La guida Debian

Osamu Aoki <osamu@debian.org>
Editor: David Sewell <dsewell@virginia.edu>
Traduzione italiana: Davide Di Lazzaro <mc0315@mclink.it>
'Autori' a pagina 257

1.08-3, dom mag 1 07:45:18 UTC 2005

Estratto

La guida Debian (`http://qref.sourceforge.net/`) è pensata per fornire un'ampio panorama del sistema Debian come una **guida dell'utente post-installazione**. Essa copre molti aspetti dell'amministrazione di sistema attraverso esempi di **comandi da shell**. Vengono proposte lezioni base, trucchi e molte altre informazioni su molti argomenti, tra cui i concetti fondamentali tipici di un sistema Debian, suggerimenti per l' installazione, il sistema Debian di gestione dei pacchetti, il kernel sotto Debian, l'affinamento del sistema, la costruzione di un gateway, editor di testo, CVS, programmazione e GnuPG per i **non sviluppatori**.

Avviso di Copyright

Indice

Capitolo 1

Prefazione

La guida Debian (http://qref.sourceforge.net/) è nata per fornire una descrizione a grandi linee del sistema Debian, intesa come guida utente post-installazione. Il lettore a cui è indirizzata è qualcuno interessato a leggere gli script di shell. Mi aspetto che il lettore abbia acquisito delle conoscenze base dei sistemi tipo Unix, prima di leggere questo documento.

Ho fatto una scelta ponderata di **non** spiegare ogni cosa in dettaglio, se questa può essere trovata in una pagina di manuale, o in una pagina info, oppure in un documento HOWTO. Al posto di dare delle spiegazioni complete, ho tentato di fornire delle informazioni pratiche, fornendo le sequenze esatte dei comandi nel testo principale, oppure negli script di esempio reperibili sotto http://www.debian.org/doc/manuals/debian-reference/examples/. Dovete comprendere il contenuto degli esempi prima di dare i comandi. Il vostro sistema potrebbe richidere sequenze di comandi lievemente diverse.

Molte delle informazioni incluse consistono in richiami ai riferimenti più autorevoli, elencati in 'Riferimenti' a pagina 249.

Questo documento è nato come "guida rapida", poi è cresciuto. Nondimeno, **Keep it short and simple (Sii semplice e conciso)** (KISS) è il mio principio guida.

Per un aiuto sulle procedure di emergenza per il vostro sistema, andate subito a 'Comandi di sopravvivenza Debian' a pagina 85.

1.1 Il documento ufficiale

L'ultima versione ufficiale del documento è reperibile negli archivi Debian come pacchetto di nome `debian-reference-it` ed anche reperibile da http://www.debian.org/doc/manuals/debian-reference/.

L'ultima versione in fase di sviluppo è http://qref.sourceforge.net/Debian/. Il progetto è ospitato presso http://qref.sourceforge.net/, dove questo documento è disponibile per essere scaricato in formato testo, HTML, PDF, SGML e PostScript.

1.2 Convenzioni utilizzate

“La guida Debian” fornisce informazioni attraverso brevi comandi `bash`. Le convenzioni utilizzate sono le seguenti:

```
# comando nell'account root
$ comando nell'account utente
... descrizione dell'azione
```

Questi esempi di comandi di shell usano `PS2="  "` Vedere ‘Bash – la shell interattiva standard **GNU**’ a pagina 226 per maggiori informazioni su Bash.

Il riferimento a:

- una **pagina di manuale** in stile UNIX è dato sotto forma di `bash(1)`.
- una **pagina TEXINFO** GNU è dato sotto forma di `info libc`.
- un **libro** è dato sotto forma di *The C Programming Language*.
- una **URL** è dato sotto forma di `http://www.debian.org/doc/manuals/debian-reference/`.
- un **file** nel sistema è dato sotto forma di `/usr/share/doc/Debian/reference/`.

Abbreviazioni:

- **LDP**: Linux Documentation Project (`http://www.tldp.org/`)
- **DDP**: Debian Documentation Project (`http://www.debian.org/doc/`)

Altre abbreviazioni vengono definite nel testo prima di essere usate.

In questo documento per i documenti LDP si forniscono solo i riferimenti alle URL. Tuttavia, dei documenti LDP ne sono stati fatti dei pacchetti per Debian; quando sono installati, i documenti sono disponibili in `/usr/share/doc/HOWTO/`.

Vedere ‘Riferimenti’ a pagina 249.

1.3 Script di esempio

Gli script di esempio (`http://www.debian.org/doc/manuals/debian-reference/examples/`) che accompagnano questo documento nel pacchetto `debian-reference-it` disponibile in `/usr/share/doc/Debian/reference/examples/`. Il punto “.” che precede il nome dei file nascosti è convertito in sottolineatura “_”. Estensioni aggiuntive sono state date quando ci sono altri script di esempio alternativi.

1.4 Impostazione di base

Se il sistema è stato installato con il minimo dei pacchetti, e volete fare l’uso migliore di questo documento, allora è consigliabile dare i seguenti comandi per installare altri pacchetti che contengono documenti utili:

```
# apt-get install info man-db doc-base dhelp apt apt-utils auto-apt \
                  dpkg less mc ssh nano-tiny elvis-tiny vim sash \
                  kernel-package \
                  manpages manpages-dev doc-debian doc-linux-text \
                  debian-policy developers-reference maint-guide \
                  apt-howto harden-doc install-doc \
                  libpam-doc glibc-doc samba-doc exim-doc cvsbook \
                  gnupg-doc
# apt-get install debian-reference # per Sarge, fate anche questo :)
```

Per Woody, aggiungete `exim-doc-html` alla lista precedente. Per Sarge, sostituite `exim-doc` con `exim4-doc-html` e `exim4-doc-info`.

1.5 Le basi delle distribuzioni Debian

Debian mantiene tre distribuzioni diverse simultaneamente. Esse sono:

- `stable` — Ottima per organizzare un server, dato che è l'unica a ricevere aggiornamenti per la sicurezza. Vedere 'La distribuzione `stable`' a pagina 6.
- `testing` — la distribuzione preferita per una workstation, siccome contiene le ultime versioni di software per desktop che ha ricevuto un minimo di prove. Vedere 'La distribuzione `testing`' a pagina 7.
- `unstable` — Sul filo del rasoi. la scelta degli sviluppatori Debian Vedere 'La distribuzione `unstable`' a pagina 7.

Quando i pacchetti in `unstable` non hanno annunci di bachi release-critical (RC) da almeno una settimana o più, vengono automaticamente passati a `testing`.

le distribuzioni Debian hanno dei nomi in codice, come descritto in 'Codice dei nomi della distribuzioni Debian' a pagina 8. Prima del rilascio di Woody, nell'agosto del 2002, le tre distribuzioni erano, rispettivamente, Potato, Woody, e Sid. Dopo il rilascio di Woody erano diventate, Woody, Sarge, e Sid. Quando verrà rilasciata Sarge, le distribuzioni `stable` ed `unstable` diventeranno Sarge e Sid; verrà quindi creata una nuova `testing` (all'inizio come copia di `stable`) e le verrà dato un nuovo nome in codice.

Sottoscrivete alla lista di messaggi a basso volume `debian-devel-announce@lists.debian.org` per annunci importanti su Debian. Vedere 'Gli archivi Debian' a pagina 5.

Se volete usare dei pacchetti con versioni più recenti di quelle rilasciate con la distribuzione che state usando, allora o aggiornate ad una distribuzione più nuova, come descritto in 'Aggiornare una distribuzione a `stable`, `testing` o `unstable`' a pagina 75, oppure aggiornate solo i singoli pacchetti. Se questi non possono essere aggiornati facilmente, allora potreste pensare ad un backport, come descritto in 'Portare un pacchetto nel sistema `stable`' a pagina 92.

Tracciare `testing` può avere come effetto collaterale di ritardare l'installazione dei pacchetti che contengono aggiornamenti per la sicurezza. Questi vengono caricati in `unstable` e migrano in `testing` solo dopo un certo ritardo.

Se mescolate le distribuzioni, p. es. `testing` con `stable` o `unstable` con `stable`, vi troverete alla fine a prendere pacchetti fondamentali, tipo `libc6` da `testing` o `unstable`, senza garanzie che non contengano bachi. Siete stati avvertiti.

Usare `testing` o `unstable` aumenta il rischio di incappare in bachi seri. Potete gestirlo o impiantando uno schema multiboot con una distribuzione Debian più stabile, o con il trucco elegante di usare `chroot`, come descritto in '`chroot`' a pagina 130. Quest'ultimo vi permetterà di far girare distribuzioni Debian diverse contemporaneamente su console differenti.

Dopo una spiegazione sui fondamentali della distribuzione Debian in 'Nozioni fondamentali della Debian' nella pagina successiva, vi verranno fornite alcune informazioni di base per aiutarvi a vivere felicemente con il software più recente, usufruendo dei vantaggi delle distribuzioni `testing` e `unstable` di Debian. Gli impazienti dovrebbero procedere immediatamente a 'Comandi di sopravvivenza Debian' a pagina 85. Felice aggiornamento!

Capitolo 2

Nozioni fondamentali della Debian

Questo capitolo fornisce le informazioni fondamentali sul sistema debian per i non-sviluppatori. Per avere informazioni più autorevoli, vedere:

- Debian Policy Manual
- Debian Developer's Reference
- Debian New Maintainers' Guide

reperibili sotto 'Riferimenti' a pagina 249.

Se state cercando una qualsiasi risposta che li riguarda senza, però, tutti i loro dettagli,andate direttamente a 'Gestione dei pacchetti in Debian' a pagina 79 o ad altri capitoli.

Questo capitolo è formato da documenti presi dalla "Debian FAQ", e profondamente riorganizzati, per permettere ad un qualsiasi amministratore di un sistema Debian di avere un solido punto di partenza.

2.1 Gli archivi Debian

2.1.1 Struttura della directory

Il software impacchettato per la debian, è disponibile in una delle numerose directory su ciascun Mirror Debian (`http://www.debian.org/mirror/`) raggiungibili tramite FTP o HTTP.

Queste sono le directory presenti su ciascun mirror, sotto la directory `/debian/`:

`/dists/`: Contiene le "distribuzioni" ed era il luogo canonico di accesso dei pacchetti disponibili nelle versioni rilasciate e pre-rilascio. Alcuni vecchi pacchetti, i files `Contents-*.gz`, ed i files `Packages.gz` sono ancora qui.

`pool/`: Nuova locazione, che contiene fisicamente tutti i pacchetti, sia quelli della versione rilasciata, che quelli pre-rilascio.

`tools/`: Utilità DOS per creare dischetti boot, partizionare il disco rigido, comprimere/decomprimere i file e lanciare Linux.

`doc/`: La documentazione base, come le FAQ, le istruzioni per la notifica dei bachi, ecc.

`indices/`: I file dei Manutentori, ed i file override.

`project/`: In gran parte materiale solo per sviluppatori, tipo:

> **`project/experimental/`:** Pacchetti e strumenti ancora in via di sviluppo, in fase alfa. I normali utenti non dovrebbero utilizzare i pacchetti qui contenuti, che possono essere pericolosi persino per i più esperti.
>
> **`project/orphaned/`:** Pacchetti lasciati dai loro vecchi manutentori e tolti dalla distribuzione.

2.1.2 Le distribuzioni Debian

Di norma sono tre le distribuzioni contenute nella directory `dists`. Sono definite come la distribuzione `stable`, la `testing` e la `unstable`. Talvolta se ne aggiunge una quarta, la `frozen`. Ogni distribuzione viene definita con un collegamento simbolico alla directory reale, tramite un nome proprio nella directory `dists`.

2.1.3 La distribuzione `stable`

Le voci dei pacchetti per la distribuzione `stable`, Debian Sarge (3.1r0), vengono inserite nella directory `stable` (collegamento simbolico a `sarge/`):

- `stable/main/`: Contiene i pacchetti che costituiscono formalmente il rilascio più recente del sistema Debian.

  Tutti i pacchetti sono totalmente complianti con le Linee guida Debian per il Software Libero (`http://www.debian.org/social_contract#guidelines`) (DFSG) (disponibile anche come `/usr/share/doc/debian/social-contract.txt` installato da `debian-doc`).

- `stable/non-free/`: Contiene i pacchetti la cui distribuzione è in qualche modo limitata, tale da richiedere ai distributori delle cautele dovute ai loro requisiti specifici di copyright.

  Per esempio alcuni pacchetti hanno licenze che ne vietano la distribuzione commerciale. Altri possono essere ridistribuiti, ma sono degli shareware.

- `stable/contrib/`: Contiene i pacchetti che sono di per sè DFSG-free e DFSG-liberi, ma dipendono in qualche modo da un pacchetto che **non è** DFSG-libero.

Ora, in aggiunta alle locazioni sopra descritte, i nuovi pacchetti sono fisicamente localizzati nella directory `pool` (‘La directory `pool`’ a pagina 9).

Lo stato attuale dei bachi della distribuzione `stable` è riportato in sulla pagina Web Problemi di Stable (`http://ftp-master.debian.org/testing/stable_probs.html`).

2.1.4 La distribuzione `testing`

Le voci dei pacchetti per la distribuzione `testing`, Debian Etch, sono registrate nella directory `testing` (collegamento simbolico a `etch`) dopo aver subito un periodo di prova in `unstable`. Ora, in aggiunta alle locazioni sopra descritte, i nuovi pacchetti sono fisicamente localizzati nella directory `pool` (‘La directory `pool`’ a pagina 9). La directory `testing` ha delle sottodirectory, `main`, `contrib` e `non-free`, che hanno le stesse funzioni che in `stable`.

I pacchetti devono essere sincronizzati in tutte le architetture per le quali sono stati compilati e non devono mostrare dipendenze tali da renderli non installabili; devono inoltre avere meno bachi release-critical delle versioni in `unstable`. In questo modo si auspica che `testing` sia sempre molto vicina ad essere candidata al rilascio. Per maggiori dettagli sul meccanismo che regola la distribuzione vedere `http://www.debian.org/devel/testing`.

Lo stato aggiornato della distribuzione `testing` è riportato presso:

- update excuses (`http://ftp-master.debian.org/testing/update_excuses.html`)
- testing problems (`http://ftp-master.debian.org/testing/testing_probs.html`)
- bug release-critical (`http://bugs.debian.org/release-critical/`)
- bug del base system (`http://bugs.qa.debian.org/cgi-bin/base.cgi`)
- bug nei pacchetti standard e dei task (`http://bugs.qa.debian.org/cgi-bin/standard.cgi`)
- altri bug e note sul bug squashing party (`http://bugs.qa.debian.org/`)

2.1.5 La distribuzione `unstable`

Le voci dei pacchetti della distribuzione `unstable`, sempre con nome in codice “Sid”, sono registrate nella directory `unstable` (collegamento simbolico a `sid/`) dopo essere state caricate nell’archivio Debian, rimanendovi finchè non vengono spostate in `testing`. I nuovi pacchetti sono fisicamente localizzati nella directory `pool` (‘La directory `pool`’ a pagina 9). La directory `unstable` ha delle sottodirectory, `main`, `contrib` e `non-free`, che hanno le stesse funzioni che in `stable`.

La distribuzione `unstable` contiene le immagini più recenti del sistema in fase di sviluppo. Gli utenti possono liberamente usare e testare questi pacchetti, ma vengono avvisati del loro precario stato di preparazione. Il vantaggio di usare `unstable` è quello di essere sempre al massimo dell'aggiornamento del progetto Debian relativo al software—siate però pronti a raccogliere i pezzi se qualcosa va storto.

Lo stato aggiornato della distribuzione `unstable` è riportato presso la pagina Web Unstable Problems (`http://ftp-master.debian.org/testing/unstable_probs.html`).

2.1.6 La distribuzione `frozen`

Una volta che la distribuzione testing è sufficientemente matura, diventa frozen; ciò significa che nessun nuovo codice viene più accettato, solo eliminazioni di bachi, se necessari. In aggiunta un nuovo albero testing viene creato nella directory `dists`, con un nuovo nome. La distribuzione frozen passa attraverso un ciclo di test (chiamato appunto "test cycles") di qualche mese caratterizzato da aggiornamenti intermittenti ed importanti stabilizzazioni.(Il recente processo di rilascio di `woody` non ha prodotto un collegamento simbolico `frozen`, così `frozen` non era una distribuzione, ma solo uno stadio di sviluppo della distribuzione `testing`.)

Viene tenuto un registro dei bug della distribuzione frozen che possono impedire il rilascio di un pacchetto o di tutta la distribuzione. Una volta che il conteggio dei bug scende al di sotto di una valore massimo prestabilito, la distribuzione frozen diventa `stable` e viene rilasciata. La precedente distribuzione stable diventa obsoleta (e finisce in archivio).

2.1.7 Codice dei nomi della distribuzioni Debian

I nomi delle directory localizzate fisicamente nella directory `dists`, come `sarge` e `etch`, sono semplicemente dei nomi in codice. Quando una distribuzione Debian è nella fase di sviluppo le viene assegnato un nome in codice e non un numero di versione. Lo scopo di questi nomi è di rendere il mirroring delle distribuzioni Debian più semplice (se, ad esempio, una directory reale come `unstable` cambiasse improvvisamente di nome in `stable`, una gran quantità di programmi dovrebbe essere nuovamente scaricata senza motivo).

Attualmente `stable` è un collegamento simbolico a `sarge` e `testing` è un collegamento simbolico a `etch`. Ciò significa che `Sarge` è la distribuzione attualmente stable e `Etch` è l'attuale testing.

`unstable` è un collegamento simbolico permanente a `sid`, dato che Sid è sempre la distribuzione unstable.

2.1.8 Nomi in codice usati in passato

I nomi in codice che sono già stati utilizzati sono: "Buzz" per la release 1.1, "Rex" per la 1.2, "Bo" per la 1.3.x, "Hamm" per la 2.0, "Slink" per la 2.1, "Potato" per la 2.2, "Woody" per la 3.0 e "Sarge" per la 3.1.

2.1.9 Da dove vengono i nomi delle distribuzioni?

Finora sono stati presi dai nomi dei personaggi del film *Toy Story* della Pixar.

- **Buzz** (Buzz Lightyear) era l'astronauta,
- **Rex** era il tirannosauro,
- **Bo** (Bo Peep) era la bambina che si prese cura della pecorella,
- **Hamm** era il porcellino salvadanaio,
- **Slink** (Slinky Dog) era il cane giocattolo,
- **Potato** era, ovviamente, Mr. Potato,
- **Woody** era il cowboy.
- **Sarge** era il "leader of The Green Plastic Army Men".
- **Etch** (Etch-a-Sketch) era la lavagna.
- **Sid** era il bambino della porta accanto che rompeva i giocattoli.

2.1.10 La directory `pool`

Storicamente i pacchetti erano contenuti nella subdirectory di `dists` corrispondente alla distribuzione di cui facevano parte. Questo portò a vari problemi, tipo un grosso consumo di banda di connessione dei mirror ogni volta che venivano fatti dei cambiamenti di grossa entità.

Ora i pacchetti vengono tenuti in una grossa "vasca" (pool), strutturata in accordo con il nome del pacchetto sorgente. Per rendere il tutto maneggevole, la vasca è suddivisa in sezioni (`main`, `contrib` e `non-free`) e per la prima lettera del nome del pacchetto sorgente. Queste directory contengono svariati file: binari per ciascuna architettura ed i pacchetti sorgente da cui i pacchetti binari sono stati generati.

E' possibile sapere dove ciascun pacchetto è situato eseguendo un comando tipo: `apt-cache showsrc` *`nomemiopacchetto`* ed andando a leggere la riga "Directory:". Per esempio, i pacchetti `apache` sono immagazzinati in `pool/main/a/apache/`. Essendo molteplici, i pacchetti `lib*` vengono trattati in maniera particolare: per esempio, i pacchetti `libpaper` sono immagazzinati in `pool/main/libp/libpaper/`.

Le directory `dists` vengono ancora utilizzate per i file indice usati da programmi tipo `apt`. Inoltre, al momento attuale le vecchie distribuzioni non sono state convertite ad usare le vasche, per cui si troveranno i percorsi contenenti distribuzioni tipo `potato` o `woody` nel campo "Filename" dell'intestazione.

Di norma non avete da preoccuparvi di ciò, poichè il nuovo `apt` e probabilmente il vecchio `dpkg-ftp` sono in grado di gestire la cosa senza problemi. Se volete maggiori informazioni, andate a vedere RFC: implementazione dei pool dei pacchetti (`http://lists.debian.org/debian-devel-announce/2000/debian-devel-announce-200010/msg00007.html`).

2.1.11 Alcune note storiche su Sid

Quando il Sid attuale non esisteva, l'organizzazione dell'archivio Debian aveva un problema principale: l'assunto che quando un'architettura veniva creata nell'attuale `unstable`, sarebbe stata rilasciata quando la distribuzione diventava la nuova `stable`. Però per molte architetture questo non era il caso, con il risultato che quelle directory dovevano essere mosse al momento del rilascio. Fatto poco pratico, poichè lo spostamento avrebbe fagocitato grosse quantità di banda.

Gli amministratori dell'archivio hanno evitato questo problema per pacchetti anni piazzando i binari delle architetture ancora non rilasciate in una directory speciale chiamata `sid`. Al momento del loro rilascio esisteva un collegamento dall'architettura a quel momento `stable` a `sid` e da quel momento in poi essa veniva creata all'interno dell'albero `unstable`, come di norma. Tutto ciò era motivo di confusione per gli utenti.

Con l'avvento della vasca dei pacchetti (vedere 'La directory `pool`' nella pagina precedente) durante lo sviluppo della distribuzione Woody i pacchetti binari cominciarono ad essere immagazzinati in una locazione canonica nella vasca, indipendentemente dalla distribuzione; in tal modo il rilascio di una distribuzione non determina più la grossa dispersione di banda sui mirror (c'è, ovviamente, un notevole consumo, ma graduale, di banda durante la fase di sviluppo).

2.1.12 Pacchetti caricati in `incoming`

I pacchetti che vengono caricati nell'archivio vengono dapprima immagazzinati in `http://incoming.debian.org/` prima di accertarsi che provengano realmente da uno sviluppatore Debian (e vengono piazzati nella sottodirectory `DELAYED` in caso di Non-Maintainer Upload (NMU)). Una volta al giorno, vengono mossi da `incoming` ad `unstable`.

In caso di emergenza, potreste voler installare i pacchetti da qui, prima che raggiungano `unstable`.

2.1.13 Recuperare un vecchio pacchetto

Mentre le distribuzioni Debian più recenti vengono tenute nella directory `debian` su ciascun Mirror Debian (`http://www.debian.org/mirror/`), gli archivi per le distribuzioni più vecchie, tipo Scollegamento , sono tenuti su `http://archive.debian.org/` o sotto la directory `debian-archive` di ciascun mirror Debian.

I vecchi pacchetti `testing` ed `unstable` sono localizzati in `http://snapshot.debian.net/`.

2.1.14 Sezioni per architettura

All'interno di ciascun albero directory principale (`dists/stable/main`, `dists/stable/contrib`, `dists/stable/non-free` `dists/unstable/main`, etc.), le voci dei pacchetti binari risiedono all'interno di sottodirectory i cui nomi indicano l'architettura per la quale sono stati compilati.

- `binary-all/`, per pacchetti architettura-indipendenti. Comprendono, per esempio, scripts Perl o pura documentazione.
- `binary-`*`piattaforma`*`/`, per pacchetti che girano su una particolare piattaforma.

Ricordate che i reali pacchetti binari per `testing` ed `unstable` non risiedono più in queste directory, ma al livello principale della directory `pool`. I file elenco (`Packages` e `Packages.gz`) sono stati comunque mantenuti, per compatibilità con il vecchio sistema.

Per sapere quali architetture sono al momento supportate, leggetevi le Note di Rilascio per ciascuna distribuzione. Possono essere trovate presso i siti delle Note di Rilascio per stable (`http://www.debian.org/releases/stable/releasenotes`) e testing (`http://www.debian.org/releases/testing/releasenotes`).

2.1.15 Il codice sorgente

Il codice sorgente è disponibile per ogni cosa contenuta nel sistema Debian. In più, i termini di licenza della maggior parte dei programmi **richiedono** che il codice venga distribuito insieme ai programmi, o che un'offerta di fornire il codice li accompagni.

Di regola il codice viene reperito nelle directory `source`, che sono in parallelo a tutte le directory dei binari architettura-specifiche, o più di recente alla directory `pool` (vedere 'La directory `pool`' a pagina 9). Per scaricare il codice sorgente senza la necessità di essere addentro alla struttura dell'archivio Debian, provate un comando tipo `apt-get source` *`nomemiopacchetto`*.

Alcuni pacchetti, in particolare `pine`, sono disponibili solamente come sorgenti, a causa delle limitazioni delle licenze. (Recentemente è stato fornito il pacchetto `pine-tracker` per facilitare l'installazione di Pine). Le procedure descritte in 'Portare un pacchetto nel sistema `stable`' a pagina 92 e 'Creare pacchetti debian' a pagina 244 dovrebbero fornire tutto il necessario per compilare un pacchetto manualmente.

Il codice sorgente potrebbe non essere disponibile, invece, per i pacchetti delle directory `contrib` e `non-free`, che formalmente non fanno parte del sistema Debian.

2.2 Il sistema di gestione dei pacchetti Debian

2.2.1 Panoramica dei pacchetti Debian

Normalmente i pacchetti contengono tutti i file necessari all'implementazione di una serie di comandi o di funzionalità. Esistono due tipi di pacchetti:

- **Pacchetti binari**, che contengono eseguibili, file di configurazione, pagine man/info, informazioni sul copyright ed altra documentazione. Questi pacchetti vengono distribuiti in un formato specifico alla Debian (vedere 'Il formato dei pacchetti Debian' in questa pagina); si riconoscono per il suffisso `.deb`. Questi pacchetti possono essere "spacchettati" usando l'utilità tutta Debian `dpkg`; i dettagli si possono vedere alla pagina di manuale corrispondente.
- **Pacchetti sorgente**, che consistono in un file `.dsc` che descrive il pacchetto sorgente (inclusi in nomi dei file seguenti), un file `.orig.tar.gz` che contiene i sorgenti originali non modificati in formato tar gzip ed in genere un file `.diff.gz` che contiene le modifiche specifiche per Debian ai sorgenti originali. L'utilità `dpkg-source` impacchetta e spacchetta questo tipo di pacchetti. Per i dettagli, ovviamente, la pagina man corrispondente.

L'installazione del software attraverso il sistema dei pacchetti utilizza delle "dipendenze", che sono state dichiarate dal responsabile (manutentore) del pacchetto. Le dipendenze vengono descritte nel file `control`, associato a ciascun pacchetto. Ad esempio, il pacchetto contenente il compilatore GNU C (`gcc`) "dipende" dal pacchetto `binutils` che include il collegamento e l'assembler. Se si prova ad installare `gcc` senza aver prima installato `binutils`, il sistema di gestione dei pacchetti (dpkg) invierà un messaggio di errore riguardo alla necessità di avere anche `binutils` e bloccherà l'installazione di `gcc`. (Questo comportamento può comunque essere scavalcato dall'utente tenace, vedere al riguardo `dpkg(8)`.) Per dettagli aggiuntivi, vedere più sotto in 'Dipendenze dei pacchetti' a pagina 16.

Gli strumenti Debian per la gestione dei pacchetti possono essere usati per:

- manipolare e gestire i pacchetti o parte di essi,
- aiutare l'utente nella frammentazione dei pacchetti che devono essere trasmessi con un mezzo di limitate capacità come un floppy,
- aiutare gli sviluppatori nella costruzione degli archivi dei pacchetti e
- aiutare gli utenti nell'installazione dei pacchetti residenti in un archivio remoto Debian.

2.2.2 Il formato dei pacchetti Debian

Un "pacchetto" Debian, od un file dell'archivio Debian contiene gli eseguibili,le librerie e tutta la documentazione associata ad un gruppo o suite di programmi correlati. I file dell'archivio Debian, di norma, hanno il suffisso `.deb`. [1]

---

[1] Il progetto `debian-installer` ha introdotto dei pacchetti i cui nomi finiscono in `.udeb`. In breve, è un formato micro-`.deb` che non necessita di seguire alla lettera le linee guida Debian, manca di alcune parti, tipo la documentazione ed è pensato per essere utlizzato solo dal `debian-installer`, il nuovo programma di installazione Debian, sviluppato per il rilascio di Sarge. I due formati, `.deb` e `.udeb` sono uguali. Il programma `udpkg`, usato per gestire i pacchetti `.udeb`, ha limitate capacità rispetto a `dpkg` e supporta minori relazioni fra i pacchetti. La differenza di nomi sta nel fatto che i manutentori dell'archivio Debian non erano contenti di avere dei `.deb` che non seguissero le regole, per cui venne scelto un nome diverso per accentuare questa diversità e per rendere meno probabile l'installazione accidentale su sistemi reali. Gli `.udeb` vengono usati in un ramdisk iniziale durante l'installazione di base, solo per creare un sistema Debian molto ristretto.

I dettagli dei pacchetti binari Debian sono descritti nella pagina di manuale `deb(5).` Il loro formato interno è soggetto a cambiamenti (tra una versione maggiore e l'altra di Debian), per cui leggete sempre `dpkg-deb(8)` prima di manipolare i `.deb` file.

Almeno fino a Sarge, gli archivi Debian sono sempre stati manipolabili anche dai normali comandi Unix, tipo `ar` e `tar`, anche quando i comandi `dpkg` non erano disponibili.

2.2.3 Convenzioni nei nomi dei pacchetti Debian

Il nome di un pacchetto Debian segue la convenzione seguente:

```
foo_ver-rev_arch.deb
```

Dove in genere *foo* sta per il nome del pacchetto. *ver* è la versione del programma originale, *rev* è il numero di revisione Debian e *arch* è l'architettura per la quale il pacchetto è stato compilato. I file vengono facilmente rinominati, naturalmente. Potete scoprire quale pacchetto è realmente contenuto in un dato file di none *filename* dando il comando seguente:

```
dpkg --info filename
```

Il numero di revisione Debian viene specificato dallo sviluppatore Debian o da chiunque compili il pacchetto. Un cambio nel numero di revisione in genere indica che qualche aspetto nel pacchetto è cambiato.

2.2.4 Mantenimento della configurazione locale

I file che sono considerati modificabili dall'amministratore locale si trovano in `/etc`. Le linee guida Debian prescrivono che tutte le modifiche ai file localmente configurabili vengano mantenute attraverso gli aggiornamenti dei pacchetti.

Se una versione predefinita di un file localmente configurabile viene fornita con il pacchetto stesso, allora il file viene etichettato come un "conffile". Il sistema di gestione dei pacchetti non aggiorna i conffile che sono stati modificati dall'amministratore dopo l'ultima installazione del dato pacchetto senza prima aver chiesto il permesso dell'amministratore stesso. D'altro canto, se il conffile non è stato modificato, allora verrà aggiornato insieme al resto del pacchetto. Ciò è sempre auspicabile, così è vantaggiorso minimizzare le modifiche ai conffile.

Per elencare i conffile appartenenti ad un dato pacchetto, lanciare:

```
dpkg --status package
```

L'elenco segue la riga "Confflies".

Per maggiori informazioni sui conffile potete leggere la sezione del Debian Policy Manual intitolata "Configuration files" (Vedere 'Riferimenti' a pagina 249).

2.2.5 Script di gestione Debian

Gli script di gestione Debian sono degli script eseguibili che vengono lanciati automaticamente prima o dopo l'installazione di un pacchetto. Insieme ad un file chiamato `control`, tutti questi file fanno parte della sezione "control" di un file Debian.

I singoli file sono:

preinst Questo script viene eseguito prima che il pacchetto venga estratto dal file Debian (`.deb`). Molti script "preinst" interrompono i servizi per i pacchetti che devono essere aggiornati fino a che la loro installazione o aggiornamento non sono completati (a seguire dell'esecuzione con successo dello script "postinst").

postinst Questo script tipicamente completa ogni configurazione richiesta da un pacchetto dopo che è stato estratto dal suo file Debian (`.deb`). Spesso gli script "postinst" richiedono all'utente determinate azioni e/o lo avvertono che, qualora accettasse le impostazioni di base, deve ricordarsi di riconfigurare il pacchetto se la situazione lo richiede. Molti script "postinst", poi, eseguono tutti i comandi necessari a lanciare o far ripartire i servizi, dopo che il pacchetto è stato aggiornato o installato.

prerm Questo script ferma tutti i demoni associati con un pacchetto. Viene eseguito prima della rimozione di file associati ad un determinato pacchetto.

postrm Modifica i collegamenti od altri file correlati ad un pacchetto e/o rimuove i files creati da esso.(Vedere anche 'Pacchetti Virtuali' nella pagina successiva.)

Tutti i file di controllo possono essere localizzati nella directory `/var/lib/dpkg/info`. I file correlati con il pacchetto `foo` iniziano, appunto, con il nome "foo" ed hanno le estensioni "preinst", "postinst", ecc. a seconda della funzione. Il file `foo.list` nella stessa directory elenca tutti i file installati con il pacchetto `foo`. (Notate che la localizzazione di questi file è interna a dpkg e può essere soggetta a modifiche.)

2.2.6 Priorità dei pacchetti

Ad ogni pacchetto viene assegnata una **priorità** dai responsabili della distribuzione, come aiuto al sistema di gestione dei pacchetti. Le priorità sono:

- **Richiesto** (Required): pacchetti necessari al corretto funzionamento del sistema.

  Comprende tutti gli strumenti necessari alla riparazione di difetti di sistema. Questi pacchetti non devono essere rimossi, pena la completa inutilizzabilità del sistema, probabilmente nemmeno con `dpkg` si riuscirebbe a mettere le cose a posto. I sistemi con solo i pacchetti Richiesti probabilmente sarebbero inutilizzabili, ma hanno abbastanza funzionalità per permettere all'amministratore di sistema di fare un boot ed installare altri programmi.

- **Importante** (Important): pacchetti che si ritrovano probabilmente su qualsiasi sistema Unix o correlato.

  Altri pacchetti necessari ad un corretto funzionamento del sistema, senza i quali non sarebbe utilizzabile. Tra questi **non** sono inclusi Emacs o X11 o TeX o qualsiasi altra grossa applicazione. Qui si parla di pacchetti che costituiscono l'infrastruttura di base.

- **Standard**: pacchetti comuni su qualsiasi sistema Linux, compreso un sistema ragionevolmente piccolo ma nemmeno troppo limitato all'interfaccia a carattere.

  Questo è ciò che viene installato di base se l'utente non seleziona altro. Non include grosse applicazioni, però include Emacs (più un pezzo di infrastruttura che un'applicazione) ed un ragionevole sottogruppo di TeX e LaTeX (se è possibile senza X).

- **Opzionale** (Optional): pacchetti che comprendono tutto quello di cui potete aver voglia di installare senza nemmeno sapere che cosa è, o se non avete delle necessità particolari.

  Comprende X11, una distribuzione completa di TeX e molte applicazioni.

- **Extra**: pacchetti che o entrano in conflitto con altri di priorità più alta, probabilmente utili se già sapete a che servono, oppure hanno requisiti speciali che li rendono non consoni come "Opzionali".

Notate le differenze fra "Priority: required", "Section: base" ed "Essential: yes" nella descrizione dei pacchetti. "Section: base" significa che il pacchetto viene installato prima tutti su un nuovo sistema. Molti dei pacchetti in "Section: base" hanno "Priority: required" o almenot "Priority: important" e molti di loro sono etichettati con "Essential: yes". "Essential: yes" significa che il pacchetto richiede di specificare un'ulteriore opzione force al sistema di gestione dei pacchetti, tipo `dpkg` quando viene rimosso dal sistema. Per esempio, `libc6`, `mawk` e `makedev` sono "Priority: required" and "Section: base" ma non "Essential: yes".

2.2.7 Pacchetti Virtuali

Il termine pacchetto virtuale è un termine generico che si applica a tutti i pacchetti di un gruppo che provvede alla medesima funzione. Per esempio, i programmi `tin` e `trn` sono entrambi dei newsreader, in grado di soddisfare qualsiasi dipendenza di un programma che richieda un newsreader su un sistema, al fine di funzionare correttamente. Entrambi, quindi, si dice che provvedano il "pacchetto virtuale" definito `news-reader`.

Allo stesso modo `exim` `exim4`, `sendmail` e `postfix` forniscono la funzionalità di un agente di trasporto posta (mail transport agent). Perciò, provvedono al pacchetto virtuale `mail transport agent`. Se uno di loro è installato, qualsiasi programma che dipenda dall'installazione di un agente di trasporto posta vedrà le proprie dipendenze soddisfatte dall'esistenza di questo pacchetto virtuale.

La Debian ha un meccanismo tale che, se più di un pacchetto che fornisce lo stesso pacchetto virtuale è installato, l'amministratore di sistema è in grado di sceglierne uno come pacchetto preferito. Il comando che viene chiamato in causa è`update-alternatives` e verrà descritto in dettaglio oltre, in 'Comandi alternativi' a pagina 96.

2.2.8 Dipendenze dei pacchetti

Il sistema dei pacchetti Debian ha una serie di dipendenze che sono utilizzate per esprimere il fatto che un pacchetto, per funzionare, o per funzionare meglio, ha bisogno dell'installazione di un altro pacchetto:

- Il Pacchetto A **Dipende** dal Pacchetto B se B deve essere assolutamente installato per eseguire A. In alcuni casi, esso noN dipende solo da B, ma da una sua specifica versione. In tal caso la dipendenza dalla versione rappresenta un limite inferiore, nel senso che A dipende da qualsiasi versione di B più recente di quella specificata.
- Il Pacchetto A **Raccomanda** il B, se il responsabile del pacchetto giudica che la maggior parte degli utenti non vorrebbe A senza le funzioni fornite anche da B.
- Il Pacchetto A **Suggerisce** B se B contiene file correlati e che migliorano le funzioni di A. La stessa relazione si esprime dichiarando che il Pacchetto B **Migliora** il Pacchetto B.
- Il Pacchetto A **è in Conflitto** con B quando A non è in grado di funzionare se B è installato nel sistema. Spesso "è in conflitto" è combinato con "Sostituisce".
- Il Pacchetto A **Sostituisce** B quando i file installati da B vengono rimossi o sovrascritti da quelli in A.
- Il Pacchetto A **Fornisce** B quando tutti i file e le funzioni di B vengono incorporate da A.

Informazioni più dettagliate possono essere trovate nel *Packaging Manual* e nel *Policy Manual*.

Notate che `dselect` ha un controllo molto più raffinato sui pacchetti contrassegnati da **Raccomanda** e **Suggerisce** rispetto ad `apt-get`, che prende semplicemente tutti i pacchetti specificati da **Dipende** e lascia quelli indicati da **Raccomanda** e **Suggerisce**. Entrambi i programmi nelle forme più moderne utilizzano come back-end APT.

2.2.9 Cosa significa "Pre-Depends"

`dpkg` conigura sempre un pacchetto da cui ne Dipende un altro prima di configurare quast'ultimo. Tuttavia, `dpkg` in genere spacchetterà il file seguendo un ordine arbitrario, indipendentemente dalle dipendenze. (Spacchettare il file vuol dire che estrarre i file e metterli al posto giusto). Se, però un pacchetto **Pre-Dipende** da un altro, allora quast'ultimo veràà spacchettato e configurato prima che quello che ne Pre-Dipende sia anche solo spacchettato. [2] L'uso di questo tipo di dipendenza è ridotto al minimo.

---

[2] Questo meccanismo è stato previsto per supportare un aggiornamento sicuro di sistemi dal formato `a.out` al formato `ELF`, dove **l'ordine** in cui i pacchetti venivano estratti risultava critico.

2.2.10 Lo stato dei pacchetti

Lo stato di un pacchetto può essere "sconosciuto", "installa", "rimuovi", "elimina" o "mantieni". Queste etichette "voglio", indicano il volere dell'utente riguardo ad un pacchetto (come indicato dalle azioni dell'utente nella sezione "Scegli" di `dselect` o dal richiamo diretto dell'utente di `dpkg`).

