Module Support, supporto dinamico per i noduli del kernel) (generico)
`module-assistant`	V:1, I:12	395	script di aiuto per creare pacchetti di moduli (specifico di Debian)
`devscripts`	V:5, I:31	2783	script di aiuto per i manutentori di pacchetti Debian (specifico di Debian)

Tabella 9.26: Elenco di pacchetti chiave da installare per la ricompilazione del kernel in un sistema Debian

Se si usa `initrd` nello Sezione 3.1.2, ci si assicuri di leggere le informazioni relative in initramfs-tools(8), update-initramfs(8), mkinitramfs(8) e initramfs.conf(5).



avvertimento
Quando si compilano i sorgenti del kernel Linux, non mettere collegamenti simbolici alle directory nell'albero dei sorgenti (ad esempio, "`/usr/src/linux*`") in "`/usr/include/linux`" e "`/usr/include/asm`". (Alcuni documenti ormai datati suggeriscono di farlo.)

Nota
Quando si compila il kernel Linux più recente nel sistema Debian `stable`, potrebbe essere necessario l'uso delle versioni backport degli strumenti piu recenti da Debian `unstable`.
module-assistant(8) (o la sua forma abbreviata `m-a`) aiuta gli utenti a compilare e installare pacchetti di moduli facilmente per uno o più kernel personalizzati.
Il DKMS (Dynamic Kernel Module Support, supporto dinamico per i moduli del kernel) è un nuova infrastruttura indipendente dalla distribuzione progettata per permettere l'aggiornamento di singoli moduli del kernel senza cambiare tutto il kernel. È utilizzata per il mantenimento dei moduli esterni all'albero dei sorgenti. Rende anche molto facile la ricompilazione dei moduli quando si aggiornano i kernel.

9.10.4 Compilare i sorgenti del kernel: il metodo raccomandato dal Team del Kernel di Debian

Per compilare pacchetti binari del kernel personalizzati a partire dai sorgenti originali del kernel, si deve usare il target "`deb-pkg` fornito.

```
$ sudo apt-get build-dep linux
$ cd /usr/src
$ wget https://mirrors.edge.kernel.org/pub/linux/kernel/v7.x/linux-7.2.8.tar.xz
$ tar --xz -xvf linux-7.2.8.tar.xz
$ cd linux-7.2.8
$ cp /boot/config-7.2.8.config
$ make menuconfig
 ...
$ make deb-pkg
```

Suggerimento
Il pacchetto linux-source-*versione* fornisce i sorgenti del kernel Linux con le patch Debian come "`/usr/src/linux-`*`versione`*`.tar.bz2`".

Per compilare pacchetti binari specifici a partire dal pacchetto Debian dei sorgenti del kernel, si devono usare i target "`binary-arch_`*`architettura_featureset_flavour`*" in "`debian/rules.gen`".

```
$ sudo apt-get build-dep linux
$ apt-get source linux
$ cd linux-7.*
$ fakeroot make -f debian/rules.gen binary-arch_i386_none_686
```

Ulteriori informazioni:

- Wiki Debian: FAQ del Kernel
- Wiki Debian: Kernel Debian
- Debian Linux Kernel Handbook: https://kernel-handbook.debian.net

9.10.5 Driver per hardware e firmware

Un driver hardware è il codice eseguito nelle CPU principali del sistema target. La maggior parte dei driver hardware è ora disponibile come software libero ed è inclusa nei normali pacchetti Debian dei kernel nell'area `main`.

- driver GPU
  - driver GPU Intel (`main`)
  - driver GPU AMD/ATI (`main`)
  - driver GPU NVIDIA (`main` per il driver nouveau e `non-free` per i driver solo binari supportati dal produttore.

Il firmware è il codice o i dati caricati sul dispositivo attaccato al sistema target (ad esempio microcodice della CPU, codice di rendering eseguito nella GPU oppure dati FPGA/CPLD, …). Alcuni pacchetti firmware sono disponibili come software libero, ma molti pacchetti firmware non lo sono dato che contengono dati binari senza sorgenti. L'installazione di questi dati del firmware è necessaria affinché il dispositivo funzioni come atteso.

- I pacchetti dei dati del firmware contengono dati caricati nella memoria volatile nel dispositivo obiettivo.
  - `firmware-linux-free` (`main`)

  - `firmware-linux-nonfree` (`non-free-firmware`)
  - `firmware-linux-*` (`non-free-firmware`)
  - `*-firmware` (`non-free-firmware`)
  - `intel-microcode` (`non-free-firmware`)
  - `amd64-microcode` (`non-free-firmware`)
- I pacchetti dei programmi di aggiornamento del firmware che aggiornano i nella memoria non volatile del dispositivo target.
  - fwupd (`main`): demone per aggiornamento del firmware che scarica dati del firmware da Linux Vendor Firmware Service.
  - gnome-firmware (`main`): frontend GTK per fwupd
  - plasma-discover-backend-fwupd (`main`): frontend Qt per fwupd

Notare che l'accesso ai pacchetti `non-free-firmware` è fornito dai supporti ufficiali di installazione per offrire un'esperienza di installazione funzionale a partire da Debian 12 Bookworm. La sezione `non-free-firmware` è descritta in Sezione 2.1.5.

Notare anche che i dati di firmware che fwupd scarica da Linux Vendor Firmware Service e che sono caricati nel kernel Linux in esecuzione potrebbero essere `non-free`.

9.11 Sistema virtualizzato

L'uso di sistemi virtualizzati permette di eseguire più istanze di un sistema simultaneamente su un singolo hardware.

Suggerimento
Vedere wiki Debian: SystemVirtualization.

9.11.1 Strumenti per virtualizzazione ed emulazione

Ci sono diversi piattaforme di strumenti per la virtualizzazione e l'emulazione.

- Pacchetti per l'emulazione hardware completa come quelli installati dal metapacchetto games-emulator.
- Emulazione per lo più a livello di CPU con alcune emulazioni di dispositivi di I/O come QEMU.
- Virtualizzazione per lo più a livello di CPU con alcune emulazioni di dispositivi di I/O come macchine virtuali basate sul Kernel (KVM).
- Virtualizzazione in contenitori a livello di SO con supporto a livello del kernel come LXC (Linux Containers), Docker, systemd-nspawn(1), ...
- Virtualizzazione dell'accesso al file system a livello di SO con override del percorso dei file nelle chiamate di libreria di sistema come chroot.
- Virtualizzazione dell'accesso al file system a livello di SO con override della proprietà dei file nelle chiamate di libreria di sistema come chroot.
- Emulazione delle API del SO come Wine
- Virtualizzazione a livello di interprete con selezione del suo eseguibile e override della libreria a runtime come virtualenv e venv per Python.

La virtualizzazione in contenitori usa Sezione 4.7.5 ed è la tecnologia che fa da backend per Sezione 7.7.

Ecco alcuni pacchetti che aiutano a configurare un sistema virtualizzato.

Vedere l'articolo di Wikipedia Comparison of platform virtual machines per una comparazione dettagliata di diverse soluzioni per la virtualizzazione di piattaforme.

pacchetto	popcon	dimensione	descrizione
coreutils	V:912, I:1000	17994	utilità principali di GNU che contengono chroot(8)
util-linux	V:913, I:1000	4826	utilità di sistema varie che contengono unshare(1)
systemd-container	V:74, I:77	2906	strumenti systemd container/nspawn che contengono systemd-nspawn(1)
schroot	V:7.2, I:9.1	2222	strumento specializzato per eseguire pacchetti Debian binari in chroot
sbuild	V:3.9, I:6.9	155	strumento per compilare pacchetti Debian binari da sorgenti Debian
debootstrap	V:4, I:44	332	avviare un sistema Debian base (scritto in sh)
mmdebstrap	V:8, I:12	592	avviare un sistema Debian (scritto in Perl)
cdebootstrap	V:0.1, I:1.3	114	avviare un sistema Debian (scritto in C)
cloud-image-utils	V:1, I:14	52	utilità di gestione di immagini cloud
cloud-guest-utils	V:5, I:23	67	utilità cloud ospiti
virt-manager	V:12, I:48	2309	Virtual Machine Manager: applicazione desktop per gestire macchine virtuali
libvirt-clients	V:45, I:71	1168	programmi per la libreria libvirt
docker.io	V:50, I:53	80705	docker: runtime per contenitori Linux
podman	V:34, I:38	85864	podman: motore per eseguire contenitori basati su OCI in Pod
podman-docker	V:2.8, I:3.4	271	motore per eseguire contenitori basati su OCI in Pod - wrapper per docker
incus	V:0.8, I:3.4	24	Incus: contenitore per sistemi e gestore di macchine virtuali
games-emulator	I:0.19	21	games-emulator: emulatori di Debian per i giochi
bochs	V:0.04, I:0.59	8469	Bochs: emulatore PC IA-32
qemu-system	I:20	69	QEMU: binari per emulazione completa del sistema
qemu-user	V:5.6, I:9.3	472213	QEMU: binari per emulazione in spazio utente
qemu-utils	V:14, I:114	12643	QEMU: utilità
qemu-system-x86	V:52, I:98	69257	KVM: virtualizzazione completa su hardware x86 convirtualizzazione assistita da hardware
virtualbox	V:2.9, I:3.7	151079	VirtualBox: soluzione per virtualizzazione i dx86 su i386 e amd64
gnome-boxes	V:1.2, I:6.1	7074	Boxes: semplice applicazione GNOME per accedere a sistemi virtuali
xen-tools	V:0.1, I:1.4	719	strumenti per gestire server virtuali XEN Debian
wine	V:13, I:54	204	Wine: implementazione della API Windows (suite standard)
dosbox	V:1, I:11	2697	DOSBox: emulatore x86 con grafica Tandy/Herc/CGA/EGA/VGA/SVGA, suono e DOS
lxc	V:10, I:13	1681	strumenti in spazio utente per contenitori Linux
python3-venv	V:8, I:159	6	venv per creare ambienti Python virtuali (libreria di sistema)
python3-virtualenv	V:7, I:37	374	virtualenv per creare ambienti Python virtuali isolati
pipx	V:8, I:50	4412	pipx per installare applicazioni Python in ambienti isolati

Tabella 9.27: Elenco di strumenti di virtualizzazione

9.11.2 Fasi del processo di virtualizzazione

Nota
I kernel Debian predefiniti hanno il supporto per KVM a partire da `lenny`.

Il tipico processo di virtualizzazione comporta diverse fasi.

- Creare un file system vuoto (un albero di file o un'immagine di disco).
  - L'albero di file può essere creato con "`mkdir -p /percorso/di/chroot`".
  - Il file immagine raw del disco può essere creato con dd(1) (vedere Sezione 9.7.1 e Sezione 9.7.5).
  - Si può usare qemu-img(1) per creare e convertire file immagine di dischi supportati da QEMU.
  - I formati di file raw e VMDK possono essere usati come formati comuni tra i vari strumenti di virtualizzazione.
- Montare l'immagine del disco con mount(8) nel file system (opzionale).
  - Per il file immagine raw sel disco, montarlo come device loop o device mapper (vedere Sezione 9.7.3).
  - Per le immagini disco supportate da QEMU, montarle come device a blocchi di rete (vedere Sezione 9.11.3).
- Popolare il file system obiettivo con i dati di sistema necessari.
  - L'uso di programmi come `debootstrap` e `cdebootstrap` aiuta questo processo (vedere Sezione 9.11.4).
  - Usare gli installatori di sistemi operativi nell'emulazione di sistemi completi.
- Eseguire un programma in un ambiente virtualizzato.
  - chroot fornisce un ambiente di virtualizzazione base sufficiente a compilare programmi, eseguire applicazioni in console ed eseguire demoni al suo interno.
  - QEMU fornisce emulazione di CPU inter-piattaforma.
  - QEMU con KVM fornisce una completa emulazione di sistema con virtualizzazione assistita da hardware.
  - VirtualBox fornisce una completa emulazione del sistema in i386 e amd64 con o senza virtualizzazione assistita da hardware.

9.11.3 Montare il file immagine di disco virtuale

Per i file immagine raw di disco, vedere Sezione 9.7.

Per altri file immagine di dischi virtuali, si può usare qemu-nbd(8) per esportarli usando il protocollo per device a blocchi di rete e montarli usando il modulo `nbd` del kernel.

qemu-nbd(8) supporta i formati di disco supportati da QEMU: raw, qcow2, qcow, vmdk, vdi, bochs, cow (copy-on-write di user-mode Linux), parallels, dmg, cloop, vpc, vvfat (VFAT virtuale) e host_device.

I device a blocchi di rete possono supportare partizioni nello stesso modo dei device loop (vedere Sezione 9.7.3). Si può montare la prima partizione di "`disk.img`" nel modo seguente.

```
# modprobe nbd max_part=16
# qemu-nbd -v -c /dev/nbd0 disk.img
...
# mkdir /mnt/part1
# mount /dev/nbd0p1 /mnt/part1
```

Suggerimento
È possibile esportare solamente la prima partizione di "`disk.img`" usando l'opzione "`-P 1`"per qemu-nbd(8).

9.11.4 Sistema chroot

Se si desidera provare un nuovo ambiente Debian da una console in terminale, è raccomandato usare chroot. Ciò permette di eseguire applicazioni in console di Debian `unstable` e `testing` senza i consueti rischi associati e senza riavviare. chroot(8) è il metodo più basilare.



Attenzione
Gli esempi seguenti presuppongono che entrambi i sistemi, quello genitore e quello chroot, condividano la stessa architettura `amd64` della CPU.

Sebbene sia possibile creare manualmente un ambiente chroot(8) usando debootstrap(1), ciò richiede un impegno non banale.

Il pacchetto sbuild per compilare pacchetti Debian da sorgenti usa l'ambiente chroot gestito dal pacchetto schroot. Viene fornito con uno script ausiliario sbuild-createchroot(1). Vediamo come funziona eseguendolo nel modo seguente.

```
$ sudo mkdir -p /srv/chroot
$ sudo sbuild-createchroot -v --include=eatmydata,ccache unstable /srv/chroot/unstable- ↩
    amd64-sbuild http://deb.debian.org/debian
 ...
```

Si può vedere come debootstrap(8) popoli i dati di sistema per l'ambiente `unstable` sotto "`/srv/chroot/unstable-amd6` per un sistema di compilazione minimale.

Si può fare il login in questo ambiente usando schroot(1).

```
$ sudo schroot -v -c chroot:unstable-amd64-sbuild
```

Si può vedere come venga creata una shell di sistema in esecuzione nell'ambiente `unstable`.

Nota
Il file "`/usr/sbin/policy-rc.d`" che termina sempre con 101 evita che, in un sistema Debian, i programmi demone vengano avviati automaticamente. Vedere "`/usr/share/doc/init-system-helpers/README.policy-rc.d.gz`".

Nota
Alcuni programmi in chroot per funzionare possono aver bisogno dell'accesso a più file nel sistema genitore di quanti ne fornisca `sbuild-createchroot`. Per esempio, può essere necessario montare con bind o copiare "`/sys`", "`/etc/passwd`", "`/etc/group`", "`/var/run/utmp`", "`/var/log/wtmp`", ecc.

Suggerimento
Il pacchetto `sbuild` aiuta a costruire un sistema chroot e compila un pacchetto dentro la chroot usando `schroot` come suo backend. È un sistema ideale per controllare le dipendenze di compilazione. Vedere ulteriori dettagli su sbuild nel wiki Debian e l'esempio di configurazione di sbuild nella "Guida per il Manutentore Debian".

Suggerimento
Il comando systemd-nspawn(1) aiuta ad eseguire un comando o un sistema operativo in un contenitore leggero in modo simile a `chroot`. È più potente dato che usa gli spazi dei nomi per virtualizzare completamente l'albero dei processi, IPC, nome host, nome di dominio e, opzionalmente, la rete e i database degli utenti. Vedere systemd-nspawn.

9.11.5 Sistemi desktop multipli

Se si desidera provare un nuovo ambiente desktop con GUI per qualsiasi SO, l'autore raccomanda l'uso di QEMU o KVM in un sistema Debian `stable` per eseguire sistemi desktop multipli in maniera sicura usando la virtualizzazione. Permettono di eseguire qualsiasi applicazione desktop, incluse quelle di Debian `unstable` e `testing`, senza i consueti rischi ad esse associati e senza riavviare.

Dato che QEMU puro è molto lento, è raccomandata la sua velocizzazione con KVM quando il sistema host lo permette.

Virtual Machine Manager noto anche come `virt-manager` è un comodo strumento GUI per gestire macchine virtuali KVM con libvirt.

L'immagine disco virtuale "`virtdisk.qcow2`" contenente un sistema Debian per QEMU si può creare con i piccoli CD dell'installatore Debian nel modo seguente.

```
$ wget https://cdimage.debian.org/debian-cd/5.0.3/amd64/iso-cd/debian-503-amd64-netinst.iso
$ qemu-img create -f qcow2 virtdisk.qcow2 5G
$ qemu -hda virtdisk.qcow2 -cdrom debian-503-amd64-netinst.iso -boot d -m 256
...
```

Suggerimento
Eseguire altre distribuzioni GNU/Linux come Ubuntu e Fedora in una virtualizzazione è un ottimo metodo per imparare trucchetti di configurazione. Anche altri sistemi operativi proprietari possono essere eseguiti tranquillamente in queste virtualizzazioni GNU/Linux.

Vedere ulteriori suggerimenti in wiki Debian: SystemVirtualization.

Capitolo 10

Gestione dei dati

In questo capitolo sono descritti strumenti e trucchi per gestire dati binari e di testo in un sistema Debian.

10.1 Condividere, copiare ed archiviare



avvertimento
L'accesso in scrittura non coordinato a device a cui si sta attivamente accedendo e a file da parte di processi diversi deve essere evitato per prevenire le race condition. Per evitare ciò devono essere usati i meccanismi di lock dei file utilizzando flock(1).

La sicurezza dei dati e la loro condivisione controllata hanno diversi aspetti.

- La creazione di archivi di dati
- L'accesso ad archivi remoti
- La duplicazione
- Il tenere traccia della cronologia delle modifiche
- La facilitazione della condivisione dei dati
- Il prevenire l'accesso non autorizzato ai file
- La rilevazione di modifiche non autorizzate ai file

Queste azioni possono essere realizzate usando una combinazione di strumenti.

- Strumenti di archiviazione e compressione
- Strumenti di copia e sincronizzazione
- file system di rete
- Supporti di archiviazione removibili
- Secure Shell
- Il sistema di autenticazione
- Strumenti per sistemi di controllo delle versioni
- Strumenti crittografici per hash e cifratura

10.1.1 Strumenti di archiviazione e compressione

Ecco una tabella riassuntiva degli strumenti di archiviazione e compressione disponibili per il sistema Debian.



avvertimento
Non impostare la variabile "$TAPE" a meno che non si sappia esattamente cosa aspettarsi. Cambia il comportamento di tar(1).

- Gli archivi tar(1) compressi con gzip usano l'estensione di file "`.tgz`" o "`.tar.gz`".
- Gli archivi tar(1) compressi con xz usano l'estensione di file "`.txz`" o "`.tar.xz`".
- La popolarità dei metodi di compressione negli strumenti FOSS come tar(1) è cambiata nel tempo nel modo seguente `gzip` → `bzip2` → `xz`
- cp(1), scp(1) e tar(1) possono avere alcune limitazioni per file speciali. cpio(1) è più versatile.
- cpio(1) è progettato per essere usato con find(1) ed altri comandi adatti per creare script di backup, dato che la porzione di selezione dei file dello script può essere testata in modo autonomo.
- La struttura interna dei file di dati di LibreOffice è quella dei file «`.jar`» che può essere aperta anche da `unzip`.
- Lo strumento multipiattaforma di fatto usato per gli archivi è `zip`. Usarlo come «`zip -rX`» per ottenere la massima compatibilità. Usare anche l'opzione «`-s`» se è importante la dimensione massima dei file.

10.1.2 Strumenti di copia e sincronizzazione

Ecco una tabella riassuntiva dei semplici strumenti di copia e backup disponibili in un sistema Debian.

La copia dei file con rsync(8) offre un insieme di funzionalità più ricco di altri strumenti.

- l'algoritmo delta-transfer, che invia solamente le differenze tra il file sorgente ed il file esistente nella destinazione
- algoritmo veloce di verifica (predefinito) che cerca i file la cui dimensione o il cui orario di ultima modifica sono cambiati
- opzioni "`--exclude`" e "`--exclude-from`" simili a quelle di tar(1)
- sintassi con "una barra / alla fine della directory sorgente" che evita di dover creare un livello aggiuntivo di directory nella destinazione

Suggerimento
Gli strumenti di controllo delle versioni VCS (Version control system) in Tabella 10.14 possono essere usati come strumenti di copia e sincronizzazione multidirezionali.

10.1.3 Esempi di invocazione per archivi

Ecco diversi modi di archiviare ed estrarre archivi con l'intero contenuto della directory "`./sorgente`", usando diversi strumenti.

GNU tar(1):

```
$ tar -cvJf archive.tar.xz ./source
$ tar -xvJf archive.tar.xz
```

pacchetto	popcon	dimensione	estensione	comando	commento
tar	V:907, I:1000	3101	.tar	tar(1)	strumento di archiviazione standard (standard de facto)
cpio	V:352, I:999	1201	.cpio	cpio(1)	strumento di archiviazione Unix in stile System V, da usare con find(1)
binutils	V:191, I:631	1131	.ar	ar(1)	strumento di archiviazione per la creazione di librerie statiche
fastjar	V:1.0, I:9.5	182	.jar	fastjar(1)	strumento di archiviazione per Java (simile a zip)
pax	V:5.1, I:9.0	167	.pax	pax(1)	nuovo strumento POSIX di archiviazione, compromesso tra tar e cpio
gzip	V:895, I:1000	262	.gz	gzip(1), zcat(1), …	utilità GNU di compressione LZ77 (standard de facto)
bzip2	V:163, I:973	113	.bz2	bzip2(1), bzcat(1), …	utilità per compressione con ordinamento dei blocchi Burrows-Wheeler con maggiore rapporto di compressione di gzip(1) (più lenta di gzip con sintassi simile)
lzma	V:0.6, I:9.6	349	.lzma	lzma(1)	utilità di compressione LZMA con maggiore rapporto di compressione di gzip(1) (deprecata)
xz-utils	V:310, I:981	1551	.xz	xz(1), xzdec(1), …	utilità di compressione XZ con maggiore rapporto di compressione di bzip2(1) (più lenta di gzip, ma più veloce di bzip2; sostituto dell'utilità di compressione LZMA)
zstd	V:334, I:830	2308	.zstd	zstd(1), zstdcat(1), …	utilità di compressione Zstandard veloce senza perdita di dati
p7zip	V:5, I:172	8	.7z	7zr(1), p7zip(1)	strumento di archiviazione file 7-Zip con alto rapporto di compressione (compressione LZMA)
p7zip-full	V:17, I:199	12	.7z	7z(1), 7za(1)	strumento di archiviazione file 7-Zip con alto rapporto di compressione (compressione LZMA e altre)
lzop	V:14, I:129	164	.lzo	lzop(1)	utilità di compressione LZO con velocità di compressione e decompressione più alta di quella di gzip(1) (più basso rapporto di compressione di gzip con sintassi simile)
zip	V:48, I:355	635	.zip	zip(1)	InfoZIP: strumento di archiviazione e compressione per DOS
unzip	V:123, I:744	399	.zip	unzip(1)	InfoZIP: strumento di estrazione di archivi e decompressione per DOS

Tabella 10.1: Elenco di strumenti di archiviazione e compressione

pacchetto	popcon	dimensione	strumento	funzione
coreutils	V:912, I:1000	17994	GNU cp	copia file e directory localmente (”-a” per modalità ricorsiva)
openssh-client	V:682, I:997	3611	scp	copia file e directory da remoto (client, ”-r” per modalità ricorsiva)
openssh-server	V:774, I:820	3615	sshd	copia file e directory da remoto (server remoto)
rsync	V:180, I:539	967		sincronizzazione e backup in remoto unidirezionale
unison	V:3, I:12	14		sincronizzazione e backup in remoto bidirezionale

Tabella 10.2: Elenco di strumenti di copia e sincronizzazione

In alternativa usare i comandi seguenti.

```
$ find ./source -xdev -print0 | tar -cvJf archive.tar.xz --null -T -
```

cpio(1):

```
$ find ./source -xdev -print0 | cpio -ov --null > archive.cpio; xz archive.cpio
$ zcat archive.cpio.xz | cpio -i
```

10.1.4 Esempi di invocazione per la copia

Ecco diversi modi di copiare l’intero contenuto della directory ”`./sorgente`”, usando diversi strumenti.

- Copia locale: directory ”`./sorgente`” → directory ”`/dest`”
- Copia remota: directory ”`./sourgente`” sull’host locale → directory ”`/dest`” sull’host ”`utente@host.dom`”

rsync(8):

```
# cd ./source; rsync -aHAXSv . /dest
# cd ./source; rsync -aHAXSv . user@host.dom:/dest
```

In alternativa si può usare la sintassi con ”una barra / alla fine della directory sorgente”.

```
# rsync -aHAXSv ./source/ /dest
# rsync -aHAXSv ./source/ user@host.dom:/dest
```

In alternativa usare i comandi seguenti.

```
# cd ./source; find . -print0 | rsync -aHAXSv0 --files-from=- . /dest
# cd ./source; find . -print0 | rsync -aHAXSv0 --files-from=- . user@host.dom:/dest
```

GNU cp(1) e openSSH scp(1):

```
# cd ./source; cp -a . /dest
# cd ./source; scp -pr . user@host.dom:/dest
```

GNU tar(1):

```
# (cd ./source && tar cf - . ) | (cd /dest && tar xvfp - )
# (cd ./source && tar cf - . ) | ssh user@host.dom '(cd /dest && tar xvfp - )'
```

cpio(1):

```
# cd ./source; find . -print0 | cpio -pvdm --null --sparse /dest
```

In tutti gli esempi contenenti "`.`", si può sostituire "`.`" con "`pippo`" per copiare i file dalla directory "`./sorgente/pippo`" alla directory "`/dest/pippo`".

In tutti gli esempi contenenti "`.`", si può sostituire "`.`" con il percorso assoluto "`/percorso/di/sorgente/pippo`" per evitare di fare "`cd ./sorgente;`". I file verranno copiati in posizioni diverse a seconda dello strumento utilizzato, come descritto qui di seguito.

- in "`/dest/pippo`": rsync(8), GNU cp(1) e scp(1)
- "`/dest/percorso/di/sorgente/pippo`": GNU tar(1) e cpio(1)

Suggerimento
rsync(8) e GNU cp(1) hanno l'opzione "`-u`" per saltare i file che sono più recenti nella destinazione.

10.1.5 Esempi di invocazione per la selezione di file

find(1) viene usato per selezionare i file per i comandi di archiviazione e copia (vedere Sezione 10.1.3 e Sezione 10.1.4) o per xargs(1) (vedere Sezione 9.4.9). Questo funzionamento può essere migliorato usando le sue opzioni di comando.

La sintassi base di find(1) può essere riassunta nel modo seguente.

- Gli argomenti condizionali sono valutati da sinistra a destra.
- Questa valutazione si ferma una volta che il risultato è determinato.
- L'operatore "**OR** logico" (specificato con "`-o`" tra condizioni) ha una precedenza più bassa dell'operatore "**AND** logico" (specificato da "`-a`" o dall'assenza di un operatore tra condizioni).
- L'operatore "**NOT** logico" (specificato da "`!`" prima di una condizione) ha una precedenza più alta di un operatore "**AND** logico".
- L'opzione "`-prune`" restituisce sempre una condizione di **VERO** logico e, se si tratta di una directory, la ricerca si ferma a questo punto.
- L'opzione "`-name` trova corrispondenze con il nome base del file tramite espressioni glob di shell (vedere Sezione 1.5.6), ma fa corrispondenza anche con il carattere iniziale "`.`" con l'uso di metacaratteri come "`*`" e "`?`". (Nuova funzionalità POSIX.)
- L'opzione "`-regex`" trova corrispondenze con il percorso completo usando, in modo predefinito, **BRE** in stile emacs (vedere Sezione 1.6.2).
- L'opzione "`-size`" trova corrispondenze con file in base alla loro dimensione (valori preceduti da "`+`" o "`-`" per cercare dimensioni, rispettivamente, più grandi o piccole del valore).
- L'opzione "`-newer`" trova corrispondenze con file più recenti di quello specificato come argomento dell'opzione.
- L'opzione "`-print0`" restituisce sempre il valore logico **VERO** e stampa il nome file completo (terminato dal carattere null) sullo standard output.

find(1) è spesso usato con uno stile di invocazione come il seguente.

```
# find /path/to \
    -xdev -regextype posix-extended \
    -type f -regex ".*\.cpio|.*~" -prune -o \
    -type d -regex ".*/\.git" -prune -o \
    -type f -size +99M -prune -o \
    -type f -newer /path/to/timestamp -print0
```

Il comando precedente si traduce nelle azioni seguenti.

1. Cercare tutti i file a partire da "`/percorso/di`"
2. Limitare globalmente la ricerca al file system da cui è richiamato e usare espressioni regolari **ERE** (vedere Sezione 1.6.2)
3. Escludere i file che corrispondono all'espressione regolare "`.*\.cpio`" o "`.*~`" dalla ricerca fermando la loro elaborazione
4. Escludere le directory che corrispondono all'espressione regolare "`.*/\.git`" dalla ricerca fermando la loro elaborazione
5. Escludere i file più grandi di 99 Megabyte (unità di 1048576 byte) dalla ricerca fermando la loro elaborazione
6. Stampare i nomi di file che soddisfano le condizioni di ricerca precedenti e che siano più recenti di "`/percorso/di/mar`

Notare nell'esempio precedente l'uso della parte di comando "`-prune -o` per escludere file.

Nota
Alcune opzioni per find(1) potrebbero non essere supportate per i sistemi *nix non Debian. In questi casi, cambiare le invocazioni in quelle adatte corrispondenti e sostituire "`-print0`" con "`-print`". Potrebbe essere necessario modificare anche comandi correlati.

10.1.6 Supporti di archiviazione

Quando si deve scegliere il supporto di archiviazione di dati informatici per un importante archivio di dati, si dovrebbe porre attenzione alle limitazioni dei supporti. Per piccoli backup di dati personali, io uso CD-R e DVD-R scegliendoli in base alla marca del produtttore e conservandoli in un ambiente fresco, all'ombra, asciutto e pulito. (I supporti di archiviazione a nastro sembrano molto popolari per gli usi professionali.)

Nota
Le casseforti a prova di fuoco sono pensate per i documenti cartacei. La maggior parte dei supporti di archiviazione di dati informatici ha una tolleranza più bassa alle alte temperature rispetto alla carta. Di solito mi affido a copie multiple cifrate sicure conservate in diverse posizioni sicure.

Durata di vita ottimistica di diversi supporti di archiviazione da dati raccolti in rete (per lo più dalle informazioni dei produttori).

- 100+ anni: carta senza acidi con inchiostro
- 100 anni: supporti ottici (CD/DVD, CD/DVD-R)
- 30 anni: supporti magnetici (nastri, dischetti floppy)
- 20 anni: supporti ottici a cambio di fase (CD-RW)

Questi tempi non tengono conto dei danni meccanici causati dal maneggiamento, ecc.

Cicli di scrittura ottimistici di diversi supporti di archiviazione da dati raccolti in rete (per lo più dalle informazioni dei produttori).

- 250,000+ cicli: unità a disco fisso
- 10,000+ cicli: memoria flash

- 1,000 cicli: CD/DVD-RW
- 1 ciclo: CD/DVD-R, carta



Attenzione
I valori di durata di vita e dei cicli di scrittura riportati non dovrebbero essere usati per prendere decisioni riguardo all'archiviazione di dati critici. Consultare le informazioni specifiche per ciascun prodotto forniti dal produttore.

Suggerimento
Dato che i CD/DVD-R e la carta hanno un solo ciclo di scrittura, prevengono per loro stessa natura le perdite accidentali di dati per sovrascrittura. Questo è un vantaggio!

Suggerimento
Se è necessario fare backup frequenti e veloci di una grande quantità di dati, un disco fisso su un host remoto connesso con una connessione veloce, potrebbe essere l'unica soluzione realistica.

Suggerimento
Se si usano supporti riscrivibili per i backup, l'uso di file system come btrfs o zfs che supportano solo istantanee in sola lettura potrebbe essere una buona idea.

10.1.7 Supporti di archiviazione removibili

Un support di archiviazione removibile può essere uno dei seguenti.

- Unità flash USB
- Unità a disco fisso
- Unità a dischi ottici
- Fotocamera digitale
- Lettore audio digitale

Possono essere connessi in uno dei modi seguenti.

- USB
- IEEE 1394 / FireWire

Gli ambienti desktop moderni, come GNOME e KDE, possono montare questi dispositivi removibili automaticamente senza una voce corrispondente in "`/etc/fstab`".

- Il pacchetto `udisks2` fornisce un demone e le utilità associate per montare e smontare questi dispositivi.
- D-bus crea eventi per dare inizio ai processi automatici.
- PolicyKit fornisce i privilegi necessari.

Suggerimento
I dispositivi montati automaticamente hanno l'opzione di mount "`uhelper=`" che viene usata da umount(8).

Suggerimento
Nei moderni ambienti desktop il montaggio automatico avviene solo quando i device dei supporti removibili non sono elencati in "`/etc/fstab`".

Il punto di mount negli ambienti desktop moderni viene scelto come "`/media/`*`nomeutente/etichetta_disco`*" che può essere personalizzato nel modo seguente.

- mlabel(1) per file system FAT
- genisoimage(1) con l'opzione "`-V`" per file system ISO9660
- tune2fs(1) con l'opzione "`-L`" per file system ext2/ext3/ext4

Suggerimento
Può essere necessario fornire come opzione di montaggio la scelta della codifica (vedere Sezione 8.1.3).

Suggerimento
L'uso del menu GUI per smontare un file system può rimuovere il suo nodo di device generato dinamicamente come "`/dev/sdc`". Se si desidera mantenere il suo nodo di device, smontarlo con il comando umount(8) dal prompt di shell.

10.1.8 Scelta del file system per la condivisione di dati

Quando si condividono dati con un altro sistema attraverso dispositivi di archiviazione removibili, quest'ultimi andrebbero formattati con un filesystem comune supportato da entrambi i sistemi. Quello che segue è un elenco delle scelte possibili per il file system.

Nota
Le informazioni date riguardo gli hard disk (HDD) sono applicabili ad altri dispositivi di archiviazione, come SSD / unità flash USB / Schede di memoria / ... . Sostituire negli esempi il nome del dispositivo, ad esempio `/dev/sda` con i nomi di dispositivo effettivi, `/dev/nvme0`, `/dev/mmcblk0`, ...

Suggerimento
Vedere Sezione 9.9.1 per la condivisione interpiattaforma di dati usando cifratura a livello di dispositivo.

Il file system FAT è supportato da quasi tutti i sistemi operativi moderni ed è piuttosto utile per scopi di scambio di dati attraverso supporti come dischi fissi removibili.

Quando si formatta un dispositivo come un disco fisso removibile con il file system FAT per la condivisione interpiattaforma di dati, le scelte seguenti dovrebbero essere quelle più sicure.

- Partizionare con fdisk(8), cfdisk(8) o parted(8) (vedere Sezione 9.6.2) creando un'unica partizione primaria e marcarla nel modo seguente.

nome del file system	scenario di uso tipico
FAT12	condivisione interpiattaforma di dati su dischetti floppy (<32MiB)
FAT16	condivisione interpiattaforma di dati su dispositivi come piccoli dischi fissi (<2GiB)
FAT32	condivisione interpiattaforma di dati su dispositivi come grandi dischi fissi (<8TiB, supportato da sistemi più recenti di MS Windows95 OSR2)
exFAT	condivisione interpiattaforma di dati su dispositivi come grandi dischi fissi (<512TiB, supportato da WindowsXP, Mac OS X Snow Leopard 10.6.5 e kernel Linux sin dal rilascio 5.4.
NTFS	condivisione interpiattaforma di dati su dispositivi come grandi dischi fissi (supportato nativamente su MS Windows NT e versioni successive e supportato da NTFS-3G attraverso FUSE in Linux)
ISO9660	condivisione interpiattaforma di dati statici su CD-R e DVD+/-R
UDF	scrittura incrementale di dati su CD-R e DVD+/-R (nuovo)
MINIX	archiviazione, efficiente in termini di spazio, di file dati unix su dischetti floppy
ext2	condivisione di dati su dispositivi come dischi fissi con sistemi Linux più vecchi
ext3	condivisione di dati su dispositivi come dischi fissi con sistemi Linux più vecchi
ext4	condivisione di dati su dispositivi come dischi fissi con sistemi Linux recenti
btrfs	condivisione di dati su dispositivi come dischi fissi con sistemi Linux recenti con istantanee in sola lettura

Tabella 10.3: Elenco di possibili scelte per il file system di dispositivi di archiviazione removibili con scenari di uso tipici

  - Tipo ”6”, cioè FAT16, per supporti più piccoli di 2GB.
  - Tipo ”c”, cioè FAT32 (LBA), per supporti più grandi.

- Formattare la partizione primaria con mkfs.vfat(8) nel modo seguente.
  - Per FAT16, semplicemente con il suo nome di device, ad esempio ”`/dev/sda1`”.
  - Per FAT32, con il suo nome di device e l’opzione esplicita, ad esempio ”`-F 32 /dev/sda1`”

Quando si usano i file system FAT o ISO9660 per la condivisione dei dati, per essere sicuri dei risultati, si dovrebbero considerare i seguenti aspetti.

- Archiviare prima i file in un file di archivio usando tar(1) o cpio(1) per mantenere i nomi di file lunghi, i collegamenti simbolici, i permessi Unix sui file originali e le informazioni sui proprietari.
- Suddividere il file di archivio in pezzi più piccoli di 2 GiB con il comando split(1) per proteggerli da limitazioni sulla dimensione dei file.
- Cifrare il file archivio per proteggere i suoi contenuti da accesso non autorizzato.

Nota
Il file system FAT, per sua stessa natura, permette una dimensione massima per i file di `(2^32 - 1) byte = (4GiB - 1 byte)`. Per alcune applicazioni su sistemi operativi a 32 bit più vecchi, la dimensione massima per i file era ancora più piccola: `(2^31 - 1) byte = (2GiB - 1 byte)`. Debian non soffre di quest’ultima limitazione.

Nota
La stessa Microsoft non raccomanda l’uso di FAT per le unità o le partizioni più grandi di 200 MB. Microsoft evidenzia le sue limitazioni, quali un uso inefficiente dello spazio su disco, nel documento ”Overview of FAT, HPFS, and NTFS File Systems”. Naturalmente per Linux si dovrebbe normalmente usare il file system ext4.

Suggerimento
Per maggiori informazioni sui file system e sull'accesso ad essi, leggere il "Filesystems HOWTO".

10.1.9 Condividere dati attraverso una rete

Quando si condividono dati con un altro sistema attraverso una rete, si dovrebbero tenere a mente i servizi comuni. Ecco alcuni suggerimenti.

servizio di rete	descrizione dello scenario di uso tipico
file system montato di rete SMB/CIFS con Samba	condivisione di file attraverso "rete Microsoft Windows", vedere smb.conf(5) e The Official Samba 3.x.x HOWTO and Reference Guide o il pacchetto `samba-doc`
file system montato di rete NFS con il kernel Linux	condivisione di file attraverso "rete Unix/Linux", vedere exports(5) e Linux NFS-HOWTO
servizio HTTP	condivisione di file tra client/server web
servizio HTTPS	condivisione di file tra client/server web con SSL (Secure Sockets Layer) cifrato o TLS (Transport Layer Security)
servizio FTP	condivisione di file tra client/server FTP

Tabella 10.4: Elenco dei servizi di rete da scegliere in base allo scenario di uso tipico

Sebbene questi file system montati in rete e metodi di trasferimento di file attraverso la rete siano piuttosto comodi per la condivisione dei dati, possono essere non sicuri. La loro connessione di rete deve essere resa sicura nel modo seguente.

- Cifrarli con SSL/TLS
- Usarli in tunnel SSH
- Usarli in tunnel VPN
- Limitarli dietro ad un firewall sicuro

Vedere anche Sezione 6.5 e Sezione 6.6.

10.2 Backup e ripristino

Tutti sanno che i computer a volte si danneggiano oppure errori umani causano danni al sistema e ai dati. Le operazioni di backup e ripristino sono una parte essenziale di un'amministrazione di sistema di successo. Tutte i possibili modi in cui si possono creare danni si verificano prima o poi.

Suggerimento
Mantenere il proprio sistema di backup semplice e fare il backup di sistema spesso. Avere dati di backup è più importante della qualità tecnica del metodo di backup.

10.2.1 Politica di backup e ripristino

Ci sono 3 fattori chiave che determinano la reale politica di backup e ripristino.

1. Sapere di cosa fare il backup ed il ripristino

- I file dati direttamente creati dall'utente: in "`~/`"
- I file dati creati da applicazioni usate dall'utente: dati in "`/var/`" (tranne "`/var/cache/`", "`/var/run/`" e "`/var/tmp/`")
- File di configurazione del sistema: dati in "`/etc/`"
- Programmi locali: dati in "`/usr/local/`" o "`/opt/`"
- Informazioni di installazione del sistema: un memorandum in puro testo sui passi chiave (partizioni, …)
- Insiemi di dati comprovati: confermati da operazioni preventive sperimentali di ripristino
  - Compito di cron come processo di un utente; file nella directory "`/var/spool/cron/crontabs`" e riavvio di cron(8). Vedere Sezione 9.4.14 per cron(8) e crontab(1).
  - Compiti temporizzati di systemd come processi utente: file nella directory "`~/.config/systemd/user`". Vedere systemd.timer(5) e systemd.service(5).
  - Compiti automatici all'avvio come processi utente: file nella directory "`~/.config/autostart`". Vedere Specifica per l'avvio automatico di applicazioni desktop.

2. Sapere come fare il backup ed il ripristino
   - Rendere sicura l'archiviazione dei dati: protezione da sovrascritture e fallimenti del sistema
   - Backup frequenti: backup pianificati
   - Backup ridondanti: mirror di dati
   - Procedura a prova di idioti: singolo facile comando di backup
3. Valutazione dei rischi e dei costi
   - Rischio della perdita di dati
     - I dati dovrebbere come minimo essere su partizioni del disco diverse, preferibilmente su dischi e macchine diversi per sopravvivere alla corruzione del file system. È meglio archiviare i dati importanti su un file system in sola lettura. 1
   - Rischio dell'accesso non consentito a dati
     - Dati sensibili relativi all'identità dell'utente, come "`/etc/ssh/ssh_host_*_key`", "`~/.gnupg/*`", "`~/.ssh/*`", "`~/.local/share/keyrings/*`", "`/etc/passwd`", "`/etc/shadow`", "`popularity-contest.conf`", "`/etc/ppp/pap-secrets`" e "`/etc/exim4/passwd.client`" dovrebbero essere archiviati in backup cifrati 2 (Vedere Sezione 9.9.)
     - Non inserire mai in modo fisso la password di login nel sistema o la passphrase di decifrazione nel codice di alcun script, nemmeno in un sistema fidato. (Vedere Sezione 10.3.6.)
   - Modi in cui le cose possono andare storte e loro probabilità
     - L'hardware (specialmente gli HDD) si rompe
     - I file system possono diventare corrotti e i dati in essi possono venire persi
     - Non ci si può fidare che i sistemi di archiviazione in remoto non abbiano falle di sicurezza.
     - Una protezione con password deboli può essere facilmente compromessa
     - Il sistema dei permessi dei file può venir compromesso
   - Risorse necessarie per il backup: umane, hardware, software, …
     - Backup automatici pianificati con compiti di cron o temporizzati di systemd

---

Suggerimento

Si possono ripristinare i dati di configurazione di debconf con "`debconf-set-selections debconf-selections`" ed i dati delle selezioni di dpkg con "`dpkg --set-selection <dpkg-selections.list`".