Il loro significato è il seguente:

- **sconosciuto** - l'utente non ha mai indicato se vuole il pacchetto
- **installa** - l'utente vuole il pacchetto installato od aggiornato
- **rimuovi** - l'utente vuole che il pacchetto sia rimosso, ma non i suoi file di configurazione.
- **elimina** - l'utente vuole il pacchetto completamente rimosso, compresi i file di configurazione.
- **mantieni** - l'utente non vuole che il pacchetto sia processato, ovvero vuole mantenere la versione attuale con lo stato corrente, qualunque essa sia.

2.2.11 Evitare l'aggiornamento dei pacchetti

Esistono due modi per evitare l'aggiornamento di un pacchetto, tramite `dpkg` o, da Woody in poi, tramite APT.

Con `dpkg`, dovete solo esportare la lista dei pacchetti selezionati con:

```
dpkg --get-selections > selections.txt
```

Dopodichè modificate il file risultante *`selections.txt`*, cambiando la riga che contiene il pacchetto da mantenere, tipo `libc6`, da:

```
libc6                          install
```

a:

```
libc6                          hold
```

Salvate il file e ricaricatelo nel database di `dpkg` con:

```
dpkg --set-selections < selections.txt
```

Se conoscete il nome del pacchetto da mantenere, basta eseguire:

```
echo libc6 hold | dpkg --set-selections
```

Questo processo evita l'aggiornamento dei pacchetti al momento dell'installazione di ciascun file.

Lo stesso risultato si ottiene tramite `dselect`. Basta accedere alla schermata [S]cegli, trovare il pacchetto da mantenere nello stato attuale e premere il tasto '=' (o 'H'). I cambiamenti saranno effettivi non appena lasciata la schermata [S]cegli.

Il sistema APT nella nuova distribuzione Woody ha un meccanismo alternativo per mantenere i pacchetti durante il processo di raccolta di un archivio, utilizzando la `Pin-Priority`. Vedere la pagina di manuale `apt_preferences(5)`, l'`http://www.debian.org/doc/manuals/apt-howto/` o il pacchetto `apt-howto`.

2.2.12 Pacchetti sorgente

I pacchetti sorgente vengono distribuiti in una directory chiamata `source` e possono essere scaricati o manualmente, oppure tramite il comando

```
apt-get source foo
```

(vedere `apt-get(8)` la pagina man su come impostare APT all'uopo).

2.2.13 Compilare pacchetti binari dai sorgenti

Per un dato pacchetto *foo* avete bisogno di tutti i *foo_**`.dsc`, *foo_**`.tar.gz` e *foo_*.diff*`.gz` (nota bene: non esiste nessun `.diff.gz` per un pacchetto Debian nativo).

Una volta presi, se avete installato il pacchetto `dpkg-dev` il seguente comando:

```
$ dpkg-source -x foo_version-revision.dsc
```

estrarrà il pacchetto in una directory denominata *foo-version*.

Date i seguenti comandi per compilare il pacchetto binario:

```
$ cd foo-versione
$ su -c "apt-get update ; apt-get install fakeroot"
$ dpkg-buildpackage -rfakeroot -us -uc
```

poi

```
# su -c "dpkg -i ../foo_version-revision_arch.deb"
```

per installarlo. Vedere 'Portare un pacchetto nel sistema `stable`' a pagina 92.

2.2.14 Creare nuovi pacchetti Debian

Per maggiori dettagli al riguardo, leggete la *New Maintainers' Guide*, reperibile nel pacchetto `maint-guide` oppure presso `http://www.debian.org/doc/manuals/maint-guide/`.

2.3 Aggiornare un sistema Debian

Uno degli scopi della Debian è di fornire un sentiero solido di ed un processo sicuro di aggiornamento. Il sistema di gestione dei pacchetti avverete l'amministratore delle modifiche importanti e talvolta gli chiede di prendere delle decisioni. Dovreste leggere anche le Note di Rilascio; vengono fornite con tutti i CD Debian e sono disponibili sul WWW presso `http://www.debian.org/releases/stable/releasenotes` oppure `http://www.debian.org/releases/testing/releasenotes`.

Una guida pratica viene fornita in 'Gestione dei pacchetti in Debian' a pagina 79. Questa sezione fornisce una panoramica generale, cominciando con gli strumenti di gestione dei pacchetti.

2.3.1 `dpkg`

E' il programma principale per la manipolazione dei pacchetti. Per ulteriori informazioni, leggere la pagina di manuale `dpkg(8)`.

`dpkg` è fornito con parecchi programmi supplementari di base.

- `dpkg-deb`: Manipola i files `.deb`. `dpkg-deb(1)`
- `dpkg-ftp`: Vecchio comando per il recupero dei pacchetti. `dpkg-ftp(1)`
- `dpkg-mountable`: Vecchio comando per il recupero dei pacchetti. `dpkg-mountable(1)`
- `dpkg-split`: Divide grossi pacchetti in files più piccoli. `dpkg-split(1)`

`dpkg-ftp` e `dpkg-mountable` sono stati resi obsoleti dall'introduzione del sistema APT.

2.3.2 APT

APT (Advanced Packaging Tool) è un'interfaccia avanzata per il sistema Debian di gestione dei pacchetti e consiste di vari programmi i cui nomi iniziano tipicamente con "apt-". `apt-get`, `apt-cache` e `apt-cdrom` sono gli strumenti da riga di comando per maneggiare i pacchetti. Funzionano anche come programmi backend per l'utente di altri strumenti, come `dselect` ed `aptitude`.

Per maggiori informazioni, installare `apt` e leggere `apt-get(8)`, `apt-cache(8)`, `apt-cdrom(8)`, `apt.conf(5)`, `sources.list(5)`, `apt_preferences(5)` (Woody), e `/usr/share/doc/apt/guide.html/index.html`.

Esistono fonti di informazione alternative, come APT HOWTO (`http://www.debian.org/doc/manuals/apt-howto/`). Può essere installato tramite `apt-howto` in `/usr/share/doc/Debian/apt-howto/`.

`apt-get upgrade` e `apt-get dist-upgrade` prendono solo i pacchetti elencati sotto "Dipende", mentre lasciano quelli sotto "Raccomanda" e "Suggerisce". Per evitare ciò, usate `dselect`.

2.3.3 `dselect`

Questo programma rappresenta un'interfaccia utente basata su menu al sistema di gestione dei pacchetti. E' particolarmente utile per prime installazioni ed aggiornamenti su larga scala. Vedere '`dselect`' a pagina 82.

Per ulteriori informazioni, installare `install-doc` e leggere `/usr/share/doc/install-doc/dselect-beginner.en.html` oppure Documentazione per dselect per Principianti (`http://www.debian.org/releases/woody/i386/dselect-beginner`).

2.3.4 Aggiornare un sistema in funzione

Il kernel (filesystem) in Debian supporta la sostituzione dei file anche mentre sono in uso. Quando i pacchetti vengono aggiornati, tutti i servizi forniti da essi vengono riavviati se sono configurati per girare nel runlevel corrente. Il sistema Debian non ha bisogno della modalità singolo utente per aggiornare un sistema in funzione.

2.3.5 File `.deb` scaricati e tenuti in cache

Se avete scaricato i pacchetti nel vostro disco rigido (cosa assolutamente non necessaria, vedere sopra per la descrizione di `dpkg-ftp` o di APT), dopo l'installazione dei pacchetti potete rimuoverli dal vostro sistema.

Se si usa APT, i file vengono tenuti nella directory `/var/cache/apt/archives`. Potete cancellarli dopo l'installazione (`apt-get clean`), oppure copiarli sulla stessa directory `/var/cache/apt/archives` di un'altra macchina, per evitare un nuovo download durante la successiva installazione.

2.3.6 Tenere una registrazione dell'aggiornamento

`dpkg` mantiene una registrazione dei pacchetti scompattati, configurati, rimossi e/o eliminati, ma (al momento) non tiene nessuna registrazione dell'attività scritta su terminale durante tali manipolazioni.

Il metodo più semplice per aggirare questo impedimento è di lanciare una qualsiasi sessione di `dpkg`, `dselect` `apt-get`, ecc. all'interno del programma `script(1)`.

2.4 La sequenza di boot della Debian

2.4.1 Il programma `init`

Come ogni buon appartenente alla famiglia degli Unix, Debian esegue il boot eseguendo il programma `init`. Il file di configurazione di `init` (che è `/etc/inittab`) specifica che il primo script da eseguire deve essere `/etc/init.d/rcS`.

Quello che accade poi dipende se è installato il pacchetto `sysv-rc` oppure `file-rc`. Quanto segue assume che sia installato `sysv-rc`. (`file-rc` il proprio script `/etc/init.d/rcS` ed usa un file invece che collegamenti simbolici nelle directory rc per controllare quali servizi siano stati avviati ed in quali runlevel.)

Il file `/etc/init.d/rcS` del pacchetto `sysv-rc` lancia tutti gli script in `/etc/rcS.d/` per eseguire l'inizializzazione, tipo controllo e montaggio dei filesystem, caricamento dei moduli, lancio dei servizi di rete, impostazione dell'orologio, e così via. Poi, per compatibilità, lancia tutti i file (tranne quelli con '.' nel filename) localizzati in `/etc/rc.boot/`. Quest'ultima è riservata all'amministratore di sistema, ed il suo utilizzo è deprecato. Vedere 'Inizializzazione del sistema' a pagina 135 e System run levels and init.d scripts (`http://www.debian.org/doc/debian-policy/ch-opersys#s-sysvinit`) nel Debian Policy Manual per maggiori informazioni.

Debian non usa una directory `rc.local` in stile BSD.

2.4.2 I Runlevel

Dopo il completamento del processo di boot, `init` lancia tutti i servizi configurati per girare nel runlevel predefinito. Questo è definito dalla riga per `id` in `/etc/inittab`. Debian arriva con `id=2`.

Debian usa i seguenti runlevel:

- 1 (modalità singolo utente),
- 2 a 5 (varie modalità multiutente) e
- 0 (arresta il sistema)
- 6 (riavvia il sistema).

I runlevel 7, 8, e 9 possono essere utilizzati, ma le loro directory rc non vengono popolate quando i pacchetti vengono installati.

Scambiate i runlevel mediante il comando `telinit`.

Quando si entra in un runlevel tutti gli script in `/etc/rc`*`runlevel`*`.d/` vengono eseguiti. La prima lettera del nome determina il **modo** in cui lo script viene lanciato: quelli che iniziano con `K` vengono lanciati con l'argomento `stop`. Quelli che iniziano per `S` vengono lanciati con l'argomento `start`. Gli script vengono eseguiti in ordine alfabetico; per cui quelli "stop" vengono lanciati prima di quelli “start” e i numeri a due cifre che seguono `K` o `S` determinano l’ordine in cui venono eseguiti.

Gli script in `/etc/rc`*`runlevel`*`.d` sono infatti semplici collegamenti simbolici agli script in `/etc/init.d/`. Essi accettano anche argomenti tipo “restart” e “force-reload”; questi ultimi metodi possono essere utilizzati dopo che un sistema è stato avviato per riavviare i servizi o forzarli a ricaricare i loro file di configurazione.

Per esempio:

```
# /etc/init.d/exim4 reload
```

2.4.3 Personalizzare i runlevel

La personalizzazione dei runlevel è un compito avanzato di amministrazione di sistema. Il suggerimento seguente vale per gran parte dei servizi.

Per abilitare il servizio *service* nel runlevel *R* create il collegamento simbolico `/etc/rc`*`R`*`.d/S`*`xyservice`* con obiettivo `../init.d/`*`service`*. Il numero di sequenza *xy* dovrebbe essere quello che è stato assegnato al servizio quando il pacchetto è stato installato.

Per disabilitare il servizio, rinominate il the collegamento simbolico in maniera che il nome inizi con `K` invece che con `S` ed il suo numero di sequenza sia 100 meno *xy*.

E’ conveniente usare un editor di runlevel, come `sysv-rc-conf` o `ksysv` per questi scopi.

E’ possibile cancellare il collegamento simbolico `S` ad un servizio in una data directory di un dato runlevel invece di rinominarlo. Ciò non disabilita il servizio, ma lo lascia in uno stato “fluttuante”, finchè il sistema di inizio `sysv-rc` è interessato: al cambio di runlevel il servizio non sarà nè lanciato nè fermato, ma verrà lasciato così com’è, che stia girando o no. Notate comunque che un servizio lasciato in uno stato tale verrà lanciato se il pacchetto corrispondente verrà aggiornato, che girasse o meno prima dell’aggiornamento. Questo è un limite noto del sistema Debian attuale. Notate anche che dovreste mantenere i collegamenti simbolici `K` di un servizio nei runlevel 0 e 6. Se cancellate tutti i collegamenti simbolici di un servizio, allora durante un aggiornamento il pacchetto corrispodente ripristinerà tutti i collegamenti simbolici al loro stato predefinito iniziale.

Not è consigliabile modificare i collegamenti simbolici in `/etc/rcS.d/`.

2.5 Supportare le differenze

Debian offre parecchie opportunità per soddisfare le esigenze (e i desideri) degli amministratori di sistema, senza per questo renderlo inutilizzabile.

- `dpkg-divert`, vedere ‘Il comando `dpkg-divert`’ a pagina 95.
- `equivs`, vedere ‘Il pacchetto `equivs`’ a pagina 96.

- `update-alternative`, vedere 'Comandi alternativi' a pagina 96.
- `make-kpkg` può accettare svariati boot loaders. Vedere `make-kpkg(1)` e 'Il metodo Debian standard' a pagina 99.

Tutti i file in `/usr/local/` appartengono all'amministratore di sistema e Debian non li toccherà. Gran parte dei file in `/etc` sono `conffiles` e Debian non li sovrascriverà in caso di aggiornamento a meno che l'amministratore non lo richieda espressamente.

2.6 Internazionalizzazione

Il sistema Debian è internazionalizzato e fornisce il supporto per la visualizzazione e la scrittura dei caratteri in molte lingue, sia da console che sotto X. Molti documenti, pagine di manuali e messaggi di sistema sono stati tradotti in numero sempre crescente di lingue. Durante l'installazione Debian chiede all'utente di scegliere la lingua di installazione (e talvolta una variante locale della stessa).

Se il vostro sistema non supporta tutte le caratteristiche della lingua di cui avete bisogno, o se dovete cambiare la lingua od installare una diversa tastiera che supporti la vostra lingua, andate a leggere 'Localizzazione (l10n)' a pagina 169.

2.7 Debian ed il kernel

Vedere 'Il kernel Linux su Debian' a pagina 99.

2.7.1 Compilare un kernel da un sorgente non-Debian

Bisogna comprendere le linee guida Debian nei confronti degli header.

Le librerie C Debian sono compilate con le versioni **stabili** più recenti degli header del **kernel**.

Ad esempio, le versione Debian-1.2 usava la versione 5.4.13 degli header. Questa pratica è in contrasto con i pacchetti sorgente del kernel distribuiti in tutti gli archivi Linux FTP, pacchetti che usano versioni persino più recenti degli header. Gli header distribuiti con i sorgenti del kernel sono localizzati in `/usr/include/linux/include/`.

Se avete bisogno di compilare un programma con header più recenti di quelli di quelli forniti da `libc6-dev`, quando compilate dovete aggiungere alla riga di comando `-I/usr/src/linux/include/`. Un problema del genere è uscito, per esempio, quando si è creato il pacchetto del demone automounter (`amd`). Quando i nuovi kernel cambiavano alcune istruzioni relative al NFS, `amd` aveva necessità di esserne al corrente. Ciò ha richiesto l'inclusione degli header più recenti.

2.7.2 Gli strumenti per compilare un kernel personalizzato.

Gli utenti che desiderano (o devono) compilare un kernel personalizzato, sono incoraggiati a scaricare il pacchetto `kernel-package`. Il pacchetto contiene lo script per compilare il pacchetto del kernel e fornisce le capacità di creare un pacchetto Debian kernel-image, semplicemente dando il comando

```
# make-kpkg kernel_image
```

dalla directory principale del kernel sorgente. L'aiuto è disponibile dando il comando

```
# make-kpkg --help
```

o tramite la pagina di manuale `make-kpkg(8)` e 'Il kernel Linux su Debian' a pagina 99.

L'utente deve scaricarsi a parte il sorgente per il kernel, sia esso il più recente o quello di scelta, dall'archivio Linux preferito, a meno che un pacchetto kernel-source-*version* non sia disponibile (dove *version* sta per la versione del kernel). Lo script di boot Debian `initrd` richiede una speciale patch del kernel, chiamata `initrd`; vedere `http://bugs.debian.org/149236`.

Le istruzioni dettagliate per usare il pacchetto `kernel-package` sono fornite nel file `/usr/share/doc/kernel-package/README.gz`.

2.7.3 Boot loader alternativi

Per utilizzare boot loader alternativi, tipo `grub` o `loadlin`, copiate il kernel compilato `bzimage` in un'altra locazione (tipo `/boot/grub` od una partizione MS-DOS).

2.7.4 Boot floppy personalizzato

Questo compito è fortemente aiutato dal pacchetto Debian `boot-floppies`, reperibile normalmente nella sezione `admin` dell'archivio FTP Debian. Gli script di shell di questo pacchetto producono dei boot floppy nel formato `syslinux`. Questi sono floppy formattati MS-DOS i cui master boot record sono stati modificati in maniera tale da fare il boot di Linux (o di qualsiasi altro S.O. sia stato definito nel file `syslinux.cfg` del floppy) direttamente. Altri script del pacchetto producono dei dischi root di emergenza e possono persino riprodurre i dischi base.

Maggiori informazioni in `/usr/doc/boot-floppies/README` dopo l'installazione del pacchetto `boot-floppies`.

2.7.5 Funzioni speciali per trattare con i moduli

Il pacchetto Debian `modconf` fornisce uno script di shell (`/usr/sbin/modconf`) che può essere utilizzato per personalizzare la configurazione dei moduli. Lo script presenta un'interfaccia a menu, chiedendo all'utente particolari circa i device drivers caricabili presenti sul proprio sistema. La risposte vengono utilizzate per personalizzare il file `/etc/modules.conf` (che elenca alias ed altri argomenti che devono essere utilizzati insieme ai vari moduli), tramite i file in `/etc/modutils/`, e `/etc/modules` (che elencano i moduli che devono essere caricati al boot).

Così come i (nuovi) file `Configure.help` ora disponibili per aiutare nella compilazione di kernel personalizzati, il pacchetto `modconf` arriva con tutta una serie di file di aiuto (in `/usr/share/modconf/`) che forniscono informazioni dettagliate sugli argomenti appropriati da dare a ciascun modulo. Vedere 'Kernel 2.4 modulare' a pagina 101 per gli esempi.

2.7.6 Disinstallare un vecchio pacchetto kernel

Si, lo script `kernel-image-NNN.prerm` controlla se il kernel attualmente in uso è lo stesso che state tentando di disinstallare. Perciò potete rimuovere pacchetti kernel che non volete più tramite il comando:

```
# dpkg --purge --force-remove-essential kernel-image-NNN
```

(sostituite *NNN* con la versione ed il numero di revisione del vostro kernel, naturalmente)

Capitolo 3

Suggerimenti per l'installazione di un sistema Debian

La documentazione ufficiale sull'installazione di Debian è localizzata presso `http://www.debian.org/releases/stable/` e `http://www.debian.org/releases/stable/installmanual`.

Per le versioni in fase di sviluppo è localizzata presso `http://www.debian.org/releases/testing/` e `http://www.debian.org/releases/testing/installmanual` (il lavoro è in corso, talvolta potrebbe non essere reperibile)

Sebbene questo capitolo sia stato inizialmente scritto durante i giorni dell'installatore di Potato, gran parte del suo contenuto è stato aggiornato all'installatore di Woody ed essi sono molto simili. Siccome Sarge ne userà uno totalmente nuovo, usate il capitolo come punto di riferimento per quest'ultimo. Anche alcuni pacchetti chiave hanno cambiato nome e priorità. Per esempio, il MTA predefinito di Sarge è `exim4` invece di `exim` e `coreutils` è stato introdotto per sostituire parecchi pacchetti. Potrebbe essere necessario aggiustare le vostre azioni.

3.1 Tracce generali per l'installazione di Linux

Non dimenticate di leggere >url id="http://www.debian.org/CD/netinst/"> se state cercando per un'immagine CD compatta dell'installatore Debian.

Usare le distribuzioni `testing` o `unstable` aumenta il rischio di incappare ne i seri bachi. Questo rischio può essere gestito impiantando uno schema multiboot con una versione di Debian più stabile, oppure usando un trucco elegante fornito da `chroot`, come descritto in '`chroot`' a pagina 130. Quest'ultimo permette di usare versioni differenti della Debian simultaneamente su console diverse.

3.1.1 Le basi della compatibilità hardware

Linux è compatibile con la maggior parte dei componenti per PC esistenti. Può essere installato praticamente in ogni configurazione. Per quanto mi riguarda, l’installazione è stata semplice come per Windows 95/98/Me. La lista di componenti compatibili cresce di giorno in giorno.

Se avete un laptop, andate su: Linux on Laptops (`http://www.linux-laptop.net/`) per indirizzi per l’installazione divisi per marca e modello.

La mia raccomandazione per i componenti di un desktop è, invece, “Siate conservativi.”

- SCSI invece di IDE per il lavoro, IDE/ATAPI HD per uso privato.
- IDE/ATAPI CD-ROM (o CD-RW).
- PCI invece di ISA. In particolare per le NIC.
- Usate NIC economiche. Tulip per PCI, NE2000 per ISA sono più che buone.
- Evitare PCMCIA (notebook) se siete alla prima installazione.
- Evitate le tastiere, mouse, ecc. USB..a meno che vogliate accettare una sfida.

Per macchine lente, bypassarne il disco rigido ed interfacciarle con macchine più veloci per eseguire l’installazione di Linux può essere una buona idea.

3.1.2 Definire i componenti del PC ed il chipset

Durante l’installazione vi verranno richieste informazioni sull’hardware o sui chipset, informazioni non sempre facili da reperire. Ecco un metodo:

1. Aprite il PC e guardate dentro.
2. Scrivete i numeri presenti sui grossi chip sulle schede grafiche, sulle schede di rete, sui chip situati vicino alle porte seriali ed alle porte IDE.
3. Scrivete i nomi scritti sul dorso delle schede PCI ed ISA.

3.1.3 Definire i componenti del PC tramite Debian

Dare i comandi seguenti all’interno di un sistema Linux già installato, fornisce qualche idea sull’hardware presente e sulla sua configurazione.

```
$ pager /proc/pci
$ pager /proc/interrupts
$ pager /proc/ioports
$ pager /proc/bus/usb/devices
```

Questi comandi possono essere dati durante il processo di installazione dalla console, premendo ALT-F2.

Dopo l’installazione iniziale, installando i pacchetti opzionali, tipo `pciutils`, `usbutils`, e `lshw`, potete ottenere informazioni più estese sul sistema.

```
$ lspci -v |pager
$ lsusb -v |pager
# lshw  |pager
```

Uso tipico degli interrupt:

- IRQ0: timer output (8254)
- IRQ1: controller tastiera
- IRQ2: cascata a IRQ8–IRQ15 su PC-AT
- IRQ3: porta seriale secondaria (io-port=0x2F8) (`/dev/ttyS1`)
- IRQ4: porta seriale primaria (io-port=0x3F8) (`/dev/ttyS0`)
- IRQ5: libero [scheda audio (SB16: io-port=0x220, DMA-low=1, DMA-high=5)]
- IRQ6: controller del floppy disk (io-port=0x3F0) (`/dev/fd0`, `/dev/fd1`)
- IRQ7: parport (io-port=0x378) (`/dev/lp0`)
- IRQ8: rtc
- IRQ9: software interrupt (int 0x0A), rediretto a IRQ2
- IRQ10: libero [network interface card (NE2000: io-port=0x300)]
- IRQ11: libero [(SB16-SCSI: io-port=0x340, SB16-IDE: io-port=0x1E8,0x3EE)]
- IRQ12: PS/2 Mouse
- IRQ13: libero (era il coprocessore matematico 80287)
- IRQ14: controller IDE primario (`/dev/hda`, `/dev/hdb`)
- IRQ15: controller IDE secondario (`/dev/hdc`, `/dev/hdd`)

Per le vecchie schede ISA non-PnP, potreste voler impostare gli IRQ5, 10 ed 11 come non-PnP dal BIOS.

Per i dispositivi USB, le classi sono elencate in `/proc/bus/usb/devices` come `Cls=`*`nn`*:

- Cls=00 : Inutilizzato
- Cls=01 : Audio (altoparlanti, ecc.)
- Cls=02 : Comunicazione (MODEM, NIC, . . .)
- Cls=03 : HID (Human Interface Device: KB, mouse, joystick)
- Cls=07 : Stampante
- Cls=08 : Memorie di massa (FDD, CD/DVD, HDD, Flash, . . .)
- Cls=09 : Hub (USB hub)
- Cls=255 : Specifico del construttore

Se la classe non è 255, Linux supporta il dispositivo.

3.1.4 Definire i componenti del PC tramite altri S.O.

Una ulteriore fonte di informazione è rappresentata dagli altri sistemi operativi.

Installate una distribuzione di Linux commerciale. I sistemi di riconoscimento dell’hardware tendono ad essere migliori della Debian, al momento. Le differenze si livelleranno all’introduzione del `debian-installer` con Sarge.

Installate Windows. La configurazione attuale del PC può essere ottenuta cliccando con il pulsante destro del mouse su “Risorse del Computer” e poi “Proprietà”, ottenendo informazioni tipo IRQ, indirizzi di porta I/O, DMA. Alcune vecchie schede ISA potrebbero aver bisogno di essere configurate sotto DOS, ed utilizzate di conseguenza.

3.1.5 Il mito di Lilo

"Lilo è limitato ai primi 1024 cilindri."Sbagliato!

Il nuovo `lilo`, usato nella Debian Potato ha il supporto lba32. Se il BIOS o la scheda madre sono abbastanza recenti da supportare lba32, `lilo` dovrebbe essere in grado di caricarsi oltre il vecchio limite dei 1024 cilindri.

Se avete tenuto il vostro vecchio `lilo.conf`, vi basterà aggiungere una riga con "lba32" da qualche parte vicino all'inizio del file.Vedere `/usr/share/doc/lilo/Manual.txt.gz`

3.1.6 GRUB

Il nuovo boot loader `grub`, proveniente dal progetto GNU HURD, può essere installato su un sistema Debian Woody:

```
# apt-get update
# apt-get install grub-doc
# mc /usr/share/doc/grub-doc/html/
... leggete il contenuto
# apt-get install grub
# pager /usr/share/doc/grub/README.Debian
... leggetelo :)
```

Per modificare il menu di GRUB, aprite `/boot/grub/menu.lst`. Guardate 'Come impostare i parametri di boot (GRUB)' a pagina 110 su come impostare i parametri di avvio durante il processo di inizializzazione, dato che sono lievemente diversi da `lilo`.

3.1.7 Scegliere i floppy di avvio

Per Potato ho apprezzato molto il set di dischi denominato IDEPCI per il desktop. Per Woody, apprezzo il set bf2.4. Entrambi i set usano una versione di `boot-floppies` per creare i floppy di avvio.

Se avete una scheda di rete PCMCIA, dovete usare il set di dischi di avvio standard (numero di floppy maggiore, ma tutti i moduli dei driver disponibili) e configurare la NIC nel setup PCMCIA; non provate ad impostare una card NIC dal dall'interfaccia di configurazione della rete standard.

Per sistemi particolari, potete creare un disco di salvataggio fatto apposta, sostituendo l'immagine del kernel denominata "linux" sul disco di salvataggio Debian con un'altra immagine compilata apposta altrove per la suddetta macchina. I dettagli dell'operazione li trovate sul `readme.txt` sul dischetto di salvataggio. Il floppy è formattato con il filesystem MSDOS, per cui potete utilizzare qualunque sistema per leggerlo e modificarlo. Ciò dovrebbe rendere la vita più semplice a coloro con particolari schede di rete, ecc. . .

Per Sarge, `debian-installer` e/o `pgi` si pensa verranno usati per creare i floppy di avvio.

3.1.8 Installazione

Seguite le istruzioni ufficiali reperibili presso `http://www.debian.org/releases/stable/installmanual` e `http://www.debian.org/releases/testing/installmanual` (il lavoro è in corso, talvolta potrebbe non essere reperibile).

Se installate un sistema tramite i `boot floppies` della distribuzione testing, potreste avere bisogno di aprire un nuovo terminale durante il processo di installazione premendo ALT-F2 e di modificare manualmente le voci di `/etc/apt/sources.list` da "stable" a "testing" per aggiustare le fonti di APT.

Io tendo ad installare `lilo` in posti tipo `/dev/hda3`, mentre `mbr` in `/dev/hda`. Ciò minimizza il rischio di sovrascrivere le informazioni di boot.

Ecco le mie scelte durante l'installazione.

- MD5 passwords "si"
- shadow passwords "si"
- Installazione "avanzata" (dselect **) e scegliere
  - Escludere emacs (se è selezionato), nvi, tex, telnet, talk(d);
  - Includere mc, vim, uno fra nano-tiny od elvis-tiny. Vedere '`dselect`' a pagina 82. Anche se siete dei fanatici di Emacs, evitatelo per ora ed accontentatevi di nano durante l'installazione. Evitate anche di installare altri grossi pacchetti, tipo TEX (Potato faceva questo) a questo stadio. Vedere 'Editor di salvataggio' a pagina 212 per le ragioni per installare nano-tiny od elvis-tiny.
- Rispondo a tutte le domande di configurazione = "y" (sostituisce l'attuale) durante le domande per l'installazione di ciscun pacchetto.
- `exim`: seleziono 2 per la macchina, dato che invio la posta tramite il server SMTP del mio provider.

Per ulteriori informazioni su dselect, vedere '`dselect`' a pagina 82.

3.1.9 Quale host ed IP usare per una LAN

Esempio di configurazione di una LAN (C subnet: 192.168.1.0/24):

```
Internet
   |
   +--- Un ISP esterno fornisce il servizio POP (raggiunto da fetchmail)
   |
  Un punto di accesso dell'ISP fornisce i servizi DHCP e SMTP
   |                     :
  Modem              (Dial-up)
   |                     :
Porta esterna della macchina gateway della LAN: eth0 (IP dato dal DHCP dell'ISP)
 usa un vecchio notebook (IBM Thinkpad, 486 DX2 50 MHz, 20 MB RAM)
 gira Linux con kernel 2.4 e file system ext3.
 gira il pacchetto "ipmasq" (con patch più sicura, NAT e firewall)
```

```
   gira il pacchetto "dhcp-client" configurato per eth0 (scavalca le impostazioni D
   gira il pacchetto "dhcp" configurato per eth1
   gira "exim" come smarthost (modo 2)
   gira "fetchmail" con un intervallo lungo (ripiego)
   gira "bind" come cache name server per Internet dalla LAN
              come name server autoritativo per il dominio LAN dalla LAN
   gira "ssh" sulle porte 22 e 8080 (connessione da ovunque)
   gira "squid" come cache server per l'archivio Debian (per APT)
  Porta interna della macchina gateway della LAN: eth1 (IP = 192.168.1.1, fisso)
                           |
           +--- LAN Switch (10 base T) ---+
           |                              |
  Alcuni clienti IP fissi sulla LAN     Alcuni clienti DHCP sulla LAN
  (IP = 192.168.1.2-127, fisso)    (IP = 192.168.1.128-200, dinamico)
```

Vedere 'Configurare la rete' a pagina 181 per i dettagli sulla configurazione del gateway server. Vedere 'Costruire un gateway router' a pagina 207 per i dettagli sulla configurazione del server gateway per la LAN.

3.1.10 Account utenti

Per avere lo stesso aspetto di "famiglia" tra le varie macchine, i primi account che imposto sono sempre gli stessi.

Il primo account che creo è sempre uno del tipo "admin" (uid=1000). Tutta la posta per root viene reindirizzata lì. Questo account è aggiunto al gruppo `adm` (vedere "'Why GNU `su` does not support the `wheel` group" (Perchè GNU `su` non supporta il gruppo `wheel`)' a pagina 138), al quale viene data una certa quota di privilegi di root tramite `su`, usando PAM, od il comando `sudo`. Vedere 'Aggiungere un account utente' a pagina 49 per i dettagli.

3.1.11 Creare i filesystem

Partizionare il disco rigido

Per limitare i danni in caso di blocco del sistema, preferisco tenere partizioni diverse per directory diverse. Per esempio,

```
  /          == (/ + /boot + /bin + /sbin)
             == 50MB+
  /tmp       == 100MB+
  /var       == 100MB+
  /home      == 100MB+
  /usr       == 700MB+ con X
  /usr/local == 100MB
```

Le dimensioni della directory `/usr` dipendono strettamente dal tipo di applicazione X window. `/usr` può essere di soli 300MB per la sola console, mentre 2GB–3GB non sono inusuali se si vogliono installare varie applicazioni di Gnome. Quando `/usr` diventa troppo grande, la cura più efficace è spostare `/usr/share/` in una partizione diversa. Con i nuovi kernel 2.4 pre-impacchettati, `/` può avere bisogno di più di 200MB di spazio.

Per esempio, la configurazione attuale della mia macchina che funge da Internet gateway, è la seguente (output del comando `df -h`):

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda3             300M  106M  179M  38% /
/dev/hda7             100M   12M   82M  13% /home
/dev/hda8             596M   53M  513M  10% /var
/dev/hda6             100M  834k   94M   1% /var/lib/cvs
/dev/hda9             596M  222M  343M  40% /usr
/dev/hda10            596M  130M  436M  23% /var/cache/apt/archives
/dev/hda11            1.5G  204M  1.2G  14% /var/spool/squid
```

(Le grosse dimensioni di `/var/spool/squid` sono per la funzione di proxy cache e per lo scarico dei pacchetti).

Per dare un’idea della struttura delle partizioni, il seguente è il risultato di `fdisk -l`:

```
# fdisk -l /dev/hda # commenti

/dev/hda1            1        41    309928+   6  FAT16 # DOS
/dev/hda2           42        84    325080   83  Linux # (not used)
/dev/hda3   *       85       126    317520   83  Linux # Main
/dev/hda4          127       629   3802680    5  Extended
/dev/hda5          127       143    128488+  82  Linux swap
/dev/hda6          144       157    105808+  83  Linux
/dev/hda7          158       171    105808+  83  Linux
/dev/hda8          172       253    619888+  83  Linux
/dev/hda9          254       335    619888+  83  Linux
/dev/hda10         336       417    619888+  83  Linux
/dev/hda11         418       629   1602688+  83  Linux
```

Ci sono alcune partizioni inutilizzate. Queste servono per installare una seconda distribuzione di Linux, o come spazio per espandere qualche directory in crescita.

Montare i filesystem

Il montaggio appropriato delle partizioni avviene mediante il seguente `/etc/fstab`:

```
# /etc/fstab: static file system information.
#
# filesystem    mount point     type    options                 dump pass
/dev/hda3       /               ext2    defaults,errors=remount-ro 0 1
/dev/hda5       none            swap    sw                         0 0
proc            /proc           proc    defaults                   0 0
/dev/fd0        /floppy         auto    defaults,user,noauto       0 0
/dev/cdrom      /cdrom          iso9660 defaults,ro,user,noauto    0 0
#
# mantenete le partizioni separate
/dev/hda7       /home           ext2    defaults                   0 2
/dev/hda8       /var            ext2    defaults                   0 2
/dev/hda6       /var/lib/cvs    ext2    defaults                   0 2
# noatime accelera l'accesso ai file in lettura
/dev/hda9       /usr            ext2    defaults,noatime           0 2
/dev/hda10      /var/cache/apt/archives ext2    defaults           0 2
# una partizione molto grande come proxy cache
/dev/hda11      /var/spool/squid ext2   rw                         0 2

# DOS avviabile di backup
/dev/hda1       /mnt/dos        vfat    rw,noauto                  0 0
# Linux avviabile di backup (non ancora fatto)
/dev/hda2       /mnt/linux      ext2    rw,noauto                  0 0
#
# nfs mounts
mickey:/        /mnt/mickey     nfs     ro,noauto,intr             0 0
goofy:/         /mnt/goofy      nfs     ro,noauto,intr             0 0
# minnie:/ /mnt/minnie smbfs ro,soft,intr,credentials={filename} 0 2
```

Per NFS, uso qui le opzioni `noauto,intr` combinate con quella di default `hard`. In tal modo, un processo che si blocca in seguito alla perdita di connessione può essere recuperato mediante Control-C.

Usare `rw,auto,soft,intr` per macchine Windows connesse tramite Samba (smbfs), può essere una buona idea. Vedere 'Configurare Samba' a pagina 40.

Per i floppy, utilizzare `noauto,rw,sync,user,exec` previene la corruzione dei file in caso di rimozione accidentale del disco prima di averlo smontato, però rallenta il processo di scrittura.

Montaggio con autofs

Punti chiave per il montaggio automatico (auto mount):

- Caricare il modulo `vfat` per permettere a `/etc/auto.misc` di contenere `–fstype=auto`:

```
# modprobe vfat # prima del tentativo di accedere al floppy
 ... oppure automatizzate il processo,
# echo "vfat" >> /etc/modules
 ... e riavviate il sistema.
```

- Impostate `/etc/auto.misc` come segue:

```
floppy -fstype=auto,sync,nodev,nosuid,gid=100,umask=000 :/dev/fd0
 ... dove gid=100 è "users".
```

- Create i collegamenti in `/home/`*`user`*, `cdrom` e `floppy`, che puntino a `/var/autofs/misc/cdrom` e `/var/autofs/misc/floppy`, rispettivamente.
- Rendete *user* membro del gruppo "users".

Montaggio del NFS

Il server esterno NFS (goofy) risiede dietro un firewall (gateway). Dato che sono l'unico ad usarla, ho delle regole di sicurezza sulla LAN molto rilassate. Per abilitare l'accesso NFS, il server NFS necessita di aggiungere `/etc/exports` come segue:

```
# /etc/exports: lista di controllo degli accessi per filesystem che possono
#               essere esportati ai client NFS. Vedere exports(5).
/       (rw,no_root_squash)
```

Ciò è richiesto per attivare il server NFS, in aggiunta all'installazione e ad attivazione del server/client NFS.

Per semplicità, in genere creo un partizione singola di circa 2GB per installazioni sperimentali e/o secondarie di linux. Opzionalmente condivido le partizioni di swap e `/tmp`. Per questi scopi lo schema a partizioni multiple è eccessivo. Se vi serve un semplice sistema a console, bastano 500MB.

3.1.12 Linee guida per la memoria DRAM

Di seguito presento alcune (grandi) linee guida per le DRAM.

```
  4MB: Il minimo necessario a far funzionale il kernel di Linux.
 16MB: Il minimo per un sistema con console.
 32MB: Il minimo per un semplice sistema X.
 64MB: Il minimo per X con GNOME/KDE.
128MB: Il giusto per X con GNOME/KDE.
256MB: (o più): Perchè no se ve le potete permettere? Le DRAM sono economiche.
```

L'opzione di boot `mem=4m` (o lilo `append="mem=4m"`) vi mostra come il sistema si comporta con soli 4MB di memoria installati. Il parametro di boot per lilo è necessario per sistemi con vecchi BIOS e più di 64MB di memoria.

3.1.13 Lo spazio di Swap

Uso la formula seguente:
- Ciascuna partizione di swap è di < 128 MB (con vecchi kernel 2.0), < 2 GB (kernel recenti)
- Totale = o (1 o 2 volte la RAM installata) o (da 128 MB a 2 GB) come guida generale
- Spalmatele su dischi diversi e montatele tutte con le opzioni `sw,pri=1` in `/etc/fstab`. Ciò assicura che il kernel faccia una sorta di RAID di partizioni swap ed offre il massimo di prestazioni di swap.
- Usate la porzione centrale del disco rigido quando possibile.

Anche se non ne avrete mai bisogno, un pò di spazio swap (128MB) è comunque desiderabile, in modo che un programma che ha un leak di memoria rallenti progressivamente il sistema prima di bloccarlo definitivamente.

3.2 Configurare Bash

Modifico gli script di inizio della shell a mio piacimento:

```
/etc/bash.bashrc        Sostituire con copia privata
/etc/profile            Mantenere la copia della distribuzione ( \w -> \W)
/etc/skel/.bashrc       Sostituire con copia privata
/etc/skel/.profile      Sostituire con copia privata
/etc/skel/.bash_profile Sostituire con copia privata
~/.bashrc               Sostituire con copia privata per tutti gli account
~/.profile              Sostituire con copia privata per tutti gli account
~/.bash_profile         Sostituire con copia privata per tutti gli account
```

Per i dettagli vedere i miei esempi. (`http://www.debian.org/doc/manuals/debian-reference/examples/`) Mi piace un sistema trasparente, così imposto `umask` a 002 o 022.

`PATH` viene impostato dai seguenti file di configurazione, in questo ordine.