1Supporti scrivibili una sola volta, come CD/DVD-R possono prevenire incidenti di sovrascrittura. (Vedere Sezione 9.8 per come scrivere sul supporto di archiviazione dalla riga di comando della shell. L'ambiente desktop GNOME con interfaccia grafica fornisce un facile accesso tramite menu: "Risorse → Creazione CD/DVD".)

2Alcuni di questi dati non possono essere rigenerati inserendo la stessa stringa di input nel sistema.

Nota
Non fare il backup dei contenuti dei pseudo file system che si trovano in `/proc`, `/sys`, `/tmp` e `/run` (vedere Sezione 1.2.12 e Sezione 1.2.13). A meno di non sapere esattamente ciò che si sta facendo, sono un'enorme mole di dati senza utilità.

Nota
Durante il backup dei dati può essere preferibile fermare alcuni demoni applicativi come l'MTA (vedere Sezione 6.2.4).

10.2.2 Suite con utilità di backup

Quello che segue è un elenco di importanti suite di utilità di backup disponibili in un sistema Debian

pacchetto	popcon	dimensione	descrizione
`bacula-common`	V:6.0, I:6.7	2501	Bacula: backup, ripristino e controllo in rete - file comuni di supporto
`bacula-client`	V:0.2, I:1.8	199	Bacula: backup, ripristino e controllo in rete - metapacchetto client
`bacula-console`	V:0.6, I:2.1	112	Bacula: backup, ripristino e controllo in rete - console testuale
`bacula-server`	I:0.59	199	Bacula: backup, ripristino e controllo in rete - metapacchetto server
`amanda-common`	V:0.6, I:2.1	9951	Amanda: Advanced Maryland Automatic Network Disk Archiver (Librerie)
`amanda-client`	V:0.5, I:2.1	1109	Amanda: Advanced Maryland Automatic Network Disk Archiver (Client)
`amanda-server`	V:0.11, I:0.22	1095	Amanda: Advanced Maryland Automatic Network Disk Archiver (Server)
`backuppc`	V:1.4, I:1.6	3088	BackupPC è un sistema ad altre prestazioni di qualità professionale per il backup di PC (basato su dischi)
`duplicity`	V:5, I:44	2662	backup incrementali (remoti)
`deja-dup`	V:28, I:41	5274	Interfaccia GUI per duplicity
`borgbackup`	V:13, I:28	3532	backup (remoto) con deduplicazione
`borgmatic`	V:3.2, I:4.4	946	strumento ausiliario per borgbackup
`rdiff-backup`	V:2.1, I:6.1	1169	backup incrementali (remoti)
`restic`	V:8, I:15	41617	backup incrementali (remoti)
`backupninja`	V:2.0, I:2.4	360	sistema di **meta-backup** leggero ed estensibile
`slbackup`	V:0.10, I:0.13	147	backup incrementali (remoti)
`backup-manager`	V:0.41, I:0.70	573	strumento di backup a riga di comando
`backup2l`	V:0.35, I:0.46	110	strumento di backup/ripristino per supporti montabile (basato su dischi) che richiede bassa manutenzione

Tabella 10.5: Elenco di suite con utilità di backup

Gli strumenti di backup hanno una propria specializzazione.

- Mondo Rescue è un sistema di backup per facilitare il ripristino veloce di un sistema completo a partire da backup su CD/DVD ecc., senza dover affrontare il normale processo di installazione del sistema.
- Bacula, Amanda e BackupPC sono suite di utilità di backup complete che sono pensate per backup regolari in rete.
- Duplicity e Borg sono utilità per backup più semplici per le postazioni di lavoro tipiche.

10.2.3 Suggerimenti per i backup

Per una postazione di lavoro personale, utilità da suite di backup complete pensate per ambienti server possono non essere adatte. Allo stesso tempo le utilità di backup esistenti per le postazioni di lavoro possono avere alcune limitazioni.

Ecco alcuni suggerimenti per rendere più facili i backup con un minimo sforzo da parte dell'utente. Queste tecniche possono essere utilizzati con qualsiasi utilità di backup.

Per illustrare la cosa, supponiamo che il nome dell'utente principale e del suo gruppo siano `penguin` e si creerà un esempio di script di backup e per istantanea "`/usr/local/bin/bkss.sh`" con:

```
#!/bin/sh -e
SRC="$1" # source data path
DSTFS="$2" # backup destination filesystem path
DSTSV="$3" # backup destination subvolume name
DSTSS="${DSTFS}/${DSTSV}-snapshot" # snapshot destination path
if [ "$(stat -f -c %T "$DSTFS")" != "btrfs" ]; then
  echo "E: $DESTFS needs to be formatted to btrfs" >&2 ; exit 1
fi
MSGID=$(notify-send -p "bkup.sh $DSTSV" "in progress ...")
if [ ! -d "$DSTFS/$DSTSV" ]; then
  btrfs subvolume create "$DSTFS/$DSTSV"
  mkdir -p "$DSTSS"
fi
rsync -aHxS --delete --mkpath "${SRC}/" "${DSTFS}/${DSTSV}"
btrfs subvolume snapshot -r "${DSTFS}/${DSTSV}" ${DSTSS}/$(date -u --iso=min)
notify-send -r "$MSGID" "bkup.sh $DSTSV" "finished!"
```

Qui viene usato solo lo strumento di base rsync(1) per facilitare il backup di sistema e lo spazio di archiviazione viene usato efficientemente da Btrfs.

Suggerimento
Notare che l'autore usa uno script di shell personale simile "bss: Btrfs Subvolume Snapshot Utility" per le sue postazioni di lavoro.

10.2.3.1 Backup con GUI

Ecco un esempio di come impostare un backup con GUI con un unico clic.

- Preparare un dispositivo USB di archiviazione da usare per il backup.
  - Formattare un dispositivo USB di archiviazione con una partizione in btrfs con etichetta chiamata "BKUP". Può essere cifrata (vedere Sezione 9.9.1).
  - Connetterlo al sistema. Il sistema desktop dovrebbe montarlo automaticamente come "`/media/penguin/BKUP`".
  - Eseguire "`sudo chown penguin:penguin /media/penguin/BKUP`" per renderlo scrivibile dall'utente.
- Creare "`~/.local/share/applications/BKUP.desktop`" seguendo le tecniche descritte in Sezione 9.4.10, come:

```
[Desktop Entry]
Name=bkss
Comment=Backup and snapshot of ~/Documents
Exec=/usr/local/bin/bkss.sh /home/penguin/Documents /media/penguin/BKUP Documents
Type=Application
```

Per ogni clic nella GUI, viene fatto il backup dei dati da "`~/Documenti`" in un dispositivo USB di archiviazione e viene creata un'istantanea in sola lettura.

10.2.3.2 Backup attivato da eventi di mount

Ecco un esempio di come impostare un backup automatico attivato da un evento di mount.

- Preparare un dispositivo USB di archiviazione da usare per il backup come in Sezione 10.2.3.1.
- Creare un file con un'unità di servizio di systemd "`~/.config/systemd/user/back-BKUP.service`", come:

```
[Unit]
Description=USB Disk backup
Requires=media-%u-BKUP.mount
After=media-%u-BKUP.mount

[Service]
ExecStart=/usr/local/bin/bkss.sh %h/Documents /media/%u/BKUP Documents
StandardOutput=append:%h/.cache/systemd-snap.log
StandardError=append:%h/.cache/systemd-snap.log

[Install]
WantedBy=media-%u-BKUP.mount
```

- Abilitare questa configurazione con unità di systemd usando:

```
$ systemctl --user enable bkup-BKUP.service
```

Per ogni evento di mount, viene fatto il backup dei dati da "`~/Documenti`" in un dispositivo USB di archiviazione e viene creata un'istantanea in sola lettura.

Si possono chiedere i nomi delle unità di mount che systemd ha attualmente in memoria al gestore del servizio dell'utente in uso con "`systemctl --user list-units --type=mount`".

10.2.3.3 Backup attivato da un evento timer

Ecco un esempio di come impostare un backup automatico attivato da un evento di mount.

- Preparare un dispositivo USB di archiviazione da usare per il backup come in Sezione 10.2.3.1.
- Creare un file con un'unità timer di systemd "`~/.config/systemd/user/snap-Documents.timer`", come:

```
[Unit]
Description=Run btrfs subvolume snapshot on timer
Documentation=man:btrfs(1)

[Timer]
OnStartupSec=30
OnUnitInactiveSec=900

[Install]
WantedBy=timers.target
```

- Creare un file di unità di servizio di systemd "`~/.config/systemd/user/snap-Documents.service`", come:

```
[Unit]
Description=Run btrfs subvolume snapshot
Documentation=man:btrfs(1)

[Service]
Type=oneshot
Nice=15
```

```
ExecStart=/usr/local/bin/bkss.sh %h/Documents /media/%u/BKUP Documents
IOSchedulingClass=idle
CPUSchedulingPolicy=idle
StandardOutput=append:%h/.cache/systemd-snap.log
StandardError=append:%h/.cache/systemd-snap.log
```

- Abilitare questa configurazione con unità di systemd usando:

```
$ systemctl --user enable snap-Documents.timer
```

Per ogni evento timer, viene fatto il backup dei dati da "`~/Documenti`" in un dispositivo USB di archiviazione e viene creata un'istantanea in sola lettura.

Si possono chiedere i nomi delle unità timer che systemd ha attualmente in memoria al gestore del servizio dell'utente in uso con "`systemctl --user list-units --type=timer`".

Per i sistemi desktop moderni questo approccio che usa systemd può offrire un controllo più dettagliato rispetto a quelli Unix tradizionali che usano at(1), cron(8) o anacron(8).

10.3 Infrastruttura di sicurezza dei dati

L'infrastruttura di sicurezza dei dati viene fornita dalla combinazione di strumenti di cifratura dei dati, strumenti message digest e strumenti di firma.

pacchetto	popcon	dimensione	comando	descrizione
gnupg	V:418, I:860	465	gpg(1)	GNU Privacy Guard - strumento OpenPGP di cifratura e firma
gpgv	V:203, I:923	577	gpgv(1)	GNU Privacy Guard - strumento OpenPGP di verifica delle firme
sq	V:1, I:33	24447	sq(1)	Sequoia-PGP - strumento OpenPGP di cifratura e firma
sqv	V:479, I:554	1919	sqv(1)	Sequoia-PGP - OpenPGP signature Sequoia-PGP - strumento OpenPGP di verifica delle firme
paperkey	V:1, I:12	58	paperkey(1)	estrae solamente le informazioni segrete da chiavi OpenPGP segrete
cryptsetup	V:34, I:77	469	cryptsetup(8), …	utilità per la cifratura dm-crypt per i device a blocchi con supporto di LUKS
coreutils	V:912, I:1000	17994	md5sum(1)	calcola e controlla message digest MD5
coreutils	V:912, I:1000	17994	sha1sum(1)	calcola e controlla message digest SHA1
openssl	V:846, I:996	2564	openssl(1ssl)	calcola message digest con "`openssl dgst`" (OpenSSL)
libsecret-tools	V:2, I:13	44	secret-tool(1)	archivia e recupera password (CLI)
seahorse	V:76, I:260	7999	seahorse(1)	strumento di gestione delle chiavi (GNOME)

Tabella 10.6: Elenco di strumenti per l'infrastruttura di sicurezza dei dati

Vedere Sezione 9.9 su dm-crypt e fscrypt che implementano un'infrastruttura di cifratura automatica dei dati usando moduli del kernel Linux.

10.3.1 Gestione delle chiavi per GnuPG

Quelli che seguono sono alcuni comandi per GNU Privacy Guard per la gestione base delle chiavi.

comando	descrizione
`gpg --gen-key`	Genera una nuova chiave
`gpg --gen-revoke mio_ID_utente`	genera una chiave di revoca per mio_ID_utente
`gpg --edit-key user_ID`	modifica la chiave in modo interattivo, ”help” per l’aiuto
`gpg -o file --export`	esporta tutte le chiavi pubbliche in un file
`gpg -o file --export-secret-keys`	esporta tutte le chiavi private in un file
`gpg --import file`	importa tutte le chiavi da un file
`gpg --send-keys ID_utente`	invia la chiave di ID_utente al server di chiavi
`gpg --recv-keys ID_utente`	riceve la chiave di ID_utente dal server di chiavi
`gpg --list-keys ID_utente`	elenca le chiavi di ID_utente
`gpg --list-sigs ID_utente`	elenca le firme di ID_utente
`gpg --check-sigs ID_utente`	controlla le firme di ID_utente
`gpg --fingerprint ID_utente`	controlla le impronte digitali di ID_utente
`gpg --refresh-keys`	aggiorna il portachiavi locale

Tabella 10.7: Elenco di comandi per GNU Privacy Guard per la gestione delle chiavi

Quelli seguenti sono i significati dei codici di fiducia.

codice	descrizione della fiducia
`-`	nessuna fiducia assegnata dal proprietario / ancora non calcolata
`e`	calcolo della fiducia fallito
`q`	informazioni insufficienti per il calcolo
`n`	non fidarsi mai di questa chiave
`m`	marginalmente affidabile
`f`	completamente fidata
`u`	definitivamente fidata

Tabella 10.8: Elenco dei significati dei codici di fiducia

Il comando seguente genera la mia chiave ”`9563FC29932C409F1A77F9C1AB8A1522D8234C6A`”.

```
$ gpg --gen-key
gpg (GnuPG) 2.4.7; Copyright (C) 2024 g10 Code GmbH
...
GnuPG needs to construct a user ID to identify your key.

Real name: Osamu Aoki
Email address: osamu@debian.org
You selected this USER-ID:
    "Osamu Aoki <osamu@debian.org>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
...
public and secret key created and signed.

pub   ed25519 2026-02-14 [SC] [expires: 2029-02-13]
      9563FC29932C409F1A77F9C1AB8A1522D8234C6A
uid                      Osamu Aoki <osamu@debian.org>
sub   cv25519 2026-02-14 [E] [expires: 2029-02-13]
```

Il comando seguente carica la mia chiave "9563FC29932C409F1A77F9C1AB8A1522D8234C6A" sul popolare server di chiavi "hkp://keys.gnupg.net".

```
$ gpg --keyserver hkp://keys.gnupg.net --send-keys 9563FC29932C409F1A77F9C1AB8A1522D8234C6A
```

Un buon server di chiavi predefinito impostato in "~/.gnupg/gpg.conf" (o nella vecchia posizione "~/.gnupg/options") si ottiene la voce seguente.

```
keyserver hkp://keys.gnupg.net
```

Il comando seguente recupera le chiavi sconosciute dal server di chiavi.

```
$ gpg --list-sigs --with-colons | grep '^sig.*\[User ID not found\]' |\
          cut -d ':' -f 5| sort | uniq | xargs gpg --recv-keys
```

In OpenPGP Public Key Server (versioni pre-0.9.6) esisteva un bug che corrompeva le chiavi con più di 2 sottochiavi. I pacchetti GnuPG più recenti (>1.2.1-2) possono gestire queste sottochiavi corrotte. Vedere l'opzione "--repair-pks-subkey-bug in gpg(1).

L'uso di SHA-1 per gli hash è stato deprecato. Se la vostra vecchia chiave openPGP basata su RSA usa SHA-1 per l'hash, correggerla usando FixKeyWithSha1.

Suggerimento
I comandi sq(1) e sqv(1) sono un insieme alternativo di comandi openPGP. Vedere la documentazione utente di sq e A Practical Introduction to using sq, Sequoia PGP's CLI.

10.3.2 Usare GnuPG su file

Quelli seguenti sono esempi di comandi per usare GNU Privacy Guard su file.

10.3.3 Usare GnuPG con Mutt

Aggiungere quanto seguie al file "~/.muttrc" per evitare che il lento GnuPG venga avviato automaticamente, permettendo allo stesso tempo di richiamarlo digitando "S" nel menu della vista indice.

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

10.3.4 Usare GnuPG con Vim

Il plugin gnupg permette di eseguire GnuPG in modo trasparente per i file con estensione ".gpg", ".asc" e ".pgp".[3]

```
$ sudo aptitude install vim-scripts
$ echo "packadd! gnupg" >> ~/.vim/vimrc
```

3Se si usa "~/.vimrc" invece di "~/.vim/vimrc", modificare di conseguenza.

comando	descrizione
`gpg -a -s file`	firma un file in file.asc con corazza ASCII
`gpg --armor --sign file`	” ”
`gpg --clearsign file`	inserisce una firma leggibile nel messaggio
`gpg --clearsign file|mail pippo@example.org`	invia un messaggio di posta firmato leggibile a `pippo@example.org`
`gpg --clearsign --not-dash-escaped patchfile`	inserisce una firma leggibile in patchfile
`gpg --verify file`	verifica un file con firma leggibile
`gpg -o file.sig -b file`	crea una firma staccata
`gpg -o file.sig --detach-sign file`	„ ”
`gpg --verify file.sig file`	verifica file con file.sig
`gpg -o cifr_file.gpg -r nome -e file`	cifratura di file nel file binario cifr_file.gpg usando la chiave pubblica indirizzata a nome
`gpg -o cifr_file.gpg --recipient nome --encrypt file`	„ ”
`gpg -o cifr_file.asc -a -r nome -e file`	cifratura di file nel file con cifratura corazzata ASCII cifr_file.asc usando la chiave pubblica indirizzata a nome
`gpg -o cifr_file.gpg -c file`	cifratura simmetrica da file a cifr_file.gpg
`gpg -o cifr_file.gpg --symmetric file`	„ ”
`gpg -o cifr_file.asc -a -c file`	cifratura simmetrica pensata per nome di file nel file con cifratura corazzata ASCII cifr_file.asc
`gpg -o file -d cifr_file.gpg -r nome`	decifratura
`gpg -o file --decrypt cifr_file.gpg`	„ ”

Tabella 10.9: Elenco di comandi GNU Privacy Guard per file

10.3.5 Somme di controllo MD5

md5sum(1) fornisce un'utilità per creare un file digest usando il metodo descritto nella rfc1321 e per verificare i file con esso.

```
$ md5sum foo bar >baz.md5
$ cat baz.md5
d3b07384d113edec49eaa6238ad5ff00  foo
c157a79031e1c40f85931829bc5fc552  bar
$ md5sum -c baz.md5
foo: OK
bar: OK
```

Nota
Il calcolo delle somme di controllo MD5 è meno dispendioso in termini di CPU di quello delle firme crittografiche di GNU Privacy Guard (GnuPG). Di solito solamente il file digest di più alto livello è firmato crittograficamente per assicurare l'integrità dei dati.

10.3.6 Portachiavi per le password

Nei sistemi GNOME lo strumento GUI seahorse(1) gestisce le password e le archivia in modo sicuro nel portachiavi `~/.local/share/keyrings/*`.

secret-tool(1) può archiviare le password nel portachiavi dalla riga di comando.

Memorizziamo le passphrase usate per l'immagine del disco cifrata con LUKS/dm-crypt

```
$ secret-tool store --label='LUKS passphrase for disk.img' LUKS my_disk.img
Password: ********
```

Questa password archiviata può essere recuperata e passata ad altri programmi, es. cryptsetup(8).

```
$ secret-tool lookup LUKS my_disk.img | \
  cryptsetup open disk.img disk_img --type luks --keyring -
$ sudo mount /dev/mapper/disk_img /mnt
```

Suggerimento
Ogni qual volta si deve fornire la password in uno script, usare `secret-tool` ed evitare di inserire direttamente nel codice la password.

10.4 Strumenti per la fusione di codice sorgente

Esistono molti strumenti per la fusione di codice sorgente. Quello che segue è un elenco di strumenti che hanno catturato la mia attenzione.

10.4.1 Estrarre differenze da file sorgenti

Si possono estrarre le differenze tra due file sorgenti e creare file diff unificati "`file.patch0`" o "`file.patch1`", a seconda della posizione del file, con le procedure seguenti.

```
$ diff -u file.old file.new > file.patch0
$ diff -u old/file new/file > file.patch1
```

pacchetto	popcon	dimensione	comando	descrizione
patch	V:102, I:727	242	patch(1)	applica un file diff ad un originale
vim	V:83, I:338	4164	vimdiff(1)	confronta 2 file uno di fianco all'altro in vim
imediff	V:0.02, I:0.25	329	imediff(1)	strumento interattivo a tutto schermo per unione di modifiche a 2/3 vie
meld	V:5, I:23	3996	meld(1)	confronta e fonde file (GTK)
wiggle	V:0.02, I:0.12	175	wiggle(1)	applica le patch respinte
diffutils	V:896, I:998	1768	diff(1)	confronta i file riga per riga
diffutils	V:896, I:998	1768	diff3(1)	confronta e fonde tre file riga per riga
quilt	V:2, I:16	880	quilt(1)	gestisce serie di patch
wdiff	V:5, I:38	580	wdiff(1)	mostra le differenze di parole tra file di testo
diffstat	V:9, I:93	79	diffstat(1)	produce un istogramma delle modifiche apportate da un diff
patchutils	V:11, I:92	269	combinediff(1)	crea una patch cumulativa da due patch incrementali
patchutils	V:11, I:92	269	dehtmldiff(1)	estrae un diff da una pagina HTML
patchutils	V:11, I:92	269	filterdiff(1)	estrae o esclude diff da un file diff
patchutils	V:11, I:92	269	fixcvsdiff(1)	aggiusta file diff creati da CVS che sono male interpretati da patch(1)
patchutils	V:11, I:92	269	flipdiff(1)	scambia l'ordine di due patch
patchutils	V:11, I:92	269	grepdiff(1)	mostra quali file siano modificati da una patch che fa corrispondenza con un'espressione regolare
patchutils	V:11, I:92	269	interdiff(1)	mostra le differenze tra due file diff unificati
patchutils	V:11, I:92	269	lsdiff(1)	mostra quali file vengano modificati da una patch
patchutils	V:11, I:92	269	recountdiff(1)	ricalcola conteggi e offset in diff unificati
patchutils	V:11, I:92	269	rediff(1)	aggiusta conteggi ed offset di un diff modificato a mano
patchutils	V:11, I:92	269	splitdiff(1)	separa due patch incrementali
patchutils	V:11, I:92	269	unwrapdiff(1)	ripristina patch il cui contenuto è stato mandato a capo automaticamente
dirdiff	V:0.1, I:1.3	167	dirdiff(1)	mostra le differenze ed apporta i cambiamenti tra alberi di directory
docdiff	V:0.01, I:0.16	554	docdiff(1)	confronta due file parola per parola / carattere per carattere
makepatch	V:0.01, I:0.14	99	makepatch(1)	genera file patch estesi
makepatch	V:0.01, I:0.14	99	applypatch(1)	applica file patch estesi

Tabella 10.10: Elenco di strumenti per la fusione di codice sorgente

10.4.2 Fondere aggiornamenti per file sorgenti

I file diff (chiamati anche file patch) sono usati per inviare aggiornamenti per un programma. Chi li riceve applica questo aggiornamento ad un altro file nel modo seguente.

```
$ patch -p0 file < file.patch0
$ patch -p1 file < file.patch1
```

10.4.3 Unione (merge) interattiva

Se si hanno due versioni di un codice sorgente, è possibile effettuare una unione a 2 vie interattiva usando imediff(1) nel modo seguente.

```
$ imediff -o file.merged file.old file.new
```

Se si hanno tre versioni di un codice sorgente, è possibile effettuare una unione a 3 vie interattiva usando imediff(1) nel modo seguente.

```
$ imediff -o file.merged file.yours file.base file.theirs
```

10.5 Git

Git oggi giorno è lo strumento più usato per il sistema di controllo di versione(VCS) dato che Git può fare tutto per la gestione del codice sia locale che remota.

Debian fornisce servizi Git liberi attraverso il servizio Debian Salsa. La sua documentazione è reperibile su https://wiki.debian Salsa.

Ecco alcuni pacchetti correlati a Git.

10.5.1 Configurazione del client Git

È possibile che si desideri impostare diverse configurazioni globali, come il nome e l'indirizzo di posta elettronica usati da Git, in "`~/.gitconfig`" nel modo seguente.

```
$ git config --global user.name "Name Surname"
$ git config --global user.email yourname@example.com
```

Si può anche personalizzare il comportamento predefinito di Git nel modo seguente.

```
$ git config --global init.defaultBranch main
$ git config --global pull.rebase true
$ git config --global push.default current
```

Se si è abituati ai comandi di CVS o Subversion, si potrebbe volere impostare alcuni alias per i comandi nel modo seguente.

```
$ git config --global alias.ci "commit -a"
$ git config --global alias.co checkout
```

La configurazione globale può essere controllata con il comando seguente.

```
$ git config --global --list
```

pacchetto	popcon	dimensione	comando	descrizione
git	V:422, I:625	52512	git(7)	Git, il sistema di controllo delle revisioni veloce, scalabile e distribuito
gitk	V:3, I:26	2045	gitk(1)	browser degli archivi Git con interfaccia utente grafica e cronologia
qgit	V:0.2, I:1.9	1431	qgit(1)	browser degli archivi Git con interfaccia utente grafica e cronologia
git-cola	V:0.8, I:4.5	5129	git-cola(1)	browser degli archivi Git con interfaccia utente grafica e cronologia
tig	V:2, I:11	1245	tig(1)	browser in console dei repository Git con cronologia
lazygit	V:1.0, I:4.1	24167	lazygit(1)	browser in console dei repository Git con cronologia
git-gui	V:1, I:16	2549	git-gui(1)	interfaccia utente grafica per Git (senza cronologia)
git-email	V:0.3, I:9.3	1229	git-send-email(1)	invia una raccolta di patch come messaggio di posta da Git
git-buildpackage	V:1.2, I:6.8	2027	git-buildpackage(1)	automatizza la creazione di pacchetti Debian con Git
dgit	V:0.12, I:0.97	757	dgit(1)	interoperabilità di git con l’archivio Debian
imediff	V:0.02, I:0.25	329	git-ime(1)	strumento ausialiario per aiutare a suddividere commit di git
stgit	V:0.08, I:0.50	6253	stg(1)	quilt sopra a git (Python)
git-doc	I:10	15192	N/D	documentazione ufficiale per Git
gitmagic	I:0.48	721	N/D	”Git Magic”, una guida per Git più semplice da capire

Tabella 10.11: Elenco di pacchetti e comandi relativi a Git

10.5.2 Comandi Git di base

Il funzionamento di Git coinvolge diversi dati.

- L'albero di lavoro che contiene i file che l'utente vede e ai quali fa le modifiche.
  - Le modifiche da registrare devono essere selezionate esplicitamente e posizionate in attesa nell'indice. Questo viene fatto con i comandi `git add` e `git rm`.
- L'indice che contiene i file pronti per il commit (staged).
  - Il commit nel repository locale dei file preparati nell'indice (staged) verrà fatto alla successiva richiesta. Ciò viene fatto con il comando `git commit`.
- Il repository locale che contiene i file di cui è stato fatto il commit.
  - Git registra la cronologia collegata dei dati di cui è fatto il commit e li organizza come rami nel repository.
  - Il repository locale può inviare dati al repository remoto con il comando `git push`.
  - Il repository locale può ricevere dati dal repository remoto con i comandi `git fetch` e `git pull`.
    * Il comando `git pull` effettua un comando `git merge` o `git rebase` dopo il comando `git fetch`.
    * Qui, `git merge` combina due rami separati della cronologia per finire in un unico punto. (Questo è il comportamento predefinito di `git pull` senza alcuna personalizzazione e può essere buono per le persone che lavorano a monte che pubblicano rami verso molte persone.)
    * Qui, `git rebase` crea un singolo ramo di cronologia in sequenza del ramo remoto seguita da quella del ramo locale. (Questo è il caso della personalizzazione `pull.rebase true` e può essere adatto per il resto delle persone.)
- Il repository remoto che contiene i file di cui è stato fatto il commit.
  - La comunicazione con il repository remoto utilizza protocolli di comunicazione sicuri, come SSH o HTTPS.

L'albero di lavoro è fuori dalla directory `.git/`. I file dentro alla directory `.git/` contengono l'indice, i dati del repository locale e alcuni file di testo di configurazione di git.

Ecco una panoramica dei principali comandi di Git.

10.5.3 Suggerimenti per Git

Ecco alcuni suggerimenti per Git.



avvertimento
Non usare stringhe per etichette contenenti spazi, anche se alcuni strumenti come gitk(1) lo permettono. Altri comandi `git` potrebbero avere problemi con esse.



Attenzione
Se viene fatto il rebase o lo squash di un ramo locale di cui è stato fatto il push sul repository remoto, fare il push di tale ramo ha rischi e richiede l'opzione `--force`. Solitamente ciò non è accettabile per il ramo `main` ma può essere accettabile per un ramo di specifico argomento prima dell'unione nel ramo `main`.



Attenzione
A partire dall'inizio del 2006, l'invocazione diretta, dalla riga di comando, di un sottocomando di `git` come "`git-xyz`" è diventata deprecata.

Comando Git	funzione
`git init`	creazione dell'archivio (locale)
`git clone URL`	clona il repository remoto in un repository locale con l'albero di lavoro
`git pull origin main`	aggiorna il ramo `main` locale in base al repository remoto `origin`
`git add .`	aggiunge il/i file nell'albero di lavoro all'indice solo per i file già presenti nell'indice
`git add -A .`	aggiunge il/i file nell'albero di lavoro all'indice per tutti i file, incluse le rimozioni
`git rm nomefile`	rimuove il/i file dall'albero di lavoro e dall'indice
`git commit`	fa il commit dei cambiamenti in attesa nell'indice nel repository locale
`git commit -a`	aggiunge tutte le modifiche all'albero di lavoro nell'indice e ne fa il commit nel repository locale (add + commit)
`git push -u origin nome_ramo`	aggiorna il repository remoto `origin` con il ramo locale `nome_ramo` (invocazione iniziale)
`git push origin nome_ramo`	aggiorna il repository remoto `origin` con il ramo locale `nome_ramo` (invocazione successiva)
`git diff treeish1 treeish2`	mostra le differenza tra il commit *treeish1* e il commit *treeish2*
`gitk`	visualizzazione GUI dell'albero della cronologia dei rami del repository

Tabella 10.12: Principali comandi Git

Suggerimento
Se c'è un file eseguibile `git-pippo` nel percorso specificato da `$PATH`, l'inserimento nella riga di comando di "`git pippo`" senza trattino invoca tale `git-pippo`. Questa è una funzionalità del comando `git`.

10.5.4 Documenti di consultazione per Git

Vedere la documentazione seguente.

- pagina man: git(1) (`/usr/share/doc/git-doc/git.html`)
- Manuale utente di Git (`/usr/share/doc/git-doc/user-manual.html`)
- Un tutorial introduttivo su git (`/usr/share/doc/git-doc/gittutorial.html`)
- Un tutorial introduttivo su git: parte seconda (`/usr/share/doc/git-doc/gittutorial-2.html`)
- Uso quotidiano di GIT con 20 comandi o poco più (`/usr/share/doc/git-doc/giteveryday.html`)
- Git Magic (`/usr/share/doc/gitmagic/html/index.html`)

10.5.5 Altri sistemi di controllo delle versioni

Un sistema di controllo delle versioni (VCS) è a volte noto come sistema di controllo delle revisioni (RCS) o gestione delle configurazione del software (SCM).

Ecco un riassunto degli altri VCS diversi da Git nel sistema Debian degni di nota.

Riga di comando di Git	funzione
`gitk --all`	per vedere la cronologia completa di Git e operare su di essa, come reimpostare HEAD ad un altro commit, scegliere una a una le patch, creare etichette e rami...
`git stash`	ottenere un albero di lavoro pulito senza perdere dati
`git remote -v`	controllare le impostazioni per il repository remoto
`git branch -vv`	controllare le impostazioni per il ramo
`git status`	mostrare lo stato dell'albero di lavoro
`git config -l`	elencare le impostazioni di Git
`git reset --hard HEAD; git clean -x -d -f`	annullare tutte le modifiche all'albero di lavoro e ripulirle completamente
`git rm --cached nomefile`	annullare l'indice con modifiche in stage cambiato con `git add nomefile`
`git reflog`	ottenere il registro dei riferimenti (utile per ripristinare commit da un ramo rimosso)
`git branch nome_nuovo_ramo HEAD@{6}`	crea un nuovo ramo dalle informazioni reflog
`git remote add new_remote URL`	aggiunge un repository remoto `nuovo_remoto` a cui punta l'URL
`git remote rename origin upstream`	rinomina il repository remoto da `origin` a `upstream`
`git branch -u upstream/nome_ramo`	imposta il tracciamento remoto al repository remoto `upstream` e al suo ramo `nome_ramo`.
`git remote set-url origin https://foo/bar.git`	cambia l'URL di `origin`
`git remote set-url --push upstream DISABLED`	disabilita il push verso `upstream` (Modificare `.git/config` per riabilitarlo)
`git remote update upstream`	prende gli aggiornamenti di tutti i rami remoti nel repository `upstream`
`git fetch upstream foo:upstream-foo`	crea un ramo locale (possibilmente orfano) `upstream-foo` come copia del ramo `foo` nel repository `upstream`
`git checkout -b topic_branch ; git push -u topic_branch origin`	crea un nuovo `ramo_topic` e ne fa il push verso `origin`
`git branch -m vecchionome nuovonome`	rinomina il nome locale del ramo
`git push -d origin branch_to_be_removed`	rimuove il ramo remoto (nuovo metodo)
`git push origin :branch_to_be_removed`	rimuove il ramo remoto (vecchio metodo)
`git checkout --orphan unconnected`	crea un nuovo ramo `unconnected`
`git rebase -i origin/main`	riordina/abbandona/fa lo squish dei commit da `origin/main` per ripulire la cronologia del ramo
`git reset HEAD^; git commit --amend`	schiaccia gli ultimi 2 commit in uno solo
`git checkout topic_branch ; git merge --squash topic_branch`	schiaccia l'intero `ramo_topic` in un commit
`git fetch --unshallow --update-head-ok origin '+refs/heads/*:refs/heads/*'`	converte un clone shallow in un clone completo di tutti i rami
`git ime`	suddivide l'ultimo commit in una serie di commit più piccoli file-per-file (è richiesto il pacchetto `imediff`)
`git repack -a -d; git prune`	reimpacchetta il repository locale in un singolo pacchetto (ciò può limitare la possibilità di ripristinare dati persi da rami cancellati, ecc.)

Tabella 10.13: Suggerimenti per Git

pacchetto	popcon	dimensione	strumento	tipo di VCS	commento
mercurial	V:3, I:25	1375	Mercurial	distribuito	DVCS in Python e un po' di C
darcs	V:0.2, I:3.1	38856	Darcs	distribuito	DVCS con algebra intelligente per le patch (lento)
tla	V:0.03, I:0.67	1022	GNU arch	distribuito	DVCS principalmente da Tom Lord (storico)
bzr	V:0.0, I:3.9	28	GNU Bazaar	distribuito	DVCS influenzato da `tla`, scritto in Python (storico)
subversion	V:9, I:54	4856	Subversion	remoto	"CVS fatto bene", un più nuovo standard per VCS remoto (storico)
cvs	V:3, I:23	4835	CVS	remoto	precedente standard per VCS remoto (storico)
tkcvs	V:0.11, I:0.83	34	CVS, …	remoto	visualizzazione GUI di alberi di archivi VCS (CVS, Subversion, RCS)
rcs	V:1.6, I:9.0	578	RCS	locale	"SCCS Unix fatto bene" (storico)
cssc	V:0.01, I:0.28	2044	CSSC	locale	clone di SCCS Unix (storico)

Tabella 10.14: Elenco di altri strumenti per sistemi di controllo delle versioni

Capitolo 11

Conversione di dati

In questo capitolo sono descritti strumenti e trucchi per convertire tra diversi formati di dati in un sistema Debian. Gli strumenti standard sono eccellenti, ma il supporto per i formati proprietari per i dati è limitato.

11.1 Strumenti di conversione di dati testuali

Quelli che seguono sono pacchetti per la conversione dei dati testuali che hanno attirato la mia attenzione.

pacchetto	popcon	dimensione	parola chiave	descrizione
`libc6`	V:936, I:1000	5483	set di caratteri	convertitore della codifica del testo tra localizzazioni usando iconv(1) (fondamentale)
`recode`	V:2, I:12	528	set caratteri+carattere fine riga	convertitore della codifica del testo tra localizzazioni (versatile, maggior numero di alias e funzionalità)
`konwert`	V:2, I:42	137	set di caratteri	convertitore della codifica tra localizzazioni (elegante)
`nkf`	V:0.4, I:8.6	360	set di caratteri	traduttore del set di caratteri per il giapponese
`tcs`	V:0.01, I:0.14	518	set di caratteri	traduttore del set di caratteri
`unaccent`	V:0.03, I:0.27	35	set di caratteri	sostituisce le lettere accentate con le equivalenti senza accento
`tofrodos`	V:1, I:12	50	carattere fine riga	convertitore del formato di testo tra DOS e Unix: fromdos(1) e todos(1)
`macutils`	V:0.04, I:0.39	319	carattere fine riga	convertitore del formato di testo tra Macintosh e Unix: frommac(1) e tomac(1)

Tabella 11.1: Elenco di strumenti di conversione di dati testuali

11.1.1 Convertire un file di testo con iconv

Suggerimento
iconv(1) viene fornito come parte del pacchetto `libc6` ed è sempre disponibile praticamente su tutti i sistemi in stile Unix, per fare la conversione della codifica dei caratteri.

Si può convertire la codifica di un file di testo con iconv(1) nel modo seguente.

```
$ iconv -f encoding1 -t encoding2 input.txt >output.txt
```

Nel trovare corrispondenze con i valori delle codifiche non viene tenuto conto delle lettere maiuscole o minuscole e vengono ignorati i caratteri "`-`" e "_". Si possono controllare le codifiche supportate con il comando "`iconv -l`".

valore della codifica	uso
ASCII	American Standard Code for Information Interchange, codifica americana standard per lo scambio di informazioni, codice a 7 bit senza caratteri accentati
UTF-8	attuale standard multilingua per tutti i sistemi operativi moderni
ISO-8859-1	vecchio standard per le lingue europee occidentali, ASCII + caratteri accentati
ISO-8859-2	vecchio standard per le lingue europee orientali, ASCII + caratteri accentati
ISO-8859-15	vecchio standard per le lingue europee occidentali, ISO-8859-1 con simbolo dell'euro
CP850	code page 850, caratteri Microsoft DOS con caratteri grafici per le lingue europee occidentali, variante di ISO-8859-1
CP932	code page 932, variante in stile Microsoft Windows di Shift-JIS per il giapponese
CP936	code page 936, variante in stile Microsoft Windows di GB2312, GBK o GB18030 per il cinese semplificato
CP949	code page 949, variante in stile Microsoft Windows di EUC-KR o Unified Hangul Code per il coreano
CP950	code page 950, variante in stile Microsoft Windows di Big5 per il cinese tradizionale
CP1251	code page 1251, codifica in stile Microsoft Windows per l'alfabeto cirillico
CP1252	code page 1252, variante in stile Microsoft Windows di ISO-8859-15 per le lingue europee occidentali
KOI8-R	vecchio standard UNIX russo per l'alfabeto cirillico
ISO-2022-JP	codifica standard per la posta elettronica in giapponese che usa solo codici a 7 bit
eucJP	vecchio standard UNIX giapponese con codici a 8 bit completamente diverso da Shift-JIS
Shift-JIS	standard JIS X 0208 Appendix 1 per il giapponese (vedere CP932)

Tabella 11.2: Elenco dei valori delle codifiche e loro uso

Nota
Alcune codifiche sono gestite solo per la conversione dei dati e non sono utilizzate come valori di localizzazione (Sezione 8.1).

Per i set di caratteri contenuti in un singolo byte, come i set di caratteri ASCII e ISO-8859, la codifica di carattere è quasi la stessa cosa del set di caratteri.

Per i set di caratteri con molti caratteri, come JIS X 0213 per il giapponese o Universal Character Set (UCS, Unicode, ISO-10646-1) per praticamente tutte le lingue, esistono molti schemi di codifica per inserirli nella sequenza dei byte dati.

- EUC e ISO/IEC 2022 (chiamato anche JIS X 0202) per il giapponese

- UTF-8, UTF-16/UCS-2 e UTF-32/UCS-4 per Unicode

In questi casi c'è una netta differenza tra il set di caratteri e la codifica di caratteri.

Il termine code page è usato come sinonimo di tabella di codifica dei caratteri per alcune tabelle specifiche di produttori.

Nota
Notare che la maggior parte dei sistemi di codifica condividono con ASCII gli stessi codici per i caratteri a 7 bit. Ci sono però alcune eccezioni. Se si stanno convertendo dati di vecchi programmi C o URL in giapponese dal formato di codifica volgarmente chiamato Shift-JIS in formato UTF-8, usare "`CP932`" come nome di codifica invece di "`shift-JIS`" per ottenere i risultati attesi: `0x5C` → "\" e `0x7E` → "~". Altrimenti questi vengono convertiti nei caratteri sbagliati.

Suggerimento
Si può anche usare recode(1) che offre più della semplice combinazione delle funzionalità di iconv(1), fromdos(1), todos(1), frommac(1) e tomac(1). Per maggiori informazioni vedere "`info recode`".

11.1.2 Controllare se un file è in UTF-8 con iconv

Si può controllare se un file di testo è nella codifica UTF-8 usando iconv(1) nel modo seguente.

```
$ iconv -f utf8 -t utf8 input.txt >/dev/null || echo "non-UTF-8 found"
```

Suggerimento
Usare l'opzione "`--verbose`" nell'esempio precedente per trovare il primo carattere non UTF-8.

11.1.3 Convertire nomi di file con iconv

Ecco uno script d'esempio per convertire in una directory la codifica dei nomi di file dai nomi creati in un vecchio sistema operativo a quelli UTF-8 moderni.

```
#!/bin/sh
ENCDN=iso-8859-1
for x in *;
 do
 mv "$x" "$(echo "$x" | iconv -f $ENCDN -t utf-8)"
done
```

La variabile "`$ENCDN`" specifica la codifica originale usata per i nomi di file nei sistemi operativi più vecchi come in Tabella 11.2.

Per scenari più complessi, montare, usando come opzione di mount(8) la codifica appropriata (vedere Sezione 8.1.3), il file system (ad esempio una partizione in un disco fisso) contenente tali nomi di file e copiare usando il comando "`cp -a`" il suo intero contenuto in un altro file system montato come UTF-8.

11.1.4 Conversione del carattere di fine riga

Il formato dei file di testo e specificatamente il codice EOL (End of line, fine riga) è dipendente dalla piattaforma.

I programmi di conversione del formato di EOL, fromdos(1), todos(1), frommac(1) e tomac(1), sono piuttosto comodi. Anche recode(1) è utile.

piattaforma	codice EOL	carattere di controllo	decimale	esadecimale
Debian (unix)	LF	^J	10	0A
MSDOS e Windows	CR-LF	^M^J	13 10	0D 0A
Macintosh di Apple	CR	^M	13	0D

Tabella 11.3: Elenco di stili per EOL per differenti piattaforme

Nota
Alcuni dati nel sistema Debian, come i dati della pagina wiki per il pacchetto `python-moinmoin`, usano CR-LF in stile MSDOS come codice EOL. Perciò la regola sopra descritta è solamente una regola generale.

Nota
La maggior parte degli editor (es. `vim`, `emacs`, `gedit`, ...) può gestire file con EOL in stile MSDOS in modo trasparente.

Suggerimento
È meglio usare "`sed -e '/\r$/!s/$/\r/'`" invece di todos(1) quando si desidere unificare lo stile EOL allo stile MSDOS da uno stile misto MSDOS e Unix (ad esempio dopo la fusione di 2 file in stile MSDOS con diff3(1)). Questo perché `todos` aggiunge CR a tutte le righe.