```
/etc/login.defs   - prima del PATH impostato per la shell
/etc/profile      ( può chiamare /etc/bash.bashrc)
~/.bash_profile   ( può chiamare ~/.bashrc)
```

3.3 Configurare il mouse

3.3.1 Mouse PS/2

In caso di un mouse PS/2 con scheda madre ATX, il flusso del segnale deve essere:

```
mouse -> /dev/psaux -> gpm -> /dev/gpmdata = /dev/mouse -> X
```

In questo caso si crea un collegamento simbolico `/dev/mouse` che punta a `/dev/gpmdata` per rendere felici alcune utilità di configurazione e per rendere facile la riconfigurazione. (Per esempio se decidete di non usare per niente il demone gpm, basta puntare il collegamento simbolico `/dev/mouse` a `/dev/psaux` dopo essersi liberati del demone `gpm`.)

Il flusso del segnale permette di deconnettere e reinizializzare la tastiera ed il mouse riavviando `gpm` dopo la riconnessione. X rimarrà funzionante!

Il protocollo del flusso del segnale tra l'output di `gpm` e l'input di X può essere implementato in due modi, come "ms3" (protocollo del mouse seriale Microsoft a 3 pulsanti) o come "raw" (usa lo stesso protocollo del mouse che è connesso) e questa scelta condiziona la scelta del protocollo usato nella configurazione di X.

Mostrerò gli esempi di configurazione usando un mouse Logitech a 3 pulsanti (mouse tradizionale nello stile Unix) PS/2 come modello.

Se siete tra gli sfortunati la cui scheda grafica non è supportata dal nuovo X4 e dovete utilizzare il vecchio X3 (alcune schede ATI 64), configurate `/etc/X11/XF86Config` invece di `/etc/X11/XF86Config-4` come negli esempi seguenti, durante l'installazione dei pacchetti di X3.

Approccio con protocollo ms3

```
/etc/gpm.conf             | /etc/X11/XF86Config-4
==========================+=======================================
device=/dev/psaux         | Section "InputDevice"
responsiveness=           |  Identifier "Configured Mouse"
repeat_type=ms3           |  Driver     "mouse"
type=autops2              |  Option     "CorePointer"
append=""                 |  Option     "Device"    "/dev/mouse"
sample_rate=              |  Option     "Protocol"  "IntelliMouse"
                          | EndSection
```

Usando questo approccio, le modifiche del tipo di mouse vanno fatte solo in `gpm.conf`, mentre la configurazione di X rimane costante. Vedere i miei script di esmpio (http://www.debian.org/doc/manuals/debian-reference/examples/).

Approccio con protocollo raw

```
/etc/gpm.conf             | /etc/X11/XF86Config-4
==========================+=======================================
device=/dev/psaux         | Section "InputDevice"
responsiveness=           |  Identifier "Configured Mouse"
repeat_type=raw           |  Driver     "mouse"
```

```
type=autops2                | Option     "CorePointer"
append=""                   | Option     "Device"   "/dev/mouse"
sample_rate=                | Option     "Protocol" "MouseManPlusPS/2"
                            | EndSection
```

Usando questo approccio, le modifiche del tipo di mouse vanno fatte sia in `gpm.conf` che aggiustando la configurazione di X.

Come adattarsi ai diversi tipi di mouse

Il dispositivo di `gpm` denominato `autops2` si suppone sia in grado di riconoscere la gran parte dei mouse PS/2 sul mercato. Sfortunatamente non sempre funziona e non è disponibile nelle versioni pre-Woody. Provate ad usare `ps2`, o `imps2` in `gpm.conf` al posto di `autops2` in questi casi. Per scoprire quali tipi di mouse gpm conosce date: `gpm -t help`. Vedere `gpm(8)`.

Se utilizzate un mouse a 2 pulsanti PS/2, impostate il protocollo di X in modo da abilitare `Emulate3Buttons`. La differenza di protocollo fra un mouse a 2 pulsanti e quello a 3 viene riconosciuta ed impostata automaticamente da `gpm` dopo aver premuto il pulsante di mezzo una volta.

Per il protocollo X con 'Approccio con protocollo raw' nella pagina precedente o senza `gpm`, usate:

- `IntelliMouse`: mouse seriale (ripetitore `gpm` con "ms3")
- `PS/2`: mouse PS/2 (provatelo sempre, prima)
- `IMPS/2`: qualsiasi mouse PS/2 (2, 3, o con rotella, migliore)
- `MouseManPlusPS/2`: Mouse Logitech PS/2
- `...`

Per saperne di più vedere Supporto per il mouse in XFree86 (`http://www.xfree86.org/current/mouse.html`).

Il mouse con rotella classico Microsoft funziona al meglio con:

```
/etc/gpm.conf               | /etc/X11/XF86Config-4
============================+========================================
device=/dev/psaux           | Section "InputDevice"
responsiveness=             |  Identifier "Configured Mouse"
repeat_type=raw             |  Driver     "mouse"
type=autops2                |  Option     "CorePointer"
append=""                   |  Option     "Device"   "/dev/mouse"
sample_rate=                |  Option     "Protocol" "IMPS/2"
                            |  Option     "Buttons" "5"
                            |  Option   "ZAxisMapping" "4 5"
                            | EndSection
```

Per alcuni nuovi notebook Toshiba ultrapiatti: Attivare `gpm` prima di PCMCIA nello script di inizializzazione del System-V. Ciò evita che il sistema si pianti. Strano, ma vero.

3.3.2 Mouse USB

Accertatevi di avere tutte le funzioni del kernel richieste attivate o alla compilazione, oppure tramite i moduli:

- Sotto "Input core support":
  - "Input core support" (CONFIG_INPUT, `input.o`),
  - "Mouse support" (CONFIG_INPUT_MOUSEDEV, `mousedev.o`),
- Sotto "USB support":
  - "Support for USB" (CONFIG_USB, `usbcore.o`),
  - "Preliminary USB device filesystem" (CONFIG_USB_DEVICEFS),
  - "UHCI" o "OHCI" (CONFIG_USB_UHCI || CONFIG_USB_UHCI_ALT || CONFIG_USB_OHCI, `usb-uhci.o` || `uhci.o` || `usb-ohci.o`),
  - "USB Human Interface Device (full HID) support" (CONFIG_USB_HID, `hid.o`), e
  - "HID input layer support" (CONFIG_USB_HIDINPUT)

Qui, i nomi in minuscolo sono i nomi dei moduli

Se non usate devfs, create un device node `/dev/input/mice` con major 13 e minor 63 come segue:

```
# cd /dev
# mkdir input
# mknod input/mice c 13 63
```

Per i tipici mouse a 3 pulsanti **USB**, le combinazioni di configurazione dovrebbero essere:

```
/etc/gpm.conf            | /etc/X11/XF86Config-4
=========================+======================================
device=/dev/input/mice   | Section "InputDevice"
responsiveness=          |  Identifier "Generic Mouse"
repeat_type=raw          |  Driver     "mouse"
type=autops2             |  Option     "SendCoreEvents" "true"
append=""                |  Option     "Device"   "/dev/input/mice"
sample_rate=             |  Option     "Protocol" "IMPS/2"
                         |  Option     "Buttons" "5"
                         |  Option   "ZAxisMapping" "4 5"
                         | EndSection
```

Vedere il Linux USB Project (`http://www.linux-usb.org/`) per maggiori informazioni.

3.3.3 Touchpad

Sebbene il touchpad di un portatile emuli un mouse PS/2 a 2 tasti come comportamento predefinito, il pacchetto `tpconfig` permette il controllo totale del dispositivo. Per esempio, `OPTIONS="--tapmode=0"` in `/etc/default/tpconfig` disabilta lo scomodo comportamento "click by tap". Impostate `/etc/gpm.conf` come segue per usare sia il touchpad che il mouse esterno USB in console:

```
device=/dev/psaux
responsiveness=
repeat_type=ms3
type=autops2
append="-M -m /dev/input/mice -t autops2"
sample_rate=
```

3.4 Configurare NFS

Impostate NFS tramite `/etc/exports`.

```
# apt-get install nfs-kernel-server
# echo "/ *.domainname-for-lan-hosts(rw,no_root_squash,nohide)" \
        >> /etc/exports
```

Per i dettagli vedere i miei esempi. (http://www.debian.org/doc/manuals/debian-reference/examples/)

3.5 Configurare Samba

Riferimenti:

- http://www.samba.org/
- il pacchetto `samba-doc`

Impostare Samba in modalità "share" è molto più semplice, dato che crea dei dischi di share sul modello di WfW. E' comunque molto meglio l'impostazione in modalità "user".

Samba può essere configurato con `debconf` o `vi`:

```
# dpkg-reconfigure --priority=low samba # in Woody
# vi /etc/samba/smb.conf
```

Per i dettagli vedere i miei esempi. (http://www.debian.org/doc/manuals/debian-reference/examples/)

L'aggiunta di un nuovo utente al file `smbpasswd` può essere fatta tramite `smbpasswd`:

```
$su -c "smbpasswd -a nomeutente"
```

Per la migliore compatibilità, usate password criptate.

Impostate il `os level` sulla base delle equivalenze di sistema seguenti (maggiore il numero, più alta la priorità come server).

```
0: Samba con attitudini molto lasse  (non sarà mai un master browser)
1: Wfw 3.1, Win95, Win98, Win/me?
16: Win NT WS 3.51
17: Win NT WS 4.0
32: Win NT SVR 3.51
33: Win NT SVR 4.0
255: Samba con poteri estesi
```

Accertatevi che gli utenti siano membri del gruppo a cui appartiene la directory che offre l'accesso condiviso e che il percorso alla directory abbia il bit di esecuzione impostato.

3.6 Configurare la stampante

Il metodo tradizionale è `lpr/lpd`. Esiste un nuovo sistema CUPS™ (Common UNIX Printing System). PDQ è un altro approccio. Vedere il Linux Printing HOWTO (`http://www.tldp.org/HOWTO/Printing-HOWTO.html`) per maggiori informazioni.

3.6.1 `lpr/lpd`

Per i vari tipi di spooler `lpr/lpd` (`lpr`, `lprng`, e `gnulpr`), impostate `/etc/printcap` come segue se sono connessi ad una stampante solo testo o PostScript (le basi):

```
lp|alias:\
        :sd=/var/spool/lpd/lp:\
 :mx#0:\
        :sh:\
 :lp=/dev/lp0:
```

Cosa significano le righe precedenti:
- Riga di intestazione: *lp* — nome dello spool, *alias* = alias
- mx#0 — dimensione massima del file illimitata
- sh — sopprime la stampa dell' header di pagina
- lp=/dev/lp0 — stampante locale, oppure porta@host se remota

Questa è una buona configurazione se siete connessi ad una stampante PostScript. In più è una buona configurazione anche se state stampando da una macchina Windows tramite Samba per qualsiasi stampante supportata da Windows (la comunicazione bidirezionale non è supportata). Dovete solo selezionare la configurazione della stampante corrispondente sulla macchina Windows.

Se non avete una stampante PostScript dovete impostare un sistema di filtro usando `gs`. Esistono molti strumenti di autoconfigurazione per l'impostazione di `/etc/printcap`. Una qualsiasi di queste combinazioni è un'opzione:
- `gnulpr`, (`lpr-ppd`) e `printtool` — Io uso questa.

- `lpr` e `apsfilter`
- `lpr` e `magicfilter`
- `lprng` e `lprngtool`
- `lprng` e `apsfilter`
- `lprng` e `magicfilter`

Per lanciare gli strumenti grafici di configurazione, tipo `printtool`, vedere 'Diventare root in X' a pagina 156 per ottenere i privilegi di root. Gli spool creati con `printtool` usano `gs` ed agiscono come stampanti PostScript. Per cui, quando vi accedete, usate i driver delle stampanti PostScript. Sul lato Windows, lo standard è "Apple LaserWriter".

3.6.2 CUPS™

Installazione del Common UNIX Printing System (o CUPS™):

```
# apt-get install cupsys cupsys-bsd cupsys-client cupsys-driver-gimpprint
# apt-get install foomatic-db-engine foomatic-db-hpijs
# apt-get install foomatic-filters-ppds foomatic-gui
```

In Sarge, usando `aptitude`, potete selezionare "Print Server Task".

KDE e Gnome forniscono un ambiente per la facile configurazione delle stampanti. In alternativa, potete configurare il sistema con qualsiasi browser se avete installato `swat` is installed:

```
$ miobrowser http://localhost:631
```

Per esempio, per aggiungere la vostra stampante collegata ad una porta qualsiasi all'elenco delle stampanti accessibili:

- cliccate "Printers" dalla pagina principale e poi "Add Printer",
- date "root" come username e la sua password,
- procedete ad aggiungere la stampante seguendo le istruzioni,
- tornate alla pagina "Printers" e cliccate "Configure Printer" e
- procedete alla configurazione delle dimensioni della carta, risoluzione, e degli altri parametri.

per maggiori informazioni `http://localhost:631/documentation.html` e `http://www.cups.org/cups-help.html`.

Per i kernel 2.4 kernel, vedere anche 'Supporto per la porta parallela' a pagina 105.

3.7 Altri consigli di configurazione per l'host

3.7.1 Installate pochi altri pacchetti dopo l'installazione iniziale

Una volta qui, avete un piccolo, ma funzionante, sistema Debian. E' il momento giusto per installare i pacchetti più grandi.

- Lanciate `tasksel`. Vedere 'Installare i task' a pagina 81.

  Potete scegliere fra i seguenti task, se ne avete bisogno:
  - Utente finale (End-user) — X window system
  - Sviluppo (Development) — C e C++
  - Sviluppo — Python
  - Sviluppo — Tcl/Tk
  - Miscellanea — ambiente TeX/LaTeX
  - Per gli altri, preferisco usare `tasksel` come guida guardando fra i loro componenti elencati sotto <Task Info> ed installandoli selettivamente attraverso `dselect`.
- Lanciate `dselect`.

  La prima cosa che dovreste fare è selezionare il vostro editor preferito e tutti i programmi di cui avete bisogno. Potete installare tutte le varianti di Emacs in un colpo solo. Vedere '`dselect`' a pagina 82 e 'Editor più popolari' a pagina 211.

  Potete anche sostituire alcuni dei pacchetti di default con quelli più completi.
  - lynx-ssh (al posto di lynx)
  - ...
- ...

In genere modifico `/etc/inittab` per un facile spegnimento.

```
...
# Cosa succede quando si preme CTRL-ALT-DEL.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -h now
...
```

3.7.2 Moduli

I moduli per i driver dei vari dispositivi vengono configurati durante l'installazione iniziale. `modconf` fornisce la possibilità di configurare i moduli in un secondo momento attraverso un'interfaccia a menu. Questo programma è estremamente utile quando alcuni moduli vengono tralasciati durante l'installazione iniziale, o quando è stato installato un nuovo kernel in un momento successivo.

Tutti i nomi dei moduli da precaricare devono essere elencati in `/etc/modules`. Io uso anche `lsmod` e `depmod` per controllarli manualmente.

Assicuratevi anche di aggiungere alcune righe a `/etc/modules` per gestire IP masquerading (ftp, ecc.) per i kernel 2.4. Vedere 'Kernel 2.4 modulare' a pagina 101, specificatamente 'Funzioni di rete' a pagina 102.

3.7.3 Impostazione base del CD-RW

Per i masterizzatori IDE e kernel 2.4, modificate i seguenti file:

```
/etc/lilo.conf  (aggiungere append="hdc=ide-scsi ignore=hdc"
                 lanciate lilo per attivarlo)
/dev/cdrom      (collegamento simbolico # cd /dev; ln -sf scd0 cdrom)
/etc/modules (aggiungere "ide-scsi" e "sg". "sr" a seguire, se necessario.)
```

Vedere 'Masterizzatori' a pagina 141 per i dettagli.

3.7.4 Grosse memorie e spegnimento automatico

Modificate `/etc/lilo.conf` come segue per impostare i parametri al boot in caso di grosse memorie (per i kernel 2.2) e per lo spegnimento automatico (per apm):

```
append="mem=128M apm=on apm=power-off noapic"
```

Lanciate `lilo` per attivare queste nuove impostazioni. <`apm=power-off` è necessario per un kernel SMP e `noapic` è necessario per ridurre i problemi del mio hardware SMP un pò bacato. Si può fare lo stesso direttamente al boot prompt. Vedere 'Altri trucchi con il boot prompt' a pagina 109.

Se apm è compilato come modulo, come avviene automaticamente nei kernel Debian 2.4, lanciate `# insmod apm power_off=1` dopo il boot, oppure impostate `/etc/modules` con:

```
# echo "apm power_off=1" >>/etc/modules
```

In alternativa, compilare il supporto ACPI ottiene lo stesso scopo con i kernel più nuovi, e sembra essere più "amichevole" con SMP (richiede una scheda madre recente). I kernels 2.4 su schede madri più recenti dovrebbero riconoscere correttamente le grosse memorie.

```
CONFIG_PM=y
CONFIG_ACPI=y
...
CONFIG_ACPI_BUSMGR=m
CONFIG_ACPI_SYS=m
```

ed aggiungete le seguenti righe a `/etc/modules` in quest'ordine.

```
ospm_busmgr
ospm_system
```

Oppure ricompilate il kernel con tutte le opzioni precedenti su "y". Nel caso dell'ACPI, non serve alcun parametro di boot.

3.7.5 Strani problemi di accesso con alcuni siti web

I kernel Linux recenti attivano l'ECN di default, cosa che può causare problemi di accesso ad alcuni siti web con dei cattivi routers. Per controllare lo stato dell'ECN:

```
# cat /proc/sys/net/ipv4/tcp_ecn
 ... oppure
# sysctl net.ipv4.tcp_ecn
```

Per disattivarlo usate:

```
# echo "0" > /proc/sys/net/ipv4/tcp_ecn
 ... oppure
# sysctl -w net.ipv4.tcp_ecn=0
```

Per disabilitare TCP ECN ad ogni boot, aprite `/etc/sysctl.conf` ed aggiungete:

```
net.ipv4.tcp_ecn = 0
```

3.7.6 Configurazione di PPP per una connessione dial-up

Installate il pacchetto `pppconfig` per impostare un accesso dial-up con PPP.

```
# apt-get install pppconfig
# pppconfig
 ... seguite le istruzioni per configurare PPP
# adduser nome_utente dip
 ... permette a nome_utente di accedere a PPP
```

L'accesso dial-up PPP può essere iniziato dall'utente (*nome_utente*):

```
$ pon nome_IP   # inizia l'accesso PPP al vostro IP
 ... gustatevi Internet
$ poff nome_ISP # termina l'accesso PPP, nome_ISP è facoltativo
```

Vedere 'Configurare un'interfaccia PPP' a pagina 185 per maggiori dettagli.

3.7.7 Altri file di configurazione da perfezionare in `/etc`

Potreste voler aggiungere un file `/etc/cron.deny`, che manca nella installazione Debian standard (potete copiare `/etc/at.deny`).

Capitolo 4

Lezioni di Debian

Questa sezione fornisce delle coordinate di base per orientarsi nel mondo Debian, mirate all'utente inesperto. Se è un pò di tempo che utilizzate un sistema Unix-like, probabilmente saprete già tutto di quello che verrà spiegato qui. In tal caso, usatela come ripasso.

4.1 Per iniziare

Dopo aver installato Debian sul vostro PC, dovete imparare alcune cose per renderlo utile. Partiamo a tambur battente con le lezioni.

4.1.1 Login come root al prompt dei comandi

Dopo il reboot vi si presenterà o la schermata grafica di accesso al sistema, oppure quella a carattere, a seconda della vostra selezione iniziale dei pacchetti. Per semplicità, se avete davanti la schermata grafica, premete Ctrl-Alt-F1 [1] per accedere alla schermata di accesso a carattere.

Supponiamo che il vostro hostname sia *foo*, il prompt del login sarà così:

```
foo login:
```

Digitate `root` , premete il tasto Enter e digitate la password scelta durante il processo di installazione. In Debian,come da tradizione del mondo Unix, la password è dipendente anche dal maiuscolo/minuscolo. Dopo di ciò, il sistema parte con il messaggio di benvenuto e vi presenta il prompt dei comandi di root in attesa di un vostro comando. [2]

```
foo login: root
```

---

[1] I tasti Ctrl-sinistro, Alt-sinistro ed F1 vanno premuti insieme.

[2] Notate che, se avete modificato il messaggio di benvenuto in in `/etc/motd`, this will be different.

```
Password:
Last login: Sun Oct 26 19:04:09 2003 on tty3
Linux foo 2.4.22-1-686 #6 Sat Oct 4 14:09:08 EST 2003 i686 GNU/Linux

Most of the programs included with the Debian GNU/Linux system are
freely redistributable; the exact distribution terms for each program
are described in the individual file in /usr/share/doc/*/copyright

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

root@foo:root#
```

A questo punto siete pronti per amministrare il sistema dal prompt dei comandi di root. L'account root è anche chiamato utente privilegiato o superuser. Tramite questo account potete fare qualsiasi cosa:

- leggere, modificare e rimuovere qualsiasi file del sistema, indipendentemente dai loro permessi
- impostare i permessi ed il proprietario di qualsiasi file del sistema
- impostare la password di qualsiasi utente senza privilegi nel sistema
- accedere a qualsiasi account senza bisogno di password

E' un'idea veramente pessima condividere l'accesso all'account di root condividendone la password. Il modo giusto di condividere i privilegi dell'amministratore è mediante l'uso di programmi come `sudo(8)`.

Notate che è considerata una buona abitudine in ambito Unix accedere prima ad un account senza privilegi, anche quando si pensa di eseguire delle attività di amministrazione di sistema. Utilizzate i comandi `sudo`, `super`, o `su -c` per guadagnare dei privilegi di root limitati quando ne avete necessità. Vedere '`sudo` – lavorare con maggiore sicurezza' a pagina 139. [3]

4.1.2 Impostare un ambiente minimale per il novizio

Credo che imparare ad usare un sistema operativo sia come imparare una nuova lingua. Sebbene le guide siano utili, uno deve fare pratica con strumenti che lo possano aiutare. In questo contesto, credo sia una buona idea installare alcuni pacchetti aggiuntivi, tipo `mc`, `vim`, `lynx`, `doc-linux-text` e `debian-policy`. [4]

```
# apt-get update
 ...
# apt-get install mc vim lynx doc-linux-text debian-policy
 ...
```

---

[3] Devo ammettere che ho utilizzato l'account di root più spesso di quanto necessario, solo perchè era più comodo ed io poco scrupoloso.

[4] Può essere una buona idea installare anche `gpm`, `emacs21` e `doc-linux-html`. Vedere 'Configurare il mouse' a pagina 36 e 'Gli Editor' a pagina 211.

Se avete questi pacchetti già installati, non accadrà nulla.

4.1.3 Aggiungere un account utente

Durante l'installazione avrete creato un account utente senza privilegi, che utilizzerete per ricevere le mail inviate all'account di root. [5] Dato che non volete usare questo account speciale per le esercitazioni che andrete a fare, dovrete creare un nuovo account utente.

Supponiamo che vogliate come username *`penguin`*, digitate:

```
root@foo:root# adduser penguin
... rispondete a tutte le domande
```

eccolo creato. [6] Prima di andare avanti, impariamo alcune cose.

4.1.4 Muoversi tra le console virtuali

In un sistema Debian standard esistono 6 pseudo-terminali disponibili, cioè potete usare lo schermo a carattere VGA del PC come 6 terminali VT-100 intercambiabili. Potete passare da uno all'altro premendo simultaneamente i tasti Alt-sinistro ed uno dei tasti F1–F6. Ciascun (pseudo)terminale permette i login indipendenti agli account. L'ambiente multiutente è una grande ed avvincente caratteristica di Unix.

Se premete accidentalmente Alt-F7 su un sistema con X Window System e lo schermo mostra l'interfaccia grafica di login, potete riguadagnare l'accesso all'interfaccia a carattere premendo Ctrl-Alt-F1. Per abituarvi, provate a muovervi tra le varie console e a tornare indietro su quella di partenza.

4.1.5 Lo spegnimento del PC

Come qualsiasi moderno sistema operativo, in cui i file vengono tenuti in memoria, il sistema Debian una procedura di arresto appropriata, prima che l'interruttore possa essere spento con sicurezza, per mantenere l'integrità dei file. Date il seguente comando dal prompt di root per spegnere il sistema:

```
# shutdown -h now
```

Questo è per la normale modalità multiutente. In modalità singolo utente, sempre da prompt di root, è invece:

---

[5] Tendo a chiamare questo account `admin`, ma potete dargli il nome che preferite.

[6] Potete aggiungere l'utente *`penguin`* al gruppo `adm` per concedere l'accesso il lettura ai molti file log contenuti in `/var/log/`. Vedere `passwd(5)`, `group(5)`, `shadow(5)`, `group(5)`, `vipw(8)` e `vigr(8)`. Per i significati ufficiali di utente e gruppo leggetevi una versione recente del documento Users and Groups (`/usr/share/doc/base-passwd/users-and-groups.html`).

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

In alternativa, potete premere Ctrl-Alt-Delete (o Ctrl-Alt-Canc per la tastiera italiana) per spegnere. [7]

Attendete finchè non appare la scritta "System halted", poi spegnete il computer. Se le funzioni APM o ACPI sono state attivate dal BIOS e su Linux, il sistema si spegnerà da solo. Per i dettagli, vedere 'Grosse memorie e spegnimento automatico' a pagina 44.

4.1.6 E' tempo di giocare

Ora siete pronti per giocare con Debian senza rischi, finchè utilizzate l'account utente senza privilegi *penguin*. [8]

Accediamo come *penguin*. Se eravate al prompt di root, premete Ctrl-D [9] per chiudere l'attività della shell di root e tornare al prompt del login. Inserite lo username appena creato *penguin* e la password corrispondente. [10] Vi si presenterà il seguente prompt dei comandi.

```
penguin@foo:penguin$
```

Da ora in poi l'esempio utilizzerà dei prompt semplificati per maggiore chiarezza. Userò:

- `#` : prompt della shell di root
- `$` : prompt della shell di un utente senza privilegi

Cominceremo ad imparare il sistema Debian nel modo più facile 'Midnight commander (MC)' in questa pagina e poi in quello più giusto 'Ambiente di lavoro Unix-like' a pagina 53.

4.2 Midnight commander (MC)

Midnight commander (MC) può essere considerato come uno di quei coltellini svizzeri multiuso, per la console Linux ed altri terminali. Ciò offre al novizio un'esperienza con una console con menu, molto più semplice per imparare i comandi Unix standard.

Usate questo comando per esplorare il sistema Debian. E' il modo migliore per imparare. Esplorate alcune locazioni chiave usando solo i tasti cursore ed il tasto Enter:

---

[7] I tasti Ctrl-sinistro, Alt-sinistro e Delete (o Canc) vanno premuti comtemporaneamente da console. In un sistema standard, ciò provoca il reboot. Potete modificare `/etc/inittab` per avere il comando `shutdown` comando con l'opzione `-h`, come descritto in 'Installate pochi altri pacchetti dopo l'installazione iniziale' a pagina 43.

[8] Ciò è possibile perchè il sistema Debian è, anche dopo l'installazione di base, già configurato con i giusti permessi dei file, prevenendo danneggiamenti del sistema da parte di utenti non privilegiati. Ovviamente potranno sempre esistere delle falle che potranno essere sfruttate, ma coloro che se ne preoccupano non dovrebbero leggere questa sezione, ma il Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`).

[9] I tasti Ctrl-sinistro e d vanno premuti insieme. Non c'è bisogno di premere il tasto del maiuscolo, anche se questi caratteri di controllo vengono definiti come "control D" con il maiuscolo.

[10] Se inserite `root` invece di *penguin* e la password corrispondente, avrete l'accesso all'account `root` account. Questa procedura vi servirà per riguadagnare l'accesso all'account `root`.

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

4.2.1 Migliorare MC

Per far cambiare ad MC la directory di lavoro in uscita, dovete modificare `~/.bashrc` (o `/etc /bash.bashrc`, chiamato da `.bashrc`), come spiegato nella sua pagina di manuale, `mc(1)`, sotto l'opzione `-P`. [11]

4.2.2 Lanciare MC

```
$ mc
```

MC si prende cura di tutte le operazioni sui file tramite il proprio menu, richiedendo il minimo sforzo da parte dell'utente. Basta premere F1 per accedere alla schermata di aiuto. Potete giocare con MC premendo i tasti cursore ed i tasti funzione. [12]

4.2.3 Il file manager in MC

Come default vengono presentate due finestre affiancate che mostrano la lista di file contenuti nelle directory correnti. Un'altra modalità utile è impostare la finestra di destra ad "informazioni", per avere tute le informazioni su file, tipo privilegi di accesso, ecc. A seguire si riportano i tasti fondamentali. Se il demone `gpm` sta girando, potete usare anche il mouse. (Ricordatevi di premere il tasto maiscolo per avere il normale comportamento sotto MC per taglia ed incolla).

- F1: Menu aiuto
- F3: File viewer interno
- F4: Editor interno
- F9: Attiva il menu a cascata
- F10: Esce da Midnight commander
- Tab: Muove tra le due finestre
- Insert: Marca il file per operazioni con più file, tipo copia
- Del: Cancella il file (Fate attenzione—impostate MC in modalità cancellazione sicura)
- Tasti cursore: Si spiegano da sè

4.2.4 Trucchi per la riga di comando in MC

- Qualsiasi comando `cd` cambierà la directory mostrata sullo schermo selezionato.

---

[11] Se non capite esattamente di cosa sto parlando, potete farlo in seguito.

[12] Se siete in un terminale, tipo `kon` e `Kterm` per il giapponese, che utilizza dei caratteri grafici speciali, aggiungendo `-a` alla riga di comando di MC può aiutare a prevenire dei problemi.

- Control-Invio o Alt-Invio copiano il nome del file sulla riga di comando. Usatelo insieme ai comandi `cp` o `mv` durante l'editing da riga di comando.
- Alt-Tab mostra le scelte per i suffissi di file.
- Si possono stabilire le directory di partenza per ciascuna finestra, come argomenti per MC; per esempio, `mc /etc /root`.
- Esc + *tastonumero* == F*n* (cioè, Esc + '1' = F1, ecc.; Esc + '0' = F10)
- Tasto Esc == tasto Alt (= Meta, M-); cioè, premete Esc + 'c' per Alt-C.

4.2.5 L'editor in MC

L'editor interno ha un sistema di taglia ed incolla interessante. Premendo F3 si marca l'inizio della selezione, un secondo F3 ne segna la fine e la evidenzia. Muovete ora il cursore. Premendo F6 l'area selezionata viene mossa dove è il cursore. Con F5 l'area verrà copiata dove è il cursore. F2 salva il file, F10 esce. Gran parte dei tasti cursori ha un funzionamento intuitivo.

Questo editor può essere lanciato direttamente per un determinato file:

```
$ mc -e file_da_modificare
$ mcedit file_da_modificare
```

Non è un editor multifinestra, si può ottenere lo stesso effetto utilizzando più console. Per copiare da una finestra all'altra, usate la combinazione Alt-F*n* per passare da una console ad un'altra e"File->Insert file" o "File->Copy to file" per muovere parti di un file in un altro.

L'editor interno può essere sostituito da qualsiasi editor esterno preferiate.

Molti programmi usano variabili d'ambiente tipo `EDITOR` o `VISUAL` per decidere quale editor usare. Se vi trovate male con `vim`, impostatele a `mcedit` aggiungendo queste righe a `~/.bashrc`:

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

Raccomando comunque di impostarle a `vim`, se possibile. Abituarsi ai comandi di `vim` sarebbe la cosa giusta da fare, dato che gli editor-Vi sono una costante nel mondo Linux/Unix. [13]

4.2.6 Il visualizzatore in MC

Molto valido. E' uno strumento notevole per la ricerca di parole nei documenti. Lo uso sempre per i file nella directory `/usr/share/doc`. Rappresenta uno dei modi più rapidi di girare tra la massa di informazioni su Linux. Può essere lanciato direttamente con:

```
$ mc -v file_da_vedere
```

---

[13] In effetti, `vi` o `nvi` sono programmi che trovate ovunque. Ho scelto `vim`, invece, per il novizio, poichè offre un aiuto in linea attraverso il tasto F1, pur essendo simile è più potente dei programmi sopracitati.

4.2.7 Le caratteristiche di lancio automatico di MC

Premete Invio su un file, ed il programma appropriato si prenderà cura del suo contenuto. E' una caratteristica di MC molto utile.

```
eseguibile:         Esegue il comando
file man, html:     Dirotta il contenuto al viewer corrispondente
file tar.gz, deb:   Sfoglia il contenuto come fosse una sottodirectory
```

Per permettere a queste utilità di svolgere il loro compito, i file da leggere non devono essere segnati come eseguibili. Cambiatene lo stato tramite il comando `chmod`, oppure attraverso il menu file di MC.

4.2.8 Il fileystem FTP virtuale di MC

MC può essere utilizzato per accedere a file tramite Internet, usando FTP. Attivate il menu premendo F9, poi attivate il file system virtuale FTP premendo 'p'. Inserite una URL sotto forma di `username:passwd@nomehost.nomedomain`, che raggiungerà una directory remota che apparirà come una locale.

Provate `http.us.debian.org/debian` come URL e scorrete l'archivio dei file Debian. Vedere 'Gli archivi Debian' a pagina 5 per come sono organizzati.

4.3 Ambiente di lavoro Unix-like

Nonostante MC vi permetta di fare qualsiasi cosa, è bene che impariate come usare gli strumenti da riga di comando invocati dal prompt della shell e che familiarizziate con l'ambiente di lavoro Unix-like. [14]

4.3.1 Combinazioni speciali di tasti

Nell'ambiente Unix-like, esistono alcune combinazioni di tasti che hanno un significato particolare. [15]

- Ctrl-U: Cancella la riga prima del cursore.
- Ctrl-H: Cancella il carattere prima del cursore.
- Ctrl-D: Termina l'input. (eshe dalla shell se la state usando)
- Ctrl-C: Termina l'esecuzione di un programma.
- Ctrl-Z: Ferma temporaneamente un programma. (mette il job in background vedere '`comando &`' a pagina 61)

---

[14] In questa lezione, il termine shell significa `bash`. Per approfondire il significato delle varie shell, vedere 'Shell' a pagina 225.

[15] Su una normale console a caratteri Linux solo il Ctrl sinistro ed il tasto Alt, funzionano come ci si aspetta.

- Ctrl-S: Interrompe l'output a schermo. [16]
- Ctrl-Q: Riattiva l'output a schermo.

la shell di default, `bash`, prevede l'editing della cronologia ed il completamento mediante il tasto tab per facilitarne l'uso interattivo.

- freccia su: Lancia la ricerca della cronologia dei comandi.
- Ctrl-R: Lancia la ricerca incrementale della cronologia dei comandi.
- TAB: Completa l'input del nome del file sulla riga di comando.
- Ctrl-V TAB: Inserisce TAB senza espandere la riga di comando.

Altre combinazioni importanti da ricordare:

- Ctrl-Alt-Del: Reboot/arresta il sistema, vedere 'Installate pochi altri pacchetti dopo l'installazione iniziale' a pagina 43.
- Tasto-sinistro-e-sposta il mouse: Seleziona e copia negli appunti.
- Tasto centrale del: Incolla il contenuto degli appunti dove è il cursore.
- Tasto Meta (terminologia Emacs) viene tradizionalmente assegnato al tasto Alt sinsitro. Alcuni sistemi possono essere configurati per utilizzare come tasto Meta il tasto Windows.

Per usare il mouse sotto la console a caretteri Linux, dovete avere `gpm` lanciato come demone. [17] Vedere 'Configurare il mouse' a pagina 36.

4.3.2 Comandi base Unix

Impariamo i comandi base Unix. [18] Provate tutti i comandi dall'account utente senza privilegi `penguin`:

- `pwd`
  - Mostra il nome della directory attuale/di lavoro.
- `whoami`
  - Mostra il nome utente attuale.
- `file foo`
  - Mostra il tipo per il dato file *foo*.
- `type -p nomecomando`
  - Mostra la localizzazione del comando `nomecomando`.
  - `which nomecomando` fa la stessa cosa. [19]
- `type nomecomando`
  - Mostra le informazioni sul comando `nomecomando`.
- `apropos parola-chiave`
  - Trova i comandi correlati a `parola-chiave`.
  - `man -k parola-chiave` fa la stessa cosa.
- `whatis nomecomando`

---

[16] Potete disabilitare questa caratteristica del terminale tramite `stty(1)`.

[17] Nell'ambiente grafico X le funzioni del mouse sono identiche quando usate il programma Xterm.

[18] Qui uso "Unix" nel suo significato generico. Qualsiasi clone di Unix in genere offre comandi equivalenti. il sistema Debian non fa eccezione. Non preoccupatevi se alcuni comandi non funzionano come desiderate. Questi esempi non sono stati pensati per essere eseguiti in questo ordine.

[19] Se si usa `alias` nella shell, il loro output sarà differente.

  - Mostra una linea di spegazione sul comando *`nomecomando`*.
- `man -a` *`nomecomando`*
  - Mostra la spiegazione sul comando *`nomecomando`*. (Unix style)
- `info` *`nomecomando`*
  - Mostra una spiegazione più lunga sul comando *`nomecomando`*. (stile GNU)
- `ls`
  - Elenca il contenuto della directory. (file non puntati e directory) [20]
- `ls -a`
  - Elenca il contenuto della directory. (tutti i file e directory)
- `ls -A`
  - Elenca il contenuto della directory. (quasi tutti i file e directory, ovvero salta “`..`” e “`.`”)
- `ls -la`
  - Elenca tutto il contenuto della directory con informazioni dettagliate. Vedere ‘Il concetto di filesystem in Debian’ a pagina 67.
- `ls -d`
  - Elenca tutte le directory sotto la directory corrente.
- `lsof` *`foo`*
  - Mostra lo stato del file *`foo`*.
- `mkdir` *`foo`*
  - Crea una nuova directory *`foo`* sotto la directory corrente.
- `rmdir` *`foo`*
  - Rimuove la directory *`foo`* nella directory corrente.
- `cd` *`foo`*
  - Cambia directory nella directory *`foo`* nella directory corrente o in quella elencata nella variabile `CDPATH`. Vedere il comando `cd` in `builtins(7)`.
- `cd /`
  - Cambia directory in quella di root.
- `cd`
  - Cambia directory nella home dell’utente attuale.
- `cd /`*`foo`*
  - Cambia directory seguendo il percorso assoluto `/`*`foo`*.
- `cd ..`
  - Va nella directory superiore.
- `cd ~`*`foo`*
  - Va nella home directory dell’utente *`foo`*.
- `cd -`
  - Va nella directory precedente.
- `</etc/motd pager`
  - Mostra il contenuto di `/etc/motd` usando il paginatore di default. Vedere ‘`comando <` *`foo`*’ a pagina 62. [21]

---

[20] Unix ha la tradizione di nascondere il file il cui nome comincia con “`.`”. Sono generalmente file che contengono informazioni di configurazione e preferenze dell’utente.

[21] Il paginatore di default di un sistema Debian di base è `more` che non ha lo scorrimento indietro. Installando il pacchetto `less` tramite `apt-get install less`, `less` diventa il paginatore di default, così potrete scorrere all’indietro tramite i tasti cursore.

- `touch junkfile`
  - Crea un file vuoto `junkfile`.
- `cp foo bar`
  - Copia un file esistente `foo` in muovo file `bar`.
- `rm junkfile`
  - Rimuove il file `junkfile`.
- `mv foo bar`
  - Rinomina un file esistente `foo` in `bar`.
- `mv foo bar/baz`
  - Muove un file esistente `foo` in una nuova locazione con un nuovo nome `bar/baz`. La directory `bar` deve esistere.
- `chmod 600 foo`
  - Rende un file esistente `foo` non-leggibile e non-scrivibile dagli altri utenti. (non-eseguibile per tutti)
- `chmod 644 foo`
  - Rende un file esistente `foo` leggibile ma non-scrivibile dagli altri utenti. (non-eseguibile per tutti)
- `chmod 755 foo`
  - Rende un file esistente `foo` leggibile ma non-scrivibile per gli altri utenti. (eseguibile per tutti)
- `top`
  - Mostra le informazioni sui processi a tutto schermo. Digitate "q" per uscire.
- `ps aux | pager`
  - Mostra le informazioni su tutti i processi in esecuzione usando un output in stile BSD. Vedere '`comando1 | comando2`' a pagina 61.
- `ps -ef | pager`
  - Mostra le informazioni su tutti i processi in esecuzione usando un output in stile Unix system-V.
- `ps aux | grep -e "[e]xim4*"`
  - Mostra tutti i processi che usano `exim` o `exim4`. Imparate le espressioni regolari dalla `grep(1)` pagina di manuale digitando `man grep`. [22]
- `ps axf | pager`
  - Mostra le informazioni su tutti i processi in esecuzione usando un output in arte ASCII.
- `kill 1234`
  - Uccide un processo idenficato dal numero: *1234*. Vedere 'Uccidere un processo' a pagina 115.
- `grep -e "modello" *.html`
  - Trova un "*modello*" in tutti i file che terminano per `.html` nella directory corrente e li mostra tutti.
- `gzip foo`
  - Comprime `foo` per creare `foo.gz` usando la codifica Lempel-Ziv (LZ77).

---

[22] `[e]` nelle espressioni regolari permettono a `grep` di evitare di trovare corrispondenze con sè stesso. Il `4*` nella espressione regolare significa 0 o più ripetizioni del carattere 4, per cui permette a `grep` di trovare corrispondenze sia con `exim` che con `exim4`. Sebbene `*` sia usato come metacarattere sia nei nomi dei file della shell che nelle espressioni regolari, ha significati differenti.

- `gunzip foo.gz`
  - Decomprime `foo.gz` per creare `foo`.
- `bzip2 foo`
  - Comprime `foo` per creare `foo.bz2` utilizzando l'algoritmo di compressione del testo Burrows-Wheeler e la codifica Huffman. (Migliore compressione di `gzip`)
- `bunzip2 foo.bz2`
  - Decomprime `foo.bz2` per creare `foo`.
- `tar -xvvf foo.tar`
  - Estrae i file dall'archivio `foo.tar`.
- `tar -xvvzf foo.tar.gz`
  - Estrae i file dall'archivio gzippato `foo.tar.gz`.
- `tar -xvvf --bzip2 foo.tar.bz2`
  - Estrae i file dall'archivio `foo.tar.bz2`. [23]
- `tar -cvvf foo.tar bar/`
  - Archivia il contenuto della cartella `bar/` in un archivio `foo.tar`.
- `tar -cvvzf foo.tar.gz bar/`
  - Archivia il contenuto della cartella `bar/` in un archivio `foo.tar.gz` compresso.
- `tar -cvvf --bzip2 foo.tar.bz2 bar/`
  - Archivia il contenuto della cartella `bar/` in `foo.tar.bz2`. [24]
- `zcat README.gz | pager`
  - Mostra il contenuto del compresso `README.gz` usando il paginatore di default.
- `zcat README.gz > foo`
  - Crea un file `foo` con il contenuto non compresso di `README.gz`.
- `zcat README.gz >> foo`
  - Appende il contenuto non compresso di `README.gz` alla fine del file `foo`. (Se non esiste, prima lo crea.)
- `find . -name modello`
  - trova i file con i nomi corrispondenti usando il `modello` della shell. (più lento)
- `locate -d . modello`
  - find i file con i nomi corrispondenti usando il `modello` della shell. (più rapido, se si usa un database generato regolarmente)

Come esercizio, attraversate le directory e sbirciate nel sistema usando i comandi sopraelencati. Se avete dubbi sui comandi, assicuratevi di aver letto le pagine di manuale. Per esempio, un buon inizio sono questi comandi:

```
$ man man
$ man bash
$ man ls
```

Questo è anche il momento giusto per lanciare `vim` e premere il tasto F1. Dovreste leggere almeno le prime 35 righe. Poi fate un pò di esercizio in linea muovendo il cursore su |`tutor`| e premendo Ctrl-]. Vedere 'Gli Editor' a pagina 211 per impararne di più sugli editor.

---

[23] Qui si è usata l'opzione `--bzip2` al posto della più breve `-j` per essere sicuri che funzioni anche con le versioni più vecchie di `tar` in Potato.

[24] Di nuovo, `--bzip2` viene usato per garantire la compatibilità.

Notate che molti comandi Unix-like, compresi quelli provenienti da GNU e BSD, mostreranno delle brevi informazioni di aiuto se invocati in uno dei modi seguenti (o senza argomenti, in alcuni casi):

```
$ nomecomando --help
$ nomecomando -h
```

Come esercizio, provate anche gli esempi in 'Trucchi per Debian' a pagina 107.

4.3.3 Esecuzione dei comandi

Avete avuto un assaggio su come usare il sistema Debian. Addentriamoci ora nei meccanismi di esecuzione dei comandi. [25]

4.3.4 Il comando semplice

Un comando semplice è una sequenza di

1. compiti della variabile (opzionale)
2. nome comando
3. argomenti (opzionale)
4. redirezioni (opzionale: `>` , `>>` , `<` , `<<` , etc.)
5. operatore di controllo (opzionale: `&&` , `||` ; <newline> , `;` , `&` , `(` , `)`)

Per comandi più complessi, con quotazioni e sostituzioni, vedere 'Processamento delle righe di comando' a pagina 229.

4.3.5 Esecuzione dei comandi e variabili d'ambiente

Una tipica esecuzione di un comando usa una sequenza di shell coma la seguente: [26]

```
$ date
Sun Oct 26 08:17:20 CET 2003
$ LC_ALL=fr_FR date
dim oct 26 08:17:39 CET 2003
```

In questo caso il programma `date` viene eseguito in primo piano. La variabile di ambiente `LC_ALL` è:

- non impostata (default del sistema, `C`) per il primo comando
- impostata a `fr_FR` (locale Francese) per il secondo comando

---

[25] Qui la realtà è stata semplificata, per venire incontro al neofit. Vedere `bash(1)` per una spiegazione più completa.

[26] Per ottenereTo il risultato seguente, dovreste installare il locale Francese, vedere 'I locale' a pagina 170. Ciò non è essenziale per la lezione. Viene fatto solo per mostrare gli effetti potenziali.

Gran parte delle esecuzioni dei comandi non sono generalmente preceduti da una definizione della variabile di ambiente. In riferimento all'esempio precedente, potete eseguire, in alternativa:

```
$ LC_ALL=fr_FR
$ date
dim oct 26 08:17:39 CET 2003
```

Come potete vedere, il risultato del comando viene influenzato dalla variabile di ambiente, che produrrà un risultato in francese. Se volete che la variabile di ambiente venga inglobata dai sottoprocessi (quando chiamate uno script della shell, per esempio), dovete "esportarla", usando:

```
$ export LC_ALL
```

4.3.6 Il percorso di ricerca dei comandi

Quando date un comando nella shell, essa lo cerca nella lista di directory contenuta nella variabile `PATH`. Il valore di `PATH` viene anche chiamato percorso di ricerca della shell.

In una installazione Debian di base, la variabile `PATH` degli account utenti può non comprendere `/sbin/`. Quindi, se volete lanciare un comando tipo `ifconfig` da `/sbin/`, dovete modificare `PATH` in maniera da includerlo. La variabile `PATH` viene di solito impostata dal file di inizializzazione `~/.bash_profile`, vedere 'Configurare Bash' a pagina 36.

4.3.7 Opzioni della riga di comando

Alcuni comandi richiedono degli argomenti. Gli argomenti che cominciano con `-` o con `--` vengono chiamati opzioni e controllano il comportamento del comando.

```
$ date
Mon Oct 27 23:02:09 CET 2003
$ date -R
Mon, 27 Oct 2003 23:02:40 +0100
```

In questo caso l'argomento `-R` modifica il comportamento di `date` per dare come risultato una stringa con una data compatibile RFC-2822.

4.3.8 Metacaratteri della shell

Spesso capita che vogliate che un comando funzioni su un gruppo di file, senza digitarli tutti. Il modello di espansione dei nomi dei file che utlizza **metacaratteri** facilita questo compito.

- `*`
  - Comprende un qualsiasi gruppo di 0 o più caratteri.
  - Non comprende un file che inizia per "`.`".
- `?`
  - Comprende esattamente un unico carattere.
- `[...]`
  - Comprende esattamente qualsiasi carattere contenuto fra le parentesi
- `[a-z]`
  - Comprende esattamente qualsiasi carattere compreso fra `a` e `z`.
- `[^...]`
  - Comprende esattamente qualsiasi carattere diverso da quelli contenuti tra parentesi (escluso "`^`").

Come esempio, provate da voi e ragionateci su:

```
$ mkdir junk; cd junk; touch 1.txt 2.txt 3.c 4.h .5.txt
$ echo *.txt
1.txt 2.txt
$ echo *
1.txt 2.txt 3.c 4.h
$ echo *.[hc]
3.c 4.h
$ echo .*
. .. .5.txt
$ echo .[^.]*
.5.txt
$ echo [^1-3]*
4.h
$ cd ..; rm -rf junk
```

4.3.9 Restituire il valore dei comandi

Ogni comando restituisce il suo stato in uscita come valore restituito.

- valore restituito = 0 se il comando è stato eseguito con successo.
- valore restituito = non-zero se il comando termina con errori.

Si può accedere al valore restituito attraverso la variabile di shell `$?` subito dopo l'esecuzione.

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

Notate che, quando il valore restituito viene usato nel contesto logico della shell, il **successo** viene trattato come il **VERO** in logica. Ciò è poco intuitivo, dato che **successo** ha valore **zero**.

Vedere 'Condizionali' a pagina 228.

4.3.10 Tipiche sequenze di comandi

Proviamo a ricordare i seguenti comandi idiomatici della shell. Vedere 'Parametri della shell' a pagina 227, 'Redirezione' a pagina 228, 'Condizionali' a pagina 228, e 'Processamento delle righe di comando' a pagina 229 dopo aver letto questi idiomi.

`comando &`

Il `comando` viene eseguito nella subshell nello **sfondo**. I lavori nello sfondo permettono all'utente di lanciare più programmi in una singola shell.

La gestione dei processi nello sfondo coinvolge i fondamentali della shell: `jobs`, `fg`, `bg`, e `kill`. Leggete le sezioni della pagina di manuale `bash(1)` sotto "SIGNALS", "JOB CONTROL", e "SHELL BUILTIN COMMANDS". [27]

`comando1 | comando2`

Lo standard output di `comando1` viene dato allo standard input di `comando2` . Entrambi i comandi possono essere eseguiti **contemporaneamente**. Questa si chiama **pipeline**.

`comando1 ; comando2`

`comando1` e `comando2` sono eseguiti **in sequenza**.

`comando1 && comando2`

`comando1` viene eseguito. Se con successo, `comando2` viene eseguito **in sequenza**. Verrà restituito un successo se sia `comando1` **che** `comando2` sono stati eseguiti con successo.

`comando1 || comando2`

`comando1` viene eseguito. Se non con successo, allora anche `comando2` viene eseguito **in sequenza**. Verrà restituito un successo se `comando1` **oppure** `comando2` sono stati eseguiti con successo.

`comando >` *`foo`*

Redireziona lo standard output di `comando` ad un file *`foo`*. (sovrascrive)

---

[27] Il sistema Debian è un sistema multi-tasking.

comando >> *foo*

Redireziona lo standard output di `comando` ad un file *`foo`*. (appende)

comando > *foo* 2>&1

Redireziona sia lo standard output che lo standard error di `comando` ad un file *`foo`*.

comando < *foo*

Redireziona lo standard input di `comando` ad un file *`foo`*. Provate:

```
$ </etc/motd pager
 ... (il saluto)
$ pager </etc/motd
 ... (il saluto)
$ pager /etc/motd
 ... (il saluto)
$ cat /etc/motd | pager
 ... (il saluto)
```

Sebbene tutte e 4 le sintassi mostrino la stessa cosa, l'ultimo esempio lancia un comando in più, `cat` e spreca risorse senza motivo.

4.3.11 Alias dei comandi

Potete impostare un alias per i comandi usati più di frequente. Per esempio:

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

Da adesso in poi, `la` funzionerà come abbreviazione di `ls -la`, che elenca tutti i file in formato esteso.

Potete identificare il percorso esatto o l'identità di un comando tramite il comando `type`. Per esempio:

```
$ 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` è stato usato di recente, mentre `file` no, per cui `ls` è “hashed”, ovvero la shell ha un registro interno per un accesso veloce alla locazione del comando `ls`.

4.4 Elaborazione del testo Unix-like

Esistono alcuni strumenti standard per l’elaborazione del testo, che vengono spesso utilizzati nei sistemi Unix-like.

- Nessuna espressione regolare viene usata:
  - `head` restituisce la prima parte dei file.
  - `tail` restituisce l’ultima parte dei file.
  - `sort` ordina le righe di testo dei file.
  - `uniq` rimuove le righe duplicate da un file ordinato.
  - `tr` traduce o cancella i caratteri.
  - `diff` confronta i file riga per riga.
- Si usano le espressioni regolari di base (Basic regular expression, BRE):
  - `grep` trova il testo corrispondente al modello dato.
  - `ed` è un editor primitivo a riga.
  - `sed` è un editor a flusso.
  - `vi` è un editor a schermo.
  - `emacs` è un editor a schermo.
- Si usano le espressioni regolari estese (Extended regular expression, ERE):
  - `egrep` trova il testo corrispondente al modello dato.
  - `awk` fa della semplice elaborazione di testo. Vedere ‘Awk’ a pagina 230.
  - `perl` fa qualsiasi concepibile elaborazione di testo. Vedere ‘Perl’ a pagina 231.

Vedere ‘Sostituzione delle espressioni regolari’ a pagina 120, ‘Parti di script per il piping’ a pagina 123, e ‘Brevi follie in Perl script’ a pagina 124 per alcuni script di esempio.

4.4.1 Le espressioni regolari

Le espressioni regolari vengono utilizzate da molti strumenti di elaborazione del testo. Sono analoghe ai metacaratteri della shell (vedere ‘Metacaratteri della shell’ a pagina 59), ma molto più complesse e potenti.

L’espressione regolare descrive il modello corrispondente da trovare ed è fatta da caratteri di testo e da **metacaratteri**. Il metacarattere è semplicemente un carattere con un significato speciale. esistono due stili maggiori, BRE ed ERE, aseconda degli strumenti utilizzati, come descritto in ‘Elaborazione del testo Unix-like’ in questa pagina.

In ERE, i **metacaratteri** comprendono “`\ . [ ] ^ $ * + ? ( ) { } |`”. L’espressione regolare significa:

- `c`
  - Corrispondenza con il non-metacarattere “`c`”.
- `\c`
  - Corrispondenza con il carattere letterale “`c`”.

- `.`
  - Corrispondenza con qualsiasi carattere che includa una nuova riga.
- `^`
  - Corrispondenza con l'inizio di una stringa.
- `$`
  - Corrispondenza con la fine di una stringa.
- `\<`
  - Corrispondenza con l'inizio di una parola.
- `\>`
  - Corrispondenza con la fine di una parola.
- `[abc...]`
  - Questa lista di caratteri ha corrispondenza con uno qualsiasi dei caratteri "`abc...`".
- `[^abc...]`
  - Questa lista di caratteri negat ha corrispondenza con qualsiasi carattere tranne quelli "`abc...`".
- `r*`
  - Ha corrispondenza con nessuna o più espressioni regolari identificate da "`r`".
- `r+`
  - Ha corrispondenza co una o più espressioni regolari identificate da "`r`".
- `r?`
  - Ha corrispondenza con nessuna od una espressione regolare identificata da "`r`".
- `r1|r2`
  - Ha corrispondenza con una delle espressioni regolari identificate da "`r1`" or "`r2`".
- `(r1|r2)`
  - Ha corrispondenza con una delle espressioni regolari identificate da "`r1`" or "`r2`" e la tratta come una espressione regolare **tra parentesi**.

In BRE i **metacaratteri** "`+ ?  ( ) { } |`" perdono il loro significato speciale; al loro posto si usano le versioni con la backslash "`\+ \?  \( \) \{ \} \|`". Perciò il costrutto `(r1|r2)` deve essere protetto come `\(r1|r2\)`. Siccome `emacs`, sebbene sia di base BRE, tratta "`+ ?`" come **metacaratteri**, non c'è necessità di proteggerli. Vedere 'Espressioni sostitutive' in questa pagina per come il costrutto viene utilizzato.

Per esempio, `grep` può essere utilizzato per eseguita una ricerca di testo mediante l'espressione regolare:

```
$ 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
```

4.4.2 Espressioni sostitutive

Per le espressioni sostitutive, i caratteri seguenti hanno significati speciali:

- `&`

  - Rappresenta la corrispondenza dell'espressione regolare. (usate `\&` in `emacs`)

- `\n`
  - Rappresenta l' *n*-esima corrispondenza dell'espressione regolare **tra parentesi**.

Nelle stringhe sostitutive in Perl si usa, `$n` al posto di `\n` e `&` non alcun significato speciale meaning.

Per esempio:

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

Ponete particolare attenzione allo stile delle espressioni regolari **tra parentesi** e a come le stringhe corrispondenti siano state usate nel processo di sostituzione del testo dai vari strumenti.

Queste espressioni regolari possono anche essere usate per i movimenti del cursore e la sostituzione del testo negli editor.

Per imparare questi comandi, leggete le loro pagine di manuale.

4.5 Il filesystem Unix-like

Nei sistemi operativi GNU/Linux e negli altri Unix-like, i **file** sono organizzati in **directory**. [28] Tutti i **file** e le **directory** sono organizzati in un unico grande albero, la gerarchia dei file, la cui radice è `/`.

Questi file e directory possono essere sparsi su vari device. Il comando `mount(8)` attacca il file system trovato su un device al grande albero. Al contrario, il comando `umount(8)` lo staccherà nuovamente.

---

[28] Le **directory** vengono chiamate **cartelle** in altri sistemi.

4.5.1 Le basi dei file Unix

I principi basilari:

- I nomi dei file dipendono dai caratteri maiuscoli o minuscoli. Ovvero, `MIOFILE` e `MioFile` sono file **differenti**.
- La root directory è definita semplicemente come `/`. Non confondete questa "root" con l'utente root. Vedere 'Login come root al prompt dei comandi' a pagina 47.
- Ciascuna directory ha un nome che può contenere sia lettere che simboli **tranne** `/`. [29] La root directory fa eccezione; il suo nome è `/` (pronunciato "slash" o "la root directory") e non può essere rinominata.
- Ciascun file o directory viene designato da un **fully-qualified filename**, **absolute filename**, o **path**, che fornisce la sequenza delle directory che devono esserre attraversate per raggiungerlo. I tre termini sono sinonimi. Tutti gli absolute filenames iniziano con la directory `/`, e c'è una `/` fra ciascuna directory o file nel filename. La prima `/` è il nome della directory, mentre le altre sono dei semplici separatori per distinguere le varie parti del. Le parole possono confondere. Prendiamo il seguente esempio:

  `/usr/share/keytables/us.map.gz`

  Questo è un fully-qualified filename; alcuni lo chiamano **path**. Comunque sia, ci si riferisce anche a `us.map.gz` da solo come un filename. [30]
- La root directory ha un certo numero di branche, come `/etc/` e `/usr/`. Queste sottodirectory a turno si dividono in altre sottodirectory, come `/etc/init.d/` e `/usr/local/`. Il loro insieme è definito **l'albero delle directory**. Potete immaginare l'absolute filename come un percorso che parte dalla base dell'albero (`/`) fino alla fine di qualche ramo (un file). Sentirete anche definire l'albero delle directory come se fosse un albero **genealogico**: per cui le sottodirectory hanno dei **genitori**, ed un path mostra la completa storia familiare di un file. Esiste anche un path relativo, cha ha inizio altrove rispetto alla root directory. Ricordate che la directory `../` si riferisce alla directory genitore.
- Non esiste alcuna directory corrispondente ad un dispositivo fisico, tipo il vostro disco rigido. In ciò sta la differenza rispetto a CP/M, DOS, e Windows, dove tutti i paths iniziano con il nome di un dispositivo, come `C:\`. Vedere 'Il concetto di filesystem in Debian' a fronte.

Le pratiche migliori e più dettagliate per la gerarchia dei file vengono descritte nelFilesystem Hierarchy Standard (`/usr/share/doc/debian-policy/fhs/fhs.txt.gz`). per iniziare, dovreste ricordare i seguenti fatti:

- `/`
  - Una semplice `/` rappresenta la root directory.
- `/etc/`
  - E' il luogo dei file di configurazione generale del sistema.
- `/var/log/`

---

[29] Anche se **potete** usare qualsiasi lettera o simbolo per il nome di un file, in pratica è una cattiva idea. E' meglio evitare quei caratteri che hanno spesso un significato speciale nella riga di comando, compresi spazi, tabulazioni, a capo ed altri caratteri speciali: `` { } ( ) [ ] ' ` " \ / > < | ; ! # & ^ * % @ $ ``. Per separare le parole in un nome, buone scelte sono il punto, il trattino e la sottolineatura. Potete anche iniziare ciascuna parola in maiuscolo, `ComeQuesta`.

[30] Esiste anche un altro utilizzo per la parole **path**. Vedere 'Il percorso di ricerca dei comandi' a pagina 59. Il suo significato viene in genere reso chiaro dal contesto.

  - E’ il luogo per i file di log del sistema.
- `/home/`
  - E’ la directory che contiene tutte le home directory di tutti gli utenti non privilegiati.

4.5.2 Il concetto di filesystem in Debian

Seguendo la tradizione Unix, il sistema Debian fa sì che i filesystem sotto i quali i dati fisici sui dischi rigidi e sugli altri dispositivi di memorie di massa e l’interazione con i dispositivi hardware tipo le console su schermo e le console remotes vengano rappresentati in maniera unificata.

Ciascun file, directory, pipe, o dispositivo fisico in un sistema Debian ha una struttura di dati chiamata **inode** che descrive gli attributi ad esso associati, come l’utente a cui appartiene (proprietario), il gruppo a cui appartiene, la data di ultimo accesso, ecc.. Vedere `/usr/include/linux/fs.h` per la definizione precisa di `struct inode` nel sistema Debian GNU/Linux.

Questa rappresentazione unificata di entità fisiche risulta molto potente, in quanto permette l’uso degli stessi comandi per lo stesso tipo di operazioni su dispositivi complatamente differenti

Tutti i vostri file potrebbero risiedere su un disco — oppure potreste averne 20, alcuni connessi a computer diversi situati altrove sulla rete. Non potreste distinguerli guerdando semplicemente l’albero delle directory, e quasi tutti i comandi lavorerebbero alla stessa maniera, non importa su quale dispositivo(i) fisico(i) i file risiedono realmente.

4.5.3 Permessi di accesso a file e directory

I permessi di accesso a file e directory vengono definiti separatamente per ciascuna delle seguenti tre categorie di utenti:

- l’ **utente** che è proprietario del file (u),
- gli altri utenti nel **gruppo** a cui il file appartiene (g) e
- tutti gli **altri** utenti (o).

Dato un file, ciascun permesso corrispondente permette:

- **read (lettura)** (r): di esaminare il contenuto del file,
- **write (scrittura)** (w): di modificare e
- **execute (esecuzione)** (x): di eseguire il file come un comando.

Data una directory, ciascun permesso corrispondente permette:

- **read** (r): di elencare i contenuti della directory,
- **write** (w): di aggiungere o rimuovere i file nella directory e
- **execute** (x): di accedere ai file nella.

In questo caso il permesso in **esecuzione** sulla directory non solo permette di leggere i file nella directory, ma anche di vedere i lori attributi, come le dimensioni e la data dell’ultima modifica.

per mostrare le informazioni sui permessi (e molto altro) dei file e delle directory si usa `ls`. Vedere `ls(1)`. Quando `ls` viene invocato con l'opzione `-l` mostrerà le seguenti informazioni, nell'ordine:

- **tipo di file** (primo carattere)
  - `-`: normale
  - `d`: directory
  - `l`: collegamento simbolico
  - `c`: character device node
  - `b`: block device node
  - `p`: named pipe
  - `s`: socket
- i **permessi** di accesso al file (i nove caratteri successivi, consistenti di tre caratteri per ciascuno, utente, gruppo ed altri, in quest'ordine)
- il **numero di hard links** al file
- il nome dell' **utente** a cui appartiene
- il nome del **gruppo** a cui il file appartiene
- le **dimensioni** del file in caratteri (bytes)
- **data ed ora** del file (mtime)
- il **nome** del file.

Per cambiare il proprietario del file, si usa `chown` dall'account di root. Per cambiarne il gruppo, si utlizza `chgrp` o dall'account del proprietario, o da quello di root. Per cambiare i permessi di accesso al file ed alla directory, si usa `chmod` dall'account del proprietario, o da quello di root. La sintassi di base per manipolare un dato file `foo` file è:

```
# chown nuovoproprietario foo
# chgrp nuovogruppo foo
# chmod  [ugoa][+-=][rwx][,...] foo
```

Vedere `chown(1)`,`chgrp(1)`,e `chmod(1)` per i dettagli.

Per esempio, per rendere proprietario di una directory l'utente *foo* e condivisa da un gruppo *bar*, eseguite i seguenti comandi dall'account di root:

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

Esistono altri tre bit di permessi speciali:

- **set user ID** (s o S invece della x di user),
- **set group ID** (s o S invece della x di group), and
- **sticky bit** (t o T invece della x di other).

In questo caso l'output di `ls -l` per detti bit viene dato in maiuscolo se la modalità nascosta per i bit di esecuzione non è impostata.

Impostare **set user ID** su un file eseguibile permette all'utente di eseguirlo con l'owner ID del file (per esempio **root**). Allo stesso modo, impostare **set group ID** su un eseguibile permette all'utente di eseguirlo con il group ID del file (per esempio **root**). Poichè queste impostazioni possono causare seri problemi di sicurezza, abilitarle richide estrema cautela.

Impostare **set group ID** su una directory abilita lo schema di creazione dei file stile BSD, dove tutti i file creati nella directory appartengono al **gruppo** della directory.

Impostare lo **sticky bit** in una directory previene la rimozione di un file in detta directory da un utente che non sia il proprietario del file. Per rendere sicuro il contenuto di un file in una directory scrivibile da tutti, come ad esempio `/tmp` o in diretory scrivibile dal gruppo, non bisogna solamente disabilitare i permessi in **scrittura** del file, ma anche impostare lo **sticky bit** nella directory. Altrimenti il file potrebbe essere rimosso e sostituito da un nuovo fiole con lo stesso nome, da qualsiasi utente con accesso in scrittura alla directory.

Ecco alcuni esempi interessanti sui permessi.

```
$ ls -l /etc/passwd /etc/shadow /dev/ppp /usr/sbin/pppd
crw-rw----    1 root     dip      108,   0 Jan 18 13:32 /dev/ppp
-rw-r--r--    1 root     root         1051 Jan 26 08:29 /etc/passwd
-rw-r-----    1 root     shadow        746 Jan 26 08:29 /etc/shadow
-rwsr-xr--    1 root     dip        234504 Nov 24 03:58 /usr/sbin/pppd
$ ls -ld /tmp /var/tmp /usr/local /var/mail /usr/src
drwxrwxrwt    4 root     root         4096 Feb  9 16:35 /tmp
drwxrwsr-x   10 root     staff        4096 Jan 18 13:31 /usr/local
drwxrwsr-x    3 root     src          4096 Jan 19 08:36 /usr/src
drwxrwsr-x    2 root     mail         4096 Feb  2 22:19 /var/mail
drwxrwxrwt    3 root     root         4096 Jan 25 02:48 /var/tmp
```

Esiste un metodo numerico alternativo di descrivere i permessi per i comandi `chmod(1)`. Questo metodo usa numeri da 3 a 4 cifre in ottale (base 8). Ogni cifra corrisponde a:

- 1a cifra facoltativa: la somma di **set user ID** (=4), **set group ID** (=2), e **sticky bit** (=1)
- 2a cifra: la somma dei permessi **lettura** (=4), **scrittura** (=2), e **esecuzione** (=1) per l' **utente**
- 3a cifra: idem per il **gruppo**
- 4a cifra: idem per **other**

Suona complicato, ma è in effetti molto semplice. Se guardate alle prime (2-10) colonne dell'output del comando `ls -l` e le leggete come una rappresentazione binaria (base 2) dei permessi dei file ("-" essendo "0" e "rwx" essendo "1"), il valore numerico diventa una rappresentazione in ottale dei permessi. [31] Provate ad esempio:

```
$ touch foo bar
$ chmod u=rw,go=r foo
$ chmod 644 bar
$ ls -l foo bar
-rw-r--r--    1 penguin  penguin  0 Nov  3 23:30  foo
-rw-r--r--    1 penguin  penguin  0 Nov  3 23:30  bar
```

---

[31] Oviamente questo metodo funziona solo per i numeri a 3 cifre.

La maschera di default dei permessi può essere impostata tramite il comando di shell `umask`. Vedere `builtins(7)`.

4.5.4 Timestamp

Per un file GNU/Linux, ci sono tre tipi di timestamp:

- **mtime**: orario di modifica (`ls -l`),
- **ctime**: orario di cambio di stato (`ls -lc`), e
- **atime**: orario dell'ultimo accesso (`ls -lu`).

Notate che **ctime** non è l'orario di creazione del file.

- La sovrascrittura di un file cambierà tutti e tre, **mtime**, **ctime**, e **atime**.
- Il cambio di permessi od utente modificherà **ctime** e **atime**.
- La lettura di un file ne cambierà **atime**.

Notate che anche una semplice lettura del file in un sistema Debian causerà una normale operazione di scrittura del file, per aggiornare le informazioni relative a **atime** nell' **inode**. Montare un filesystem con l'opzione `noatime` farà si che il sistemi salti questa operazione, risultando un tempo di accesso del file più breve in lettura. Vedere `mount(8)`.

Usate il comando `touch(1)` per cambiare i timestamp dei file esistenti.

4.5.5 Collegamenti

Due sono i metodi per associare un dato file *foo* con un diverso nome *bar*.

- un **collegamento** è un nome duplicato di un file esistente (`ln foo bar`),
- un **collegamento simbolico**, o "symlink", è un file speciale che punta ad un altro file (`ln -s foo bar`).

Vedere il seguente esempio per i cambiamenti nella conta dei collegamenti e le sottili differenze nel risultato del comando `rm`.

```
$ echo "Contenuto Originale" > foo
$ ls -l foo
-rw-r--r--    1 osamu    osamu          4 Feb  9 22:26 foo
$ ln foo bar     # hard link
$ ln -s foo baz  # symlink
$ ls -l foo bar baz
-rw-r--r--    2 osamu    osamu          4 Feb  9 22:26 bar
lrwxrwxrwx    1 osamu    osamu          3 Feb  9 22:28 baz -> foo
-rw-r--r--    2 osamu    osamu          4 Feb  9 22:26 foo
$ rm foo
$ echo "Nuovo Contenuto" > foo
$ cat bar
Contenuto Originale
$ cat baz
Nuovo Contenuto
```

Il collegamento simbolico ha sempre i permessi nominali di accesso impostati su “rwxrwxrwx”, come mostrato nell’esempio precedente, con i permessi effettivi dettati dai permessi del file a cui punta.

La directory `.` è collegata alla directory nella quale appare, per cui la conta dei collegamenti di qualsiasi nuova directory parte da 2. La directory `..` e collegata alla directory genitore, per cui la conta dei collegamenti della directory aumenta all’aggiungere nuove sottodirectory.

4.5.6 Named pipe (FIFO)

Una named pipe è un file che agisce come una pipe. Inserite qualcosa in un file da un lato ed esce dall’altro. Da qui il nome FIFO, o First-In-First-Out: la prima cosa che infilate nella pipe è anche la prima ad uscirne.

Se avviate un processo di scrittura verso una named pipe, il processo non terminerà finchè l’informazione che viene scritta non è letta dalla pipe. Se avviate un processo di lettura dalla pipe, il processo aspetterà prima di terminare che non ci sia altro da leggere. Le dimesioni della pipe sono sempre zero — non memorizza dati, si limita ad unire due processi, come il carattere | della shell. Comunque, poichè questa pipe ha un nome, i due processi non devono necessariamente stare sulla riga di comando, nè essere lanciati dallo stesso utente.

Provate facendo quanto segue:

```
$ cd; mkfifo miapipe
$ echo "hello" >miapipe & # gira nello sfondo
[1] 5952
$ ls -l miapipe
prw-r--r--    1 penguin penguin  0 2003-11-06 23:18 miapipe
$ cat miapipe
hello
[1]+  Done                    echo hello >miapipe
$ ls miapipe
prw-r--r--    1 penguin penguin  0 2003-11-06 23:20 miapipe
$ rm miapipe
```

4.5.7 Socket

Il socket è simile alla named pipe (FIFO) è permette ai processi di scambiarsi informazioni. per il socket questi processi non devono girare allo stesso tempo, nè devono essere figli dello stesso genitore. Questo è il traguardo di qualsiasi comunicazione interprocesso. Lo scambio di informazioni può avvenire attraverso una rete, fra host differenti.

4.5.8 Device

Un device fa riferimento ad un dispositivo fisico o virtuale presente nel sistema, come il disco rigido, la scheda grafica, lo schermo o la tastiera. Esempio di dispositivo virtuale è la console, rappresentata da `/dev/console`.

Esistono due tipi di device:

- **character device**
  - Vi può accedere un carattere alla volta, cioè la più piccola unità di dati che vi può essere scritta è un carattere (byte).
- **block device**
  - Qui vi si può accedere in unità maggiori, chiamte blocchi, che contengono un numero di caratteri. Il disco rigido è un block device.

Potete leggere e scrivere sui device, anche se il file potrebbe contenere dati binari, incomprensibili ai comuni mortali. Scrivere direttamente i dati su questi file può essere utile per diagnosticare problemi nelle connessioni hardware. Per esempio, inviare un file di testo al device della stampante `/dev/lp0` oppure inviare i comandi del modem alla porta seriale appropriata `/dev/ttyS0`. Attenzione che, se eseguite imprudentemente, queste manovre possono portare a disastri notevoli.

`/dev/null` ecc.

`/dev/null` è un device speciale che elimina qualsiasi cosa gli si vada a scrivere. Se c'è qualcosa che non volete, gettatela in `/dev/null`. E' fondamentalmente un pozzo senza fondo. Se andate a leggere `/dev/null`, ottenete immediatamente un carattere end-of-file (EOF).

`/dev/zero` is simile, solo che se andate a leggerlo, ottenete il carattere `\0` (diverso dal numero zero in ASCII). Vedere 'File fantoccio' a pagina 130.

Numeri dei nodi dei device

I numeri dei nodi dei device lanciando `ls` come:

```
$ ls -l /dev/hda /dev/ttyS0 /dev/zero
brw-rw----    1 root     disk       3,   0 Mar 14  2002 /dev/hda
crw-rw----    1 root     dialout    4,  64 Nov 15 09:51 /dev/ttyS0
crw-rw-rw-    1 root     root       1,   5 Aug 31 03:03 /dev/zero
```

Qui,

- `/dev/hda` ha come numero primario 3 e come secondario 0. E' accessibile in lettura/scrittura dall'utente che appartiene al gruppo `disk`,
- `/dev/ttyS0` ha come numero primario 4 e come secondario 64. E' accessibile in lettura/scrittura dall'utente che appartiene al gruppo `dialout` group e
- `/dev/zero` ha come numero primario 1, secondario 5. E' accessibile in lettura/scrittura da chiunque.

Nei vecchi sistemi il processo di installazione crea i nodi dei device tramite il comando `/sbin/MAKEDEV` command. Vedere `MAKEDEV(8)`.

Nei sistemi più recenti il filesystem sotto `/dev` viene popolato automaticamente in maniera analoga al filesystem `/proc`.

4.5.9 Il filesystem `/proc`

Il filesystem `/proc` è uno pseudo-filesystem e contiene informazioni sul sistema e sui processi in corso.

La gente in genere si spaventa quando nota un file in particolare - `/proc/kcore` - che è generalmente enorme. Esso è (più o meno) una copia della memoria del vostro computer. Viene utilizzato per il debug del kernel. In pratica non esiste da nessuna parte, per cui non preoccupatevi delle sue dimensioni.

Vedere 'Mettere a punto il kernel tramite il filesystem proc' a pagina 105 and `proc(5)`.

4.6 Il sistema X Window

Vedere 'X' a pagina 145.

4.6.1 Lanciare il sistema X Window

X Window può essere lanciato automaticamente con `xdm`-come demone per il login grafico, o digitando quanto segue da console.

```
$ exec startx
```

4.6.2 Il menu di X Window

Poichè l'ambiente di X è in grado di ospitare molti window manager, le loro interfacce variano molto. Ricordate che cliccando col pulsante destro del mouse sulla root window evidenzierà il menu delle scelte. Questo è sempre disponibile.

- Per avviare la shell, lanciate Xterm dal menu:
  - "XShells" –> "XTerm".
- Per scorrere graficamente le pagine web, lanciate Mozilla dal menu:
  - "Apps" –> "Net" –> "Mozilla Navigator".
- Per visualizzare graficamente file in PDF, lanciate xpdf del menu:
  - "Apps" –> "Viewers" –> "Xpdf".

Se vi manca la voce nel menu, installate i pacchetti corrispondenti. Vedere 'Iniziare la gestione dei pacchetti Debian' a pagina 80.

4.6.3 Sequenze di tasti per X Window

Alcune combinazioni da ricordare quandi si è in X:

- Ctrl-Alt-F1 fino a F6: Passa ad altri pseudo-terminali (da X-window, DOSEMU, ecc.)
- Alt-F7: Torna ad X
- Ctrl-Alt-meno: Modifica la risoluzione dello schermo in X (il meno si riferisce al tastierono numerico)
- Ctrl-Alt-più: Modifica la risoluzione in X dalla parte opposta (il più si riferisce al tastierino numerico)
- Ctrl-Alt-Backspace: Termina il server X
- Alt-X, Alt-C, Alt-V: Le normali combinazioni Windows/Mac per Taglia, Copia, Incolla con Ctrl- in alcuni programmi tipo Netscape Composer vengono sostituite da Alt-.

4.7 Studi ulteriori

In questo momento, vi raccomando la lettura delle guide fondamentali da The Linux Documentation Project: Guides (`http://www.tldp.org/guides.html`):

- “The Linux System Administrators’ Guide”,
  - Copre tutti gli aspetti del mantenimento di un sistema in funzione, gestione degli account utente, backup, configurazione del sistema.
  - pacchetto: `sysadmin-guide`
  - file: `/usr/share/doc/sysadmin-guide/html/index.html`
  - web: `http://www.tldp.org/LDP/sag/index.html`
- “The Linux Network Administrator’s Guide, Second Edition”,
  - Riferimento per l’amministrazione di rete in ambiente Linux.
  - pacchetto: `(not available)`
  - file: `(notapplicable)`
  - web: `http://www.tldp.org/LDP/nag2/index.html`
- *Linux: Rute User’s Tutorial and Exposition*
  - Un bel libro e versione on line che copre l’amministrazione dei sistemi GNU/Linux.
  - Di Paul Sheer
  - Pubblicato da Prentice Hall
  - Pacchetto: `rutebook` (da `non-free`)
  - File: `/usr/share/doc/rutebook/`
  - Web: `http://www.icon.co.za/~psheer/book/index.html.gz`

Vedere ‘Supporto per Debian’ a pagina 249 per ulteriori fonti di apprendimento.