11.1.5 Conversione di tabulazioni

Esistono alcuni programmi popolari specializzati nella conversione dei codici di tabulazione.

funzione	`bsdmainutils`	`coreutils`
espande le tabulazioni in spazi	"`col -x`"	`expand`
de-espande gli spazi in tabulazioni	"`col -h`"	`unexpand`

Tabella 11.4: Elenco di comandi di conversione di TAB dai pacchetti `bsdmainutils` e `coreutils`

indent(1) nel pacchetto `indent` riformatta completamente gli spazi bianchi nei programmi C.

Anche programmi editor come `vim` ed `emacs` possono essere usati per la conversione di TAB. Per esempio con `vim`, è possibile espandere i TAB con la sequenza di comandi "`:set expandtab`" e "`:%retab`". Si può annullare questa azione con la sequenza di comandi "`:set noexpandtab`" e "`:%retab!`".

11.1.6 Editor con auto-conversione

Gli editor moderni intelligenti come il programma`vim` sono piuttosto bravi e gestiscono bene qualsiasi sistema di codifica e formato di file. Per una migliore compatibilità questi editor andrebbero usati con la localizzazione UTF-8 in console con capacità UTF-8.

Un vecchio file di testo Unix in lingua europea occidentale, "`u-file.txt`" salvato nella codifica latin1 (iso-8859-1) può essere modificato con `vim` semplicemente con il comando seguente.

```
$ vim u-file.txt
```

Ciò è reso possibile dal fatto che il meccanismo di rivelazione automatica della codifica dei file di `vim` presuppone come prima cosa che la codifica sia UTF-8 e, se ciò fallisce, presuppone che sia latin1.

Un vecchio file di testo Unix in lingua polacca, "`pu-file.txt`" salvato nella codifica latin1 (iso-8859-1) può essere modificato con `vim` con il comando seguente.

```
$ vim '+e ++enc=latin2 pu-file.txt'
```

Un vecchio file di testo Unix in giapponese, "`ju-file.txt`" salvato nella codifica eucJP può essere modificato con `vim` con il comando seguente.

```
$ vim '+e ++enc=eucJP ju-file.txt'
```

Un vecchio file di testo MS-Windows in giapponese, "`jw-file.txt`" salvato nella cosiddetta codifica shift-JIS (più precisamente CP932) può essere modificato con `vim` con il comando seguente.

```
$ vim '+e ++enc=CP932 ++ff=dos jw-file.txt'
```

Quando un file viene aperto con le opzioni "`++enc`" e "`++ff`", l'uso di "`:w`" nella riga di comando di Vim lo salva nel formato originale sovrascrivendo il file originale. Si può anche specificare nella riga di comando di Vim il formato e il nome con cui salvare il file, ad esempio "`:w ++enc=utf8 nuovo.txt`".

Fare riferimento a mbyte.txt per il "supporto di testi multi-byte" nell'aiuto in linea di `vim` e a Tabella 11.2 per i valori delle localizzazione usati con "`++enc`".

La famiglia di programmi `emacs` può svolgere funzioni equivalenti.

11.1.7 Estrazione del testo puro

Il comando seguente legge una pagina web mettendola in un file di testo. È molto utile quando si copiano configurazioni dal Web o per applicare strumenti di testo base Unix come grep(1) sulla pagina web.

```
$ w3m -dump https://www.remote-site.com/help-info.html >textfile
```

In modo analogo, si possono estrarre dati in testo puro da altri formati con gli strumenti seguenti.

pacchetto	popcon	dimensione	parola chiave	funzione
`w3m`	V:10, I:129	2853	`html→text`	conversione da HTML a testo semplice con il comando "`w3m -dump`"
`html2text`	V:3, I:66	298	`html→text`	convertitore avanzato da HTML a testo semplice (ISO 8859-1)
`lynx`	V:29, I:455	2031	`html→text`	conversione da HTML a testo semplice con il comando "`lynx -dump`"
`elinks`	V:2, I:15	1791	`html→text`	conversione da HTML a testo semplice con il comando "`elinks -dump`"
`links`	V:2, I:20	2321	`html→text`	conversione da HTML a testo semplice con il comando "`links -dump`"
`links2`	V:0.8, I:9.4	5466	`html→text`	conversione da HTML a testo semplice con il comando "`links2 -dump`"
`catdoc`	V:16, I:170	682	`MSWord→text,TeX`	converte file MSWord in testo puro o TeX
`antiword`	V:0.9, I:6.3	596	`MSWord→text,ps`	converte file MSWord in testo puro o ps
`unhtml`	V:0.04, I:0.49	40	`html→text`	rimuove i tag marcatori da file HTML
`odt2txt`	V:1, I:19	60	`odt→text`	converte da OpenDocument Text in testo puro

Tabella 11.5: Elenco di strumenti per estrarre dati in testo puro

11.1.8 Evidenziare e formattare dati in puro testo

È possibile evidenziare e formattare dati in testo puro usando gli strumenti seguenti.

pacchetto	popcon	dimensione	parola chiave	descrizione
vim-runtime	V:17, I:355	38849	evidenziare	MACRO Vim per convertire codice sorgente in HTML con ":source $VIMRUNTIME/syntax/html.vim"
cxref	V:0.01, I:0.21	1191	c→html	convertitore per programmi C in latex e HTML (linguaggio C)
src2tex	V:0.02, I:0.16	1799	evidenziare	converte molti codici sorgenti in TeX (linguaggio C)
source-highlight	V:0.4, I:2.9	2131	evidenziare	converte molti codici sorgenti in file HTML, XHTML, LaTeX, Texinfo, sequenze di escape per colori ANSI e DocBook con evidenziazione (C++)
highlight	V:0.4, I:2.8	1412	evidenziare	converte molti codici sorgenti in file HTML, XHTML, RTF, LaTeX, TeX o XSL-FO con evidenziazione (C++)
grc	V:0.9, I:5.7	208	text→color	colorazione generica per tutto (Python)
pandoc	V:11, I:47	209559	text→any	convertitore universale di marcatori (Haskel)
python3-docutils	V:13, I:51	2013	text→any	formattatore di documenti ReStructured Text in XML (Python)
markdown	V:0.4, I:5.7	56	text→html	formattatore Markdown di documenti di testo in (X)HTML (Perl)
asciidoctor	V:0.5, I:4.6	101	text→any	formattatore AsciiDoc di documenti di testo in XML/HTML (Ruby)
python3-sphinx	V:7, I:27	3234	text→any	sistema di pubblicazione di documenti basati su ReStructured Text (Python)
hugo	V:0.8, I:5.0	66708	text→html	sistema di pubblicazione di siti statici basato su Markdown (Go)

Tabella 11.6: Elenco di strumenti per evidenziare dati in testo puro

11.2 Dati XML

XML (Extensible Markup Language) è un linguaggio a marcatori per documenti contenenti informazioni strutturate.

Vedere informazioni introduttive su XML.COM.

- "Cosa è XML?"
- "Cosa è XSLT?"
- "Cosa è XSL-FO?"
- "Cosa è XLink?"

11.2.1 Suggerimenti base per XML

I testi XML hanno un aspetto simile all'HTML. Permettono di gestire formati di output multipli per un documento. Un facile sistema per XML è il pacchetto `docbook-xsl` che è stato usato per questo documento.

Ogni file XML inizia con una dichiarazione XML standard come la seguente.

```
<?xml version="1.0" encoding="UTF-8"?>
```

La sintassi di base per un elemento XML usa un marcatore come il seguente.

```
<name attribute="value">content</name>
```

Gli elementi XML senza contenuto sono indicati nella seguente forma breve.

```
<name attribute="value" />
```

La parte ”`attributo="valore"`” negli esempi precedenti è opzionale.

In XML una sezione di commento è marcata nel modo seguente.

```
<!-- comment -->
```

A parte l’aggiunta dei marcatori, XML richiede solo una minima conversione dei contenuti usando entità predefinite per i caratteri seguenti.

entità predefinita	**carattere in cui convertire**
`"`	" : virgolette
`'`	' : apostrofo
`<`	< : minore-di
`>`	> : maggiore-di
`&`	& : e-commerciale

Tabella 11.7: Elenco di entità predefinite per XML



Attenzione
”<” e ”&” non possono essere usati in attributi o elementi.

Nota
Quando vengono usate entità definite in stile SGML, ad esempio ”`&qualche-tag;`”, la prima definizione prevale su tutte le altre. La definizione di entità è espressa nella forma ”`<!ENTITY qualche-tag "valore entità">`”.

Nota
Fintanto che i marcatori XML sono usati in modo coerente con un certo set dei nomi di tag (qualche dato usato come contenuto o valore di attributo), la conversione in un altro XML è un compito banale usando XSLT (Extensible Stylesheet Language Trasformations.

11.2.2 Elaborazione XML

Sono disponibili molti strumenti per elaborare file XML, come l’XSL (Extensible Stylesheet Language).

Fondamentalmente, una volta creato un file XML ben formato, lo si può convertire in qualsiasi formato usando XSLT (Extensible Stylesheet Language Transformation).

XSL-FO (Extensible Stylesheet Language for Formatting Object), linguaggio per fogli di stile estensibile per la formattazione di oggetti, è pensato per essere una soluzione per la formattazione. Il pacchetto `fop` è una novità nell’archivio

pacchetto	popcon	dimensione	parola chiave	descrizione
docbook-xml	V:15, I:407	2126	xml	DTD (Document Type Definition) XML per DocBook
docbook-xsl	V:15, I:144	14823	xml/xslt	fogli di stile XSL per elaborare XML DocBook con XSLT in vari formati di output
xsltproc	V:15, I:71	83	xslt	elaboratore a riga di comando XSLT (XML→ XML, HTML, testo semplice, ecc.)
xmlto	V:0.5, I:8.0	124	xml/xslt	convertitore da-XML-a-tutto con XSLT
fop	V:0.7, I:8.0	281	xml/xsl-fo	converte file XML DocBook in PDF
dblatex	V:0.9, I:5.4	4635	xml/xslt	converte file DocBook con XSLT in documenti DVI, PostScript, PDF
dbtoepub	V:0.04, I:0.54	37	xml/xslt	convertitore da XML DocBook a .epub

Tabella 11.8: Elenco di strumenti XML

Debian `main` a causa delle sue dipendenze dal linguaggio di programmazione Java. Perciò il codice LaTeX è solitamente generato dall'XML usando XSLT ed il sistema LaTeX viene usato per creare file adatti alla stampa come DVI, PostScript e PDF.

Dato che XML è un sottoinsieme di SGML (Standard Generalized Markup Language), può essere elaborato dagli strumenti completi disponibili per SGML, come DSSSL (Document Style Semantics and Specification Language).

pacchetto	popcon	dimensione	parola chiave	descrizione
openjade	V:1, I:21	1066	`dsssl`	elaboratore standard DSSSL ISO/IEC 10179:1996 (più recente)
docbook-dsssl	V:0.5, I:7.4	2594	`xml/dsssl`	fogli di stile DSSSL per elaborare XML DocBook con DSSSL in vari formati di output
docbook-utils	V:0.3, I:5.2	287	`xml/dsssl`	utilità per file DocBook, inclusa la conversione in altri formati (HTML, RTF, PS, man, PDF) con comandi `docbook2*` con DSSSL

Tabella 11.9: Elenco di strumenti DSSSL

Suggerimento
A volte `yelp` di GNOME è utile per leggere file XML DocBook direttamente dato che li visualizza in modo decente in X.

11.2.3 Estrazione di dati XML

Si possono esstrarre dati HTML o XML da altri formati usando gli strumenti seguenti.

11.2.4 Pulizia di dati XML

I file HTML non-XML possono essere convertiti in XHTML che è un'istanza di XML ben strutturato. XHTML può essere elaborato con strumenti XML.

La sintassi dei file XML e la correttezza deli URL in essi può essere verificata.

Una volta che è stato generato codice XML corretto, si può usare la tecnologia XSLT per estrarre dati in base al contesto dei marcatori

pacchetto	popcon	dimensione	parola chiave	descrizione
man2html	V:0.1, I:1.2	142	manpage → html	convertitore da pagine man a HTML (supporto CGI)
doclifter	V:0.00, I:0.05	487	troff → xml	convertitore da troff a XML DocBook
texi2html	V:0.2, I:2.8	1847	texi → html	convertitore da texinfo a HTML
info2www	V:0.8, I:1.4	76	info → html	convertitore da GNU info a HTML (supporto CGI)
wv	V:0.2, I:2.3	733	MSWord → any	convertitore di documenti da Microsoft Word a HTML, LaTeX, ecc.
unrtf	V:0.3, I:2.8	159	rtf → html	convertitore di documenti da RTF a HTML, ecc.
wp2x	V:0.00, I:0.08	200	WordPerfect → any	da file WordPerfect 5.0 e 5.1 a TeX, LaTeX, troff, GML e HTML

Tabella 11.10: Elenco di strumenti di estrazione di dati XML

pacchetto	popcon	dimensione	funzione	descrizione
libxml2-utils	V:60, I:207	213	xml ↔ html ↔ xhtml	strumento XML a riga di comando con xmllint(1) (controllo di sintassi, riformattazione, eliminazione sporcizia, …)
tidy	V:0.9, I:6.9	79	xml ↔ html ↔ xhtml	controllore della sintassi e riformattatore per HTML
weblint-perl	V:0.05, I:0.90	32	lint	strumento di controllo della sintassi e dello stile di base per HTML
linklint	V:0.04, I:0.44	343	controllo dei collega-menti	controllore veloce di collegamenti e strumento per manutenzione di siti web

Tabella 11.11: Elenco di strumenti per belle stampe XML

11.3 Impaginazione

Per semplici impaginazioni si può usare il programma Unix troff sviluppato da AT&T. Viene di solito usato per creare pagine man.

TeX, creato da Donald Knuth è uno strumento di impaginazione molto potente ed è lo standard di fatto. LaTeX, scritto originariamente da Leslie Lamport, permette un accesso alla potenza di TeX a più alto livello.

pacchetto	popcon	dimensione	parola chiave	descrizione
`texlive`	V:1, I:27	55	(La)TeX	sistema TeX per impaginazione, anteprima e stampa
`lilypond`	V:0.7, I:5.4	16924	`(La)TeX`	impaginatore di musica
`groff`	V:2, I:23	16536	`troff`	sistema di formattazione di testi GNU troff

Tabella 11.12: Elenco di strumenti per impaginazione

11.3.1 Impaginazione roff

Tradizionalmente il sistema di elaborazione di testi Unix principale è roff. Vedere roff(7), groff(7), groff(1), grotty(1), troff(1), groff_mdoc(7), groff_man(7), groff_ms(7), groff_me(7), groff_mm(7) e "`info groff`".

Si può leggere o stampare un buon tutorial e documento di consultazione sulla macro "`-me`" in "`/usr/share/doc/groff/`", dopo aver installato il pacchetto `groff`.

Suggerimento
"`groff -Tascii -me -`" produce output in puro testo con codici di escape ANSI. Se si desidera produrre un output in stile pagine man con molti "^H" e "_", usare invece "`GROFF_NO_SGR=1 groff -Tascii -me -`".

Suggerimento
Per rimuovere i "^H" e "_" da un file di testo generato con `groff`, filtrarlo con "`col -b -x`".

11.3.2 TeX/LaTeX

La distribuzione software TeX Live offre un sistema TeX completo. Il metapacchetto `texlive` fornisce una buona selezione dei pacchetti TeX Live che dovrebbe essere sufficiente per la maggior parte dei compiti più comuni.

Ci sono molti documenti consultabili disponibili per TeX e LaTeX.

- teTeX HOWTO: The Linux-teTeX Local Guide
- tex(1)
- latex(1)
- texdoc(1)
- texdoctk(1)
- "The TeXbook", di Donald E. Knuth, (Addison-Wesley)
- "LaTeX - A Document Preparation System", di Leslie Lamport, (Addison-Wesley)

- ”The LaTeX Companion”, di Goossens, Mittelbach, Samarin, (Addison-Wesley)

Questo è l’ambiente di impaginazione più potente. Molti elaboratori SGML lo usano come backend per l’elaborazione del testo. Lyx, fornito dal pacchetto `lyx`, e GNUTeXmacs, fornito dal pacchetto `texmacs`, offrono un bell’ambiente WYSIWYG per LaTeX; inoltre molti usano Emacs e Vim come scelta di editor per i sorgenti.

Sono disponibili molte risorse in rete.

- The TEX Live Guide - TEX Live 2007 (”`/usr/share/doc/texlive-doc-base/english/texlive-en/live.html`”) (pacchetto `texlive-doc-base`)
- A Simple Guide to Latex/Lyx
- Word Processing Using LaTeX

Quando i documenti crescono di dimensioni, a volte TeX può generare errori. Per risolvere queto problema si deve aumentare la dimensione di pool in ”`/etc/texmf/texmf.cnf`” (o in modo più corretto, modificare ”`/etc/texmf/texmf.d/9`” ed eseguire update-texmf(8)).

Nota
Il sorgente TeX di ”The TeXbook” è disponibile all’indirizzo www.ctan.org tex-archive site for texbook.tex. Questo file contiene la maggior parte delle macro necessarie. Mi è stato detto che si può elaborare questo documento con tex(1) dopo aver commentato le righe da 7 a 10 ed aggiungendo ”`\input manmac \proofmodefalse`”. È caldamente raccomandabile comprare questo libro (e tutti gli altri libri di Donald E. Knuth) invece di usare le versioni in rete, ma il sorgente è un meraviglioso esempio di input TeX!

11.3.3 Fare una bella stampa di una pagina di manuale

Si può stampare una pagina di manuale in PostScript in un bel modo usando uno dei comandi seguenti.

```
$ man -Tps some_manpage | lpr
```

11.3.4 Creare una pagina di manuale

Benché sia possibile scrivere una pagina di manuale (pagina man) in formato troff semplice, ci sono alcuni pacchetti di aiuto per farlo.

pacchetto	popcon	dimensione	parola chiave	descrizione
`docbook-to-man`	V:0.6, I:5.3	189	`SGML` → `manpage`	convertitore da SGML DocDook in macro roff man
`help2man`	V:0.5, I:6.1	542	`text` → `manpage`	generatore automatico di pagine man da --help
`info2man`	V:0.02, I:0.18	134	`info` → `manpage`	convertitore da GNU info a POD o pagine man
`txt2man`	V:0.04, I:0.61	112	`text` → `manpage`	converte testo in puro ASCII nel formato delle pagine man

Tabella 11.13: Elenco di pacchetti che aiutano a creare una pagina man

11.4 Dati stampabili

In un sistema Debian i dati stampabili sono presentati in formato PostScript. CUPS (Common Unix Printing System) usa Ghostscript come suo programma backend per la rasterizzazione per le stampanti non-PostScript.

I dati stampabili possono anche essere prodotti nel formato PDF nei sistemi Debian recenti.

I file PDF possono essere visualizzati e i campi nei loro moduli possono essere riempiti usando strumenti GUI per visualizzazione come Evince e Okular (vedere Sezione 7.4) e i browser moderni come Chromium.

I file PDF possono essere modificati usando alcuni strumenti grafici come LibreOffice, Scribus e Inkscape (vedere Sezione 11.6).

Suggerimento
Si può leggere un file PDF con GIMP e convertirlo in formato PNG usando una risoluzione maggiore di 300 dpi. Ciò può essere usato come immagine di sfondo per LibreOffice, per produrre una stampa modificata come si desidera con uno sforzo minimo.

11.4.1 Ghostscript

Il cuore centrale della manipolazione dei dati da stampare è l'interprete PostScript (PS) Ghostscript che genera immagini raster.

pacchetto	popcon	dimensione	descrizione
`ghostscript`	V:165, I:559	177	interprete PostScript/PDF GPL Ghostscript
`ghostscript-x`	V:0, I:13	88	interprete PostScript/PDF GPL Ghostscript - supporto per display X
`libpoppler162`	V:14, I:22	5313	libreria di rendering per PDF risultato di un fork del visualizzatore di PDF xpdf
`libpoppler-glib8t64`	V:77, I:323	612	libreria di rendering per PDF (libreria condivisa basata su GLib)
`poppler-data`	V:161, I:582	13086	CMaps per la libreria di rendering per PDF (per supporto CJK: Adobe-*)

Tabella 11.14: Elenco di interpreti PostScript Ghostscript

Suggerimento
La configurazione di Ghostscript può essere visualizzata con "`gs -h`".

11.4.2 Unire due file PS o PDF

È possibile unire due file PostScript (PS) o PDF (Portable Document Format) usando gs(1) di Ghostscript.

```
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pswrite -sOutputFile=bla.ps -f foo1.ps foo2.ps
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=bla.pdf -f foo1.pdf foo2.pdf
```

Nota
Il formato PDF che è un formato ampiamente usato per dati stampabili multipiattaforma, è essenzialmente il formato PS compresso con alcune funzionalità ed estensioni aggiuntive.

Suggerimento
Per manipolare dalla riga di comando documenti PostScript sono utili psmerge(1) e altri comandi nel pacchetto `psutils`. Anche pdftk(1), nel pacchetto `pdftk`, è utile per manipolare documenti PDF.