Capitolo 5

Aggiornare una distribuzione a `stable`, `testing` o `unstable`

Le note ufficiali di rilascio per l'aggiornamento si trovano presso `http://www.debian.org/releases/stable/releasenotes` e `http://www.debian.org/releases/testing/releasenotes` (lavoro in corso).

L'aggiornamento alle distribuzioni `stable`, `testing` o `unstable` può richiedere vari passaggi, nel seguente ordine:

- Aggiornamento a Woody (se il sistema è antecedente ad esso)
- Aggiornamenti a stable
- Aggiornamenti a testing
- Aggiornamenti a unstable

Debian non supporta gli aggiornamenti che saltino i rilasci intermedi.

5.1 Aggiornare da Potato a Woody

Questa procedura separatamente perchè l'APT di Potato non aveva tutte le funzioni descritte nella pagina di manuale attuale. `apt_preferences(5)`.

Dopo aver inserito i puntatori per Woody in `/etc/apt/sources.list`, aggiornate APT ed i pacchetti fondamentali alle versioni di Woody facendo quanto segue.

```
# apt-get update
# apt-get install libc6 perl libdb2 debconf
# apt-get install apt apt-utils dselect dpkg
```

Poi aggiornate il resto del sistema a Woody.

```
# apt-get upgrade
# apt-get dist-upgrade
```

5.2 Preparazione per l’aggiornamento

Potete eseguire l’aggiornamento da una distribuzione all’altra prendendo i pacchetti dalla rete. Ciò può essere fatto come segue.

Procuratevi una lista pulita di depositi per `stable`:

```
# cd /etc/apt
# cp -f sources.list sources.list.old
# :>sources.list
# apt-setup noprobe
```

Se volete aggiornare a `testing`, allora aggiungete i riferimenti a `testing` a questa nuova lista. Se volete farlo a `unstable`, aggiungete i riferimenti ad `unstable`.

```
# cd /etc/apt
# grep -e "^deb " sources.list  >srcs
# :>sources.list
# cp -f srcs sources.list
# sed -e "s/stable/testing/" srcs >>sources.list
# sed -e "s/stable/unstable/" srcs >>sources.list
# apt-get update
# apt-get install apt apt-utils
```

Vedere ‘Iniziare la gestione dei pacchetti Debian’ a pagina 80 per l’arte della messa a punto di `/etc/apt/sources.list` e `/etc/apt/preferences`.

5.3 Aggiornamento

Dopo aver impostato appropriatamente `/etc/apt/sources.list` e `/etc/apt/preferences` come descritto sopra, potete iniziare l’aggiornamento.

Notate che il tracciamento della distribuzione `testing` di Debian può avere come effetto colaterale il ritardo nell’installazione dei pacchetti contenenti correzioni per la sicurezza, poicè detti pacchetti vengono caricati su `unstable` e solo dopo un lasso di tempo migrano in `testing`.

Vedere ‘Gestione dei pacchetti in Debian’ a pagina 79 per le basi e ‘Aggiornamento con APT: risoluzione dei problemi’ a pagina 86 se incontrate problemi.

5.3.1 Usare `dselect`

Se un sistema ha molti pacchetti, inclusi i pacchetti `-dev`, ecc., il metodo seguente tramite `dselect` è il metodo di scelta per un più raffinato controllo sui pacchetti.

```
# dselect update  # da fare sempre prima di un aggiornamento
# dselect select  # sceglie i pacchetti addizionali
```

Quando `dselect` parte, tutti i pacchetti correnti verranno selezionati. `dselect` potrà mostrarvi pacchetti addizionali basati su `Dipende`, `Suggerisce`, e `Raccomanda`. Se non volete altri pacchetti, basta pigiare `Q` per uscire di nuovo da `dselect`.

```
# dselect install
```

Dovrete rispondere ad alcune domande sulla configurazione dei pacchetti durante questa perte del processo, per cui tenete con voi i vostri appunti e prendetevi del tempo per questa parte. Vedere '`dselect`' a pagina 82.

Usate `dselect`. **Funziona sempre :)**

5.3.2 Usare `apt-get`

```
# apt-get update
# apt-get -t stable upgrade
# apt-get -t stable dist-upgrade
# apt-get -t testing upgrade
# apt-get -t testing dist-upgrade
# apt-get -t unstable upgrade
# apt-get -t unstable dist-upgrade
```

Una volta che il vostro sistema ha raggiunto Sarge, è consigliabile usare `aptitude` al posto di `apt-get`. (`aptitude` accetta molte delle opzioni che accetta anche `apt-get`, comprese quelle sopra.)

Per aggiornare e seguire le impostazioni date da dselect:

```
# apt-get dselect-upgrade
```

Vedere 'Dipendenze dei pacchetti' a pagina 16.

Capitolo 6

Gestione dei pacchetti in Debian

`aptitude` è attualmente l'interfaccia preferita per APT, l'Advanced Package Tool. Ricorda quali pacchetti avete deliberatamente installato e quali in base alle dipendenze; questi ultimi vengono disinstallati automaticamente da `aptitude` queo non sono più necessari ai pacchetti installati deliberatamente. Ha funzionalità avanzate di filtro sui pacchetti, ma possono essere difficili da configurare.

`synaptic` è attualmente l'interfaccia grafica Gtk preferita per APT. Le sue funzionalità di filtro sono più facili da usare rispetto ad `aptitude`. Ha anche il supporto sperimentale per le Debian Package Tags (`http://debtags.alioth.debian.org/`).

Per ridurre il carico sui depositi Debian e per velocizzare i vostri download, dovreste prendere i pacchetti dai mirror Debian.

Se dovete installare lo stesso pacchetto su più macchine sulla vostra rete locale, impostate un proxy HTTP locale useo `squid` per i pacchetti scaricati con APT. Impostate, se necessario, la variabile d'ambiente `http_proxy`, oppure il valore di `http` in `/etc/apt/apt.conf`.

Sebbene l'a funzionalità di pinning di APT, descritta in `apt_preferences(5)` sia potente, i suoi effetti possono essere difficili da comprendere e gestire. Consideratela una Funzione Avanzata.

L'uso del metodo descritto in '`chroot`' a pagina 130 è preferibile per assicurare al contempo stabilità del sistema ed accesso alle versioni più recenti del software.

Questo capitolo fa riferimento ad un sistema successivo a Woody Alcune funzionalità potrebbero richiedere un sistema Sarge o successivo.

6.1 Introduzione

Se la lettura di tutta la documentazione per gli sviluppatori è troppo per voi, leggete questo capitolo per primo ed iniziate a gustare in pieno il potere della Debian con le distribuzioni `testing/unstable` :-)

6.1.1 Strumenti principali di gestione dei pacchetti

```
dpkg      - Installatore dei pacchetti Debian
apt-get   - Interfaccia a riga di comeo per APT
aptitude  - Interfaccia avanzata testo e riga di comeo per APT
synaptic  - Interfaccia grafica Gtk di APT
dselect   - installatore Debian dei pacchetti
tasksel   - Installatore dei task
```

Questi strumenti non non sono tutti alternativi fra loro. Per esempio, `dselect` usa sia APT che `dpkg`.

APT usa `/var/lib/apt/lists/*` per tenere traccia dei pacchetti disponibili, mentre `dpkg` usa `/var/lib/dpkg/available`. Se avete installato dei pacchetti useo direttamente `aptitude` od un'altra interfaccia ad APT, e volete usare `dselect` per installare dei pacchetti, assicuratevi di aggiornare il file `/var/lib/dpkg/available` tramite la voce `[A]ggiorna` del menu di `dselect` (o con il comeo "`dselect update`").

`apt-get` installa automaticamente tutti i pacchetti dai quali un pacchetto richiesto Dipende. Non installa i pacchetti che Raccomea o Suggerisce.

`aptitude`, al contrario, può essere configurato per installare i pacchetti che un pacchetto richiesto Raccomea o Suggerisce.

`dselect` presenta all'utente una lista di pacchetti che un pacchetto richiesto Raccomea o Suggerisce e permette una loro selezione caso per caso. Vedere 'Dipendenze dei pacchetti' a pagina 16.

6.1.2 Strumenti utili

```
dpkg-reconfigure  - riconfigura un pacchetto già installato
                    (se utilizza debconf)
dpkg-source       - gestisce il pacchetto con il file sorgente
dpkg-buildpackage - automatizza la creazione di un pacchetto
apt-cache         - controlla l'archivio dei pacchetti nella cache locale
```

6.2 Iniziare la gestione dei pacchetti Debian

6.2.1 Impostare APT

Impostate `sources.list` come descritto in 'Preparazione per l'aggiornamento' a pagina 76. [1] Fate riferimento anche a 'Suggerimenti per l'installazione di un sistema Debian' a pagina 27, 'Aggiornare una distribuzione a `stable`, `testing` o `unstable`' a pagina 75 e 'Editor di salvataggio' a pagina 212.

---

[1] Se state traccieo `testing` o `unstable`, potete rimuovere i riferimenti a `stable` da `/etc/apt/sources.list` e da `/etc/apt/preferences`. Ciò perchè `testing` origina come una copia di `stable`.

6.2.2 Installare i task

E' possibile installare un gruppo di pacchetti tipicamente richiesto per un determinato utilizzo del sistema Debian. Questo gruppo di pacchetti è chiamato "task".

Il modo più semplice per installare i task al momento dell'installazione iniziale è useo `tasksel`. Notate che dovete eseguire

```
dselect update
```

prima di utilizzarlo.

Anche `aptitude` può installare i task ed è lo strumento racomeato per questo scopo. Permette di deselzionare singoli pacchetti all'interno dei task prima di procedere all'installazione.

6.2.3 `aptitude`

`aptitude` è un nuovo installatore di pacchetti basato su menu, simile a `dselect`, ma costruito da zero sopra APT. Può essere utilizzato in alternativa ad `apt-get` per gran parte dei comandi. Vedere `aptitude(1)` e `/usr/share/doc/aptitude/README`.

Una volta cominciato ad usare `aptitude`, è meglio continuare ad usarlo invece che come metodo alternativo di installazione dei pacchetti; altrimenti perdete il vantaggio di avere `aptitude` che mantiene una traccia di tutti i pacchetti installati deliberatamente.

`aptitude` in modalità a tutto schermo accetta comandi sotto forma singole lettere, di norma in minuscolo. Tasti degni di nota sono:

Tasto	Azione
F10	Menu
?	Aiuto per i comandi
u	Aggiorna le informazioni relative all'archivio dei pacchetti
+	Marca il pacchetto nuovo da installare o da aggiornare
-	Marca il pacchetto da rimuovere (mantiene la configurazione)
_	Marca il pacchetto da eliminare (rimuove la configurazione)
=	Pone il pacchetto in attesa
U	Marca tutti i pacchetti aggiornabili da aggiornare
g	Scarica ed installa i pacchetti selezionati
q	Esce dalla schermata corrente e salva le modifiche
x	Esce dalla schermata corrente e non salva le modifiche
Enter	Mostra le informazioni su un pacchetto
C	Mostra il changelog di un pacchetto
l	Cambia i limiti per il pacchetto mostrato
/	Cerca la prima corrispondenza
\	Ripete l'ultima ricerca

Come `apt-get`, `aptitude` installa i pacchetti da cui un pacchetto selezionato Dipende. `aptitude` offre anche l'opzione di prendere tutti i pacchetti che un dato pacchetto Raccomea o Suggerisce. Potete modificare questo comportamento tramite `F10 -> Options -> Dependency heling` nel suo menu.

Altri vantaggi di `aptitude` sono:
- `aptitude` offre l'accesso a tutte le versioni di un pacchetto.
- `aptitude` registra le azioni in `/var/log/aptitude`.
- `aptitude` facilita il tenere traccia di software obsoleto elenceolo in "Obsolete e Locally Created Packages".
- `aptitude` include un sistema abbastanza potente per la ricerca di pacchetti particolari e per limitare i pacchetti mostrati. Gli utenti familiari con `mutt` ci si abitueranno rapidamente, poichè mutt è stata l'ispirazione per la sintassi. Vedere "SEARCHING, LIMITING, e EXPRESSIONS" in `/usr/share/doc/aptitude/README`.
- `aptitude` nella modalità a tutto schermo ha la funzione di `su` incastonata e può essere fatto girare da normale utente finchè non si ha realmente bisogno dei privilegi amministrativi.

6.2.4 `dselect`

Nelle versioni stabili da Potato in su, `dselect` era il principale strumento di gestione dei pacchetti. Per Sarge, considerate invece di usare `aptitude`.

Una volta lanciato, `dselect` seleziona automaticamente tutti i pacchetti "Essenziali", "Importanti", e "Steard".

`dselect` presenta un'interfaccia utente un pò strana. Comunque sia, sono in tanti ad essersi abituati. Quattro sono i comandi ambigui (Maiuscolo significa MAIUSCOLO!):

Tasto	Azione
Q	Esci. Conferma la selezione corrente ed esce comunque (prevale sulle dipendenze)
R	Torna indietro. Non intendevo farlo.
D	Dannazione! Non mi interessa cosa ne pensa dselect. Fallo e basta!
U	Riporta tutto allo stato sUggerito

Useo `D` e `Q`, potete scegliere delle selezioni che presentano conflitti di dipendenze a vostro rischio e pericolo. Usate questi comand con cura.

Aggiungete una linea con l'opzione "expert" in `/etc/dpkg/dselect.cfg` per ridurne il "rumore".

Se la vostra macchine è lenta nell'esecuzione di `dselect`, prendete in considerazione di eseguire `dselect` su un'altra macchina (più veloce) per trovare i pacchetti che vi servono, poi usate `apt-get install` sulla macchina lenta per installarli.

6.2.5 Tracciare una distribuzione con APT

Per tracciare la distribuzione `testing` man mano che cambia, modificate il vostro `/etc/apt/preferences` in maniera che somigli a quanto segue:

```
Package: *
Pin: release a=testing
Pin-Priority: 800

Package: *
Pin: release a=stable
Pin-Priority: 600
```

Notate che il tracciamento della distribuzione `testing` può avere come effetto il ritardo nell'installazione degli aggiornamenti per la sicurezza. Tali pacchetti vengono prima caricati in `unstable` e poi migrano in `testing` solo dopo un lasso di tempo.

Vedere `apt_preferences(5)` per esempi più complessi che permettono, per esempio, di tracciare `testing` mentre si installano pacchetti selezionati da `unstable`.

Esempi di che bloccano alcuni pacchetti a determinate versioni, mentre si tracciano altri pacchetti via via che vengono rilasciati, sono disponibili nella sottodirectory degli esempi (`http://www.debian.org/doc/manuals/debian-reference/examples/`) come `preferences.testing` e `preferences.unstable`.

Se mischiate le distribuzioni, p.es. `testing` con `stable` o `unstable` con `stable`, vi troverete infine a prendere pacchetti fondamentali, come `libc6` da `testing` o `unstable` e non c'è garanzia che essi non conterranno bachi. Siete stati avvertiti.

Un altro esempio, `preferences.stable`, forza tutti i pacchetti ad essere degradati a `stable`.

Il declassamento di un **pacchetto** da una versione più recente ad una meno non è ufficialmente supportato in Debian. Tuttavia, potreste aver bisogno di declassare un dato pacchetto per reinstallarne una versione che sia funzionante, queo quella più nuova non lo è. Queste versione precedenti le trovate localmente in `/var/cache/apt/archives/` o remotamente in `http://snapshot.debian.net/`. Vedere anche 'Salvataggio mediante `dpkg`' a pagina 86.

Anche il declassamento di una **distribuzione** ad una meno recente non è ufficialmente supportato e quasi sicuramente causerà problemi. Tuttavia, se siete disperati, può valer la pena tentarlo come ultima risorsa.

6.2.6 I comandi di `aptitude`, `apt-get` e `apt-cache`

Mentre tracciate `testing`, come descritto sopra, potete gestire il sistema con i comandi seguenti:

- `aptitude upgrade` (o `apt-get upgrade` o `aptitude dist-upgrade` o `apt-get dist-upgrade`)

  Tracciano la distribuzione `testing` — aggiornano tutti i pacchetti del sistema dopo aver installato le versioni dei pacchetti da cui Dipende, dalla distribuzione `testing`. [2]

- `apt-get dselect-upgrade`

  Traccia `testing` — aggiorna tutti i pacchetti in accordo con la selezioni di `dselect`.

- `aptitude -R -G install` *`pacchetto`* (o `apt-get install` *`pacchetto`*)

  Installa *pacchetto* ed i pacchetti da cui Dipende da `testing`.

- `aptitude -r -G install` *`pacchetto`*

  Installa *pacchetto* ed i pacchetti da cui Dipende e che Raccomanda da `testing`.

- `aptitude -r -g install` *`pacchetto`*

  Installa *pacchetto* ed i pacchetti da cui Dipende e che raccomanda o Suggerisce da `testing`.

- `aptitude install` *`pacchetto`*`/unstable`

  Installa *pacchetto* da `unstable`, mentre installa le sue dipendenze da `testing`.

- `aptitude install -t unstable` *`pacchetto`*

  Installa *pacchetto* da `unstable`, mentre installa le sue dipendenze anche da `unstable` impostando la Pin-Priority di `unstable` a 990.

- `apt-cache policy` *`foo bar ...`*

  Controlla lo stato dei pacchetti *foo bar ...*.

- `aptitude show` *`foo bar ...`* `| less` (o `apt-cache show` *`foo bar ...`* `| less`)

  Controlla le informazioni per i pacchetti *foo bar ...*.

- `aptitude install` *`foo=2.2.4-1`*

  Installa la data versione *2.2.4-1* del pacchetto *foo*.

- `aptitude install` *`foo bar-`*

  Installa il pacchetto *foo* e rimuove *bar*

- `aptitude remove` *`bar`*

  Rimuove il pacchetto *bar* ma non i suoi file di configurazione.

---

[2] La differenza fra `upgrade` e `dist-upgrade` è evidente solo queo nuove versioni dei pacchetti hanno relazioni di dipendenza diverse rispetto alle vecchie versioni. Vedere `apt-get(8)` per i dettagli. `aptitude upgrade` e `aptitude dist-upgrade` lanciano `aptitude` in modalità riga di comando. Potete passare a tutto schermo premendo il tasto `e`.

- `aptitude purge` *`bar`*

  Rimuove *bar* insieme ai suoi file di configurazione.

Nell'esempio sopra, dando ad `apt-get` l'opzione `-u` gli fa scrivere una lista di tutti i pacchetti da aggiornare e aspettare l'utente prima di intraprendere le azioni. Quanto segue gli fa sempre `apt-get` fare così:

```
$ cat >> /etc/apt/apt.conf << .
// Mostra sempre i pacchetti che devono essere aggiornati (-u)
APT::Get::Show-Upgraded "true";
.
```

Utilizzare l'opzione `--no-act` per simulare le azioni senza realmente installare, rimuovere, ecc. alcun pacchetto.

6.3 Comandi di sopravvivenza Debian

Conoscendoli, potrete vivere una vita di **eterni** "upgrade" :-)

6.3.1 Controllo dei bachi in Debian e ricerca di aiuto

Se avete dei problemi con un pacchetto in particolare, controllate prima questi siti e poi, eventualmente, cercate aiuto o segnalate un baco. (`lynx`, `links` e `w3m` funzionano tutti bene):

```
$ lynx http://bugs.debian.org/
$ lynx http://bugs.debian.org/nome-pacchetto  # se conoscete il nome del pacchett
$ lynx http://bugs.debian.org/numerobaco     # se conoscete il numero del baco se
```

Cercate su Google (www.google.com) includendo nelle frasi da ricercare "site:debian.org".

Se avete ancora dubbi, leggete i manuali. Impostate `CDPATH` come segue:

```
export CDPATH=.:/usr/local:/usr/share/doc
```

e digitate

```
$ cd nomepachetto
$ pager README.Debian # se esiste
$ mc
```

Maggiori fonti di supporto sono descritte in 'Supporto per Debian' a pagina 249.

6.3.2 Aggiornamento con APT: risoluzione dei problemi

Problemi di dipendenze possono accadere quando si esegue l'aggiornamento in `unstable` o `testing`, come descritto in 'Aggiornamento' a pagina 76. Molte volte ciò accade poichè un pacchetto da aggiornare presenta delle nuove dipendenze che non vengono soddisfatte. Problemi del genere vengono risolti usando

```
# apt-get dist-upgrade
```

Se questo non funziona, allora ripetete uno dei seguenti comandi finchè il problema non si risolve da sè:

```
# aptitude -f upgrade        # continua l'aggiornamento nonostante l'errore
... oppure
# aptitude -f dist-upgrade   # continua il dist-upgrade nonostante l'errore
```

Alcuni script di aggiornamento realmente difettosi possono causare problemi ricorrenti. Per risolvere tali situazioni è in genere buona cosa ispezionare gli script `/var/lib/dpkg/info/packagename.`*`{post-,pre-}{inst,rm}`* del pacchetto responsabile e poi eseguire:

```
# dpkg --configure -a    # configura tutti i pacchetti installati parzialmente
```

Se uno script lamenta la mancanza di un file di configurazione, date un'occhiata in `/etc` per il file di configurazione corrispondente. Se ne esiste uno con l'estensione `.dpkg-new` (o qualcosa di simile), `mv` per rimuovere il suffisso.

Problemi di dipendenze possono accadere installando i pacchetti in `unstable` o `testing`. Esistono dei metodi per aggirare le dipendenze.

```
# aptitude -f install pacchetto # ignora le dipendenze difettose
```

Un metodo alternativo consiste nell'utilizzare il pacchetto `equivs`. Vedere `/usr/share/doc/equivs/README.Debian` e 'Il pacchetto `equivs`' a pagina 96.

6.3.3 Salvataggio mediante `dpkg`

Se siete in un vicolo cieco con APT, potete scaricare i pacchetti dai mirror Debian ed installarli con `dpkg`. Se non avete accesso alla rete, potete cercare le copie dei pacchetti in `/var/cache/apt/archives/`.

```
# dpkg -i fetchmail_6.2.5-4_i386.deb
```

Se l'installazione di un pacchetto fallisce a causa di violazioni delle dipendenze e avete realmente bisogno di quel pacchetto, potete scavalcare i controlli sulle dipedenze usando le opzioni di `dpkg` `--ignore-depends`, `--force-depends` ed altre ancora. Vedere `dpkg(8)` per i dettagli.

6.3.4 Recuperare i dati sulla selezione dei pacchetti

Se `/var/lib/dpkg/status` diventa illeggibile per una qualsivoglia ragione, il sistema Debian perde i dati sulla selezione dei pacchetti, soffrendone in maniera severa. Cercate il vecchio file `/var/lib/dpkg/status-old`, oppure `/var/backups/dpkg.status.*`.

Tenere `/var/backups/` in una partizione separata può essere una buona idea, dato che questa directory contiene molti dei dati fondamentali del sistema.

Se il vecchio file `/var/lib/dpkg/status` non è disponibile, potete ancora recuperare le informazioni dalle directory in `/usr/share/doc/`.

```
# ls /usr/share/doc | \
  grep -v [A-Z] | \
  grep -v '^texmf$' | \
  grep -v '^debian$' | \
  awk '{print $1 " install"}' | \
  dpkg --set-selections
# dselect --expert # reinstalla il sistema, deseleziona se necessario
```

6.3.5 Recupero del sistema dopo danneggiamento di `/var`

Dato che la directory `/var` contiene dati che vengono regolarmente aggiornati, tipo la posta, è più suscettibile di corruzione di, p. es., `/usr/` Metterla in una partizione separata limita i rischi. Se accade l'irreparabile, bisogna ricostruirla per salvare il sistema Debian.

Procuratevi una directory `/var` con il contenuto ridotto all'osso, da un sistema Debian minimale e funzionante, basato su una versione uguale o più vecchia, per esempio `var.tar.gz` (`http://people.debian.org/~osamu/pub/`) e piazzatela nella root directory del sistema danneggiato. Poi,

```
# cd /
# mv var var-old      # se vi è rimasto qualcosa di utile
# tar xvzf var.tar.gz # usate il file preso da Woody
# aptitude            # o dselect
```

Ciò dovrebbe restituirvi un sistema funzionante. Potete velocizzare la procedura di recupero delle selezioni dei pacchetti usando la tecnica descritta in 'Recuperare i dati sulla selezione dei pacchetti' in questa pagina. ([FIXME]: This procedure needs more experiments to verify.)

6.3.6 Installare un pacchetto su un sistema non bootabile

Fate il boot in Linux mediante un floppy o CD di salvataggio Debian, o tramite una partizione alternativa se avete un sistema con multiboot in Linux. Vedere 'Avviare il sistema' a pagina 107. Montate il sistema non bootabile su `/target` ed usate il modo chroot di installazione di `dpkg`.

```
# dpkg --root /target -i packagefile.deb
```

Poi configurate il tutto e risolvete i problemi.

A proposito, se tutto quello che impedisce il boot è un `lilo` malfunzionante, potete fare il boot con un disco di salvataggio Debian standard. Al prompt del boot, ammettendo che la partizione root della vostra installazione è su `/dev/hda12` e volete un runlevel 3, digitate:

```
boot: rescue root=/dev/hda12 3
```

A questo punto avrete accesso ad un sistema pressochè perfettamente funzionante con il kernel su dischetto. (Potranno esserci inconvenienti minori dovuti alla mancanza di moduli o di funzioni particolari del kernel.)

6.3.7 Cosa fare se `dpkg` non funziona

Un `dpkg` malfunzionante può rendere impossibile l'installazione di qualsiasi file `.deb`. Una procedura come la seguente vi aiuterà ad uscire da questa situazione. (Nella prima riga, potete sostituire "links" con il vostro browser da terminale preferito.)

```
$ links http://http.us.debian.org/debian/pool/main/d/dpkg/
... scarica un valido dpkg_version_arch.deb
$ su
password: *****
$ ar x dpkg_version_arch.deb
# mv data.tar.gz /data.tar.gz
# cd /
# tar xzfv data.tar.gz
```

Per `i386`, `http://packages.debian.org/dpkg` può anche essere utilizzata come URL.

6.4 Comandi del "paradiso" Debian

L'**Illuminazione** ottenuta con questi comandi salverà una persona dall'eterno ciclo del karma dell'inferno degli aggiornamenti, permettendogli di raggiungere il **nirvana** Debian. :-)

6.4.1 Informazioni su di un file

Per scoprire a quale pacchetto un file appartiene:

```
$ dpkg {-S|--search} modello # trova i modelli nei pacchetti installati
$ wget http://ftp.us.debian.org/debian/dists/sarge/Contents-i386.gz
$ zgrep -e pattern Contents-i386.gz
      # trova l'elenco dei file con la stessa radice (o il file) nell'archivio De
```

Potete utilizzare anche dei comandi specifici per i pacchetti:

```
# aptitude install dlocate
  # è in conflitto con slocate (la versione sicura di locate)
$ dlocate filename # alternativa rapida a dpkg -L e dpkg -S
...
# aptitude install auto-apt # strumento per l'installazione a richiesta dei pacch
# auto-apt update          # crea un database per auto-apt
$ auto-apt search modello
 # effettua una ricerca completa nell'archivio pacchetti
```

6.4.2 Informazioni su di un pacchetto

Cerca le informazioni negli archivi dei pacchetti e le visualizza. Accertatevi che APT sia indirizzato verso l'archivio(i) appropriato(i) modificando `/etc/apt/sources.list`. Se volete vedere come si comportano i pacchetti in `testing/unstable` rispetto a quelli correntemente installati utilizzate `apt-cache policy`— un comando molto utile.

```
# apt-get   check           # aggiorna la cache e controlla le dipendenze
$ apt-cache search  testo # cerca un pacchetto a partire dalla descrizione
$ apt-cache policy  pacchetto # informazioni su priorità e distribuzione di un pa
$ apt-cache show -a pacchetto # mostra la descrizione di un pacchetto per ogni di
$ apt-cache showsrc pacchetto # mostra la descrizione del pacchetto sorgente corr
$ apt-cache showpkg pacchetto # Informazioni per il debug
# dpkg  --audit|-C          # cerca i pacchetti parzialmente installati
$ dpkg {-s|--status} pacchetto... # descrizione del pacchetto installato
$ dpkg -l pacchetto ... # stato del pacchetto installato (1 linea)
$ dpkg -L pacchetto ... # elenca i file installati per un dato pacchetto
```

`apt-cache showsrc` non è documentato al momento del rilascio di Woody, ma funziona :)

E' sempre possibile trovare informazioni sui pacchetti in (Io uso `mc`):

```
/var/lib/apt/lists/*
/var/lib/dpkg/available
```

Il confronto dei file seguenti fornisce informazioni su cosa è realmente accaduto nelle ultime sessioni di installazione.

```
/var/lib/dpkg/status
/var/backups/dpkg.status*
```

6.4.3 Installazione automatica con APT

Per una installazione automatica, senza controllo alcuno, aggiungete la seguente riga a `/etc/apt/apt.conf`:

```
Dpkg::Options {"--force-confold";}
```

Questo è l'equivalente di `aptitude -y install` *`nomepacchetto`*. o `apt-get -q-y install` *`nomepacchetto`*. Siccome in questo modo si risponde "si" a tutte le domande, usate questo trucco con accortezza. Vedere `apt.conf(5)` e `dpkg(1)`.

Si può configurare qualsiasi pacchetto in un secondo momento seguendo 'Riconfigurare i pacchetti già installati' in questa pagina.

6.4.4 Riconfigurare i pacchetti già installati

Usate i seguenti comandi per riconfigurare un pacchetto già installato.

```
# dpkg-reconfigure --priority=medium pacchetto [...]
# dpkg-reconfigure --all  # riconfigura tutti i pacchetti
# dpkg-reconfigure locales # genera qualsiasi locale extra
# dpkg-reconfigure --p=low xserver-xfree86 # riconfigura il server X
```

Eseguite questo tramite `debconf` se avete la necessità di cambiarne la configurazione dei messaggi in modo permanente.

Alcuni programmi hanno degli script speciali di configurazione. [3]

```
apt-setup     - crea /etc/apt/sources.list
install-mbr   - installa un manager di Master Boot Record
tzconfig      - imposta il fuso orario locale
gpmconfig     - imposta il demone per mouse gpm
sambaconfig   - configura Samba (Woody usa debconf)
eximconfig    - configura Exim (MTA)
texconfig     - configura teTeX
apacheconfig  - configura Apache (httpd)
cvsconfig     - configura CVS
sndconfig     - configura il sistema audio
...
update-alternatives - imposta i comandi predefiniti, p.e. vim per vi
update-rc.d         - Gstione degli script di inzializzazione del System-V
update-menus        - Il sistema dei menu Debian
...
```

---

[3] Alcuni script `*config` stanno scomparendo nelle nuove versioni di Sarge e la funzionalità di configurazione dei pacchetti viene trasferita al sistema `debconf`.

6.4.5 Rimozione e purga dei pacchetti

Rimuove un pacchetto mantenendone la configurazione:

```
# aptitude remove pacchetto ...
# dpkg   --remove pacchetto ...
```

Rimuove un pacchetto e la sua configurazione:

```
# aptitude purge pacchetto ...
# dpkg   --purge        pacchetto ...
```

6.4.6 Mantenere vecchi pacchetti

Per esempio, per mantenere `libc6` e `libc6-dev` con `dselect` e `aptitude install` *`pacchetto`* si può procedere come segue:

```
# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections
```

`aptitude install` *`pacchetto`* non verrà impedito da quell'"hold". Per mantenere un pacchetto forzandone un declassamento automatico con `aptitude upgrade` *`pacchetto`* oppure `aptitude dist-upgrade`, aggiungete le seguenti linee ad `/etc/apt/preferences`:

```
Package: libc6
Pin: release a=stable
Pin-Priority: 2000
```

Qui la voce "`Package:`" non può usare termini tipo "`libc6*`". Se dovete mantenere tutti i pacchetti binari correlati al pacchetto sorgente `glibc` in una versione sincronizzata, dovete elencarli esplicitamente.

Il comando seguente mostra i pacchetti tenuti in sospeso (non aggiornati):

```
dpkg --get-selections "*"|grep -e "hold$"
```

6.4.7 Sistema misto `stable/testing/unstable`

`apt-show-versions` può elencare le versioni dei pacchetti disponibili per ciascuna distribuzione.

```
$ apt-show-versions | fgrep /testing | wc
... quanti pacchetti avete da testing
$ apt-show-versions -u
... elenco di pacchetti aggiornabili
$ aptitude install `apt-show-versions -u -b | fgrep /unstable`
... aggiorna tutti i pacchetti da unstable alle loro versioni più recenti
```

6.4.8 Potare i file della cache

L'installazione dei pacchetti tramite APT lascia i loro file archiviati in `/var/cache/apt/archives` e questi vanno rimossi.

```
# aptitude autoclean # rimuove solo pacchetti inutili
# aptitude clean     # rimuove tutti i pacchetti nella cache
```

6.4.9 Salvare/copiare la configurazione del sistema

Per fare una copia locale dello stato della selezione dei pacchetti:

```
$ dpkg --get-selections "*" >lemieselezioni   # oppure usare \*
```

`"*"` fa includere anche i nomi dei pacchetti marcati con "elimina".

Potete poi trasferire questo file su un altro computer ed installarlo con:

```
# dselect update
# dpkg --set-selections <lemieselezioni
# apt-get -u dselect-upgrade # o dselect install
```

6.4.10 Portare un pacchetto nel sistema `stable`

Se si eseguono degli aggiornamenti parziali di un sistema `stable`, può essere buona cosa ricreare un pacchetto all'interno del proprio ambiente a partire dai sorgenti. In tal modo si eviteranno dei massicci aggiornamenti di pacchetti, dovuti alle loro dipendenze. Per prima cosa, aggiungete le linee seguenti a `/etc/apt/sources.list`:

```
deb-src http://http.us.debian.org/debian testing \
 main contrib non-free
deb-src http://non-us.debian.org/debian-non-US testing/non-US \
 main contrib non-free
deb-src http://http.us.debian.org/debian unstable \
 main contrib non-free
deb-src http://non-us.debian.org/debian-non-US unstable/non-US \
 main contrib non-free
```

Ogni linea che inizia con `deb-src` qui è divisa in due parti per ragioni di stampabilità, ma in `sources.list` dovranno essere ciascuna su un'unica riga.

Poi recuperate i sorgenti e costruite un pacchetto locale:

```
$ apt-get update  # aggiorna la lista dei pacchetti sorgente
$ apt-get source pacchetto
$ dpkg-source -x pacchetto.dsc
$ cd versione-pacchetto
  ... controlla i pacchetti necessari (Build-depends nel file
      .dsc) e li installa. Avete bisogno anche del pacchetto "fakeroot".

$ dpkg-buildpackage -rfakeroot

...oppure (senza "signature")
$ dpkg-buildpackage -rfakeroot -us -uc utilizzate "debsign" in seguito, se necess

...poi per installare il pacchetto
$ su -c "dpkg -i pacchetto.deb"
```

Normalmente, per soddisfare le dipendenze si ha la necessità di installare alcuni pacchetti con il suffisso "-dev". `debsign` è contenuto nel pacchetto `devscripts`. `auto-apt` può aiutare nel soddisfare queste dipendenze. `fakeroot` evita l'uso non necessario del root account.

In Woody, questi problemi di dipendenze possono essere semplificati. Per esempio per compilare solo il sorgente del pacchetto `pine`:

```
# apt-get build-dep pine
# apt-get source -b pine
```

6.4.11 Archivio locale dei pacchetti

Per creare un archivio locale dei pacchetti che sia compatibile con APT e `dselect`, bisogna creare `Packages` ed i file dei pacchetti devono essere organizzati in un particolare albero directory.

Si può creare un deposito locale `deb` simile ad un archivio ufficiale Debian in questo modo:

```
# aptitude install dpkg-dev
# cd /usr/local
# install -d pool # i pacchetti sono fisicamente localizzati qui
# install -d dists/unstable/main/binary-i386
# ls -1 pool | sed ’s/_.*$/ priority section/’ | uniq > override
# editor override # adjust priority e section
# dpkg-scanpackages pool override /usr/local/ \
```

```
   > dists/unstable/main/binary-i386/Packages
# cat > dists/unstable/main/Release << EOF
Archive: unstable
Version: 3.0
Component: main
Origin: Local
Label: Local
Architecture: i386
EOF
# echo "deb file:/usr/local unstable main" \
   >> /etc/apt/sources.list
```

In alternativa, un altro metodo per creare un archivio locale `deb`, molto alla bruta:

```
# aptitude install dpkg-dev
# mkdir /usr/local/debian
# mv /dove/è/pacchetto.deb /usr/local/debian
# dpkg-scanpackages /usr/local/debian /dev/null | \
  gzip - > /usr/local/debian/Packages.gz
#  echo "deb file:/usr/local/debian ./" >> /etc/apt/sources.list
```

Questi archivi possono essere raggiunti da remoto tramite HTTP od FTP, modificeo le voci in `/etc/apt/sources.list` di conseguenza.

6.4.12 Convertire od installare un pacchetto binario di altra distribuzione

`alien` permette la conversione dei pacchetti binari nei formati Red Hat `rpm`, Stampede `slp`, Slackware `tgz`, e Solaris `pkg`, in pacchetti Debian `deb`. Se volete usare un pacchetto proveniente da un'altra distribuzione di Linux al posto di quello installato sul vostro sistema, potete utilizzare `alien` per convertirlo nel formato che più preferire. `alien` supporta anche pacchetti LSB.

6.4.13 Comando di installazione automatica

`auto-apt` è uno strumento di installazione dei pacchetti a richiesta

```
$ sudo auto-apt update
 ... aggiorna il database
$ auto-apt -x -y run
Entering auto-apt mode: /bin/bash
Exit the command to leave auto-apt mode.
$ less /usr/share/doc/med-bio/copyright # access non-existing file
 ...  Installa il pacchetto che fornisce questo file.
 ... Installa anche le dipendenze
```

6.4.14 Verificare i file dei pacchetti installati

`debsums` permette la verifica dei file dei pacchetti installati contro gli MD5 checksums. Alcuni pacchetti non hanno MD5 checksums. Una soluzione temporanea per gli amministratori di sistema:

```
# cat >>/etc/apt/apt.conf.d/90debsums
DPkg::Post-Install-Pkgs {"xargs /usr/bin/debsums -sg";};
^D
```

da Joerg Wendle <joergle@debian.org> (non testata).

6.4.15 Ottimizzare `sources.list`

In breve, sforzi sovrumani per creare un `sources.list` ottimizzato, non hanno prodotto per me miglioramenti, da una località negli . Ho scelto manualmente un sito vicino usando `apt-setup`.

`apt-spy` crea automaticamente `sources.list`, basandosi su latenza e larghezza di banda. `netselect-apt` crea un `sources.list` ancora più completo, ma usa un metodo meno efficace per la scelta del mirror migliore (comparazione dei tempi di ping).

```
# aptitude install apt-spy
# cd /etc/apt ; mv sources.list sources.list.org
# apt-spy -d testing -l sources.apt
```

6.5 Altre particolarità di Debian

6.5.1 Il comando `dpkg-divert`

Le **deviazioni** dei files rappresentano un modo di forzare `dpkg` a non installare un file nella sua posizione predefinita, ma in una posizione **deviata**. Le **deviazioni** possono essere utilizzate tramite gli script dei pacchetti Debian per muovere un file quando causa un conflitto. Gli amministratori di sistema possono usare una deviazione anche per scavalcare un file di configurazione di un pacchetto, oppure quando alcuni file (non marcati come conffiles) devono essere salvaguardati da `dpkg`, durante l'installazione di una nuova versione di un dato file (vedere 'Mantenimento della configurazione locale' a pagina 13).

```
# dpkg-divert [--add]  filename # aggiunge la "deviazione"
# dpkg-divert --remove filename # rimuove la "deviazione"
```

Di solito è una buona idea non usare `dpkg-divert` a meno che non sia strettamente necessario.

6.5.2 Il pacchetto `equivs`

Se compilate un programma dai sorgenti, è meglio debianizzarlo in un pacchetto (`*.deb`). Usate `equivs` solo come ultima risorsa.

```
Package: equivs
Priority: extra
Section: admin
Description: Aggira le dipendenze dei pacchetti Debian. E' un pacchetto
 vuoto che può essere usato per creare dei pacchetti Debian contenenti
 solo le informazioni sulle dipendenze.
```

6.5.3 Comandi alternativi

Per lanciare `vim` con il comando `vi`, usate `update-alternatives`:

```
# update-alternatives --display vi
...
# update-alternatives --config vi
  Selection    Comme
-----------------------------------------------
      1        /usr/bin/elvis-tiny
      2        /usr/bin/vim
*+    3        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 2
```

Le alternative del sistema in Debian sono contenute in `/etc/alternatives` come collegamenti simbolici.

Per impostare il vostro ambiente X Window preferito, usate invece `update-alternatives` su `/usr/bin/x-session-manager` e `/usr/bin/x-window-manager`. Per i dettagli, vedere 'Personalizzare le X session' a pagina 151.

`/bin/sh` è un collegamento simbolico a `/bin/bash` o `/bin/dash`. E' più sicuro usare `/bin/bash` per mantenere la compatibilità con vecchi script in contaminati da bash, ma più corretto utilizzare `/bin/dash` per rinforzare la compatibilità POSIX. L'aggiornamento al kernel 2.4 tende ad impostare il collegamento su `/bin/dash`

6.5.4 Uso dei runlevel

Una volta installati, gran parte dei pacchetti Debian configura i propri servizi per girare nei runlevel da 2 a 5. Quindi, non esistono differenze fra i runlevel 2, 3, 4 e 5 su un sistema Debian non personalizzato; Debian lascia all'amministratore la possibilità di personalizzare i runlevel, come descritto in ‘Personalizzare i runlevel’ a pagina 22. Ciò differisce dal modo in cui i runlevel sono usati da altre distribuzioni popolarin GNU/Linux. Una modifica che potreste voler fare e disabilitare `xdm` o `gdm` nel runlevel 2 in maniera da non lanciare l’X display manager al termine della sequenza di boot; potete lanciarlo passando al runlevel 3.

Per maggiori informazioni sui runlevel, vedere ‘I Runlevel’ a pagina 21.

6.5.5 Demoni di servizio disabilitati

Gli sviluppatori Debian hanno molto a cuore la sicurezza del sistema. Molti demoni di servizio sono installati con il minimo dei servizi abilitati.

Se avete dei dubbi (su Exim, DHCP, . . .) controllate `ps aux` oppure il contenuto di `/etc/init.d/*` e di `/etc/inetd.conf`. Controllate anche `/etc/hosts.deny` in ‘Restrizione tramite PAM’ a pagina 137. Anche il comando `pidof` è utile (vedere `pidof(8)`).

X11 non permette connessioni TCP/IP (remote) predefinite nelle versioni più recenti di Debian. Vedere ‘Connessione TCP/IP ad X’ a pagina 154. Anche l’X forwarding in SSH è disabilitato. Vedere ‘Connessioni ad un X server remoto – `ssh`’ a pagina 155.

Capitolo 7

Il kernel Linux su Debian

Debian ha i suoi modi per la ricompilazione del kernel e dei relativi moduli. Vedere anche 'Debian ed il kernel' a pagina 23.

7.1 Ricompilare il kernel

Utilizzare `gcc`, `binutils` e `modutils` da Debian `unstable` può essere di aiuto nel compilare l'ultimissimo kernel Linux. Vedere `/usr/share/doc/kernel-package/README.gz`, in particolare la parte finale, per le informazioni ufficiali.

Dato che la compilazione del kernel è qualcosa in continua evoluzione, è un soggetto difficile da trattare, persino lo sviluppatore più ammirato può fare confusione:

Manoj Srivastava ha scritto: `--initrd` richiede una patch cramfs solo per Debian.

Herbert Xu ha scritto: No, non ne ha bisogno, tutto quello che serve per usare un filesystem diverso da CRAMFS è di impostare MKIMAGE in `/etc/mkinitrd/mkinitrd.conf`.

Siate accorti, e fate affidamento sul `/usr/share/doc/kernel-package/README.gz` di Manoj e Kent. Accertatevi di ottenere l'ultima versione di unstable del pacchetto `kernel-package` se state per compilare la versione più recente del kernel.

`initrd` non serve per un kernel compilato solo per una macchina. Io lo uso perchè voglio che sia quasi lo stesso di quello fornito da kernel-image. Se usate `initrd`, leggetevi`mkinitrd(8)` e `mkinitrd.conf(5)`. Vedere anche `http://bugs.debian.org/149236`.

7.1.1 Il metodo Debian standard

Controllate anche i rapporti sui bachi per `kernel-package` `gcc`, `binutils` e `modutils`. Usatene le versioni più recenti, se necessario.

La compilazione di un proprio kernel dai sorgenti in un sistema Debian richiede una cura speciale. Per compilare più "kernel-images", utilizzate il nuovo `--append_to_version` con `make-kpkg`.

```
# apt-get install debhelper modutils kernel-package libncurses5.dev
# apt-get install kernel-source-2.4.18 # usate la versione più recente
# apt-get install fakeroot
# vi /etc/kernel-pkg.conf    # inserisco il mio nome ed email
$ cd /usr/src                # la directory di compilazione
$ tar -bzip2 -xvf kernel-source-2.4.18.tar.bz2
$ cd kernel-source-2.4.18    # se è il vostro sorgente
$ cp /boot/config-2.4.12-386.config
        # usa il config corrente come default
$ make menuconfig            # personalizzate a vostro piacimento
$ make-kpkg clean            # deve essere eseguito (per: man make-kpkg)
$ fakeroot make-kpkg --append_to_version -486 --initrd \
        --revision=rev.01 kernel_image \
 modules_image # modules_image è per pcmcia.cs*, ecc.
$ cd ..
# dpkg -i kernel-image*.deb pcmcia-cs*.deb # install
```

`make-kpkg kernel_image` esegue in realtà `make oldconfig` ed `make dep` Se non usate initrd, non mettete `-initrd`.

Se si vogliono i moduli da pcmcia-cs o nessun modulo pcmcia, si deve selezionare "General setup —>" e "PCMCIA/CardBus support —>" in `make menuconfig` ed impostando la configurazione come "< > PCMCIA/CardBus support" (cioè, deselezionate la casella).

Su una macchina SMP, impostate CONCURRENCY_LEVEL in accordo con `kernel-pkg.conf(5)`.

7.1.2 Il metodo classico

Scaricate i sorgenti originali da:
- Linux: `http://www.kernel.org/`
- pcmcia-cs: `http://pcmcia-cs.sourceforge.net/`

oppure utilizzate i sorgenti equivalenti da debian ed eseguite i seguenti comandi:

```
# cd /usr/src
# tar xfvz linux-versione.tar.gz
# rm -rf linux
# ln -s linux-versione linux
# tar xfvz pcmcia-cs-versione.tar.gz
# ln -s pcmcia-cs-versione pcmcia
# cd linux
# make menuconfig
... configurate a vostro piacimento ...
# make dep
# make bzImage
```

```
... modifiche per lilo / grub ...
... muovete /usr/src/linux/arch/i386/boot/bzImage in boot ...
... /sbin/lilo o qualunque altra cosa serva per grub
# make modules; make modules_install
# cd ../pcmcia
# make config
# make all
# make install
... aggiungete i nomi dei moduli richiesti a /etc/modules
# shutdown -r now
... fate il boot con il nuovo kernel ...
```

7.1.3 I Kernel header

Molti programmi “normali” non hanno bisogno dei kernel header, anzi, possono corrompersi se li utilizzate direttamente; dovrebbero invece essere compilati rispetto agli headers con cui **`glibc` è stato compilato**, ovvero le versioni contenute in `/usr/include/linux` e `/usr/include/asm` del sistema Debian.

Per cui non ponete i collegamenti simbolici alle directory in `/usr/src/linux` da `/usr/include/linux` e `/usr/include/asm`, come diversamente suggerito in alcuni documenti ormai obsoleti.

Se avete **bisogno** di kernel header particolari per qualche applicazione kernel-specifica, modificate il(i) makefile in modo da includere nei percorsi dei puntatori a `*dir-del-particolare-kernel-header*/include/linux` e `*dir-del-particolare-kernel-header*/include/asm`.

7.2 Kernel 2.4 modulare

I nuovi kernel debian 2.4 forniti dai pacchetti kernel-image-2.4.*NN* sono altamente modularizzati. Per farli funzionare come avete deciso, dovrete essere sicuri che i moduli siano stati attivati.

Sebbene abbia molti esempi per `/etc/modules` nella sezione seguente da utilizzare come correzione rapida, so che il modo giusto per correggere questi problemi correlati ai moduli è di fornire un alias per il dispositivo in un file contenuto in `/etc/modutils/`, essendoci abbastanza alias disponibili con i kernel attuali. Alcuni moduli possono essere auto attivati dai programmi di riconoscimento hardware, tipo `discover`. Vedere anche ‘Pacchetti per il riconoscimento hardware per X’ a pagina 147

Vedere ‘Funzioni speciali per trattare con i moduli’ a pagina 25 e `Documentation/*.txt` nel sorgente Linux source per informazioni più accurate.

7.2.1 PCMCIA

`/etc/modules` deve contenere i seguenti driver, affinchè PCMCIA funzioni:

```
# Driver ISA PnP
isa-pnp
# Driver PCMCIA di basso livello
# yenta_socket # non sembra essere necessario nel mio caso.
```

Il resto viene configurato dagli script PCMCIA (del pacchetto `pcmcia-cs`), da `depmod` e da `kmod`. Penso che `isa-pnp` mi sia stato necessario perchè il mio laptop è un vecchio ISA-PCMCIA. I modelli più recenti con CardBus-PCMCIA non lo richiedono.

Voce del generoso Miquel van Smoorenburg `<miquels@cistron.nl>`:

“Ho semplicemente rimosso tutta la roba correlata con pcmcia dal mio laptop, qui al lavoro, compreso cardmgr, ecc., ed ho installato un kernel 2.4 con supporto per cardbus, insieme al nuovo pacchetto `hotplug` da woody.

Se avete delle PC card a 32 bit non avete bisogno del pacchetto pcmcia, il kernel 2.4 ha i servizi per esse già inseriti. Il driver tulip standard dovrebbe funzionare bene con la vostra card dcollegamenti.

—Mike.”

Vedere Linux PCMCIA HOWTO (`http://www.tldp.org/HOWTO/PCMCIA-HOWTO.html`) e ‘Configurazione di rete e PCMCIA’ a pagina 203.

7.2.2 SCSI

[NON TESTATO]`/etc/modules` deve contenere i seguenti driver affinchè SCSI funzioni:

```
# SCSI core
scsi_mod
# Driver generico SCSI
sg
# Disco SCSI
sd_mod
# Tutti gli altri moduli hardware necessari
...
```

Forse `depmod` può prendersi cura di alcuni dei moduli summenzionati.

7.2.3 Funzioni di rete

`/etc/modules` deve contenere i seguenti driver per le funzioni extra di rete:

```
# net/ipv-4
ip_gre
ipip

# net/ipv-4/netfilter
# iptable (nell'ordine)
ip_tables
ip_conntrack
ip_conntrack_ftp
iptable_nat
iptable_filter
iptable_mangle
#
ip_nat_ftp
ip_queue
#
ipt_LOG
ipt_MARK
ipt_MASQUERADE
ipt_MIRROR
ipt_REDIRECT
ipt_REJECT
ipt_TCPMSS
ipt_TOS
ipt_limit
ipt_mac
ipt_mark
ipt_multiport
ipt_owner
ipt_state
ipt_tcpmss
ipt_tos
ipt_unclean
#
#ipchains
#ipfwadm
```

Quanto sopra non è ottimizzato al meglio. `depmod` potrebbe prendersi cura di alcuni dei moduli.

7.2.4 Il filesystem EXT3 (> 2.4.17)

L'attivazione di un filesystem "journaled" con il tipo EXT3 FS richiede i seguenti passi, usando un pacchetto kernel-image Debian precompilato (> 2.4.17):

```
# cd /etc; mv fstab fstab.old
# sed 's/ext2/ext3,ext2/g' <fstab.old >fstab
# vi /etc/fstab
... impostate il filesystem di root ad "auto" invece di "ext3,ext2"
# cd /etc/mkinitrd
# echo jbd >>modules
# echo ext3 >>modules
# echo ext2 >>modules
# cd /
# apt-get update; apt-get install kernel-image-2.4.17-686-smp
... installa l'ultimo kernel ed imposta il boot
    (lilo viene lanciato a questo punto)
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... Per tutti i filesystem EXT2 convertiti a EXT3
# shutdown -r now
```

A questo punto il journaling EXT3 è attivato. Avere le voci "type" in `fstab` come `ext3,ext2` assicura, in caso il kernel non supportasse EXT3, un tranquillo ritorno ad EXT2 per le partizioni non-root.

Se avete un kernel 2.4 già installato e non lo volete reinstallare, date gli stessi comandi `apt-get` di cui sopra e poi:

```
# mkinitrd -o /boot/initrd.img-2.4.17-686-smp /lib/modules/2.4.17-686-smp
# lilo
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... per tutti i filesystem EXT2 convertiti a EXT3
# shutdown -r now
```

A questo punto il journaling EXT3 è attivato.

Qualora `/etc/mkinitrd/modules` non fosse stato impostato al lancio di `mkinitrd` e si volessero aggiungere alcuni moduli al boot:

```
... al prompt di initrd per entrare in shell (entro 5 sec.), premete INVIO
# insmod jbd
# insmod ext3 # modprobe ext3 si prende cura di tutto
# insmod ext2
# ^D
... continuate il boot
```

Sui messaggi al boot (`dmesg`), potrà apparire "cramfs: wrong magic", messaggio noto per essere inoffensivo. Questo problema è stato risolto in Sarge (10/2002). Vedere `http://bugs.`

debian.org/135537 ed il EXT3 File System mini-HOWTO (http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html) oppure `/usr/share/doc/HOWTO/en-txt/mini/extra/ext3-mini-HOWTO.gz` per ulteriori informazioni.

Alcuni sistemi hanno sperimentato dei gravi blocchi del kernel, quando EXT3 è attivato (fino al 2.4.17) ma, per quanto mi riguarda, non ho avuto problemi.

7.2.5 Supporto nel kernel 2.4 per Realtek RTL-8139

Per qualche motivo, il modulo per RTL-8139 non si chiama più rtl8139, ma 8139too. All'aggiornamento del kernel da 2.2 a 2.4, dovrete modificare a mano `/etc/modules` per inserire il nuovo nome del modulo.

7.2.6 Supporto per la porta parallela

In `kernel-image-2.4.*`, il supporto per la porta parallela è fornito come modulo. Attivatelo con:

```
# modprobe lp
# echo lp >> /etc/modules
```

Vedere `Documentation/parport.txt` nel sorgente Linux.

7.3 Mettere a punto il kernel tramite il filesystem proc

Il comportamento del kernel Linux può essere modificato al volo usando il filesystem proc.

Per le informazioni di base sulle modifiche ai parametri del kernel attraverso il filesystem `/proc`, leggete `Documentation/sysctl/*` nella directory Linux.

Potete vedere alcuni esempi di manipolazione dei parametri del kernel in `/etc/init.d/networking` e 'Strani problemi di accesso con alcuni siti web' a pagina 45.

Vedere `sysctl.conf(5)` su come impostare la configurazione del kernel all'avvio attraverso `/proc`, con lo script `/etc/init.d/procps.sh` generalmente lanciato da `/etc/rcS.d/S30procps.sh`.

7.3.1 Troppi file aperti

Il kernel può dare un messaggio del tipo "Too many open files" (troppi file aperti). Ciò è dovuto al basso valore predefinito (8096) per `file-max`. Per risolvere il problema, date il seguente comando da root (oppure mettetelo in uno script di inizio in `/etc/rcS.d/*`.

```
# echo "65536" >/proc/sys/fs/file-max # per i kernel 2.2 e 2.4
# echo "131072" >/proc/sys/fs/inode-max # solo per kernel 2.2
```

oppure mettete quanto segue in `/etc/sysctl.conf` per avere una modifica permanente:

```
file-max=65536   # per kernel 2.2 e 2.4
inode-max=131072 # solo per 2.2
```

7.3.2 Intervalli di flush del disco rigido

Potete modificare gli intervalli di flush mediante il filesystem proc. Quanto segue riduce gli intervalli dai cinque secondi predefiniti ad un secondo.

```
# echo "40 0 0 0 100 30000 60 0 0"  > /proc/sys/vm/bdflush
```

Ciò può avere un minimo impatto negativo nella performance I/O. Però assicura il contenuto dei file, tranne che per l'ultimo secondo, più breve dei cinque predefiniti. Questo è vero anche per i filesystem journaled.

7.3.3 Vecchie macchine lente e con poca memoria

Per alcuni vecchi sistemi con poca memoria, può ancora essere utile abilitare l'over-commit della memoria tramite il filesystem proc.

```
# echo 1 > /proc/sys/vm/overcommit_memory
```

7.4 Il kernel 2.6 con udev

Udev è un rimpiazzo dinamico per `/dev`. I nomi dei device devon essere molto corti. Devfs, usato nel kernel 2.4, è obsoleto.

Potete abilitarlo installando il kernel Debian 2.6 `kernel-image-2.6.`*`NN`* con il pacchetto `udev`.

Capitolo 8

Trucchi per Debian

8.1 Avviare il sistema

Per informazioni dettagliate sul prompt del boot, vedere il BootPrompt-HOWTO (`http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html`) dal LDP.

8.1.1 "Ho scordato la password di root!" (1)

E' possibile fare il boot del sistema ed accedere all'account di root, anche se non se ne conosce la password, basta avere accesso alla tastiera. (Premesso che che non esistono altre password richieste dal BIOS, o dal boot-loader tipo `lilo` che possono impedirvi l'accesso al sistema).

Questa procedura non richiede alcun disco di boot, nè cambiamenti al BIOS. In questo contesto, "Linux" è l'etichetta per lanciare il kernel in una installazione Debian standard.

Alla schermata di boot di `lilo`, non appena appare `boot:` (in alcuni sistemi si deve premere il tasto maiuscolo per prevenire il boot automatico) e se `lilo` usa il framebuffer dovete premere TAB per vedere le opzioni che digitate), date:

```
boot: Linux init=/bin/sh
```

Il sistema avvia il kernel ed esegue `/bin/sh` invece dello standard `init`. A questo punto avete ottenuto i privilegi di root e la shell di root. Siccome, però `/` è montata in sola lettura e molte altre partizioni non sono state ancora montate, avete bisogno di eseguire quanto segue per avere un sistema ragionevolmente funzionante.

```
init-2.03# mount -n -o remount,rw /
init-2.03# mount -avt nonfs,noproc,nosmbfs
init-2.03# cd /etc
init-2.03# vi passwd
init-2.03# vi shadow
```

(Se il secondo campo di dati all'interno di `/etc/passwd` è "x" per ogni username, il vostro sistema usa le shadow password, per cui dovrete modificare `/etc/shadow`.) Per disabilitare la password di root, modificate il secondo campo nel file password in maniera che risulti vuoto. Ora potete fare il reboot ed il log in come root senza una password. Quando fa il boot nel runlevel 1, Debian (almeno dopo Potato) richiede una password, mentre alcune distribuzioni più vecchie no.

E' buona cosa avere un piccolo editor in `/bin` nel caso in cui `/usr` non fosse accessibile (vedere 'Editor di salvataggio' a pagina 212).

Considerate anche l'installazione del pacchetto `sash`. Se il sistema diventa non più avviabile, eseguite:

```
boot: Linux init=/bin/sash
```

`sash` funziona come sostituto interattivo di `sh` persino quando `/bin/sh` è inutilizzabile. Ha un collegamento statico ed include molte utilità di base al suo interno (digitate "help" al prompt per una lista di riferimento).

8.1.2 "Ho scordato la password di root!" (2)

Fate il boot da qualunque set di dischi di emergenza boot/root. Se, per esempio, */dev/hda3* è la partizione di root originale, i seguenti comandi permetteranno di aprire il file password facilmente come sopra.

```
# mkdir fixit
# mount /dev/hda3 fixit
# cd fixit/etc
# vi shadow
# vi passwd
```

Il vantaggio di questo approccio rispetto al metodo precedente è che non richiede la conoscenza della password di `lilo` (se esiste). Però, bisogna essere in grado di impostare il BIOS, se non lo è già, in modo che il boot del sistema sia da floppy o da CD.

8.1.3 Non riesco a fare il boot

Nessun problema, persino se non vi siete presi la briga di fare un dischetto di boot durante l'installazione. Se `lilo` non funziona, usate come disco di boot il disco di installazione di Debian (il primo) ed eseguite il boot da questo disco. Al prompt del boot, assumendo che la partizione di root del vostro Linux è su */dev/hda12* e che volete il runlevel 3, digitate:

```
boot: rescue root=/dev/hda12 3
```

A questo punto siete entrati in un sistema pressochè funzionante in toto, basato sul kernel del dischetto. (Potrete avere piccole noie, dovute alla mancanza di alcuni moduli).

Vedere anche 'Installare un pacchetto su un sistema non bootabile' a pagina 87 se avete un sistema danneggiato.

Se volete crearvi un vostro boot floppy, leggete il `readme.txt` sul rescue disk.

8.1.4 "Voglio disabilitare X al boot!"

Avere `unstable/sid` è divertente, ma `xdm`, `gdm`, `kdm`, e `wdm` difettosi lanciati durante il boot possono fare molto male.

Primo, guadagnate la shell di root digitando quanto segue al prompt del boot:

```
boot: Linux vga=normal s
```

Dove *Linux* è l'etichetta per l'immagine del kernel che andate a lanciare, "vga=normal" vi assicura che `lilo` si esegue in uno schermo normale VGA ed "s" (o "S") è il parametro dato ad `init` per invocare la modalità singolo utente. Al prompt date la password di root.

Esistono vari modi per disabilitare tutti i demoni che lanciano X:

- eseguite `update-rc.d ?dm stop 99 1 2 3 4 5 6 .`
- inserite "exit 0" all'inizio di tutti i file `/etc/init.d/?dm`.
- rinominate tutti i file `/etc/rc2.d/S99?dm` `/etc/rc2.d/K99?dm`.
- rimuovete tutti i file `/etc/rc2.d/S99?dm`.
- eseguite `:>/etc/X11/default-display-manager`

Qui il numero in `rc2.d` deve corrispondere al runlevel specificato in `/etc/inittab`. `?dm` significa tutti gli `xdm`, `gdm`, `kdm`, e `wdm`.

L'unico, vero modo Debian è il primo della lista. L'ultimo è semplice, ma funziona solo sulla Debian e richiede una nuova impostazione in seguito tramite `dpkg-reconfigure`. Gli altri sono metodi generici per disabilitare i demoni.

Avete sempre la possibilità di lanciare X con il comando `startx` da qualsiasi shell.

8.1.5 Altri trucchi con il boot prompt

Potete fare il boot in un particolare runlevel e configurazione tramite il prompt di `lilo`. Dettagli si trovano nel BootPrompt-HOWTO (`http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html`) (LDP).

Se volete fare il boot nel runlevel 4, date il seguente comando al prompt di `lilo`.

```
boot: Linux 4
```

Se si vuole il boot in modalità singolo utente, conoscendo la password di root, potete usare uno dei seguenti esempi al prompt di `lilo`.

```
boot: Linux S
boot: Linux 1
boot: Linux -s
```

Se si vuole il boot con meno memoria di quanta il sistema ne abbia (mettiamo 48M in un sistema con 64M), basta dare il seguente comando al prompt di `lilo`:

```
boot: Linux mem=48M
```

Evitate di specificare più memoria di quanta ne abbiate, altrimenti il kernel si pianterà inevitabilmente. Se si hanno più di 64M, per esempio 128M, con vecchi kernel o BIOS bisogna specificare il comando `mem=128M` al prompt, od includere una riga simile in `/etc/lilo.conf`, altrimenti non verrà utilizzata memoria sopra i 64MB.

8.1.6 Come impostare i parametri di boot (GRUB)

GRUB è un nuovo boot manager proveniente dal progetto Hurd, molto più flessibile di Lilo, ma con un modo di maneggiare i parametri di boot lievemente differente.

```
grub> find /vmlinuz
grub> root (hd0,0)
grub> kernel /vmlinuz root=/dev/hda1
grub> initrd /initrd
grub> boot
```

Dovete conoscere come Hurd nomina i device:

```
the Hurd/GRUB       Linux              MSDOS/Windows
 (fd0)               /dev/fd0           A:
 (hd0,0)             /dev/hda1          C: (solitamente)
 (hd0,3)             /dev/hda4          F: (solitamente)
 (hd1,3)             /dev/hdb4          ?
```

Vedere `/usr/share/doc/grub/README.Debian` e `/usr/share/doc/grub-doc/html/` per i dettagli.

8.2 Registrazione degli eventi

8.2.1 Registrare gli eventi della shell

L'amministrazione di sistema in ambiente Unix richiede dei compiti molto più elaborati che in un ambiente di un normale PC. E' importante conoscere le modalità di configurazione di base, per poter risolvere i problemi del sistema. Le interfacce grafiche di configurazione basate su X sono carine e convenienti, ma comunque limitate, specialmente in casi di emergenza.

In questo contesto, la registrazione degli eventi della shell è una buona pratica, specialmente come root.

Emacs: Usate M-x `shell` per iniziare la registrazione in un buffer e C-x C-w per scrivere il buffer su un file.

Shell: Usate il comando `screen` con "^A H" come decritto in 'Scambio di console con `screen`' a pagina 127 od il comando `script`.

```
$ script
Script started, file is typescript
 ... fate quello che dovete ...
 Ctrl-D
$ col -bx <typescript >savefile
$ vi savefile
```

Si può usare quanto segue invece di `script`:

```
$ bash -i 2>&1 | tee typescript
```

8.2.2 Registrare gli eventi di X

Se dovete registrare l'immagine di una applicazione di X, incluso `xterm`, usate `gimp` (GUI). Può catturare ogni singola finestra, oppure lo schermo intero. Alternative sono `xwd` (`xbase-clients`), `import` (`imagemagick`), o `scrot` (`scrot`).

8.3 Copia ed archiviazione di una intera sottodirectory

8.3.1 Comandi base per copiare una intera sottodirectory

Se avete necessità di riarrangiare la struttura dei file, muovete il contenuto, compresi i collegamenti con:

```
Metodo standard:
# cp -a /partenza/directory /destinazione/directory # richiede GNU cp
# (cd /partenza/directory && tar cf - . ) | \
        (cd /destinazione/directory && tar xvfp - )
Se è coinvolto un collegamento, è necessario un metodo più "pedante":
# cd /percorso/alla/vecchia/directory
# find . -depth -print0 | afio -p -xv -0a /mount/point/della/nuova/directory
Da remoto:
# (cd /origine/directory && tar cf - . ) | \
        ssh user@host.dom (cd /destinazione/directory && tar xvfp - )
Se non ci sono collegamenti:
# scp -pr user1@host1.dom:/origine/directory \
          user2@host2.dom:/destinazione/directory
```

Qui `scp` <==> `rcp` e `ssh` <==> `rsh`.

Le seguenti informazioni comparative su come copiare una intera sottodirectory sono state presentate da Manoj Srivastava <srivasta@debian.org> nella lista debian-user@lists.debian.org.

8.3.2 `cp`

Tradizionalmente, `cp` non è mai stato un gran candidato per tali scopi, poichè non dereferenzia i collegamenti simbolici, nè preserva i collegamenti. Un'altro fattore da considerare è lo spezzettamento dei files (files con buchi).

GNU `cp` ha superato questi limiti; comunque su un sistema non GNU, `cp` potrebbe avere ancora problemi. In più, usando `cp` non si possono creare archivi che siano piccoli e facilmente trasportabili.

```
% cp -a . newdir
```

8.3.3 `tar`

Tar è andato oltre alcuni dei problemi che aveva `cp` con i collegamentos simbolici. Comunque sia, sebbene `cpio` sia in grado di trattare file speciali,il tradizionale `tar` no.

Il modo in cui `tar` tratta collegamenti multipli ad un file è di porre solo una copia del collegamento sul nastro, ma il nome collegato a quella copia è il *solo* utilizzabile per recuperare il file; `cpio` invece, mette una copia per ciascun collegamento, ma per recuperare il file potete usare uno qualsiasi dei nomi.

Nella transizione da Potato a Woody, il comando `tar` ha cambiato l'opzione per i file `.bz2`, per cui usate `--bzip2` negli script al posto della forma breve `-I` (Potato) o `-j` (Woody).

8.3.4 **`pax`**

E' l'utilità POSIX Portable Archive Interchange (IEEE Std 1003.2-1992, pagine 380–388 (sezione 4.48) e pagine 936–940 (sezione E.4.48)), nuova di zecca. `pax` legge, scrive ed elenca i membri di un file archivio, e ne copia le directory secondo le gerarchie. Le operazioni di `pax` sono indipendenti dal formato dell'archivio e supportano una grossa varietà di formati diversi.

Le implementazioni di `pax` sono ancora molto nuove.

```
# apt-get install pax
$ pax -rw -p e . newdir
 oppure
$ find . -depth  | pax -rw -p e  newdir
```

8.3.5 **`cpio`**

`cpio` copia i file dentro o fuori un archivio `cpio` o `tar`. L'archivio può essere un'altro file sul disco rigido, un nastro magnetico, oppure una pipe.

```
$ find . -depth -print0 | cpio --null --sparse -pvd new-dir
```

8.3.6 **`afio`**

`afio` è un metodo migliore per trattare con archivi in formato `cpio`. In genere è più veloce di `cpio`, ha maggiori opzioni per i nastri e tratta in maniera più gentile con le corruzioni dei dati in entrata. Supporta gli archivi multi-volume durante le operazioni in modalità interattiva. `afio` può creare archivi molto più sicuri di quelli compressi con `tar` o `cpio`. `afio` è anche ottimo come "motore per l'archiviazione" negli script di backup.

```
$ find . -depth -print0 | afio -px -0a new-dir
```

Tutti i miei backup su nastro usano `afio`.

8.4 Backup differenziali e sincronizzazione dei dati

I backup differenziali e la sincronizzazione dei dati possono essere implementati con vari metodi:

- `rcs`: backup con storico, solo testo
- `rdiff-backup`: backup con storico. I collegamenti simbolici sono OK.

- `pdumpfs`: backup con storico all'interno di un filesystem. Collegamenti simbolici OK.
- `rsync`: sincronizzazione ad 1 via
- `unison`: sincronizzazione a 2 vie
- `cvs`: sincronizzazione multivia con server backup e storico, solo testo, maturo. Vedere 'Concurrent Version System (CVS)' a pagina 217.
- `arch`: sincronizzazione multivia con server backup e storico, non esiste qualcosa come una "working directory".
- `subversion`: sincronizzazione multivia con server backup e storico, Apache.

La combinazione di uno dei metodi menzionati con uno dei metodi di archiviazione descritti in 'Copia ed archiviazione di una intera sottodirectory' a pagina 111 associati ad un job automatico regolare descritto in 'Programmare gli eventi (`cron`, `at`)' a pagina 126 creeranno un ottimo sistema per il backup.

Mostrerò tre utilità facili da usare.

8.4.1 Backup differenziale con rdiff

`rdiff-backup` fornisce dei backup semplici e validi con lo storico differenziale per file di qualsiasi tipo, compresi i collegamenti simbolici. Per salvare gran parte di `~/` su `/mnt/backup`:

```
$ rdiff-backup --include ~/tmp/keep --exclude ~/tmp  ~/ /mnt/backup
```

Per recuperare i dati vecchi tre giorni da questo archivio in `~/old`:

```
$ rdiff-backup -r 3D /mnt/backup ~/old
```

Vedere `rdiff-backup(1)`.

8.4.2 Backup giornalieri con **`pdumpfs`**

`pdumpfs` è un sistema semplice per il backup giornaliero, simile a `dumpfs` di Plan), che conserva ogni snapshot giornaliero. Potete accedere in ogni monento agli snapshot passati per recuperare un determinato file di un tale giorno. Fate il backup della vostra directory home con `pdumpfs` e `cron`!

`pdumpfs` produce uno snapshot `YYYY/MM/DD` nella directory di destinazione. Tutti i file sorgenti vengono copiati nella directory di snapshot per la prima volta. La seconda volta e le succesive, `pdumpfs` copia solo i file aggiornati o nuovi, immagazzinando quelli immodificati come collegamenti ai file dello snapshot del giorno precedente per salvare spazio su disco.

```
$ pdumpfs src-dir dest-dir [dest-basename]
```

Vedere `pdumpfs(8)`.

8.4.3 Backup differenziali regolari con RCS

`Changetrack` registra regolarmente i cambiamenti ai file di testo di configurazione negli archivi RCS. Vedere `changetrack(1)`.

```
# apt-get install changetrack
# vi changetrack.conf
```

8.5 Recuperare un sistema bloccato

8.5.1 Uccidere un processo

Lanciate `top` per vedere quale processo si comporta in maniera “sconveniente”. Premete “P” per ordinarli per utilizzo di cpu, “M” per consumo di memoria e “k” per uccidere il processo.In alternativa potete usare i comandi in stile BSD `ps aux | less` od in stile System V `ps -efH | less`. La sintassi del System V mostra l’ID dei processi padri `PPID` che può essere usato per uccidere i figli zombie (defunti).

Usate `kill` per uccidere (o mandare il segnale a) un processo per ID del processo, `killall` per fare lo stesso con il nome del comando a cui il processo appartiene. Segnali più frequentemente usati:

```
 1: HUP,  riavvia il demone
15: TERM, uccisione normale
 9: KILL, uccisione "dura"
```

8.5.2 Alt-SysRq

L’assicurazione contro i malfunzionamenti di sistema è un’opzione di compilazione del kernel, denominata “Magic SysRQ key”. Premendo `ALT-SysRq` su un sistema i386, seguito da uno dei tasti `r 0 k e i s u b`, fa la “magia”.

Un’r’aw riavvia la tastiera quando cose tipo X si bloccano. Cambiare il loglevel a 0 riduce i messagi di errore. sa’k’ (system attention key) uccide tutti i processi sulla console virtuale attuale. t’e’rminate uccide tutti i processi su terminale corrente tranne init. k’I’ll uccide tutti i processi, tranne `init`.

‘S’ync, ‘’u’mount, e re‘b’oot sono utili per uscire da situazioni veramente spinose.

I kernel debian standard non sono compilati al momento con questa opzione attivata. Ricompilate il kernel per farlo. Informazioni dettagliate le trovate in `/usr/share/doc/kernel-doc-`*`version`*`/Documentation/sysrq.txt.gz` o `/usr/src/`*`kernel-version`*`/Documentation/sysrq.txt.gz`.

8.6 Alcuni piccoli comandi da ricordare

8.6.1 Pager

`less` è il pager (browser del contenuto dei file). Premete 'h' per aiuto. Fa molto di più di `more`. `less` può essere potenziato mediante l'esecuzione di `eval $(lesspipe)` o `eval $(lessfile)` nello script di inizializzazione della shell. Per ulteriori notizie guardate in `/usr/share/doc/lessf/LESSOPEN`. L'opzione `-R` permette l'output di caratteri raw ed abilita le sequenze ANSI con escape colorate. Vedere `less(1)`.

`w3m` può essere un pager alternativo utile per alcuni sistemi di codifica (EUC).

8.6.2 Memoria libera

`free` e `top` danno buone informazioni sulle risorse di memoria. Non vi preoccupate sulle dimensioni della memoria "used" nella riga "Mem:", ma leggete quella sotto. (38792 nell'esempio qui sotto).

```
$ free -k # per una macchina con 256MB
             total       used       free     shared    buffers cached
Mem:        257136     230456      26680      45736     116136 75528
-/+ buffers/cache:      38792     218344
Swap:       264996          0     264996
```

L'esatto ammontare della memoria fisica può essere confermato da `grep '^Memory' /var/log/dmesg`, che in questo caso dà "Memory: 256984k/262144k available (1652k kernel code, 412k reserved, 2944k data, 152k init)".

```
Total         = 262144k = 256M (1k=1024, 1M=1024k)
Free to dmesg = 256984k = Total - kernel - reserved - data - init
Free to shell = 257136k = Total - kernel - reserved - data
```

Circa 5MB non sono utilizzabili dal sistema, perchè a disposizione del kernel.

8.6.3 Impostare l'ora (BIOS)

```
# date MMDDhhmmCCYY
# hwclock --utc --systohc
# hwclock --show
```

Questo imposterà l'orologio del sistema e dell'hardware a MM/GG hh:mm, CCYY. Le ore sono allineate al fuso locale, ma l'orologio dell'hardware utilizza lo UTC.

Se l'ora dell'hardware (BIOS) è impostata su GMT, modificate le impostazioni a `UTC=yes` in `/etc/default/rcS`.

8.6.4 Impostare l'ora (NTP)

Riferimento: Managing Accurate Date and Time HOWTO (`http://www.tldp.org/HOWTO/TimePrecision-HOWTO/index.html`).

Impostare l'ora tramite una connessione Internet permanente

Impostazione automatica dell'ora esatta tramite un server remoto:

```
# ntpdate server
```

Se avete una connessione permanente, è bene averlo in `/etc/cron.daily`.

Impostare l'ora tramite una connessione Internet sporadica

Usate il pacchetto `chrony`.

8.6.5 Come controllare le funzionalità della console, come lo screensaver

Per disabilitare lo screensaver, usate i comandi seguenti:

In console:

```
# setterm -powersave off
```

Lanciate la console kon2 (kanji) con:

```
# kon -SaveTime 0
```

In X:

```
# xset s off
 or
# xset -dpms
 or
# xscreensaver-command -prefs
```

Leggete le pagine di manuale corrispondenti per controllare le altre funzionalità della console. Vedere anche `stty(1)` per modificare e stampare le impostazioni della riga di comando.

8.6.6 Ricerca dei database amministrativi

Glibc offre `getent(1)` per la ricerca di voci inserite nei database amministrativi, cioè, passwd, group, hosts, services, protocols, o networks.

```
getent database [chiave...]
```

8.6.7 Disabilitare il suono (beep)

Potete sempre staccare gli altoparlanti del PC ;-) Per la shell Bash:

```
echo "set bell-style none">> ~/.inputrc
```

8.6.8 Messaggi di errore sulla console

Per placare i messaggi di errore sullo schermo, il primo posto da vedere è `/etc/init.d/klogd`. Impostate `KLOGD="-c 3"` in questo script ed eseguite `/etc/init.d/klogd restart`. metodo alternativo è eseguire `dmesg -n3`.

I livelli di errore significano:
- 0: KERN_EMERG, il sistema è inutilizzabile
- 1: KERN_ALERT, un'azione va presa immediatamente
- 2: KERN_CRIT, condizioni critiche
- 3: KERN_ERR, condizioni di errore
- 4: KERN_WARNING, condizioni di allarme
- 5: KERN_NOTICE, condizione normale ma significativa
- 6: KERN_INFO, informazioni
- 7: KERN_DEBUG, massaggi di debug

Se c'è un particolare ed inutile messaggio di errore che vi perseguita, considerate l'ipotesi di una banalissima patch per il kernel, tipo `shutup-abit-bp6` (disponibile presso in the examples subdirectory (`http://www.debian.org/doc/manuals/debian-reference/examples/`)).

Un altro posto da guardare può essere `/etc/syslog.conf`; controllate per vedere se qualche messaggio viene inviato alla console.

8.6.9 Impostare i caratteri corretti per la console

Agli schermi di console nei sistemi simil-Unix si accede generalmente tramite librerie (n)curses. Queste offrono all'utente un metodo indipendente dal terminale di aggiornamento dei caratteri, con una ottimizzazione ragionevole. Vedere `ncurses(3X)` e `terminfo(5)`.