11.4.3 Utilità per dati stampabili

Quello che segue è un elenco di pacchetti con utilità per dati stampabili che hanno attirato la mia attenzione.

pacchetto	popcon	dimensione	parola chiave	descrizione
`poppler-utils`	V:169, I:476	804	`pdf`→`ps`, `text`	utilità PDF: `pdftops`, `pdfinfo`, `pdfimages`, `pdftotext`, `pdffonts`
`psutils`	V:4, I:48	34	`ps`→`ps`	strumenti di conversione di documenti PostScript
`poster`	V:0.1, I:1.6	57	`ps`→`ps`	crea grandi poster da pagine PostScript
`enscript`	V:1, I:10	2138	testo→ps, html, rtf	converte testo ASCII in PostScript, HTML, RTF o Pretty-Print
`a2ps`	V:0.7, I:6.7	4109	`text`→`ps`	convertitore e creatore di belle stampe ”Da tutto a PostScript”
`pdftk`	V:1, I:20	28	`pdf`→`pdf`	strumento di conversione di documenti PDF: `pdftk`
`html2ps`	V:0.2, I:1.7	256	`html`→`ps`	converte da HTML aPostScript
`gnuhtml2latex`	V:0.03, I:0.57	26	`html`→`latex`	convertitore da HTML a latex
`latex2rtf`	V:0.1, I:1.9	495	`latex`→`rtf`	converte documenti da LaTeX a RTF leggibili da MS Word
`ps2eps`	V:2, I:31	95	`ps`→`eps`	converte da PostScript a EPS (Encapsulated PostScript)
`e2ps`	V:0.01, I:0.11	104	`text`→`ps`	convertitore da testo a PostScript con supporto per la codifica giapponese
`impose+`	V:0.1, I:1.6	118	`ps`→`ps`	Utilità PostScript
`trueprint`	V:0.00, I:0.07	148	`text`→`ps`	belle stampe di molti tipi di codice sorgente (C, C++, Java, Pascal, Perl, Pike, Sh e Verilog) in PostScript. (linguaggio C)
`pdf2svg`	V:0.2, I:2.9	33	`pdf`→`svg`	convertitore da PDF al formato SVG (Scalable Vector Graphics)
`pdftoipe`	V:0.01, I:0.43	69	`pdf`→`ipe`	convertitore da PDF al formato XML di IPE

Tabella 11.15: Elenco di utilità per dati stampabili

11.4.4 Stampare con CUPS

Entrambi i comandi lp(1) e lpr(1) forniti da CUPS (Common Unix Printing System) forniscono opzioni per stampe personalizzate dei dati stampabili.

Si possono stampare 3 copie fascicolate di un file usando uno dei comandi seguenti.

```
$ lp -n 3 -o Collate=True filename
```

```
$ lpr -#3 -o Collate=True filename
```

Si può personalizzare ulteriormente l’operazione di stampa usando opzioni come ”`-o number-up=2`”, ”`-o page-set=eve`
”`-o page-set=odd`”, ”`-o scaling=200`”, ”`-o natural-scaling=200`”, ecc., documentate in Stampa dalla riga di comando ed opzioni relative.

11.5 Conversione dei dati di posta

Quelli che seguono sono pacchetti per la conversione dei dati di posta che hanno attirato la mia attenzione.

pacchetto	popcon	dimensione	parola chiave	descrizione
sharutils	V:2, I:27	1436	posta	shar(1), unshar(1), uuencode(1), uudecode(1)
mpack	V:0.7, I:7.4	109	MIME	codifica e decodifica di messaggiMIME: mpack(1) e munpack(1)
tnef	V:0.3, I:3.7	103	ms-tnef	spacchettamento di allegati MIME di tipo "application/ms-tnef" che è un formato esclusivo di Microsoft
uudeview	V:0.2, I:1.7	105	mail	codificatore e decodificatore per i seguenti formati: uuencode, xxencode, BASE64, quoted printable e BinHex

Tabella 11.16: Elenco di pacchetto che aiutano a convertire dati di posta

Suggerimento
Se il software del programma di posta può essere configurato per usarlo, anche il server IMAP4 (Internet Message Access Protocol, versione 4) può essere usato per spostare la posta da sistemi proprietari.

11.5.1 Nozioni di base sui dati di posta

I dati di posta (SMTP) dovrebbero essere limitati a serie di 7 bit di dati. Perciò dati binari e testi a 8 bit sono codificati in un formato a 7 bit con MIME (Multipurpose Internet Mail Extensions) e la selezione del set di caratteri (vedere Tabella 11.2).

Il formato standard per l'archiviazione della posta è mbox strutturato seguendo la RFC2822 (che aggiorna RFC822). Vedere mbox(5) (fornito dal pacchetto `mutt`).

Per le lingue europee, per la posta viene di solito usato "`Content-Transfer-Encoding: quoted-printable`" con il set di caratteri ISO-8859-1 dato che non ci sono molti caratteri a 8 bit. Se il testo europeo è codificato in UTF-8, è probabile venga usato "`Content-Transfer-Encoding: quoted-printable`" dato che sono per la maggior parte dati a 7 bit.

Per il giapponese, per la posta viene tradizionalmente usato "`Content-Type: text/plain; charset=ISO-2022-JP`" per mantenere il testo a 7 bit. Ma sistemi Microsoft più vecchi possono inviare dati di posta in Shift-JIS senza una dichiarazione appropriata. Se il testo giapponese è codificato in UTF-8, è probabile venga usato Base64 dato che contiene molti dati a 8 bit. La situazione delle altre lingue asiatiche è simile.

Nota
Se i dati di posta non-Unix sono accessibili da un software di posta non-Debian che può comunicare con il server IMAP4, potrebbe essere possibile spostarli eseguendo un proprio server IMAP4.

Nota
Se si usano altri formati di archiviazione, spostarli nel formato mbox è un buon primo passo. Un versatile programma client come mutt(1) può essere di aiuto in questo caso.

Si può suddividere il contenuto di una casella di posta nei singoli messaggi usando procmail(1) e formail(1).

Ciascun messaggio di posta può essere spachettato usando munpack(1), dal pacchetto `mpack`, (o con altri strumenti specializzati) per ottenere i contenuti codificati MIME.

11.6 Strumenti per dati grafici

Sebbene programmi con interfaccia utente grafica come gimp(1) siano molto potenti, strumenti a riga di comando come imagemagick(1) sono piuttosto utili per manipolare in modo automatico immagini usando script.

Il formato standard di fatto per i file immagini delle fotocamere digitali è EXIF (Exchangeable Image File Format) che è il formato per file immagine JPEG con tag aggiuntivi con metadati. Può contenere informazioni come la data, l'orario e le impostazioni della fotocamera.

I diritti sulla compressione senza perdita di dati Lempel-Ziv-Welch (LZW) sono scaduti. Le utilità GIF (Graphics Interchange Format) che usano il metodo di compressione LZW sono ora disponibili liberamente sul sistema Debian.

Suggerimento
Qualsiasi fotocamera digitale o scanner con un supporto di archiviazione removibile funziona in Linux attraverso i lettori di archiviazione USB dato che segue le regole progettuali per i file system delle fotocamere e usa il file system FAT. Vedere Sezione 10.1.7.

11.6.1 Strumenti per dati grafici (metapacchetto)

I seguenti metapacchetti sono un buon punto di inizio per cercare strumenti per dati grafici usando aptitude(8). Un altro punto di partenza può essere "Panoramica dei pacchetti per i manutentori degli strumenti per fotografia di Debian".

pacchetto	popcon	dimensione	parola chiave	descrizione
education-graphics	I:0.34	26	svg,jpeg,…	metapacchetto per insegnare grafica e arti pittoriche.
open-font-design-toolkit	I:0.04	9	ttf,ps,…	metapacchetto per creazione di tipi di caratteri open

Tabella 11.17: Elenco di strumenti per dati grafici (metapacchetto)

Suggerimento
Cercare ulteriori strumenti per le immagini usando l'espressione regolare "`~Gworks-with::image`" in aptitude(8) (vedere Sezione 2.2.6).

11.6.2 Strumenti per dati grafici (GUI)

Quelli che seguono sono pacchetti di strumenti GUI per la conversione, la modifica e l'organizzazione di dati grafici che hanno attirato la mia attenzione.

11.6.3 Strumenti per dati grafici (CLI)

Quelli che seguono sono pacchetti di strumenti CLI per la conversione, la modifica e l'organizzazione di dati grafici che hanno attirato la mia attenzione.

pacchetto	popcon	dimensione	parola chiave	descrizione
gimp	V:34, I:211	33315	image(bitmap)	GNU Image Manipulation Program, programma GNU di manipolazione di immagini
xsane	V:9, I:128	1512	image(bitmap)	frontend X11 basato su GTK per SANE (Scanner Access Now Easy)
scribus	V:1, I:12	32942	ps/pdf/SVG/...	editor DTP Scribus
libreoffice-draw	V:89, I:418	11418	image(vector)	suite per ufficio LibreOffice - disegno
inkscape	V:12, I:76	113353	image(vector)	editor SVG (Scalable Vector Graphics)
dia	V:1, I:16	3846	image(vector)	editor di diagrammi (Gtk)
xfig	V:0.5, I:8.5	7951	image(vector)	strumento per la generazione interattiva di figure in X11
gocr	V:0.5, I:3.6	549	image→text	software OCR libero
eog	V:23, I:129	10535	image(Exif)	visualizzatore di immagini grafiche Eye of GNOME
gthumb	V:3, I:11	5162	image(Exif)	navigatore e visualizzatore di immagini (GNOME)
geeqie	V:3, I:11	3367	image(Exif)	visualizzatore d'immagini che usa GTK
shotwell	V:14, I:246	6334	image(Exif)	organizzatore di foto digitali (GNOME)
gwenview	V:41, I:114	6057	image(Exif)	visualizzatore di immagini (KDE)
kamera	I:114	992	image(Exif)	supporto per fotocamere digitali in applicazioni KDE
digikam	V:1.5, I:8.3	333	image(Exif)	applicazione per la gestione delle foto digitali per KDE
darktable	V:4, I:11	36092	image(Exif)	tavolo luminoso e camera oscura virtuali per fotografi
hugin	V:0.5, I:5.3	6524	image(Exif)	ricucitore di foto per panorami
librecad	V:1, I:13	9164	DXF,...	editor di dati CAD 2D
freecad	V:1, I:20	112	DXF,...	editor di dati CAD 3D
blender	V:2, I:20	171893	blend,TIFF,...	editor di contenuti 3D per animazioni, ecc.
mm3d	V:0.02, I:0.22	4123	ms3d,obj,dxf,...	editor di modelli 3D basato su OpenGL
fontforge	V:0.6, I:5.3	4736	ttf,ps,...	editor di tipi di carattere per caratteri PS, TrueType e OpenType
xgridfit	V:0.02, I:0.08	878	ttf	programma per gridfitting e hinting di tipi di carattere TrueType

Tabella 11.18: Elenco di strumenti per dati grafici (GUI)

pacchetto	popcon	dimensione	parola chiave	descrizione
imagemagick	V:7, I:272	80	image(bitmap)	programmi di manipolazione immagini
graphicsmagick	V:1.0, I:8.3	5945	image(bitmap)	programmi di manipolazione di immagini (fork di imagemagick)
netpbm	V:27, I:282	8435	image(bitmap)	strumenti di conversione di dati grafici
libheif-examples	V:0.3, I:3.6	528	heif→jpeg(bitmap)	convertire il formato High Efficiency Image File (HEIF) nei formati JPEG, PNG o Y4M con il comando heif-convert(1)
icoutils	V:3, I:33	221	png↔ico(bitmap)	converte icone e puntatori MS Windows da e verso il formato PNG (favicon.ico)
pstoedit	V:2, I:37	1075	ps/pdf→image(vector)	convertitore di file PostScript e PDF in grafica vettoriale modificabile (SVG)
libwmf-bin	V:5, I:80	149	Windows/immagine(vettoriale)	strumenti di conversione di metafile windows (dati di grafica vettoriale)
fig2sxd	V:0.01, I:0.16	158	fig→sxd(vector)	converte file XFig nel formato di Draw di OpenOffice.org
unpaper	V:2, I:16	417	image→image	strumento di post-elaborazione per pagine scansionate per OCR
tesseract-ocr	V:8, I:34	2209	image→text	software OCR libero basato sul motore OCR commerciale di HP
tesseract-ocr-eng	V:9, I:35	4032	image→text	dati per motore OCR: file di lingua tesseract-ocr per testi in inglese
ocrad	V:0.3, I:2.2	608	image→text	software OCR libero
exif	V:3, I:49	335	image(Exif)	utilità a riga di comando per mostrare informazioni EXIF in file JPEG
exiv2	V:1, I:18	436	image(Exif)	strumento di manipolazione di metadati EXIF/IPTC
exiftran	V:1, I:11	81	image(Exif)	trasforma immagini JPEG di fotocamere digitali
exiftags	V:0.2, I:2.5	309	image(Exif)	utilità per leggere i tag EXIF da un file JPEG di una fotocamera digitale
exifprobe	V:0.2, I:2.1	506	image(Exif)	legge metadati da immagini digitali
dcraw	V:0.7, I:7.2	428	image(Raw)→ppm	decodifica immagini raw di fotocamere digitali
findimagedupes	V:0.11, I:0.96	76	image→fingerprint	trova immagini simili visivamente o duplicati
ale	V:0.00, I:0.14	850	image→image	fonde immagini per migliorarne la fedeltà o creare mosaici
imageindex	V:0.1, I:1.1	143	image(Exif)→html	genera gallerie HTML statiche da immagini
outguess	V:0.1, I:1.0	230	jpeg,png	strumento steganografico universale
jpegoptim	V:0.6, I:5.8	59	jpeg	ottimizza file JPEG
optipng	V:2, I:39	187	png	ottimizza file PNG, con compressione senza perdita
pngquant	V:1, I:11	62	png	ottimizza file PNG, con compressione senza perdita

Tabella 11.19: Elenco di strumenti per dati grafici (CLI)

pacchetto	popcon	dimensione	parola chiave	descrizione
alien	V:1, I:12	150	rpm/tgz→deb	convertitore di pacchetti estranei in pacchetti Debian
freepwing	V:0.00, I:0.01	447	EB→EPWING	convertitore da ”Electric Book” (popolare in Giappone) in un singolo formato JIS X 4081 (un sottoinsieme di EPWING V1)
calibre	V:7, I:24	71216	any→EPUB	convertitore e gestione di una biblioteca per libri elettronici

Tabella 11.20: Elenco di strumenti di conversione di dati vari

11.7 Conversioni di dati vari

Ci sono molti altri programmi per convertire dati. I pacchetti seguenti, trovati usando l’espressione regolare ”~Guse::conver in aptitude(8) (vedere Sezione 2.2.6), hanno catturato la mia attenzione.

Si possono estrarre i dati dal formato RPM anche nel modo seguente.

```
$ rpm2cpio file.src.rpm | cpio --extract
```

Capitolo 12

Programmazione

Vengono forniti in questo capitolo alcune informazioni base da cui partire per imparare a programmare su un sistema Debian abbastanza da seguire il codice sorgente impacchettato. Quello che segue è un elenco dei pacchetti importanti per la programmazione e dei corrispettivi pacchetti di documentazione.

Guide di riferimento in linea sono disponibili digitando "`man nome`" dopo aver installato i pacchetti `manpages` e `manpages-dev`. Le guide di riferimento in linea per gli strumenti GNU sono disponibili digitando "`info nome_programma`", dopo aver installato i pertinenti pacchetti di documentazione. Può essere necessario includere gli archivi "`contrib` e `non-free`, oltre all'archivio `main`, dato che alcune documentazioni GFDL non sono considerate conformi alle DFSG.

Considerare l'uso degli strumenti dei sistemi di controllo di versione. Vedere Sezione 10.5.



avvertimento
Non usare "`test`" come nome di un file di prova eseguibile. "`test`" è un comando interno della shell.



Attenzione
I programmi software compilati direttamente dai sorgenti andrebbero installati in "`/usr/local`" o "`/opt`" per evitare conflitti.

Suggerimento
Esempi di codice per creare la "Canzone 99 bottiglie di birra" dovrebbe dare buone idee per praticamente tutti i linguaggi di programmazione.

12.1 Script shell

Uno script di shell è un file di testo con il bit di esecuzione impostato e contiene i comandi nel formato seguente.

```
#!/bin/sh
 ... command lines
```

La prima riga specifica l'interprete di shell che legge ed esegue il contenuto di questo file.

Leggere script di shell è il modo **migliore** per capire come funzioni un sistema *nix. In questa sezione vengono forniti alcune nozioni di riferimento e promemoria per la programmazione di shell. Vedere "Errori in shell" (https://www.greenend.org rjk/2001/04/shell.html) per imparare dagli errori.

A differenza della modalità interattiva della shell (vedere Sezione 1.5 e Sezione 1.6), gli script di shell usano spesso parametri, costrutti condizionali e cicli.

12.1.1 Compatibilità con la shell POSIX

Molti script di sistema possono essere interpretati da una qualsiasi delle shell POSIX (vedere Tabella 1.13).

- La shell POSIX non interattiva predefinita "`/usr/bin/sh`" è un collegamento simbolico che punta a `/usr/bin/dash` ed è usata da molti programmi di sistema.
- La shell POSIX interattiva predefinita è `/usr/bin/bash`.

Evitare di scrivere uno script di shell con **bashismi** o **zshismi** per renderlo portabile tra tutte le shell POSIX. Si può controllare uno script con checkbashisms(1).

Buono: POSIX	**Da evitare: bashismo**
`if [ "$foo" = "$bar" ] ; then …`	`if [ "$foo" == "$bar" ] ; then …`
`diff -u file.c.orig file.c`	`diff -u file.c{.orig,}`
`mkdir /foobar /foobaz`	`mkdir /foo{bar,baz}`
`funcname() { … }`	`function funcname() { … }`
formato ottale: "`\377`"	formato esadecimale: "`\xff`"

Tabella 12.1: Elenco di bashismi tipici

Il comando "`echo`" deve essere usato con le precauzioni seguenti dato che la sua implementazione è diversa negli svariati comandi interni della shell ed esterni.

- Evitare l'uso di qualsiasi opzione di comando tranne "`-n`".
- Evitare l'uso di sequenze di escape nelle stringhe dato che la loro gestione è variabile.

Nota
Sebbene l'opzione "`-n`" **non** faccia veramente parte della sintassi POSIX, è generalmente accettata.

Suggerimento
Se è necessario inserire sequenze di escape nella stringa in output, usare il comando "`printf`" al posto del comando "`echo`".

12.1.2 Parametri di shell

Negli script di shell vengono spesso usati parametri speciali.

Le nozioni base da ricordare riguardanti la **espansione dei parametri** sono le seguenti.

I due punti "`:`" in tutti gli operatori nell'elenco precedente sono di fatto opzionali.

- **con** "`:`" l'operatore = controlla che il suo operando **esista** e sia **non nullo**
- **senza** "`:`" l'operatore = controlla solo che il suo operando **esista**

parametro di shell	valore
`$0`	nome della shell o dello script di shell
`$1`	primo (1°) argomento di shell
`$9`	nono (9°) argomento di shell
`$#`	numero di parametri posizionali
`"$*"`	`"$1 $2 $3 $4 …"`
`"$@"`	`"$1" "$2" "$3" "$4" …`
`$?`	stato d'uscita del comando più recente
`$$`	PID dello script di shell
`$!`	PID del compito sullo sfondo avviato più recentemente

Tabella 12.2: Elenco di parametri di shell

forma della espressione con parametri	valore se `var` è impostata	valore se `var` non è impostata
`${var:-string}`	"`$var`"	"`string`"
`${var:+string}`	"`string`"	"`null`"
`${var:=string}`	"`$var`"	"`stringa`" (ed esegue "`var=stringa`")
`${var:?string}`	"`$var`"	invia con echo "`stringa`" allo **stderr** (ed esce con stato di errore)

Tabella 12.3: Elenco di espansioni di parametri di shell

forma della sostituzione di parametri	risultato
`${var%suffix}`	rimuove il più piccolo modello di suffisso
`${var%%suffix}`	rimuove il più grande modello di suffisso
`${var#prefix}`	rimuove il più piccolo modello di prefisso
`${var##prefix}`	rimuove il più grande modello di prefisso

Tabella 12.4: Elenco di sostituzioni chiave di parametri di shell

12.1.3 Costrutti condizionali della shell

Ogni comando restituisce uno **stato di uscita** che può essere usato in costrutti condizionali.

- Successo: 0 (”Vero”)
- Errore: non 0 (”Falso”)

Nota
”0” nel contesto condizionale della shell significa ”Vero”, mentre ”0” nel contesto condizionale in C significa ”Falso”.

Nota
”`[`” è l’equivalente del comando ”`test` che valuta i propri argomenti sino a ”`]`” come un’espressione condizionale.

Le **espressioni condizionali** di base che è bene ricordare sono le seguenti.

- ”*`comando`* `&&` *`se_successo_esegue_anche_questo_comando`* `||` `true`”
- ”*`comando`* `||` *`se_non_successo_esegue_anche_questo_comando`* `||` `true`”
- Una porzione su più righe di script come la seguente

```
if [ conditional_expression ]; then
 if_success_run_this_command
else
 if_not_success_run_this_command
fi
```

In questo caso il ”`|| true`” finale era necessario per assicurare che lo script non termini accidentalmente a tale riga quando la shell è invocata con l’opzione ”`-e`”.

equazione	**condizione perché venga restituito il valore logico ”vero”**
`-e` *`file`*	*file* esiste
`-d` *`file`*	*file* esiste ed è una directory
`-f` *`file`*	*file* esiste ed è un file regolare
`-w` *`file`*	*file* esiste ed è scrivibile
`-x` *`file`*	*file* esiste ed è eseguibile
`file1` `-nt` *`file2`*	*file1* è più recente di *file2* (data di modifica)
`file1` `-ot` *`file2`*	*file1* è più vecchio di *file2* (data di modifica)
`file1` `-ef` *`file2`*	*file1* e *file2* sono sullo stesso device e stesso numero inode

Tabella 12.5: Elenco di operatori per paragonare file in espressioni condizionali

Gli operatori **aritmetici** di comparazione di interi nelle espressioni condizionali sono ”`-eq`”, ”`-ne`”, ”`-lt`”, ”`-le`”, ”`-gt`” e ”`-ge`”.

12.1.4 Cicli di shell

Ci sono diverse espressioni per cicli usabili nella shell POSIX.

equazione	condizione perché venga restituito il valore logico ”vero”
`-z str`	la lunghezza di *str* è zero
`-n str`	la lunghezza di *str* è diversa da zero
`str1 = str2`	*str1* e *str2* sono uguali
`str1 != str2`	*str1* e *str2* non sono uguali
`str1 < str2`	se ordinate, *str1* viene prima di *str2* (dipendente dalla localizzazione)
`str1 > str2`	se ordinate, *str1* viene dopo di *str2* (dipendente dalla localizzazione)

Tabella 12.6: Elenco di operatori per paragonare stringhe in espressioni condizionali

- ”`for x in pippo1 pippo2 … ; do comando ; done`” ripete il ciclo assegnando gli elementi nell’elenco ”`pippo1 pippo2 …`” alla variabile ”x” ed eseguendo ”`comando`”.
- ”`while condizione ; do comando ; done`” ripete ”`comando`” fintanto che ”`condizione`” è vera.
- ”`until condizione ; do comando ; done`” ripete ”`comando`” fintanto che ”`condition`” è non vera.
- ”`break`” permette di uscire dal ciclo.
- ”`continue`” permette di riprendere dalla successiva iterazione del ciclo.