Su un sistema Debian molte sono le voci predefinite:

```
$ toe | less                     # tutte le voci
$ toe /etc/terminfo/ | less      # voci riconfigurabili dall'utente
```

Esportate la vostra selezione come variabile d'ambiente `TERM`.

Se la voce terminfo per `xterm` non funziona con un `xterm` non-Debian, cambiate il tipo di terminale da "xterm" ad una delle versioni con minori opzioni tipo "xterm-r6", quando accedete da remoto ad un sistema Debian. Vedere `/usr/share/doc/libncurses5/FAQ` per ulteriori informazioni. "dumb" è il minimo comun denominatore per terminfo.

8.6.10 Riportare la console allo stato normale

Quando lo schermo impazzisce dopo `$ cat qualsiasi-file-binario` (potreste non essere in grado di vedere il comando dato mentre lo state digitando):

```
$ reset
```

8.6.11 Convertire file di testo da DOS a Unix

Converte file un di testo DOS (fine riga = `^M^J`) in un file Unix di testo (end-of-line = `^J`).

```
# apt-get install sysutils
$ dos2unix dosfile
```

8.6.12 Convertire un file di testo con `recode`

Quanto segue converte i file di testo fra gli stili di fine riga DOS, Mac, e Unix:

```
$ recode /cl../cr <dos.txt >mac.txt
$ recode /cr.. <mac.txt >unix.txt
$ recode ../cl <unix.txt >dos.txt
```

`recode`, libero, converte i file tra i vari set di caratteri e surface con:

```
$ recode charset1/surface1..charset2/surface2 \
  <input.txt >output.txt
```

I set caratteri comunemente usati sono (vedere anche 'Introduzione ai locale' a pagina 171) [1] :

- `us` — ASCII (7 bit)
- `l1` — ISO Latin-1 (ISO-8859-1, Europa occidentale, 8 bit)

---

[1] `recode` permette degli alias più convenienti di `iconv`.

- `EUCJP` — EUC-JP per Giapponese (Unix)
- `SJIS` — Shift-JIS per Giapponese (Microsoft)
- `ISO2022JP` — Mail encoding per Giapponese (7 bit)
- `u2` — UCS-2 (Universal Character Set, 2 byte)
- `u8` — UTF-8 (Universal Transformation Format, 8 bit)

Surface comuni sono [2] :

- `/cr` — A capo come fine riga (Mac text)
- `/cl` — A capo con nuova riga come fine riga (DOS text)
- `/` — Nuova riga come fine riga (Unix text)
- `/d1` — Bytewise decimal dump leggibile
- `/x1` — Bytewise hexidecimal dump leggibile
- `/64` — Testo codificato base64
- `/QP` — Testo codificato Quoted-Printable

Per maggiori notizie, leggere la descrizione pertinente in `info recode`.

Esistono anche degli strumenti di conversione più specializzati:

- conversione del set di caratteri:
  - `iconv` — locale encoding conversions
  - `konwert` — fancy encoding conversions
- conversione dei file binari:
  - `uuencode` e `uudecode` — per Unix.
  - `mimencode` — per la posta.

8.6.13 Sostituzione delle espressioni regolari

Sostituisce tutte le istanze *FROM_REGEX* con *TO_TEXT* in tutti i file *FILES* . . . :

```
$ perl -i -p -e 's/FROM_REGEX/TO_TEXT/g;' FILES ...
```

`-i` sta per “in-place editing”, `-p` è per “implicit loop over *FILES* . . . ”. se la sostituzione è complessa, potete recuperare più facilmente gli errori usando il parametro `-i.bak` al posto di `-i`; così si mantiene ogni file originale, aggiungendo `.bak` come estensione.

8.6.14 Modificare un file “in situ” tramite uno script

Lo script seguente rimuove le righe 5–10 e le righe 16–20 “in situ”.

```
#!/bin/bash
ed $1 <<EOF
```

---

[2] Fine riga:

- A capo significa ASCII 13, ASCII 0xD, ^M, or \r .
- Nuova riga significa ASCII 10, ASCII 0xA, ^J, or \n .

```
16,20d
5,10d
w
q
EOF
```

Qui i comandi di `ed` sono gli stessi del modo comando di `vi`. La modifica dei file in questo modo rende facile la creazione di script.

8.6.15 Estrarre le differenze e fondere gli aggiornamenti da un file sorgente

Seguendo una delle seguenti procedure, si estrarranno le differenze dal file sorgente creando dei file diff unificati *file.patch0* o *file.patch1* a seconda della localizzazione del file:

```
$ diff -u file.vecchio file.nuovo1 > file.patch0
$ diff -u vecchio/file nuovo1/file > file.patch1
```

Il file diff (chiamato anche file patch) è usato per inviare gli aggiornamenti di un programma. Chi lo riceve applicherà questo aggiornamento ad un altro *file* con:

```
$ patch -p0 file < file.patch0
$ patch -p1 file < file.patch1
```

Se avete tre versioni dello stesso codice sorgente, potete farle confluire più efficacemente tutte insieme usando `diff3`:

```
$ diff3 -m file.mio file.old file.tuo > file
```

8.6.16 Convertire grossi file in file più piccoli

```
$ split -b 650m file   # divide il file in pezzetti di 650 MB
$ cat x* >largefile    # riunisce i file in un unico file
```

8.6.17 Estrarre dati da un file tabella

Consideramo un file di testo chiamato `DPL` in cui tutti i nomi dei project leader Debian precedenti e la data di inizio della carica siano elencati in un formato con separazioni date da spazi.

```
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
```

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

```
$ awk '{ print $3 }' <DPL                   # mese di inzio
August
April
January
January
April
April
March
$ awk '($1=="Ian") { print }' <DPL          # DPL di nome Ian
Ian     Murdock   August  1993
Ian     Jackson   January 1998
$ awk '($2=="Perens") { print $3,$4 }' <DPL # Quando Perens ha iniziato
April 1996
```

Anche le shell come la Bash possono essere usate per questi file:

```
$ while read first last month year; do
    echo $month
  done <DPL
... lo stesso output del primo esempio con Awk
```

Qui ilcomando interno `read` usa i caratteri in $IFS (internal field separators) per dividere le righe in parole.

Se cambiate IFS in ":", potete processare elegantemente `/etc/passwd` con la shell:

```
$ oldIFS="$IFS"   # salva i vecchi valori
$ IFS=":"
$ while read user password uid gid rest_of_line; do
    if [ "$user" = "osamu" ]; then
      echo "$user's ID is $uid"
    fi
  done < /etc/passwd
osamu's ID is 1001
$ IFS="$oldIFS"   # ripristina i vecchi valori
```

(Se si usa Awk per la stessa cosa, usate `FS=“:”` per impostare il separatore di campo)

IFS viene anche usato dalla shell per dividere i risultati della espansione dei parametri, sostituzione dei comandi ed espansione aritmetica. Questi non sono possibili all’interno di parole tra virgolette, semplici o doppie. il valore predefinito di IFS è <space>, <tab>, e <newline> combinati.

Fate attenzione nell’uso di questi trucchi con IFS. Possono accadere strane cose quando la shell inetrpreta alcune parti degli script come proprio **input**.

```
$ IFS=":,"                        # usa ":" e "," come IFS
$ echo IFS=$IFS,   IFS="$IFS"     # echo è un Bash built-in
IFS=  , IFS=:,
$ date -R                         # solo l'output di un comando
Sat, 23 Aug 2003 08:30:15 +0200
$ echo $(date -R)                 # sub shell --> input alla shell principale
Sat  23 Aug 2003 08 30 36 +0200
$ unset IFS                       # riporta IFS al predefinito
$ echo $(date -R)
Sat, 23 Aug 2003 08:30:50 +0200
```

8.6.18 Parti di script per il piping

Gli script seguenti fanno cose utili come parti di una pipe.

```
find /usr | egrep -v "/usr/var|/usr/tmp|/usr/local"
                     # trova tutti i file /usr escludendone alcuni
xargs -n 1 command   # lancia i comandi per tutti gli oggetti da stdin
xargs -n 1 echo |     # divide oggetti separati da spazi in righe
xargs echo      |    # unisce tutte le linee in un'unica riga
grep -e pattern|     # estrae le righe contenenti pattern
cut -d: -f3 -|
        # estrae il terzo campo separato da : (file passwd, ecc.)
awk '{ print $3 }' | # estrae il terzo campo separato da spazi bianchi
awk -F'\t' '{ print $3 }' |
       # estrae il terzo campo separato da tabulazione
col -bx |            # rimuove il backspace ed espande le tabs in spazi
expand -|            # espande le tabs
sort -u|             # trova e rimuove i doppioni

tr '\n' ' '|         # concatena le righe in una riga unica
tr '\r' ''|          # rimuove il CR (carriage return, a capo)
tr 'A-Z' 'a-z'|      # converte il maiuscolo in minuscolo
sed 's/^/# /'|       # commenta 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
```

8.6.19 Pezzi di script ricorsivi

Il metodo seguente di agire ricorsivamente su ciascun file cercando corrispondenze `*.ext` assicura la gestione corretta dei nomi bizzarri dei file, tipo quelli con gli spazi, ed esegue il processo equivalente:

- Shell loop (E' un input multiriga con `PS2="> "`) [3]:

```
for x in *.ext; do
  if test -f "$x"; then
    command "$x"
  fi
done
```

- combibnazione di `find` e `xargs`:

```
find . -type f -maxdepth 1 -name '*.ext' -print0 | \
 xargs -0 -n 1 command
```

- un comando `find` con l'opzione `-exec`:

```
find . -type f -maxdepth 1 -name '*.ext' \
 -exec command '{}' \;
```

- un breve script di shell con `find` e l'opzione `-exec`:

```
find . -type f -maxdepth 1 -name '*.ext' \
 -exec sh -c "command '{}' && echo 'successful'" \;
```

8.6.20 Brevi follie in Perl script

Sebbene tutti gli script Awk possano essere riscritti con Perl usando `a2p(1)`, gli ad una riga di Awk sono convertiti meglio a script in perl ad una riga manualmente. Per esempio

```
awk '($2=="1957") { print $3 }' |
```

---

[3] Se digitate tutto su una riga, dove aggiungere dei punti e virgola, "`;`", per marcare la fine dei comandi della shell.

può essere scritto in uno dei seguenti modi:

```
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"' |
```

Siccome tutti gli spazi tra gli argomenti di `perl` nelle righe sopra possono essere rimossi traendo vantaggio dalla conversione automatica tra numeri e stringhe in Perl:

```
perl -lane 'print$F[2]if$F[1]eq+1957' |
```

Vedere `perlrun(1)` per le opzioni da riga di comando. Per altri pazzi script in Perl, `http://perlgolf.sourceforge.net` può essere interessante.

8.6.21 Catturare il testo od un archivio delle ML da una pagina web

Il seguente comando legge la pagina web e la copia in un file di testo. Molto utile quando si copiano delle configurazioni dalla rete.

```
$ lynx -dump http://www.remote-site.com/help-info.html >textfile
```

Anche `links` e `w3m` possono essere usati, con piccole differenze nel rendering.

Se si tratta di un archivio di una lista di messaggi, usate `munpack` per ottenere i contenuti mime dal testo.

8.6.22 Stampare bene una pagina Web

Quanto segue stampa una pagina Web in un file o stampante PostScript.

```
$ apt-get install html2ps
$ html2ps URL | lpr
```

Vedere '`lpr/lpd`' a pagina 41. Controllate anche i pacchetti `a2ps` e `mpage` per la creazione di file PostScript.

8.6.23 Stampare bene una pagina di manuale

Quanto segue stampa una pagina di manuale in un file o stampante PostScript.

```
$ man -Tps some-man-page | lpr
$ man -Tps some-man-page | mpage -2 | lpr
```

8.6.24 Unire due file Postscript o PDF

Si possono unire due file Postscript o PDF.

```
$ 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
```

8.6.25 Cronometrare un comando

Mostra il tempo utilizzato da un processo.

```
# time qualsiasi-comando >/dev/null
real    0m0.035s       # tempo sull'orologio a muro (tempo realmente passato)
user    0m0.000s       # tempo in modalità utente
sys     0m0.020s       # tempo in modalità kernel
```

8.6.26 Il comando `nice`

Usate `nice` (dal pacchetto GNU `shellutils`) per impostare il valore nice di un comando quando viene lanciato. `renice` (`bsdutils`) o `top` possono dare il renice ad un processo. Un valore di nice di 19 rappresenta il processo più lento (priorità bassa); valori negativi sono "not-nice", con -20 che rappresenta un processo molto veloce (alta priorità). Solo il superuser può impostare valori di nice negativi.

```
# nice  -19 top                                        # molto nice
# nice --20 cdrecord -v -eject speed=2 dev=0,0 disk.img # molto veloce
```

Talvolta un valore estremo di nice fa più danni che bene al sistema. Usate questo comando con cautela.

8.6.27 Programmare gli eventi (`cron`, `at`)

Usate `cron` e `at` per programmare i task in Linux. Vedere `at(1)`,`crontab(5)`,`crontab(8)`.

Eseguite il comando `crontab -e` per creare o modificare un file crontab per impostare degli eventi regolarmente programmati. Ecco un esempio di file crontab:

```
# usa /bin/sh per eseguire i comandi, non importa cosa dice /etc/passwd
SHELL=/bin/sh
# invia ogni output a 'paul', non importa a chi appartiene crontab
```

```
MAILTO=paul
# Min Hour DayOfMonth Month DayOfWeek comando
# esecuzione alle 00:05, ogni giorno
5  0  *  * *   $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# esecuzione alle 14:15 il primo giorno del mese -- output inviato a paul
15 14 1  * *   $HOME/bin/monthly
# esecuzione alle 22:00 i giorni feriali (Lun-Ven) (1-5), annoia Joe. % per una r
0 22 *   * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%.%%
23 */2 1 2 *   echo "run 23 minutes after 0am, 2am, 4am ..., on Feb 1"
5  4 *   * sun echo "run at 04:05 every sunday"
# esecuzione alle 03:40 il primo lunedi di ogni mese
40 3 1-7 * *   [ "$(date +%a)" == "Mon" ] && command -args
```

Usate il comando `at` per programmare gli eventi che accadono una sola volta:

```
$ echo 'command -args'| at 3:40 monday
```

8.6.28 Scambio di console con `screen`

Il programma `screen` vi permette di far girare terminali virtuali multipli ciascuno con la propria shell interattiva, su un singolo terminale fisico, o su una finestra di emulazione di terminale. Anche se usate le console virtuali di Linux o finestre xterm multiple, vale comunque la pena di esplorare `screen` per la sua ricchezza di opzioni, che includono

- storico per lo scrollback,
- copia-incolla
- registrazione dell'output,
- digraph entry, e
- l'abilità di **staccare** un'intera sessione di `screen` dal vostro terminale e di riattacarla in seguito.

Scenario: accesso remoto

Se accedete frequentemente ad una macchina Linux da terminale remoto o tramite un programma di terminale VT100,, `screen` renderà la vostra vita molto più semplice con la funzione **detach**.

1 Supponiamo che siate connessi mediante una connessione dialup e che state conducendo una sessione di `screen` con editors e altri programmi aperti su molte finestre.
2 Improvvisamente dovete abbandonare il terminale, ma non volete perdere tutto il lavoro alla deconnessione.
3 Basta digitare `^A d` per **staccare** la sessione, poi deconnettersi. (Oppure, più veloce ancora, digitate `^A DD` per dire a `screen` di staccare la sessione e deconnettersi da solo.)
4 Quando vi riconnettete di nuovo, date il comando `screen -r`, e `screen` magicamente **riattaccherà** tutte le finestre che avevate aperto.

Comandi tipici di `screen`

Una volta lanciato `screen`, tutto l'input della tastiera viene indirizzato alla finestra corrente, tranne il tasto di comando, di default `^A`. Tutti i comandi per `screen` vengono dati premendo `^A` più un tasto singolo [più alcuni parametri]. Comandi utili:

```
^A ?     mostra una schermata di aiuto (mostra i comandi da tastiera)
^A c     crea una nuova finestra e passa ad essa
^A n     va alla finestra successiva
^A p     va alla finestra precedente
^A 0     va alla finestra numero 0
^A w     mostra una lsta di finestre
^A a     manda un Ctrl-A alla finestra corrente come input da tastiera
^A h     scrive una copia della finestra corrente su file
^A H     inizia/termina la registrazione della finestra corrente su file
^A ^X    blocca il terminale (protetto da password)
^A d     stacca la sessione dal terminale
^A DD    stacca la sessione ed esce
```

Questo è solo un piccolo assaggio dei comandi e caratteristiche di `screen`. Se c'è qualcosa che volete che `screen` faccia, è probabile che lo sappia fare! Vedere `screen(1)` per i dettagli.

Backspace e/o Ctrl-H durante una sessione di `screen`

Se notate che il backspace e/o Ctrl-H non funzionano appropriatamente mentre state facendo girare `screen`, aprite `/etc/screenrc`, trovate la riga

```
bindkey -k kb stuff "\177"
```

e commentatela (cioè, aggiungete "#" come primo carattere).

Programma equivalente a `screen` per X

Vedetevi `xmove`. Vedere `xmove(1)`.

8.6.29 Testare la rete: le basi

Installate i pacchetti `netkit-ping`, `traceroute`, `dnsutils`, `ipchains` (per Kernel 2.2), `iptables` (per Kernel 2.4), e `net-tools` e:

```
$ ping yahoo.com              # prova la connessione internet
$ traceroute yahoo.com        # traccia i pacchetti IP
$ ifconfig                    # prova la configurazione dell'host
```

```
$ route -n                   # prova la configurazione di routing
$ dig [@dns-server.com] host.dom [{a|mx|any}] |less
      # controlla i record DNS di host.dom tramite dns-server.com
      # per {mx|any} record
$ ipchains  -L -n |less      # controlla il packet filter (kernel 2.2)
$ iptables -L -n |less       # controlla il packet filter (kernel 2.4)
$ netstat -a                 # scopre tutte le porte aperte
$ netstat -l --inet          # trova le porte in ascolto
$ netstat -ln --tcp          # trova le porte tcp in ascolto (numerico)
```

8.6.30 Eliminare la posta dallo spool locale

Per eliminare la posta:

```
# exim -q    # elimina la posta in attesa
# exim -qf   # elimina tutta la posta
# exim -qff  # elimina persino la posta "congelata"
```

`-qff` può essere migliore come opzione per lo script `/etc/ppp/ip-up.d/exim`. Per Sarge, sostituite `exim` con `exim4`.

8.6.31 Rimuovere la posta congelata dallo spool locale

Per rimuovere la posta bloccata dallo spool locale assieme al messaggio di errore nell'invio:

```
# exim -Mg `mailq | grep frozen | awk '{ print $3 }'`
```

Per Sarge, sostituite `exim` con `exim4`.

8.6.32 Riinviare il contenuto delle caselle di posta

Se la vostra directory home è piena e `procmail` non è riuscito ad inviare la posta, dovete inviarla a mano alle varie caselle di posta della directory home, da `/var/mail/`*`username`*. Dopo aver fatto spazio nella vostra directory home, lanciate:

```
# /etc/init.d/exim stop
# formail -s procmail </var/mail/username
# /etc/init.d/exim start
```

Per Sarge, sostituite `exim` con `exim4`.

8.6.33 Ripulire il contenuto di un file

Per ripulire un file dal suo contenuto, come ad esempio un file di log, non usate `rm` per cancellarlo, e poi crearne un nuovo vuoto, poichè ci potrebbero essere ancora degli accessi al file nell'intervallo fra i comandi. Quanto segue è il modo sicura di ripulire un file dal suo contenuto.

```
$ :>file-da-essere-ripulito
```

8.6.34 File fantoccio

I comandi seguenti creano dei file fantoccio o vuoti:

```
$ dd if=/dev/zero    of=nomefile bs=1k count=5 # 5KB di zeri
$ dd if=/dev/urandom of=nomefile bs=1m count=7 # 7MB di contenuto casuale
$ touch nomefile # crea un file di 0B (se il file esiste, aggiorna mtime)
```

Per esempio, i seguenti comandi eseguiti dalla shell del boot floppy Debian cancelleranno tutto il contenuto del disco rigido `/dev/hda`.

```
# dd if=/dev/urandom of=/dev/hda ; dd if=/dev/zero of=/dev/hda
```

8.6.35 `chroot`

Il programma `chroot`, `chroot(8)`, ci permette di far girare istanze diverse dell'ambiente GNU/Linux su un sistema singolo e simultaneamente senza dover fare il reboot.

Si può far girare un programma che utilizza molta memoria come `apt-get` o `dselect` su una macchina host mentre si monta via NFS una macchina satellite sull'host in lettura/scrittura e con il chroot point che agisce come punto di montaggio della macchina satellite.

Far girare versioni diverse di Debian con `chroot`

Si può creare facilmente un ambiente chroot Debian tramite il comando `debootstrap` in Woody. Per esempio, per creare Sid sotto chroot in */sid-root* mentre si ha a disposizione un accesso veloce ad Internet:

```
main # cd / ; mkdir /sid-root
main # debootstrap sid /sid-root http://ftp.debian.org/debian/
... guardatelo mentre si scarica l'intero sistema
main # echo "proc-sid /sid-root/proc proc none 0 0" >> /etc/fstab
main # mount proc-sid /sid-root/proc -t proc
```

```
main # cp /etc/hosts /sid-root/etc/hosts
main # chroot /sid-root /bin/bash
chroot # apt-setup # set-up /etc/apt/sources.list
chroot # vi /etc/apt/sources.list # punta su unstable
chroot # dselect  # potete usare aptitude, installate mc e vim :-)
```

A questo punto dovreste avere un sistema Debian completo e funzionante, dove giocare senza paura di danneggiare la vostra installazione Debian principale.

Il trucco di `debootstrap` può anche essere usato per installare la Debian su un sistema senza usare il disco di installazione Debian ma uno di una qualsiasi altra distribuzione GNU/Linux. Vedere `http://www.debian.org/releases/stable/i386/ch-preparing#s-linux-upgrade`.

Impostare il login per `chroot`

Digitare `chroot /sid-root /bin/bash` è semplice, ma tiene in giro ogni sorta di variabile d'ambiente che magari non vorreste, inoltre ha altri problemi. Un approccio sicuramente migliore è di lanciare un altro processo di login su un terminale virtuale separato, da dove fare direttamente il login in chroot.

Dato che su un sistema Debian standard da `tty1` a `tty6` girano le consoles Linux e su `tty7` ci gira il X Window System, impostiamo per esempio `tty8` per una console "chrooted". Dopo aver creato il sistema chroot come descritto in 'Far girare versioni diverse di Debian con `chroot`' a fronte, digitate dalla shell di root del sistema principale:

```
main # echo "8:23:respawn:/usr/sbin/chroot /sid-root "\
       "/sbin/getty 38400 tty8"  >> /etc/inittab
main # init q    # ricarica init
```

Impostare X per `chroot`

Volete far girare gli ultimissimi X e GNOME con sicurezza nel vostro chroot? Ciò è interamente possibile! Il seguente esempio farà girare GDM sul terminale virtuale `vt9`.

Per prima cosa installate un sistema chroot usando il metodo descritto in 'Far girare versioni diverse di Debian con `chroot`' nella pagina precedente. Da root del sistema principale, copiate i file chiave di configurazione nel sistema chroot.

```
main # cp /etc/X11/XF86Config-4 /sid-root/etc/X11/XF86Config-4
main # chroot /sid-root # od usate la console di chroot
chroot # apt-get install gdm gnome x-window-system
chroot # vi /etc/gdm/gdm.conf # date s/vt7/vt9/ nella sezione [servers]
chroot # /etc/init.d/gdm start
```

Qui `/etc/gdm/gdm.conf` è stato modificato per lanciare la console virtuale da `vt7` a `vt9`.

A questo punto potete andare facilmente avanti e indietro tra gli ambienti X completi del vostro sistema chroot e principale semplicemente saltando fra i terminali virtuali; p.es. usando Ctrl-Alt-F7 e Ctrl-Alt-F9. Buon divertimento!

[FIXME] Add a comment and link to the init script of the chrooted gdm.

Far girare altre distribuzioni con `chroot`

Si può creare facilmente un ambiente chroot con un'altra distribuzione. Installate un sistema in una o più partizioni separate mediante l'installer dell'altra distribuzione. Se la sua partizione di root è in `/dev/hda9`.

```
main # cd / ; mkdir /altra-dist
main # mount -t ext3 /dev/hda9 /altra-dist
main # chroot /altra-dist /bin/bash
```

Il resto è simile a 'Far girare versioni diverse di Debian con `chroot`' a pagina 130, 'Impostare il login per `chroot`' nella pagina precedente, e 'Impostare X per `chroot`' nella pagina precedente.

Compilare i pacchetti con `chroot`

Esiste un pacchetto chroot più specializzato, `pbuilder`, che costruisce un sistema chroot e compila un pacchetto al suo interno. E' un sistema ideale per controllare se le dipendenze di compilazione di un pacchetto sono corrette e per essere certi che dipendenze non necessarie o sbagliate non esistano nel pacchetto risultante.

8.6.36 Come controllare i collegamenti

potete controllare se due file sono lo stesso file con due collegamenti tramite:

```
$ ls -li file1 file2
```

8.6.37 `mount` il file immagine del disco rigido

Se *`file.img`* contiene l'immagine del contenuto di un disco rigido ed il disco originale aveva una configurazione che dà *xxxx* = (bytes/settore) * (settori/cilindro), allora quanto segue lo monterà in `/mnt`:

```
# mount -o loop,offset=xxxx file.img /mnt
```

Notate cha molti dischi rigidi hanno 512 bytes/settore.

8.6.38 Samba

Le basi per ottenere un file da Windows:

```
# mount -t smbfs -o username=mionome,uid=mio_uid,gid=mio_gid \
        //server/share /mnt/smb  # monta i file Windows su Linux
# smbmount //server/share /mnt/smb \
        -o "username=mionome,uid=mio_uid,gid=mio_gid"
# smbclient -L 192.168.1.2 # elenca le condivisioni su un computer
```

I PC collegati tramite Samba possono essere controllati sotto Linux tramite:

```
# smbclient -N -L ip_address_del_vostro_PC | less
# nmblookup -T "*"
```

8.6.39 Utilità per filesystem estranei

Molti filesystem estranei hanno il supporto per il kernel Linux, per cui possono essere raggiunti semplicemente montando i dispositivi che li contengono. Per alcuni filesystem, esistono anche degli strumenti specializzati per accedervi senza montarne i dispositivi. Ciò si ottiene con programmi che girano nello user space, per cui il supporto specifico nel kernel per i filesystem non è necessario.

- `mtools`: per filesystem MSDOS filesystem (MS-DOS, Windows)
- `cpmtools`: per CP/M
- `hfsutils`: per HFS (Macintosh nativo)
- `hfsplus`: per HFS+ (moderno Macintosh)

Per creare e controllare filesystem MS-DOS FAT, `dosfstools` è utile.

8.7 Errori tipici da notare

Ecco alcuni esempi di azioni pericolose. Il loro impatto negativo negativo verrà potenziato se state usando un account privilegiato: `root`.

8.7.1 `rm -rf .*`

In “`rm -rf .*`”, “`.*`” si espande ad includere “`.`” e “`..`”, e se vi capita di avere i privilegi in scrittura sulla directory superiore, finirete col rimuovere anche tutte le directory **vicine** alla directory corrente.

- “`rm -rf .`” : rimuove tutto nella directory corrente e la directory stessa.
- “`rm -rf *`” : rimuove tutti i file non-punto e le directory non-punto nella directory corrente
- “`rm -rf .[^.]*`”: rimuove tutti i file punto e directory punto nella directory corrente.
- “`rm -rf .*`” : rimuove tutto nella directory superiore e la directory stessa.

8.7.2 `rm /etc/passwd`

Perdere dei file importanti come `/etc/passwd` a causa della propria stupidità è pesante. Il sistema Debian ne effettua dei backup regolari in `/var/backups/`. Quando recuperate questi file, potreste impostare manualmente i giusti permessi.

```
# cp /var/backups/passwd /etc/passwd
# chmod 644 /etc/passwd
```

Vedere anche 'Recuperare i dati sulla selezione dei pacchetti' a pagina 87.

Capitolo 9

Messa a punto del sistema Debian

Questo capitolo descrive solo i principi basilari della configurazione del sistema tramite l'interfaccia a carattere. Un prerequisito di questo capitolo è aver letto 'Suggerimenti per l'installazione di un sistema Debian' a pagina 27.

Per coloro interessati alla sicurezza, si raccomanda caldamente di leggere il Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`), che si può anche reperire come pacchetto `harden-doc`.

9.1 Inizializzazione del sistema

debian; usa il sistema di script System V. Vedere 'Il programma `init`' a pagina 21 per un'introduzione.

9.1.1 Personalizzare gli script di inizializzazione

Il modo più semplice di controllare il comportamento di uno script di inizializzazione è modificando i compiti assegnati alle variabili d'ambiente nel file con lo stesso nome dello script nella directory `/etc/default/*`. [1] Per esempio, `/etc/default/hotplug` può essere utlizzato per controllare il comportamento di `/etc/init.d/hotplug`. Il file `/etc/init.d/rcS` può essere utilizzato per personalizzare i default al boot per `motd`, `sulogin`, etc.

Se non riuscite ad ottenere il comportamento che volete modificando dette variabili, potete allora modificare gli script stessi: sono tutti file di configurazione.

---

[1] I file in `/ect/default/` contengono **solo** i compiti delle variabili d'ambiente. Ciascun file è così strettamente correlato allo script corrispondente che questi compiti scavalcano qualsiasi impostazione predefinita della variabile nello script. La scelta del nome della directory è peculiare a Debian (`http://lists.debian.org/debian-devel/2003/debian-devel-200308/msg02114.html`). E' grosso modo l'equivalente di `/etc/sysconfig` di Red Hat ed altre distribuzioni.

9.1.2 Personalizzare i file di log

La modalità del log del sistema può essere configurata tramite `/etc/syslog.conf`. Date un'occhiata al pacchetto `colorize` per un programma che assegna colori differenti ai vari file di log. Vedere anche `syslogd(8)` e `syslog.conf(5)`.

9.1.3 Ottimizzare l'accesso hardware

Esistono alcune configurazioni hardware la cui ottimizzazione e lasciata da Debian alla cura dell'amministratore di sistema.

- `hdparm`
  - Ottimizzazione dell'accesso all'hard disk. Molto efficiente.
  - Pericolosa. Dovete leggere prima `hdparm(8)`.
  - `hdparm -tT /dev/hda` per provare la velocità di accesso al disco.
  - `hdparm -q -c3 -d1 -u1 -m16 /dev/hda` per accelerare un moderno sistema IDE. (Può essere pericoloso.)
- `setcd`
  - Ottimizzazione dell'accesso al lettore CD
  - `setcd -x 2` per rallentare a velocità 2x.
  - Vedere `setcd(1)`.
- `setserial`
  - Collezione di strumenti per la gestione delle porte seriali.
- `scsitools`
  - Collezione di strumenti per la gestione di hardware SCSI.
- `memtest86`
  - Collezione di strumenti per la gestione dell'hardware di memoria.
- `hwtools`
  - Collezione di strumenti per la gestione dell'hardware a basso livello.
    - `irqtune`: modifica la priorità dell'IRQ delle periferiche per concedere alte priorità e servizi veloci alle periferiche (tipo porte seriali, modem) che ne hanno bisogno. In tal modo è possibile ottenere accelerazioni 3X del throughput delle seriali/modem.
    - `scanport`: effettua la scansione dello spazio I/O da 0x100 a 0x3ff cercando periferiche ISA installate.
    - `inb`: un trucchetto piccolo e veloce che legge una porta I/O port e ne mostra il valore in esadecimale e binario.
- `schedutils`
  - Utilità del Linux scheduler.
  - sono compresi `taskset`, `irqset`, `lsrt`, e `rt`.

  - Insieme a `nice` e `renice` (non incluso), permettono un controllo completo dei parametri di scheduling dei processi.

Montare un filesystem con l'opzione `noatime` è anch'esso molto efficace nel velocizzare l'accesso in lettura ai file. Vedere `fstab(5)` e `mount(8)`.

Alcuni componenti hardware possono essere messi a punto direttamente dal kernel stesso, tramite il filesystem proc. Vedere 'Mettere a punto il kernel tramite il filesystem proc' a pagina 105.

Esistono molte utilità di configurazione hardware-specifiche in Debian. Molte di loro sono specifiche per i bisogni dei PC portatili. Ecco alcuni pacchetti interssanti disponibili in Debian:

- `tpconfig` - Programma per la configurazione del touchpad
- `apmd` - Utilità per Advanced Power Management (APM)
- `acpi` - mostra le informazioni sui device ACPI
- `acpid` - Utilità per usare ACPI
- `lphdisk` - prepara la partizione per l'ibernazione per i Phoenix NoteBIOS
- `sleepd` - mette il portatile in sleep durante l'inattività
- `noflushd` - permette il rallentamento dei dischi inattivi
- `big-cursor` - cursori del mouse più grandi per X
- `acme` - Abilita i "multimedia buttons" dei portatili
- `tpctl` - Strumenti di configurazione hardware per IBM ThinkPad
- `mwavem` - Supporto per i modem Mwave/ACP
- `toshset` - Accede a gran parte delle interfacce per l'hardware dei portatitili Toshiba
- `toshutils` - Toshiba laptop utilities
- `sjog` - Programma per usare il "Jog Dial" dei portatili Sony Vaio
- `spicctrl` - Programma per il controllo della luminosità dello schermo dei Sony Vaio

ACPI è un'infrastruttura più nuova di APM per la gestione dei profili energetici.

Alcuni di questi pacchetti richiedono speciali moduli del kernel. In molti casi sono già inclusi nei sorgenti più recenti. In caso di problemi potreste dover applicare da voi l'ultima patch.

9.2 Controllare gli accessi

9.2.1 Restrizione tramite PAM

PAM (Pluggable Authentication Modules) forniscono il controllo sul login.

```
/etc/pam.d/*             # file di controllo PAM
/etc/pam.d/login         # file di controllo PAM per il login
/etc/security/*          # parametri dei moduli PAM
/etc/securetty           # controlla il login di root da console (login)
/etc/login.defs          # controlla i vari comportamenti del login (login)
```

Se volete dei terminali senza password, ma insicuri, modificate a vostro rischio e pericolo il contenuto di `/etc/pam.d/login` come segue.

```
#auth        required   pam_unix.so nullok
auth       required pam_permit.so
```

Trucchetti simili possono essere applicati a `xdm`, `gdm`, . . . , per un accesso senza password alla console X.

Al contrario, se volete applicare una buona politica di password, installate `cracklib2` e modificate `/etc/pam.d/passwd` come segue.

```
password required pam_cracklib.so retry=3 minlen=6 difok=3
```

Le password utilizzabili solo una volta per l'attivazione degli account possono anche essere utili. Per fare ciò, usate il comando `passwd` con l'opzione `-e`. Vedere `passwd(1)`.

Il numero massimo di processi può essere impostato con `ulimit -u 1000` nella Bash shell oppure tramite `/etc/security/limits.conf` da PAM. Altri parametri, come `core` possono essere impostati allo stesso modo. Il valore iniziale di `PATH` può essere impostato tramite `/etc/login.defs` prima che intervenga lo script di inizializzazione della shell.

La documentazione per PAM è contenuta nel pacchetto `libpam-doc`. La *Linux-PAM System Administrator's Guide* copre gli aspetti della configurazione di, quali moduli sono disponibili, ecc. La documentazione comprende, inoltre, la *The Linux-PAM Application Developers' Guide* e la *The Linux-PAM Module Writers' Guide*.

9.2.2 "Why GNU **su** does not support the **wheel** group" (Perchè GNU **su** non supporta il gruppo **wheel**)

Questa è la famosa frase al termine della vecchia pagina di `info su`, di Richard M. Stallman. Per non preoccuparsi: l'attuale `su` in Debian usa PAM, così che potete ridurre la capacità di usare `su` a qualsiasi gruppo usando `pam_wheel.so` in `/etc/pam.d/su`. Quanto segue imposta il gruppo `adm` group di un sistema Debian come equivalente del gruppo `wheel` di BSD, permettendo `su` senza la password per i suoi membri.

```
# configurazione anti-RMS in /etc/pam.d/su
auth       required   pam_wheel.so group=adm

# I membri del gruppo wheel possono dare su senza la password
auth       sufficient pam_wheel.so trust group=adm
```

9.2.3 Il significato dei vari gruppi

Alcuni gruppi interessanti:

- il gruppo `root` è il gruppo wheel di default per `su` se `pam_wheel.so` è utilizzato senza l'argomento `group=`.

- `adm` può leggere il file di log.
- `cdrom` può essere usato localmente per fornire ad una serie di utenti l'accesso ad un drive CD-ROM.
- `floppy` può essere usato localmente per fornire ad una serie di utenti l'accesso ad un drive floppy.
- `audio` può essere usato localmente per fornire ad una serie di utenti l'accesso ad un device audio.
- `src` è proprietario del codice sorgente, compresi i file in `/usr/src`. può essere usato localmente per fornire ad un utente la possibilità di modificare il codice sorgente del sistema.
- `staff`: la sua appartenenza è utile per personale di helpdesk o amministratori di sistema aggiunti, dando loro la possibilità di fare modifiche in `/usr/local` e di creare delle directory in `/home`.

Per una lista completa, leggete la sezione "FAQ" del Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`), che può anche essere reperita come pacchetto `harden-doc` in Woody. Anche il nuovo pacchetto `base-passwd` (>3.4.6) contiene una lista autorevole: `/usr/share/doc/base-passwd/users-and-groups.html`.

9.2.4 sudo – lavorare con maggiore sicurezza

L'uso che faccio di `sudo` è fondamentalmente a protezione dalla mia stessa stupidità. Reputo l'uso di `sudo` un'alternativa migliore al continuo uso del sistema come root. Per voi le cose possono essere diverse.

Installate `sudo` ed attivatelo impostando le opzioni in `/etc/sudoers` (`http://www.debian.org/doc/manuals/debian-reference/examples/`). Controllate anche le caratteristiche del gruppo `sudo` in `/usr/share/doc/sudo/OPTIONS`.

La configurazione di esempio fornisce ai membri del gruppo "staff" accesso a qualsiasi comando come root tramite `sudo` nonchè accesso ai membri di "src" a comandi selezionati come root, sempre tramite `sudo`.

Il vantaggio di `sudo` è che richiede solo la password utente, e che l'attività e monitorizzata. E' un modo carino per dare un minimo di autorità ad un amministratore alle prime armi. Per esempio:

```
$ sudo chown -R myself:mygrp .
```

Ovviamente se conoscete la password di root (come la maggior parte degli utenti casalinghi) qualunque comando può essere eseguito come root da un account utente:

```
$ su -c "shutdown -h now"
Password:
```

(So che dovrei restringere i privilegi dell'account admin in `sudo`. Dato che è il mio server casalingo, non me ne sono ancora preso la briga.)

Per un programma differente che permetta agli utenti ordinari di eseguire comandi con privilegi di root, vedere il pacchetto `super`.

9.2.5 Controllare l'accesso ai servizi

Il *super-server* Internet `inetd`, viene lanciato al boot da `/etc/rc2.d/S20inetd` (per RUNLEVEL=2), che è un collegamento simbolico a `/etc/init.d/inetd`. In breve, `inetd` permette il lancio di un demone per richiamarne altri, riducendo il carico sul sistema.

Ogni volta che arriva la richiesta di un servizio, il suo protocollo ed il servizio vengono identificati guardando nel database contenuto in `/etc/protocols` e in `/etc/services`. `inetd` cerca, poi, un normale servizio Internet nel database `/etc/inetd.conf`, oppure un servizio basato su Sun-RPC in `/etc/rpc.conf`.

Per la sicurezza del sistema, accertatevi di disabilitare i servizi inutilizzati in `/etc/inetd.conf`. I servizi Sun-RPC devono essere attivi per il NFS ed altri programmi basati su RPC.