Suggerimento
L’iterazione numerica in stile linguaggio C può essere realizzata usando seq(1) come ”`pippo1 pippo2 …`”

Suggerimento
Vedere Sezione 9.4.9.

12.1.5 Variabile d’ambiente della shell

Alcune popolari variabili d’ambiente per il normale prompt dei comandi della shell possono non essere disponibili nell’ambiente di esecuzione di uno script utente.

- Per ”`$USER`”, usare ”`$(id -un)`”
- Per ”`$UID`”, usare ”`$(id -u)`”
- Per ”`$HOME`”, usare ”`$(getent passwd "$(id -u)"|cut -d ":" -f 6)`” (questo funziona anche con Sezione 4.5.2)

12.1.6 La sequenza di elaborazione della riga di comando di shell

A grandi linee la shell elabora uno script nel modo seguente.

- La shell legge una riga.
- La shell raggruppa parte della riga come **un unico elemento** se è racchiusa in `"…"` o `'…'`.
- La shell spezza le altre parti della riga in **elementi** in base ai caratteri seguenti.
  - Spazi bianchi: *`spazio tabulazione a capo`*
  - Metacaratteri: `< > | ; & ( )`

- La shell controlla, per ciascun elemento non racchiuso tra "…" o '…', la presenza di **parole riservate** per regolare il proprio comportamento.
  - **Parole riservate**: `if then elif else fi for in while unless do done case esac`
- La shell espande gli **alias** se non sono racchiusi in "…" o '…'.
- La shell espande il carattere **tilde** se non è racchiuso in "…" o '…'.
  - "~" → directory home dell'utente attuale
  - "~*utente*" → directory home di *utente*
- La shell espande **parametri** nei loro valori, se non sono racchiusi in '…'.
  - **Parametro**: "$PARAMETRO" o "${PARAMETRO}"
- La shell espande **sostituzioni di comandi**, se non sono racchiuse in '…'.
  - "$(comando)" → output di "comando"
  - "` command `" → output di "comando"
- La shell espande **glob di nomi percorso** nei nomi di file corrispondenti, se non sono racchiusi in "…" o '…'.
  - * → qualsiasi carattere
  - ? → un carattere
  - […] → uno qualunque dei caratteri in "…"
- La shell cerca **comando** tra le cose seguenti e lo esegue.
  - Definizione di **funzione**
  - comando **interno**
  - **file eseguibile** in "$PATH"
- La shell si sposta alla riga seguente e ripete nuovamente questo processo dall'inizio di questa sequenza.

Virgolette singole all'interno di virgolette doppie non hanno alcun effetto.

L'esecuzione di "`set -x`" nella shell o l'invocazione della shell con l'opzione "`-x`" fanno sì che la shell stampi tutti i comandi eseguiti. Ciò è piuttosto utile per il debug.

12.1.7 Programmi di utilità per script di shell

Per far sì che il proprio programma di shell sia il più portabile possibile tra i sistemi Debian, è una buona idea limitare i programmi di utilità a quelli forniti dai pacchetti **essenziali**.

- "`aptitude search ~E`" elenca i pacchetti **essenziali**.
- "`dpkg -L nome_pacchetto |grep '/man/man.*/'`" elenca le pagine man per i comandi forniti dal pacchetto *nome_pacchetto*.

Suggerimento
Sebbene `moreutils` possa non esistere al di fuori di Debian, offre piccoli programmi interessanti. Quello più degno di nota è sponge(8) che è piuttosto utile quando si desidera sovrascrivere il file originale.

Vedere Sezione 1.6 per esempi.

pacchetto	popcon	dimensione	descrizione
dash	V:929, I:999	207	piccola e veloce shell conforme a POSIX per sh
coreutils	V:912, I:1000	17994	Utilità GNU di base
grep	V:782, I:1000	1297	GNU grep, egrep e fgrep
sed	V:769, I:1000	987	GNU sed
mawk	V:495, I:998	299	awk piccolo e veloce
debianutils	V:935, I:996	228	utilità varie specifiche di Debian
bsdutils	V:421, I:999	338	utilità di base per 4.4BSD-Lite
bsdextrautils	V:761, I:870	382	utilità aggiuntive da 4.4BSD-Lite
moreutils	V:18, I:39	232	utilità Unix aggiuntive

Tabella 12.7: Elenco di pacchetti contenenti piccoli programmi di utilità per script di shell

pacchetto	popcon	dimensione	documentazione
dash	V:929, I:999	207	sh: piccola e veloce shell conforme a POSIX per sh
bash	V:886, I:999	7308	sh: "info bash" fornito da bash-doc
mawk	V:495, I:998	299	AWK: awk piccolo e veloce
gawk	V:247, I:294	3289	AWK: "info gawk" fornito da gawk-doc
perl	V:663, I:990	840	Perl: perl(1) e pagine HTML fornite da perl-doc e perl-doc-html
libterm-readline-gnu-perl	V:2, I:27	439	Estensione Perl per la libreria GNU ReadLine/History: perlsh(1)
libreply-perl	V:0.01, I:0.07	171	REPL per Perl: reply(1)
libdevel-repl-perl	V:0.02, I:0.50	237	REPL per Perl: re.pl(1)
python3	V:720, I:975	81	Python: python3(1) e pagine HTML fornite da python3-doc
tcl	V:24, I:173	20	Tcl: tcl(3) e le pagine dettagliate di manuale fornite da tcl-doc
tk	V:18, I:167	20	Tk: tk(3) e le pagine dettagliate di manuale fornite da tk-doc
ruby	V:72, I:154	32	Ruby: ruby(1), erb(1), irb(1), rdoc(1), ri(1)

Tabella 12.8: Elenco di pacchetti relativi ad interpreti

12.2 Creazione di script in linguaggi interpretati

Quando si desidera automatizzare un compito in Debian, si dovrebbe prima creare uno script per esso con un linguaggio interpretato. Le linee guida per la scelta del linguaggio interpretato sono:

- Usare `dash` se il compito è semplice e combina programmi CLI con un programma della shell.
- Usare `python3` se il compito non è semplice e si sta cercando di scriverlo da zero.
- Usare `perl`, `tcl`, `ruby`, ... se è disponibile del codice preesistente in Debian che usa uno di questi linguaggi e che deve essere ritoccato per fare il compito.

Se il codice risultante è troppo lento, si può riscrivere solo la porzione critica per la velocità d'esecuzione in un linguaggio compilato e chiamarla dal linguaggio interpretato.

12.2.1 Fare il debug di codice di linguaggi interpretati

La maggior parte degli interpreti offre un controllo di base della sintassi e funzionalità di tracciamento del codice.

- "**dash -n** *script.sh*" - Controllo della sintassi di uno script di Shell
- "**dash -x** *script.sh*" - Tracciamento di uno script di Shell
- "**python -m py_compile** *script.py*" - Controllo della sintassi di uno script Python
- "**python -mtrace --trace** *script.py*" - Tracciamento di uno script Python
- "**perl -I ../libpath -c** *script.pl*" - Controllo della sintassi di uno script Perl
- "**perl -d:Trace** *script.pl*" - Tracciamento di uno script Perl

Per testare codice per `dash`, provare Sezione 9.1.4 che è adatto ad ambienti interattivi in stile `bash`.

Per testare codice per `perl` provare l'ambiente REPL per Perl che crea un ambiente REPL (=READ + EVAL + PRINT + LOOP) in stile Python per Perl.

12.2.2 Programma GUI con script di shell

Uno script di shell può essere migliorato per creare un attraente programma GUI. Il trucco è di usare uno dei cosiddetti programmi di dialogo invece di interazioni tristi che usano i comandi `echo` e `read`.

pacchetto	popcon	dimensione	descrizione
`x11-utils`	V:228, I:555	651	xmessage(1): mostra un messaggio o richiesta in una finestra (X)
`whiptail`	V:307, I:997	61	mostra riquadri di dialogo amichevoli da script di shell (newt)
`dialog`	V:8, I:77	520	mostra riquadri di dialogo amichevoli da script di shell (ncurses)
`zenity`	V:68, I:341	193	mostra riquadri di dialogo grafici da script di shell (GTK)
`ssft`	V:0.01, I:0.16	80	Shell Scripts Frontend Tool, strumento per frontend per script di shell (contenitore per zenity, kdialog, e dialog con gettext)
`gettext`	V:49, I:210	20549	"`/usr/bin/gettext.sh`": traduce messaggi

Tabella 12.9: Elenco di programmi per dialoghi

Ecco un esempio di programma con GUI per dimostrare come è semplice facile farlo con uno script di shell.

Questo script usa `zenity` per selezionare un file (in maniera predefinita `/etc/motd`) e visualizzarlo.

L'avviatore GUI per questo script può essere creato seguendo Sezione 9.4.10.

```
#!/bin/sh -e
# Copyright (C) 2021 Osamu Aoki <osamu@debian.org>, Public Domain
# vim:set sw=2 sts=2 et:
DATA_FILE=$(zenity --file-selection --filename="/etc/motd" --title="Select a file to check ↩
    ") || \
  ( echo "E: File selection error" >&2 ; exit 1 )
# Check size of archive
if ( file -ib "$DATA_FILE" | grep -qe '^text/' ) ; then
  zenity --info --title="Check file: $DATA_FILE" --width 640  --height 400 \
    --text="$(head -n 20 "$DATA_FILE")"
else
  zenity --info --title="Check file: $DATA_FILE" --width 640  --height 400 \
    --text="The data is MIME=$(file -ib "$DATA_FILE")"
fi
```

Questo tipo di approccio ad un programma GUI con script di shell è utile solamente per casi di scelta semplice. Se si sta scrivendo un programma con funzioni complesse, considerare l'idea di scriverlo con una piattaforma con più potenzialità.

12.2.3 Azioni personalizzate per un gestore di file con GUI

I programmi di gestione di file con GUI possono essere estesi per effettuare alcune azioni comuni su file selezionati usando pacchetti di estensione aggiuntivi. È anche possibile fargli fare azioni personalizzate specifiche aggiungendo script specifici dell'utente.

- Per GNOME, vedere NautilusScriptsHowto.
- Per KDE, vedere Creating Dolphin Service Menus.
- Per Xfce, vedere Thunar - Custom Actions e https://help.ubuntu.com/community/ThunarCustomActions.
- Per LXDE, vedere Custom Actions.

12.2.4 Pazzie con corti script Perl

Per elaborare dati, `sh` deve generare sottoprocessi che eseguono `cut`, `grep`, `sed`, ecc., ed è lento. D'altro canto, `perl` ha capacità interne per elaborare i dati, perciò è veloce. Molti script di manutenzione del sistema in Debian usano `perl`.

Si consideri il seguente pezzetto di script AWK e i suoi equivalenti in Perl.

```
awk '($2=="1957") { print $3 }' |
```

Ciò equivale ad una qualsiasi delle righe seguenti.

```
perl -ne '@f=split; if ($f[1] eq "1957") { print "$f[2]\n"}' |
```

```
perl -ne 'if ((@f=split)[1] eq "1957") { print "$f[2]\n"}' |
```

```
perl -ne '@f=split; print $f[2] if ( $f[1]==1957 )' |
```

```
perl -lane 'print $F[2] if $F[1] eq "1957"' |
```

```
perl -lane 'print$F[2]if$F[1]eq+1957' |
```

L'ultima è una sorta di indovinello; sfrutta le seguenti caratteristiche di Perl.

- Gli spazi bianchi sono opzionali.
- Esiste una conversione automatica da numero a stringa.
- Trucchi di esecuzione di Perl attraverso opzioni della riga di comando: perlrun(1)
- Variabili speciali di Perl: perlvar(1)

La flessibilità è il punto di forza di Perl. Al contempo, ciò permette di creare codici criptici e ingarbugliati. Perciò fare attenzione.

12.3 Scrivere codice in linguaggi compilati

pacchetto	popcon	dimensione	descrizione
gcc	V:156, I:561	36	Compilatore di GNU C
libc6-dev	V:258, I:580	12863	Libreria GNU C: librerie di sviluppo e file header
g++	V:61, I:529	13	Compilatore di GNU C++
libstdc++-16-dev	V:3, I:19	24398	Libreria GNU Standard C++ v3 (file di sviluppo)
cpp	V:337, I:720	18	Preprocessore GNU C
gettext	V:49, I:210	20549	Utilità GNU Internationalization
glade	V:0.2, I:2.6	1613	Costruttore dell'interfaccia utente GTK
valac	V:0.2, I:3.0	533	Linguaggio in stile C# per il sistema GObject
flex	V:6, I:67	1247	Veloce generatore di analizzatori lessicali compatibile con LEX
bison	V:7, I:71	3122	generatore di parser compatibile con YACC
susv2	I:0.03	16	scarica le specifiche "The Single UNIX Specifications v2"
susv3	I:0.05	16	scarica le specifiche "The Single UNIX Specifications v3"
susv4	I:0.02	16	scarica le specifiche "The Single UNIX Specifications v4"
golang	I:21	11	Compilatore del linguaggio di programmazione Go
rustc	V:5, I:20	13195	Linguaggio di programmazione Rust
gfortran	V:4, I:49	15	Compilatore GNU Fortran 95
fpc	I:2.2	101	Free Pascal

Tabella 12.10: Elenco di pacchetti relativi al compilatore

Qui sono inclusi Sezione 12.3.3 e Sezione 12.3.4 per indicare come programmi simili a compilatori possono essere scritti in linguaggio C compilando descrizioni di più alto livello in linguaggio C.

12.3.1 C

Si può impostare l'ambiente appropriato per compilare programmi scritti nel linguaggio di programmazione C nel modo seguente.

```
# apt-get install glibc-doc manpages-dev libc6-dev gcc build-essential
```

Il pacchetto `libc6-dev`, cioè la libreria GNU C, fornisce la libreria standard C che è una raccolta di file header e routine di libreria usati dal linguaggio di programmazione C.

Vedere come documenti di riferimento per C i seguenti.

- "`info libc`" (documento di riferimento per le funzioni della libreria C)
- gcc(1) e "`info gcc`"
- each_C_library_function_name(3)
- Kernighan & Ritchie, "The C Programming Language", 2nd edition (Prentice Hall)

12.3.2 Semplice programma in C (gcc)

Un semplice esempio "`esempio.c`" può essere compilato con una libreria "`libm`" in un eseguibile "`eseg_esempio`" nel modo seguente.

```
$ cat > example.c << EOF
#include <stdio.h>
#include <math.h>
#include <string.h>

int main(int argc, char **argv, char **envp){
        double x;
        char y[11];
        x=sqrt(argc+7.5);
        strncpy(y, argv[0], 10); /* prevent buffer overflow */
        y[10] = '\0'; /* fill to make sure string ends with '\0' */
        printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
        return 0;
}
EOF
$ gcc -Wall -g -o run_example example.c -lm
$ ./run_example
        1, 2.915, ./run_exam,     (null)
$ ./run_example 1234567890qwerty
        2, 3.082, ./run_exam, 1234567890qwerty
```

In questo esempio, l'uso di "`-lm`" è necessario per fare il link alla libreria "`/usr/lib/libm.so`" nel pacchetto `libc6` per sqrt(3). La libreria reale è in "`/lib/`" con nome file "`libm.so.6`", che è un collegamento simbolico a "`libm-2.7.so`".

Si guardi l'ultimo elemento nel testo di output: ci sono più di 10 caratteri anche se è stato specificato "`%10s`".

L'uso di funzioni che operano su puntatori di memoria senza controlli sui limiti, come sprintf(3) e strcpy(3) è deprecato per prevenire exploit di tipo buffer overflow che sfruttano gli effetti di superamento dei limiti di grandezza dei dati. Usare invece snprintf(3) e strncpy(3).

12.3.3 Flex - un Lex migliorato

Flex è un veloce generatore di analizzatori lessicali compatibile con Lex.

Un tutorial per flex(1) viene fornito da "`info flex`".

Molti esempi semplici possono essere trovati in "`/usr/share/doc/flex/examples/`". 1

12.3.4 Bison - Yacc migliorato

Svariati pacchetti Debian forniscono un generatore di parser LR lookahead o parser LALR combatibile con Yacc.

pacchetto	popcon	dimensione	descrizione
`bison`	V:7, I:71	3122	generatore GNU di parser LALR
`byacc`	V:0.1, I:2.9	263	generatore Berkeley di parser LALR
`btyacc`	V:0.01, I:0.04	247	generatore di parser backtracking basato su `byacc`

Tabella 12.11: Elenco di generatori di parser LALR compatibili con Yacc

1Potrebbero essere necessarie alcune regolazioni per far sì che funzionino nel sistema attuale.

Un tutorial per bison(1) viene fornito da "`info bison`".

È necessario fornire i propri "`main()`" e "`yyerror()`". "`main()`" chiama "`yyparse()`" che a sua volta chiama "`yylex()`", solitamente creato con Flex.

Ecco un esempio di come creare un semplice programma per calcolatrice nel terminale.

Creare `example.y`:

```
/* calculator source for bison */
%{
#include <stdio.h>
extern int yylex(void);
extern int yyerror(char *);
%}

/* declare tokens */
%token NUMBER
%token OP_ADD OP_SUB OP_MUL OP_RGT OP_LFT OP_EQU

%%
calc:
 | calc exp OP_EQU    { printf("Y: RESULT = %d\n", $2); }
 ;

exp: factor
 | exp OP_ADD factor  { $$ = $1 + $3; }
 | exp OP_SUB factor  { $$ = $1 - $3; }
 ;

factor: term
 | factor OP_MUL term { $$ = $1 * $3; }
 ;

term: NUMBER
 | OP_LFT exp OP_RGT  { $$ = $2; }
  ;
%%

int main(int argc, char **argv)
{
  yyparse();
}

int yyerror(char *s)
{
  fprintf(stderr, "error: '%s'\n", s);
}
```

Creare `example.l`:

```
/* calculator source for flex */
%{
#include "example.tab.h"
%}

%%
[0-9]+ { printf("L: NUMBER = %s\n", yytext); yylval = atoi(yytext); return NUMBER; }
"+"    { printf("L: OP_ADD\n"); return OP_ADD; }
"-"    { printf("L: OP_SUB\n"); return OP_SUB; }
"*"    { printf("L: OP_MUL\n"); return OP_MUL; }
"("    { printf("L: OP_LFT\n"); return OP_LFT; }
```

```
")"    { printf("L: OP_RGT\n"); return OP_RGT; }
"="    { printf("L: OP_EQU\n"); return OP_EQU; }
"exit" { printf("L: exit\n");   return YYEOF; } /* YYEOF = 0 */
.      { /* ignore all other */ }
%%
```

Poi eseguire quanto segue dal prompt di shell per testarlo:

```
$ bison -d example.y
$ flex example.l
$ gcc -lfl example.tab.c lex.yy.c -o example
$ ./example
1 + 2 * ( 3 + 1 ) =
L: NUMBER = 1
L: OP_ADD
L: NUMBER = 2
L: OP_MUL
L: OP_LFT
L: NUMBER = 3
L: OP_ADD
L: NUMBER = 1
L: OP_RGT
L: OP_EQU
Y: RESULT = 9

exit
L: exit
```

12.4 Strumenti di analisi statica del codice

Strumenti simili a lint possono aiutare nell'analisi statica del codice.

Strumenti simili ad Indent possono aiutare nella revisione manuale del codice riformattando in modo coerente il codice sorgente.

Strumenti simili a Ctags possono aiutare nella revisione manuale del codice generando un file indice (o tag) dei nomi trovati nel codice sorgente.

Suggerimento
Configurare il proprio editor preferito (`emacs` o `vim`) in modo che usi plugin per motori per lint asincroni aiuta nella scrittura di codice. Questi plugin stanno diventando molto potenti poiché sfruttano il Language Server Protocol. Dato che sono in continuo sviluppo, usare il codice originale a monte, invece dei pacchetti Debian, potrebbe essere una buona opzione.

12.5 Debug

Il debug è un'importante fase del processo di programmazione. Sapere come fare il debug dei programmi rende buoni utenti Debian in grado di creare segnalazioni di bug sensate.

12.5.1 Esecuzione base di gdb

Lo strumento di debug principale in Debian è gdb(1) che permette di ispezionare un programma mentre viene eseguito.

Installare `gdb` e i programmi correlati nel modo seguente.

pacchetto	popcon	dimensione	descrizione
vim-ale	I:0.74	2833	motore per lint asincrono per Vim 8 e NeoVim
vim-syntastic	I:1.9	1379	trucchi per controllo della sintassi per vim
elpa-flycheck	V:0.1, I:1.6	1130	controllo della sintassi al-volo moderno per Emacs
elpa-relint	V:0.00, I:0.04	150	strumento per trovare errori in regexp per Emacs Lisp
cppcheck-gui	V:0.11, I:0.97	7973	strumento per l'analisi statica del codice C/C++ (GUI)
shellcheck	V:4, I:17	22860	strumento lint per script di shell
pyflakes3	V:2, I:14	20	controllo passivo di programmi Python 3
pylint	V:4, I:19	2104	strumento di controllo statico del codice Python
perl	V:663, I:990	840	interprete con controllore interno statico del codice: `B::Lint`(3perl)
rubocop	V:0.1, I:1.0	4502	strumento di analisi statica del codice Ruby
clang-tidy	V:2, I:11	20	strumento per ling per C++ basato su Clang
splint	V:0.05, I:0.79	2325	strumento per controllare staticamente la presenza di bug in programmi C
flawfinder	V:0.05, I:0.43	187	strumento per esaminare codice sorgente C/C++ e per cercare punti deboli per la sicurezza
black	V:3, I:15	5868	formattatore di codice Python senza compromessi
perltidy	V:0.4, I:2.9	3086	strumento per rientri e riformattazione di script Perl
indent	V:0.3, I:4.7	438	programma per formattazione di codice sorgente in linguaggio C
astyle	V:0.2, I:2.5	770	strumento per rientri in codice sorgente per C, C++, Objective-C, C# e Java
bcpp	V:0.02, I:0.22	114	abbellitore di C(++)
xmlindent	V:0.07, I:0.73	52	riformattatore di flussi XML
global	V:0.2, I:1.5	1923	strumenti per ricerca e navigazione di codice sorgente
exuberant-ctags	V:1, I:12	341	costruzione di file indice di tag di definizioni di codice sorgente
universal-ctags	V:1, I:12	4238	costruzione di file indice di tag di definizioni di codice sorgente

Tabella 12.12: Elenco di strumenti per l'analisi statica del codice

pacchetto	popcon	dimensione	documentazione
gdb	V:92, I:162	13472	"`info gdb`" fornito da `gdb-doc`
ddd	V:0.3, I:4.9	5094	"`info ddd`" fornito da `ddd-doc`

Tabella 12.13: Elenco di pacchetti relativi al debug

```
# apt-get install gdb gdb-doc build-essential devscripts
```

Un buon tutorial per gdb può essere trovato su:

- “`info gdb`”
- “Debugging with GDB” in `/usr/share/doc/gdb-doc/html/gdb/index.html`
- “tutorial sul web”

Quello che segue è un piccolo esempio d’uso di gdb(1) su di un ”`programma`” compilato con l’opzione ”`-g`” per produrre informazioni di debug.

```
$ gdb program
(gdb) b 1                # set break point at line 1
(gdb) run args           # run program with args
(gdb) next               # next line
...
(gdb) step               # step forward
...
(gdb) p parm             # print parm
...
(gdb) p parm=12          # set value to 12
...
(gdb) quit
```

Suggerimento
Molti comandi gdb(1) possono essere abbreviati. L’espansione del tasto di tabulazione funziona come nella shell.

12.5.2 Fare il debug di pacchetti Debian

Dato che tutti i binari installati in un sistema Debian dovrebbero avere, in maniera predefinita, le informazioni di debug rimosse, la maggior parte dei simboli di debug non sono presenti nei normali pacchetti. Per poter fare il debug dei pacchetti Debian con gdb(1), è necessario installare i pacchetti `*-dbgsym` (es. `coreutils-dbgsym` in the case of `coreutils`). I pacchetti sorgente generano automaticamente pacchetti `*-dbgsym` insieme ai pacchetti binari normale e tali pacchetti di debug vengono messi nell’archivio separato debian-debug. Per maggiori informazioni fare riferimento agli articoli nel Wiki Debian.

Se un pacchetto di cui si deve fare il debug non fornisce il proprio pacchetto `*-dbgsym`, è necessario installarlo dopo averlo ricompilato nel modo seguente.

```
$ mkdir /path/new ; cd /path/new
$ sudo apt-get update
$ sudo apt-get dist-upgrade
$ sudo apt-get install fakeroot devscripts build-essential
$ apt-get source package_name
$ cd package_name*
$ sudo apt-get build-dep ./
```

Correggere i bug se necessario.

Spostare la versione del pacchetto ad una che non crei conflitti con le versioni ufficiali di Debian, ad esempio una che termini con ”`+debug1`” quando si ricompilano versioni di cui esiste un pacchetto, o una che termini con ”`~pre1`” quando si ricompilano versioni non ancora rilasciate in pacchetti nel modo seguente.

```
$ dch -i
```

Compilare ed installare i pacchetti con i simboli di debug nel modo seguente.

```
$ export DEB_BUILD_OPTIONS="nostrip noopt"
$ debuild
$ cd ..
$ sudo debi package_name*.changes
```

È necessario controllare gli script di compilazione del pacchetto ed assicurarsi di usare "`CFLAGS=-g -Wall`" per la compilazione di binari.

12.5.3 Ottenere un backtrace

Quando un programma va in crash, è una buona idea inviare un segnalazione di bug riportando le informazioni di backtrace.

Il backtrace può essere ottenuto da gdb(1) utilizzando uno dei seguenti approcci:

- Approccio "crash in GDB":
  - Eseguire il programma da GDB.
  - Mandare in crash il programma.
  - Digitare "`bt`" al prompt di GDB.
- Approccio "prima il crash":
  - Aggiornare il file "**/etc/security/limits.conf**" in modo che includa quanto segue:

    ```
    * soft core unlimited
    ```

  - Digitare "`ulimit -c unlimited`" al prompt di shell.
  - Eseguire il programma da questo prompt di shell.
  - Mandare in crash il programma per produrre un file core dump.
  - Caricare il file del core dump in GDB con "`gdb gdb ./program_binary core`".
  - Digitare "`bt`" al prompt di GDB.

Per situazioni con cicli infiniti o tastiera bloccata, si può forzare il crash del programma premendo `Ctrl-\` o `Ctrl-C` o eseguendo "`kill -ABRT` *`PID`*". (Vedere Sezione 9.4.12.)

Suggerimento
Spesso si vede un backtrace in cui una o più delle prime righe sono in "`malloc()`" o "`g_malloc()`". Quando ciò accade è probabile che il backtrace non sia molto utile. Il metodo più semplice per trovare informazioni utili è di impostare la variabile d'ambiente "`$MALLOC_CHECK_`" al valore 2 (malloc(3)). Lo si può fare mentre si esegue `gdb` nel modo seguente.

```
$ MALLOC_CHECK_=2 gdb hello
```

12.5.4 Comandi gdb avanzati

12.5.5 Controllare le dipendenze dalle librerie

Per scoprire le dipendenze di un programma da librerie, usare ldd(1) nel modo seguente.

comando	descrizione degli scopi del comando
`(gdb) thread apply all bt`	ottenere un backtrace per tutti i thread di un programma multi-thread
`(gdb) bt full`	ottenere i parametri nello stack delle chiamate di funzione
`(gdb) thread apply all bt full`	ottenere un backtrace e parametri: combinazione delle due opzioni precedenti
`(gdb) thread apply all bt full 10`	ottenere un backtrace e i parametri per le prime dieci chiamate nello stack per eliminare l'output irrilevante
`(gdb) set logging on`	scrivere un registro dell'output di `gdb` in un file (il file predefinito è "`gdb.txt`")

Tabella 12.14: Elenco di comandi gdb avanzati

```
$ ldd /usr/bin/ls
        librt.so.1 => /lib/librt.so.1 (0x4001e000)
        libc.so.6 => /lib/libc.so.6 (0x40030000)
        libpthread.so.0 => /lib/libpthread.so.0 (0x40153000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Affinché ls(1) funzioni in un ambiente "chroot", le librerie in questione devono essere disponibili nell'ambiente "chroot".

Vedere Sezione 9.4.6.

12.5.6 Strumenti per tracciamento di chiamate dinamiche

In Debian sono disponibili diversi strumenti di tracciamento di chiamate dinamiche. Vedere Sezione 9.4.

12.5.7 Fare il debug di errori X

Se un programma gnome `preview1` ha ricevuto un errore X, si dovrebbe leggere un messaggio del tipo seguente.

```
The program 'preview1' received an X Window System error.
```

Se ciò avviene, si può provare ad eseguire il programma con "`--sync`" ed interrompere alla funzione "`gdk_x_error`" per ottenere un backtrace.

12.5.8 Strumenti per rilevazione di memory leak

In Debian sono disponibili svariati strumenti di rilevazione di memory leak.

12.5.9 Disassemblatore di binari

Si può disassemblare codice binario con objdump(1) nel modo seguente.

```
$  objdump -m i386 -b binary -D /usr/lib/grub/x86_64-pc/stage1
```

Nota
Per disassemblare codice in modo interattivo si può usare gdb(1).

pacchetto	popcon	dimensione	descrizione
libc6-dev	V:258, I:580	12863	mtrace(1): funzionalità di debug malloc in glibc
valgrind	V:4, I:32	89318	strumento di debug e profilazione per la memoria
electric-fence	V:0.2, I:2.2	69	strumento di debug malloc(3)
libdmalloc5	V:0.01, I:0.57	373	libreria per il debug dell'allocazione di memoria
duma	V:0.00, I:0.05	297	libreria per rilevare buffer overrun e underrun in programmi C e C++
leaktracer	V:0.01, I:0.27	56	tracciatore di memory leak per programmi C++

Tabella 12.15: Elenco di strumenti per rilevazione di memory leak

12.6 Strumenti di compilazione

pacchetto	popcon	dimensione	documentazione
make	V:145, I:562	1762	"info make" fornito da make-doc
autoconf	V:27, I:193	2244	"info autoconf" fornito da autoconf-doc
automake	V:26, I:192	1958	"info automake" fornito da automake1.10-doc
libtool	V:23, I:176	1276	"info libtool" fornito da libtool-doc
cmake	V:22, I:123	59735	cmake(1), multipiattaforma, sistema make open source
ninja-build	V:8, I:54	456	ninja(1), piccolo sistema di compilazione, il più vicino in spirito a Make
meson	V:6, I:29	4402	meson(1), sistema di compilazione ad alta produttività sulla base di ninja
xutils-dev	V:0.6, I:6.7	1495	imake(1), xmkmf(1), ecc.

Tabella 12.16: Elenco di pacchetti relativi alla compilazione

12.6.1 Make

Make è un'utilità per mantenere gruppi di programmi. Quando make(1) viene eseguito, make legge il file di regole, "Makefile" e aggiorna il file target se dipende da file prerequisiti che sono stati modificati dall'ultima volta che esso stesso è stato modificato oppure se il file target non esiste. L'esecuzione di questi aggiornamenti può avvenire in modo concorrente.

La sintassi del file di regole è la seguente.

```
target: [ prerequisites ... ]
 [TAB]  command1
 [TAB]  -command2 # ignore errors
 [TAB]  @command3 # suppress echoing
```

Qui "[TAB] è il codice di TAB. Ciascuna riga è interpretata dalla shell dopo la sostituzione delle variabili di make. Usare "\" alla fine di una riga per continuare lo script. Usare "$$" per inserire "$" per valori di ambiente per uno script di shell.

Regole implicite per il target ed i prerequisiti possono essere scritte, per esempio, nel modo seguente.

```
%.o: %.c header.h
```

In questo caso il target contiene il carattere "%" (esattamente un carattere). Il "%" fa corrispondenza con qualsiasi sottostringa non vuota nei nomi di file dei target effettivi. Similmente i prerequisiti usano "%" per mostrare come i loro nomi trovino corrispondenza nei nomi dei target effettivi.

Eseguire "make -p -f/dev/null" per vedere le regole interne automatiche.

variabile automatica	valore
$@	target
$<	primo prerequisito
$?	tutti i prerequisiti più recenti
$^	tutti i prerequisiti
$*	"%" nome base con corrispondenza con il modello target

Tabella 12.17: Elenco di variabili automatiche di make

espansione di variabile	descrizione
`foo1 := bar`	espansione valida una volta sola
`foo2 = bar`	espansione ricorsiva
`foo3 += bar`	accoda

Tabella 12.18: Elenco di espansioni delle variabili di make

12.6.2 Autotools

Autotools è una suite di strumenti per programmazione progettata per assistere nella creazione di pacchetti di codice sorgente portabili su molti sistemi simil-UNIX.

- Autoconf è uno strumento per produrre uno script di shell chiamato "`configure`" da "`configure.ac`".
  - "`configure`" è successivamente usato per produrre "`Makefile`" da un modello "`Makefile.in`".
- Automake è uno strumento per produrre "`Makefile.in`" da "`Makefile.am`".
- Libtool è uno script di shell per affrontare il problema della portabilità del software quando si compilano librerie condivise da codice sorgente.

12.6.2.1 Compilare ed installare un programma



avvertimento
Non sovrascrivere mai file di sistema quando si installano programmi compilati in proprio.

Debian non tocca i file in "`/usr/local/`" o "`/opt`". Perciò se si compila un programma dai sorgenti, installarlo in "`/usr/local/`" in modo che non interferisca con Debian.

```
$ cd src
$ ./configure --prefix=/usr/local
$ make # this compiles program
$ sudo make install # this installs the files in the system
```

12.6.2.2 Disinstallare un programma

Se si hanno i sorgenti originali e questi usano autoconf(1)/automake(1), e se ci si ricorda la configurazione usata, eseguire quanto segue per disinstallare un programma.

```
$ ./configure all-of-the-options-you-gave-it
$ sudo make uninstall
```

In alternativa, se si è assolutamente certi che il processo di installazione mette i file solo in "`/usr/local/`" e lì non c'è nulla di importante, si può cancellare tutto ciò che contiene con la riga di comando seguente.

```
# find /usr/local -type f -print0 | xargs -0 rm -f
```

Se non si è sicuri di dove siano installati i file, si dovrebbe prendere in considerazione l'uso di checkinstall(8) dal pacchetto `checkinstall`, che fornisce un percorso di disinstallazione pulita. Ora permette di creare un pacchetto Debian con l'opzione "`-D`".

12.6.3 Meson

Il sistema di compilazione del software si sta evolvendo:

- Autotools basato su Make è stato lo standard di fatto per le infrustrutture di compilazione portabili sin dagli anni '90. È estremamente lento.
- CMake inizialmente rilasciato nel 2000 ha migliorato significativamente la velocità, ma era ancora costruito sulla base dell'intrinsicamente lento Make. (Ora Ninja può essere il suo backend.)
- Ninja inizialmente rilasciato nel 2012 è pensato per rimpiazzare Make per una velocità di compilazione ulteriormente migliorata ed è progettato per avere i propri file di input generati da un sistema di compilazione di più alto livello.
- Meson rilasciato inizialmente nel 2013 è il nuovo popolare e veloce sistema di compilazione di più alto livello che usa Ninja come suo backend.

Vedere la documentazione in "The Meson Build system" e "The Ninja build system".

12.7 Web

Pagine web dinamiche interattive di base possono essere create nel modo seguente.

- Le interrogazioni vengono presentate all'utente del browser usando moduli HTML.
- La compilazione e il cliccare sulle voci nel modulo invia una delle stringhe URL seguenti con i parametri codificati dal browser al web server.
  - "`https://www.foo.dom/cgi-bin/program.pl?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`"
  - "`https://www.foo.dom/cgi-bin/program.py?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`"
  - "`https://www.foo.dom/program.php?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`"
- "%nn" nell'URL viene sostituito dal carattere con valore esadecimale `nn`.
- Viene impostata la variabile d'ambiente: "`QUERY_STRING="VAR1=VAL1 VAR2=VAL2 VAR3=VAL3"`".
- Il programma CGI (uno qualsiasi dei "`program.*`") sul server web è eseguito con la variabile d'ambiente "`$QUERY_STRING`
- Lo `stdout` del programma CGI viene inviato al browser web ed è presentato come pagina web dinamica interattiva.

Per ragioni di sicurezza è bene non creare a mano nuovi metodi per analizzare i parametri CGI. Per loro esistono moduli Perl e Python comprovati. PHP è fornito con queste funzionalità. Quando è necessaria l'archiviazione dei dati client vengono usati i cookie HTTP. Quando è necessaria l'elaborazione dei dati lato client, viene spesso usato Javascript.

Per maggiori informazioni vedere CGI (Common Gateway Interface), Apache Software Foundation e JavaScript.

Cercare "CGI tutorial" su Google digitando l'URL codificato https://www.google.com/search?hl=en&ie=UTF-8&q=CGI+tutoria direttamente nell'indirizzo del browser è un buon modo per vedere lo script CGI in azione sul server di Google.

12.8 Traduzione di codice sorgente

Esistono programmi per convertire codice sorgente.

pacchetto	popcon	dimensione	parola chiave	descrizione
perl	V:663, I:990	840	AWK → PERL	converte codice sorgente da AWK a PERL:a2p(1)
f2c	V:0.1, I:1.9	443	FORTRAN → C	converte codice sorgente da FORTRAN 77 a C/C++: f2c(1)
intel2gas	V:0.00, I:0.16	178	NASM → GAS	convertitore da NASM (formato Intel) a GAS (GNU Assembler)

Tabella 12.19: Elenco di strumenti per la traduzione di codice sorgente

12.9 Creare pacchetti Debian

Se si desidera creare un pacchetto Debian, leggere i documenti seguenti.

- Capitolo 2 per capire le basi del sistema dei pacchetti
- Sezione 2.7.13 per capire le basi del processo di port
- Sezione 9.11.4 per capire le basi delle tecniche chroot
- debuild(1) e sbuild(1)
- Sezione 12.5.2 per la ricompilazione per il debug
- Guide for Debian Maintainers (il pacchetto `debmake-doc`)
- Debian Developer's Reference (pacchetto `developers-reference`)
- Manuale Debian Policyl (pacchetto `debian-policy`)

Ci sono pacchetti come `debmake`, `dh-make`, `dh-make-perl`, ecc., che aiutano nella creazione dei pacchetti.

Appendice A

Appendice

Ecco il dietro le quinte di questo documento.

A.1 Il labirinto Debian

Il sistema Linux è una piattaforma informatica molto potente per un computer in rete. Tuttavia imparare come usare tutte le sue funzionalità non è semplice. Impostare la coda di stampa LPR con una stampante non PostScript è stato un buon esempio di problema su cui si può inciampare. (Ora non ci sono più problemi dato che le installazioni più recenti usano il nuovo sistema CUPS.)

Esiste una mappa dettagliata completa chiamata ”CODICE SORGENTE”. È molto accurata, ma estremamente difficile da capire. Esistono anche documenti di consultazione chiamati HOWTO e mini-HOWTO; sono più facili da comprendere ma tendono a dare troppi dettagli e a perdere di vista il quadro generale. Io ho a volte dei problemi a trovare la giusta sezione in un lungo HOWTO quando ho bisogno di trovare un paio di comandi da eseguire.

Spero che questa ”Debian Reference (versione 2.152)” (2026-10-10 14:56:25 UTC) possa fornire un buon punto di partenza per le persone nel labirinto Debian.

A.2 Storia del copyright

La Debian Reference è stata iniziata da Osamu Aoki <osamu at debian dot org> come promemoria personale per l’amministrazione di sistema. Molti dei contenuti derivano dalle conoscenze ottenute dalla mailing-list debian-user e da altre risorse Debian.

Seguendo un suggerimento di Josip Rodin, che è stato molto attivo nel DDP (Debian Documentation Project), è stata creata la ”Debian Reference (versione 1, 2001-2007) come parte della documentazione del DDP.

Dopo 6 anni, Osamu ha realizzato che la ”Debian Reference (versione 1)” era sorpassata ed ha iniziato a riscrivere molti dei suoi contenuti. La nuova ”Debian Reference (versione 2)” è stata rilasciata nel 2008.

L’autore ha aggiornato la ”Debian Reference (versione 2)” per affrontare nuovi argomenti (Systemd, Wayland, IMAP, PipeWire, Linux kernel 5.10) e per rimuovere argomenti sorpassati (SysV init, CVS, Subversion, SSH protocol 1, kernel Linux pre-2.5). I riferimenti a situazioni del rilascio Jessie 8 (2015-2020) o precedenti sono stati per lo più rimossi.

Questa ”Debian Reference (versione 2.152)” (2026-10-10 14:56:25 UTC) copre per lo più i rilasci Debian Trixie (`=stable`) e Forky (`=testing`).

Si può rintracciare l’origine e l’ispirazione del tutorial nelle fonti seguenti.

- ”Linux User’s Guide” di Larry Greenfield (dicembre 1996)

  - reso obsoleto dal ”Debian Tutorial”
- ”Debian Tutorial” di Havoc Pennington. (11 dicembre, 1998)
  - scritto in parte da Oliver Elphick, Ole Tetlie, James Treacy, Craig Sawyer e Ivan E. Moore II
  - reso obsoleto da ”Debian GNU/Linux: Guide to Installation and Usage”
- ”Debian GNU/Linux: Guide to Installation and Usage” di John Goerzen ed Ossama Othman (1999)
  - reso obsoleto dalla ”Debian Reference (versione 1)”

Si può rintracciare parte dell’origine e dell’ispirazione per la descrizione dei pacchetti e degli archivi nei documenti seguenti.

- ”Debian FAQ” (versione marzo 2002, quando era mantenuta da Josip Rodin)

Si può rintracciare parte dell’origine e dell’ispirazione per gli altri contenuti nei documenti seguenti.

- ”Debian Reference (versione 1)” di Osamu Aoki (2001–2007)
  - resa obsoleta da questa nuova ”Debian Reference (versione 2)” nel 2008.

La precedente ”Debian Reference (versione 1)” è stata creata con il contributo di molti.

- l’enorme contributo di Thomas Hood sugli argomenti di configurazione della rete
- il significativo contributo di Brian Nelson sugli argomenti relativi ad X ed a VCS
- l’aiuto sugli script build e molte correzioni nei contenuti da parte di Jens Seidel
- l’accurata revisione di David Sewell
- molti aiuti da parte di chi ha tradotto, contribuito e segnalato bug

Molte pagine di manuale e info del sistema Debian, oltre a pagine web di autori a monte e documenti Wikipedia sono stati usati come riferimenti principali per la stesura di questo documento. Nella misura in cui Osamu Aoki ha ritenuto rientrasse nei limiti dell’uso legittimo (fair use), molte parti de esse, specialmente le definizioni dei comandi, sono state riprese letteralmente dopo attenti sforzi editoriali per inserirle nello stile e nello scopo di questo documento.

La descrizione dello strumento di debug gdb è stata estesa usando i contenuti del Debian wiki sul backtrace con il consenso di Ari Pollak, Loïc Minier e Dafydd Harries.

I contenuti della ”Debian Reference (versione 2.152)” (2026-10-10 14:56:25 UTC) sono per la maggior parte miei tranne per quanto specificato in precedenza. Anche questi sono stati aggiornati grazie a contributi.

L’autore, Osamu Aoki, ringrazia tutti coloro che con il loro aiuto hanno reso questo documento possibile.

A.3 Formato del documento

I sorgenti del documento originale inglese sono attualmente scritti in file XML DocBook. Questi sorgenti XML Docbook vengono convertiti in HTML, testo semplice, PostScript e PDF. (Alcuni formati possono non essere saltati per la distribuzione.)