Talvolta `inetd` non lancia il server interessato direttamente, ma lancia il wrapper del demone TCP/IP `tcpd` con il nome del server come argomento in `/etc/inetd.conf`. In questo caso `tcpd` lancia il server appropriato dopo aver registrato la richiesta e fatto alcuni controlli addizionali, utilizzando `/etc/hosts.deny` e `/etc/hosts.allow`.

Se avete problemi con l'accesso remoto in un sistema Debian recente, commentate "ALL: PARANOID" in `/etc/hosts.deny`, se esiste.

Per i dettagli, vedere `inetd(8)`, `inetd.conf(5)`, `protocols(5)`, `services(5)`, `tcpd(8)`, `hosts_access(5)`, e `hosts_options(5)`.

Per maggiori informazioni su Sun-RPC, vedere `rpcinfo(8)` , `portmap(8)`, e `/usr/share/doc/portmap/portmapper.txt.gz`.

9.2.6 Centralizzare l'autenticazione – LDAP

Usate il Lightweight Directory Access Protocol (LDAP) Riferimenti:

- OpenLDAP (`http://www.openldap.org/`)
- OpenLDAP Admin Guide nel pacchetto `openldap-guide`
- LDP: LDAP Linux HOWTO (`http://www.tldp.org/HOWTO/LDAP-HOWTO/index.html`)
- LDP: LDAP Implementation HOWTO (`http://www.tldp.org/HOWTO/LDAP-Implementation-HOWTO/index.html`)
- OpenLDAP, extensive use reports (`http://portal.aphroland.org/~aphro/ldap-docs/ldap.html`)
- Open LDAP with Courier IMAP and Postfix (`http://alinux.washcoll.edu/docs/plc/postfix-courier-howto.html`)

9.3 Masterizzatori

I masterizzatori con interfaccia ATAPI/IDE sono di recente diventati un'opzione molto popolare. Rappresentano un supporto valido per il backup e l'archiviazione per l'utente casalingo che ha bisogno di capacità di < 640 MB. Per le informazioni più autorevoli, consultate il LDP CD-Writing-HOWTO (`http://www.tldp.org/HOWTO/CD-Writing-HOWTO.html`).

9.3.1 Introduzione

Primo, ogni danneggiamento di dati inviati al masterizzatore porterà ad un danno irreparabile del CD. Prendete il masterizzatore con il buffer più grande possibile. Se i soldi non sono un problema, prendetene uno con interfaccia SCSI. Se potete scegliere quale interfaccia IDE connettere, prendete quello su bus PCI (sulla scheda madre), piuttosto che su bus ISA (schede SB16, ecc.). etc.).

Quando il masterizzatore è connesso su IDE, deve essere comandato dal driver IDE-SCSI, non dall'IDE-CD. Deve essere attivato anche il driver generico SCSI. Esistono due approcci per ottenere ciò, per i kernel delle moderne distribuzioni (aggiornato a Marzo 2001).

Per il kernel 2.6, dovreste usare il driver ordinario IDE ed accedere al masterizzatore direttamente con il nome del device, tipo `/dev/hdx`. In questo modo, potete usare il DMA.

9.3.2 Approccio 1: moduli + `lilo`

Aggiungete la riga seguente a `/etc/lilo.conf` se state usando un kernel Debian standard. Se usate opzioni multiple, separatele da spazi:

```
append="hdx=ide-scsi ignore=hdx"
```

Dove la collocazione del masterizzatore, a cui si accede tramite il driver ide-scsi, è indicata da `hdx`, dove *x* rappresenta uno dei seguenti:

```
hda          se master sulla prima porta IDE
hdb          se slave sulla prima porta IDE
hdc          se master sulla seconda porta IDE
hdd          se slave sulla seconda porta IDE
hde ... hdh  per un drive su porta IDE esterna, ATA66/100
```

Date i comandi seguenti per l'attivazione dopo la configurazione:

```
# lilo
# shutdown -h now
```

9.3.3 Approccio 2: ricompilare il kernel

Debian usa `make-kpkg` per creare il kernel. Usate il nuovo `--append_to_version` con make-kpkg per produrre più kernel. Vedere 'Il kernel Linux su Debian' a pagina 99.

Usate le seguenti impostazioni attraverso `make menuconfig`:

- bzImage
- Escludete il driver IDE CD (non è obbligatorio, ma rende le cose più semplici)
- Compilate ide-scsi e sg, direttamente nel kernel o come moduli

9.3.4 Procedure post-configurazione

Il supporto del kernel per il CD-writer può essere attivato al boot con il seguente:

```
# echo ide-scsi >>/etc/modules
# echo sg       >>/etc/modules
# cd /dev; ln -sf scd0 cdrom
```

L'attivazione manuale può essere fatta con:

```
# modprobe ide-scsi
# modprobe sg
```

Dopo il reboot, potete controllarne l'installazione con:

```
$ dmesg|less
# apt-get install cdrecord
# cdrecord -scanbus
```

[Da Warren Dodge] Talvolta sorgono conflitti fra `ide-scsi` e `ide-cd` se ci sono sia un lettore che un masterizzatore nel sistema. Provate ad aggiungere la riga seguente al vostro `/etc/modutils/aliases`, lanciando poi `update-modules`, e facendo il reboot.

```
pre-install      ide-scsi      modprobe ide-cd
```

Ciò determina il caricamento del driver IDE prima di `ide-scsi`. Il driver IDE `ide-cd` prende il controllo del CD_ROM ATAPI—qualsiasi che non gli sia stato detto di **ignorare**. Ciò lascia solo i device ignorati al controllo di `ide-scsi`.

9.3.5 Il file CD-image (avviabile)

Per creare un CD-ROM con files contenuti nella `target-directory/` come `cd-image.raw` (avviabile, con il formato Joliet TRANS.TBL abilitato; se non lo volete avviabile, togliete le opzioni `-b` e `-c`), inserite un boot floppy nel drive e,

```
# dd if=/dev/fd0 target-directory/boot.img
# mkisofs -r -V volume_id -b boot.img -c bootcatalog -J -T \
        -o cd-image.raw target_directory/
```

Una trucco divertente è fare dei CD-ROM avviabili in DOS. Se una normale immagine DOS di un boot floppy è contenuta in *`boot.img`*, il CD-ROM eseguirà il boot come se nel primo floppy drive (A:) ci fosse un floppy DOS. Più interessante potrebbe essere farlo con freeDOS.

Il file CD-image può essere ispezionato montandolo sul device loop.

```
# mount -t iso9660 -o ro,loop cd-image.raw /cdrom
# cd /cdrom
# mc
# umount /cdrom
```

9.3.6 Scrivere sul masterizzatore (R, RW):

Primo test con (in caso di velocità 2x)

```
# nice --10 cdrecord -dummy speed=2 dev=0,0 disk.img
```

Se tutto va bene, scrivete sul CD-R con

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Oppure, sul CD-RW con

```
# nice --10 cdrecord -v -eject blank=fast speed=2 dev=0,0 disk.img
```

Alcune periferiche CD-RW funzionano meglio con

```
# nice --10 cdrecord -v blank=all speed=2 dev=0,0 disk.img
```

seguito da

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Sono necessari due passaggi per prevenire i timeout SCSI durante la fase di blanking dall'interferire con la fase di masterizzazione. Il valore dell'argomento di `nice` può richiedere qualche aggiustamento.

9.3.7 Creare un file immagine di un CD

Alcuni CD-R e CD commerciali hanno dei settori spuri alla fine, cosa che rende impossibile la copia tramite `dd` (il CD di Windows 98 è uno di questi). Il pacchetto `cdrecord` ha il comando `readcd`. Usatelo per copiare qualsiasi CD in un file immagine. Se è un disco di dati, montatelo ed eseguite `mount` per vedere le dimensioni reali. Dividete il numero ottenuto (in blocchi = 1024 bytes) per 2 per avere il numero reale di settori del CD (2048 bytes). Lanciate `readcd` con le opzioni ed usate l'immagine risultante per bruciare il CD-R/RW.

```
# readcd dev=target,lun,scsibusno # select function 11
```

Impostate tutti e tre i parametri a 0 per la maggior parte dei casi. Talvolta il numero di settori dato da `readcd` è eccessivo! Utilizzate il numero ottenuto da un montaggio reale per i migliori risultati.

Dovrebbe essere notato che l'uso di `dd` ha alcuni problemi se usato su CD-ROM. La prima esecuzione del comando `dd` può causare un messaggio di errore e produrre un'immagine del disco più piccola con perdita della coda. La seconda esecuzione di `dd` può produrre un'immagine più grande, con dati spuri attaccati alla fine su alcuni sistemi se le dimensioni dei dati non sono specificate. Solo la seconda esecuzione di `dd` con le dimensioni corrette dei dati specificate e senza espellere il CD dopo un messaggio di errore, sembra evitare questi problemi. Se, per esempio, le dimensioni dell'immagine mostrate da `df` sono di `46301184` blocchi, usate il comando seguente due volte per ottenere l'immagine corretta (queste sono mie informazioni empiriche):

```
# dd if=/dev/cdrom of=cd.img bs=2048 count=$((46301184/2))
```

9.3.8 CD con le immagini Debian

Per le informazioni più aggiornate sui CD Debian visitate il Sito Debian CD (`http://www.debian.org/CD/`).

Se avete una connessione Internet veloce, potete prendere in considerazione l'installazione via rete usando:

- alcune immagini floppy (`http://www.debian.org/distrib/floppyinst`).
- una immagine minima su CD avviabile (`http://www.debian.org/CD/netinst/`).

Se non avete una connessione veloce, considerate l'acquisto dei CD dai Venditori di CD (`http://www.debian.org/CD/vendors/`).

Non sprecate banda scaricando i CD immagine standard a meno di non essere un tester di CD immagine (anche con il nuovo metodo jigdo).

Un'mmagine degna di nota è KNOPPIX - Live Linux Filesystem On CD (`http://www.knopper.net/knoppix/index-en.html`). Questo CD farà il boot in un sistema Debian perfettamente funzionante senza installarsi sul disco rigido.

9.3.9 Backup del sistema su CD-R

Per copiare i file di configurazione chiave e di dati sul CD-R, usate lo script di "backup" backup (`http://www.debian.org/doc/manuals/debian-reference/examples/`). Vedete anche 'Copia ed archiviazione di una intera sottodirectory' a pagina 111 e 'Backup differenziali e sincronizzazione dei dati' a pagina 113.

9.3.10 Copiare CD musicali su CD-R

Non testato da me:

```
# apt-get install cdrecord cdparanoia
# cdparanoia -s -B
# cdrecord dev=0,0,0 speed=2 -v -dao -eject defpregap=1 -audio *.wav
```

oppure,

```
# apt-get install cdrdao #disk at once
# cdrdao read-cd --device /dev/cdrom --paranoia-mode 3 mio_cd # legge il cd
# cdrdao write --device /dev/cdrom --speed 8 mio_cd    # scrive un nuovo CD
```

`cdrdao` esegue una copia conforme (senza pause, ecc. . .)

9.3.11 Masterizzare i DVD-R, DVD-RW e DVD+RW

Per masterizzare i DVD, avete due approcci:

- Usate `growisofs` con `mkisofs`.
- Ricompilate `cdrecord` con l'opzione dvd per creare un pacchetto locale come da `/usr/share/doc/cdrecord/README.DVD.Debian`.

9.4 X

L'ambiente grafico X è fornito da Xfree86 (`http://www.xfree86.org/`). Due sono le versioni principali del server X disponibili su un sistema Debian: Le serie di XFree86 Versione 3.3 (XF3) e XFree86 Versione 4.x (XF4) sono basate sulle specifiche X11R6 di X.Org (`http://www.x.org/`).

Per le basi di X, fate riferimento a `X(7)`, l'LDP XWindow-User-HOWTO (`http://www.tldp.org/HOWTO/XWindow-User-HOWTO.html`), ed il Remote X Apps mini-HOWTO (`http://www.tldp.org/HOWTO/mini/Remote-X-Apps.html`). Per una guida per l'utente Debian-specifica, leggete `/usr/share/doc/xfree86-common/FAQ.gz` fornito dal pacchetto `xfree86-common`. Questa contiene una interessante ed autorevole revisione dei problemi chiave di Branden Robinson.

'X server' a fronte è un programma su host locale che mostra una finestra X e/o un desktop sul monitor di un utente (CRT, LCD) ed accetta input da tastiera e mouse.

'X client' a pagina 150 è un programma su un host (locale o remoto) su cui girano applicazioni compatibili con X window.

Queste definizioni ribaltano il concetto normale di "server" e "client".

Molteplici sono i modi di far accettare al "server X" (lato schermo) connessioni remote da un "client X" (lato applicazione):

- il metodo `xhost`
  - meccanismo della lista degli host (molto poco sicuro).
  - protocollo non criptato (soggetto ad attacchi tipo eavesdropping).
  - Se possibile, non utilizzatelo.
  - Vedere 'Connessioni ad un X server remoto – `xhost`' a pagina 154 and `xhost(1x)`.
- il metodo **`xauth`**
  - meccanismo MIT dei cookie magici (non sicuro, ma migliore di `xhost`).
  - protocollo non criptato (soggetto ad attacchi tipo eavesdropping).
  - usatelo solo per connessioni locali, dato che chiede meno CPU di `ssh -X`.
  - Vedere 'Diventare root in X' a pagina 156 e `xauth(1x)`.
- `xdm`, `wdm`, `gdm`, `kdm`, `...`
  - meccanismo MIT dei cookie magici (non sicuro, come `xauth`).
  - Vedere `xdm(1x)` e `Xsecurity(7)` per le basi del controllo degli accessi allo schermo X.
  - Vedere `wdm(1x)`, `gdm(8)`, e `kdm.options(5)` per maggiori informazioni, se sono installati.
  - Vedere 'Personalizzare i runlevel' a pagina 22 su come disabilitare `xdm` per avere una cosole Linux al boot senza rimuovere il pacchetto `xdm`.
- il metodo **`ssh -X`**
  - meccanismo del port forwarding tramite shell sicura (**sicuro**).
  - procollo criptato (uno spreco di risorse se usato localmente).
  - usatelo per connessioni remote.
  - Vedere 'Connessioni ad un X server remoto – `ssh`' a pagina 155.

Tutti i metodi di connessione remota, tranne `ssh`, richiedono l'abilitazione della connessione TCP/IP sul server X. Vedere 'Connessione TCP/IP ad X' a pagina 154.

9.4.1 Pacchetti del sistema X

Per facilitare l'installazione di X esistono alcuni (meta)pacchetti in Woody.

`x-window-system-core` Questo metapacchetto fornisce i componenti essenziali per una singola postazione con l' X Window System. Fornisce le librerie X, un X server `xserver-xfree86`, una serie di font ed un gruppo di X client base e di utilità.

`x-window-system` Fornisce in sostanza tutti i componenti dell' X Window System così come sono stati sviluppati dall' XFree86 Project, ed una serie di programmi accessori storicamente popolari. (Da notare che dipende da `x-window-system-core`, `twm`, e `xdm`, cioè non è necessario installare `x-window-system-core` se installate questo.)

`xserver-common-v3` Files ed utilità comuni agli X server XFree86 3.x (XF3)

`xserver-*` Pacchetti supplementari per XF3 server per supportare hardware non supportato dal nuovo XF4 server per qualsiasi ragione. Alcune vecchie ATI mach64 non sono supportate in XF4, alcune schede video si piantano brutalmente nella versione di XF4 per Woody, ecc. (Per una lista di pacchetti disponibili, usate `apt-cache search xserver- | less`. Tutti questi XF3 server dipendono da `xserver-common-v3`.)

Per la maggior parte dei casi il pacchetto da installare è `x-window-system`. (Se volete il login da console, assicuratevi di diasbilitare `xdm` come descritto in "'Voglio disabilitare X al boot!"' a pagina 109.)

9.4.2 Pacchetti per il riconoscimento hardware per X

Per abilitare il riconoscimento dell' hardware durante la fase di configurazione di X, installate i seguenti pacchetti prima di installare X:

- `discover` — sistema di identificazione hardware.
- `mdetect` — Strumento di autoidentificazione del mouse.
- `read-edid` — strumento per la raccoglta di informazioni sull'hardware per monitor VESA PnP.

9.4.3 X server

Vedere `XFree86(1x)` per informazioni sull' X server.

Invocare l' X server da una console locale:

```
$ startx -- :<display> vtXX
p.es.:
$ startx -- :1 vt8 -bpp 16
... parte su vt8 connesso ao localhost:1 in modalità 16 bpp
```

Gli argomenti dati dopo `--` sono per l' X server.

Notate che, quando si usa uno script `~/.xserverrc` per la personalizzazione dell'avvio dell' X server, bisogna essere certi di `exec` l' X server reale. Non fare ciò può causare rallentamenti nell'avvio e nell'uscita del server. Per esempio:

```
#!/bin/sh
exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Configurare l' X server (versione 4)

Per (ri-)configurare l' XF4 server,

```
# dpkg-reconfigure --priority=low xserver-common
# dpkg-reconfigure --priority=low xserver-xfree86
```

genererà il file `/etc/X11/XF86Config-4` e configurerà X usando lo script `dexconf`.

Configurare l' X server (versione 3)

Per (ri-)configurare l' X3 server,

```
# dpkg-reconfigure --priority=low xserver-common-v3
# dpkg-reconfigure --priority=low xserver-mach64
```

genererà il file `/etc/X11/XF86Config` e configurerà X usando lo script `xf86config-v3`.

Configurazione manuale dell' X server

Per aggiungere delle personalizzazioni utente a `/etc/X11/XF86Config-4`, **non modificate il file di configurazione compreso fra il testo**:

```
### BEGIN DEBCONF SECTION
[snip]
### END DEBCONF SECTION
```

Invece, **aggiungete le personalizzazioni prima del testo**. Per esempio, per usare una scheda video personalizzata, aggiungete qualcosa di simile al testo seguente all' *inizio* del file:

```
Section "Device"
  Identifier        "Custom Device"
  Driver            "ati"
  Option            "NoAccel"
EndSection

Section "Screen"
  Identifier  "Custom Screen"
  Device      "Custom Device"
  Monitor     "Generic Monitor"
  DefaultDepth 24
  Subsection "Display"
    Depth       8
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
  Subsection "Display"
    Depth       16
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
  Subsection "Display"
    Depth       24
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
EndSection

Section "ServerLayout"
    Identifier       "Custom"
    Screen           "Custom Screen"
    InputDevice      "Generic Keyboard" "CoreKeyboard"
    InputDevice      "Configured Mouse" "CorePointer"
EndSection
```

Per Sarge (`testing` al momento di scrivere), se volete mantenere le personalizzazioni dell'utente a `/etc/X11/XF86Config` attraverso gli aggiornamenti, lanciate da root il comando seguente:

```
# cp /etc/X11/XF86Config-4 /etc/X11/XF86Config-4.custom
# md5sum /etc/X11/XF86Config-4 > /var/lib/xfree86/XF86Config-4.md5sum
# dpkg-reconfigure xserver-xfree86
```

Per ottenere un **abbellimento dei caratteri**, dovete modificare `/etc/X11/XF86Config-4` come descritto in 'Font TrueType in X' a pagina 158.

Controllate anche le altre parti della vostra confiurazione di X. Impostazioni errate per il monitor possono dare più mal di testa dei caratteri brutti, quindi assicuratevi che il refresh rate sia il più alto che il vostro monitor possa gestire (85 Hz è ottimo, 75 Hz buono, 60 Hz doloroso).

9.4.4 X client

La maggior parte dei programmi X client possono essere lanciati con un comando tipo questo:

```
client $ xterm -geometry 80x24+30+200 -fn 6x10 -display hostname:0 &
```

Dove gli argomenti opzionali vogliono dire:

- `-geometry` *`WIDTH`*`x`*`HEIGHT`*`+`*`XOFF`*`+`*`YOFF`*: le dimensioni iniziali e la posizione della finestra.
- `-fn` *`FONTNAME`*: il font da usare per il testo. *`FONTNAME`* può essere:
  - a14: Font di grandezza normale
  - a24: Font di grandezza aumentata
  - … (controllate i font disponibili con `xlsfont`.)
- `-display` *`displayname`*: il nome dell' X server da usare. *`displayname`* può essere:
  - *`hostname`*`:`*`D.S`* significa schermo *`S`* sul display *`D`* dell' host *`hostname`*; l' X server per questo display è in ascolto sulla porta TCP 6000+D.
  - *`host`*`/unix:`*`D.S`* significa schermo *`S`* sul display *D* dell' host *`host`*; l' X server per questo display è in ascolto sullo UNIX domain socket `/tmp/.X11-unix/XD` (in tal modo è raggiungibile solo da *`host`*).
  - *`:D.S`* è equivalente a *`host`*`/unix:`*`D.S`*, dove *`host`* è il nome dell'host locale.

Il *`displayname`* predefinito per il programma X client program (lato applicazione) può essere impostato tramite la variabile d'ambiente DISPLAY. Per esempio, prima di lanciare un programma X client l'esecuzione di uno dei seguenti comandi permette ciò:

```
$ export DISPLAY=:0
        # il predefinito, macchina locale, utilizzando il primo schermo X
$ export DISPLAY=hostname.fulldomain.name:0.2
$ export DISPLAY=localhost:0
```

Il suo lancio può essere personalizzato tramite `~/.xinitrc`. Per esempio:

```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xclock -g 50x50-0+0 -bw 0 &
xload -g 50x50-50+0 -bw 0 &
xterm -g 80x24+0+0 &
xterm -g 80x24+0-0 &
twm
```

Come descritto in 'Personalizzare le X session' nella pagina successiva, ciò scavalca ogni normale esecuzione di `Xsession`, quando lanciato da `startx`. Usate questo approccio solo come ultima risorsa. Vedere `xsetroot(1x)`, `xset(1x)`, e 'X resources' a pagina 155.

9.4.5 Le X session

Una X session (X server + X client) può essere lanciata da:

- `startx`: script wrapper per `initx` per lanciare l' X server e client dalla console di Linux. Se `~/.xinitrc` non esiste, `/etc/X11/Xsession` viene lanciato tramite `/etc/X11/xinit/xinitrc`.
- `xdm`, `gdm`, `kdm`, o `wdm`: demoni di X display manager per lanciare l' X server e client e per controllare il login da un'interfaccia grafica. `/etc/X11/Xsession` viene eseguito direttamente.

Si può avere l'accesso alla console come da "'Voglio disabilitare X al boot!"' a pagina 109.

Personalizzare le X session

Lo script di inizializzazione predefinito, `/etc/X11/Xsession` è, in effetti, una combinazione di `/etc/X11/Xsession.d/50xfree86-common_determine-startup` e di `/etc/X11/Xsession.d/99xfree86-common_start`.

L'esecuzione di `/etc/X11/Xsession` viene in qualche modo influenzata da `/etc/X11/Xsession.options` ed è fondamentalmente l'esecuzione del programma che viene trovato per primo nella seguente lista dal comando `exec`:

1 `~/.xsession` o `~/.Xsession`, se è definito.
2 `/usr/bin/x-session-manager`, se è definito.
3 `/usr/bin/x-window-manager`, se è definito.
4 `/usr/bin/x-terminal-emulator`, se è definito.

L'esatto significato di questi comandi è determinato dal sistema alternativo Debian descritto in 'Comandi alternativi' a pagina 96. Per esempio:

```
# update-alternatives --config x-session-manager
... oppure
# update-alternatives --config x-window-manager
```

Per rendere qualsiasi X window manager quello predefinito mentre si mantengono installati i session manager di GNOME e KDE, sostituite `/etc/X11/Xsession.d/50xfree86-common_determine-startup` con quello allegato al secondo bug report in `http://bugs.debian.org/168347` (Spero che venga presto incluso.) e modificate `/etc/X11/Xsession.options` come segue per disabilitare l' X session manager:

```
# /etc/X11/Xsession.options
#
# configuration options for /etc/X11/Xsession
# See Xsession.options(5) for an explanation of the available options.
# Default enabled
allow-failsafe
allow-user-resources
```

```
allow-user-xsession
use-ssh-agent
# Default disabled (enable them by uncommenting)
do-not-use-x-session-manager
#do-not-use-x-window-manager
```

Senza le summenzionate modifiche al sistema, `gnome-session` e `kdebase` sono i pacchetti che contengono questi X session manager. Rimuovendoli si permette all'X window manager preferito di essere quello predefinito. (Idee migliori?)

Su un sistema in cui `/etc/X11/Xsession.options` contiene una riga `allow-user-xsession` non preceduta da altri caratteri, qualsiasi utente che definisse un proprio `~/.xsession` o `~/.Xsession` sarebbe in grado di personalizzare le azioni di `/etc/X11/Xsession`.

L'ultimo comando nel file `~/.xsession` dovrebbe essere sotto forma di `exec` *`some-window/session-manager`* per lanciare il vostro X window/session manager preferito.

Un buon esempio del file `~/.xsession` si trova in `/usr/share/doc/xfree86-common/examples/xsession.gz`.

Io lo uso per impostare il window manager, accesso allo schermo e lingua per ciascun utente. Vedere 'Lanciare una X session per un utente' in questa pagina, 'Diventare root in X' a pagina 156, e 'Esempio per un sistema multilingue per X window system' a pagina 174.

Se volete avere lanciare automaticamente parecchi programmi X client, guardate gli esempi di 'X client' a pagina 150 ed invocateli da `~/.xsession` invece che da `~/.xinitrc`.

Risorse addizionali di X specifiche per utente possono essere inserite in `~/.Xresources`. Vedere 'X resources' a pagina 155.

Mappature del mouse e della tastiera specifiche per ciascun utente in X possono essere definite nello script di lancio dell'utente. Vedere 'Mappatura della tastiiera e dei tasti del puntatore in X' a pagina 156.

Lanciare una X session per un utente

Seguendo il principio descritto in 'Personalizzare le X session' nella pagina precedente, X session/window manager specifici per l'utente possono essere attivati installando il pacchetto indicato ed impostando il contenuto alla fine del file `~/.xsession` come segue. (Amo `blackbox/fluxbox` per lo stile semplice e la velocità.):

- default X session manager.
  - Vedere 'Comandi alternativi' a pagina 96.
  - `exec /usr/bin/x-session-manager`
- default X window manager.

  - Vedere 'Comandi alternativi' a pagina 96.
  - `exec /usr/bin/x-window-manager`

- GNOME session manager. (caricato)
  - Installare il pacchetto: `gnome-session`
  - `exec /usr/bin/gnome-session`

- KDE session manager. (caricato)
  - Installare il pacchetto: `kdebase` (oppure `kdebase3` per KDE3)
  - `exec /usr/bin/kde2`

- Blackbox window manager. (leggero).
  - Installare il pacchetto: `blackbox`
  - `exec /usr/bin/blackbox`

- Fluxbox window manager (leggero, il nuovo blackbox)
  - Installare il pacchetto: `fluxbox`
  - `exec /usr/bin/fluxbox`

- Xfce window manager (Mac OS-X, SUN CDE–simile)
  - Installare il pacchetto: `xfce`
  - `exec /usr/bin/xfwm`

- IceWM window manager. (leggero, alternativa a GNOME)
  - Installare il pacchetto: `icewm`
  - `exec /usr/bin/X11/icewm`

- FVWM2 window manager virtuale (leggero, tipo Win95)
  - Installare il pacchetto: `fvwm`
  - `exec /usr/bin/fvwm2`

- Windowmaker window manager (in qualche modo simile a NexT)
  - Installare il pacchetto: `wmaker`
  - `exec /usr/bin/wmaker`

- Enlightenment window manager (caricato).
  - Installare il pacchetto: `enlightenment`
  - `exec /usr/bin/enlightenment`

Vedere Window Manager per X (`http://www.xwinman.org`).

Impostazione degli ambienti KDE e GNOME

Per un setup completo degli ambienti KDE o GNOME sono utili i seguenti metapacchetti:

- KDE: installare il pacchetto `kde`

- GNOME: installare il pacchetto `gnome`

L'installazione di questi pacchetti con programmi in grado di maneggiare pacchetti marcati con `Raccomanda`, tipo `dselect` ed `aptitude`, fornisce una scelta di software rispetto alla semplice installazione con `apt-get`.

Se volete comunque il login da console, disabilitate gli X display manager, tipo `kdm`, `gdm`, e `wdm`, che possono essere richiamati dal gioco delle dipendenze, come descritto in "'Voglio disabilitare X al boot!"' a pagina 109.

Se volete GNOME come sistema predefinito invece che KDE, configurate `x-session-manager` come descritto in 'Comandi alternativi' a pagina 96.

9.4.6 Connessione TCP/IP ad X

Siccome una connessione remota TCP/IP senza cifratura è vulnerabile ad attacchi tipo eavesdropping, l'impostazione predefinita per X nelle versioni Debian più recenti disabilita il socket TCP/IP. Prendete in considerazione `ssh` per una connessione remota ad X (vedere 'Connessioni ad un X server remoto – `ssh`' a fronte).

Il metodo qui descritto non è consigliato, a meno di essere in un ambiente altamente sicuro, dietro un buon firewall, solo con utenti fidati. Usate il seguente comando per verificare le impostazioni del vostro X server rispetto al socket TCP/IP:

```
# find /etc/X11 -type f -print0 | xargs -0 grep nolisten
/etc/X11/xinit/xserverrc:exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Rimuovete `-nolisten` per ripristinare l'ascolto TCP/IP sull' X server.

9.4.7 Connessioni ad un X server remoto – `xhost`

`xhost` permette l'accesso basato sugli hostname. Altamente insicuro. Quanto segue disabilita il controllo degli host e permette le connessioni da ovunque se una connessione TCP/IP è permessa (vedere 'Connessione TCP/IP ad X' in questa pagina):

```
$ xhost +
```

Potete riabilitare il controllo dell'host con:

```
$ xhost -
```

`xhost` non fa distinzioni fra utenti differenti sull'host remoto. In più gli hostname (in realtà gli indirizzi) possono essere contraffatti.

Tale metodo va evitato, anche con criteri relativi agli host più restrittivi, se siete su una rete non fidata (tipo l'accesso dialup con PPP ad Internet). Vedere `xhost(1x)`.

9.4.8 Connessioni ad un X server remoto – `ssh`

L'uso di `ssh` permette una connessione sicura da un server X locale ad un application server remoto.

- Impostate i parametri di `X11Forwarding` e `AllowTcpForwarding` a `yes` in `/etc/ssh/sshd_config` sull'host remoto.
- lanciate l' X server sull'host locale.
- Aprite un `xterm` sul'host locale.
- Lanciate `ssh` per stabilire una connessione col sito remoto.

```
localname @ localhost $ ssh -q -X -l loginname remotehost.domain
Password:
.....
```

- Eseguire comandi di applicazioni X sul sito remoto.

```
loginname @ remotehost $ gimp &
```

Questo metodo permette la visione dell'output del client X remoto come fosse connesso localmente attraverso uno UNIX domain socket.

9.4.9 L'emulatore di terminale di X – `xterm`

Per conoscere tutto su `xterm` guardate a `http://dickey.his.com/xterm/xterm.faq.html`.

9.4.10 X resources

Molti programmi per X più vecchi, tipo `xterm`, usano il cosiddetto X resource database per configurare il loro aspetto. Per immagazzinare le risorse specifiche per l'utente si usa il file `~/.Xresources`. Questo file viene fatto confluire automaticamente al login nelle X resources di default. Le X resources predefinite per il sistema sono contenute in `/etc/X11/Xresources/*` e i loro application defaults in `/etc/X11/app-defaults/*`. Usate queste impostazioni come punto di partenza.

Ecco alcune impostazioni utili da aggiungere al vostro file `~/.Xresources`:

```
! Imposta il font ad un più leggibile 9x15
XTerm*font: 9x15
```

```
! Mostra la barra di scorrimento
XTerm*scrollBar: true

! Imposta la grandezza del buffer a 1000 linee
XTerm*saveLines: 1000

! Schermo kterm più grande
KTerm*VT100*fontList: -*-fixed-medium-r-normal--24-*,\
 -*-gothic-medium-r-normal--24-*,\
 -*-mincho-medium-r-normal--24-*
```

Affinchè queste impostazioni abbianno effetto immediato, fatele confluire nel database con il comando:

```
xrdb -merge ~/.Xresources
```

Vedere `xrdb(1x)`.

9.4.11 Mappatura della tastiiera e dei tasti del puntatore in X

Il programma `xmodmap` viene usato per modificare e mostrare la la modifier map della tastiera la keymap table usate dalle applicazioni client per convertire i keycode in keysyms in X.

```
$ xmodmap -pm
 ... mostra la modifier map attuale
$ xmodmap -pk | pager
 ... mostra la keymap tabl attuale
$ xmodmap -e "pointer = 3 2 1" # imposta il mauose per la mano sinistra
$ xmodmap ~/.xmodmaprc # imposta la tastiera come in ~/.xmodmaprc
```

E' in genere lanciato dallo script di inizio della sessione di ciascun utente, `~/.xsession`.

Per avvere il `keycode`, lanciate `xev` in X e premete i tasti. Per sapere cosa significa `keysym`, guardate alla definizione MACRO in `/usr/include/X11/keysymdef.h`. Tutte le istruzioni `#define` in questo file sono chiamate con `XK_` davanti ai nomi `keysym`.

Vedere `xmodmap(1x)`.

9.4.12 Diventare root in X

Se un programma con interfaccia grafica richiede l'esecuzione da root, usate le seguenti procedure per visualizzare l'output del programma sull'X server dell'utente. **Non tentate mai di lanciare un X server direttamente dal root account** per evitare possibili rischi dal punto di vista della sicurezza.

Lanciate l' X server come un utente normale, ed aprite una console `xterm`. Poi:

```
$ XAUTHORITY=$HOME/.Xauthority
$ export XAUTHORITY
$ su root
Password:*****
# printtool &
```

Quando usate questo trucco con `su` verso un utente non root, assicuratevi che `~/.Xauthority` sia leggibile dal gruppo di detto utente non root.

Per automatizzare questa sequenza di comandi, create un file `~/.xsession` dall'accout utente che contenga le seguenti righe:

```
# Questo fa funzionare X quando lancio su verso l'account root.
if [ -z "$XAUTHORITY" ]; then
        XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi
unset XSTARTUP
# Se si desidera un window/session manager particolare, decommentate la riga segu
# e modificatela a vostro piacimento.
#XSTARTUP=/usr/bin/blackbox
# Questo lancia il programma x-window/session-manager
if [ -z "$XSTARTUP" ]; then
  if [ -x /usr/bin/x-session-manager ]; then
    XSTARTUP=x-session-manager
  elif [ -x /usr/bin/x-window-manager ]; then
    XSTARTUP=x-window-manager
  elif [ -x /usr/bin/x-terminal-emulator ]; then
    XSTARTUP=x-terminal-emulator
  fi
fi
# lancia automaticamente l'X window/session manager selezionato
exec $XSTARTUP
```

Poi lanciate `su` (non `su -`) in una finestra `xterm` dell'utente. I programmi con interfaccia grafica lanciati da questo `xterm` possono visualizzare l'output sull'X window dell'utente mentre girano con i privilegi di root. Il trucco funziona finchè si usa il file `/etc/X11/Xsession` di default. Se un utente fa le proprie modifiche tramite `~/.xinit` o `~/.xsession`, la variabile d'ambiente sopra descritta `XAUTHORITY` dovrà essere impostata allo stesso modo in questi script.

In alternativa, si può usare, `sudo` per automatizzare la sequenza di comandi:

```
$ sudo xterm
... oppure
$ sudo -H -s
```

In questo caso `/root/.bashrc` dovrà contenere:

```
if [ $SUDO_USER ]; then
sudo -H -u $SUDO_USER xauth extract - $DISPLAY | xauth merge -
fi
```

Questo funziona bene anche se la directory home dell'utente è montata su NFS mount, poichè root non legge il file `.Xauthority`.

Esistono anche parecchi altri pacchetti specializzati all'uopo: `kdesu`, `gksu`, `gksudo`, `gnome-sudo`, e `xsu`. Altri metodi possono essere usati per raggiungere lo stesso risultato: creare un collegamento simbolico tra `/root/.Xauthority` ed il corrispettivo dell'utente; usare lo script sux (`http://fgouget.free.fr/sux/sux-readme.shtml`); or mettere "`xauth merge ~`*`USER_RUNNING_X`*`/.Xauthority`" nello script di inizializzazione di root.

Vedere anche sulla lista di messaggi debian-devel (`http://lists.debian.org/debian-devel/2002/debian-devel-200207/msg00259.html`).

9.4.13 Font TrueType in X

Lo standard `xfs` in XFree86-4 funziona bene con i fonts TrueType. Se usate XFree86-3 dovete installare un server di terzi, tipo `xfs-xtt`.

Dovete solo assicurarvi che con qualsiasi applicazione vogliate usare, i font TrueType siano collegati a libXft o libfreetype (se usate file .deb precompilati non dovete nemmeno preoccuparvi di questo, probabilmente).

Per prima cosa, impostate l'infrastruttura per il supporto dei font:_

- Installate i pacchetti `x-ttcidfont-conf` e `defoma`. Ciò automatizza la generazione dei file `fonts.scale` e `fonts.dir`.
  ```
  # apt-get install x-ttcidfont-conf
  ```
- Modificate `/etc/X11/XF86Config-4` in `Section "Files"` come segue:
  ```
  Section "Files"
      FontPath  "/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType"
      FontPath  "/usr/share/fonts/truetype"
      FontPath  "/usr/lib/X11/fonts/CID"
      FontPath  "/usr/lib/X11/fonts/Speedo"
      FontPath  "/usr/lib/X11/fonts/misc"
      FontPath  "/usr/lib/X11/fonts/cyrillic"
      FontPath  "/usr/lib/X11/fonts/100dpi:unscaled"
      FontPath  "/usr/lib/X11/fonts/75dpi:unscaled"
      FontPath  "/usr/lib/X11/fonts/Type1"
  EndSection
  ```
  la prima riga imposta XFree86 ad usare qualsiasi font TrueType installiate da un pacchetto Debian. La voce relativa al font Type1 è stata spostata in basso, poichè XFree86 non fa un bel lavoro nel rendere i font Type1. Il trucco `:unscaled` per i font bitmap non

dovrebbe essere più necessario con il nuovo XF4, ma lo ho incluso lo stesso, tanto per essere sicuro. Per mantenere le modifiche manuali a `/etc/X11/XF86Config-4` seguite le istruzioni in 'Configurazione manuale dell' X server' a pagina 148.

Poi installate i pacchetti di font DFSG:

- Western TrueType font:
  - `ttf-bitstream-vera`: font TrueType di alta qualità, creati day Bitstream, Inc. [2]
  - `ttf-freefont`: font gratuiti TrueType di alta qualità che comprende il gruppo di caratteri UCS.
  - `ttf-thryomanes`: font TrueType unicode che copre Latino, Graco, Cirillico ed IPA.
- Asian font:
  - `tfm-arphic-bsmi00lp`: Chinese Arphic "AR PL Mingti2L Big5" TrueType font TeX font metric data
  - `tfm-arphic-bkai00mp`: Chinese Arphic "AR PL KaitiM Big5" TrueType font TeX font metric data
  - `tfm-arphic-gbsn00lp`: Chinese Arphic "AR PL SungtiL GB" TrueType font TeX font metric data
  - `tfm-arphic-gkai00mp`: Chinese Arphic "AR PL KaitiM GB" TrueType font TeX font metric data
  - `ttf-baekmuk`: serie di font TrueType Korean Baekmuk
  - `hbf-jfs56`: Chinese Jianti Fangsong 56x56 bitmap font (GB2312) per CJK
  - `hbf-cns40-b5`: Chinese Fanti Song 40x40 bitmap font (Big5) per CJK
  - `hbf-kanji48`: Japanese Kanji 48x48 bitmap font (JIS X-0208) perr CJK

Siccome i font **Free** sono talvolta limitati, installarne o condividerne di commerciali è un'opzione per gli utenti Debian. Per rendere facile tale processo, sono stati creati alcuni pacchetti ad hoc:

- `ttf-commercial`
- `msttcorefonts (>1.1.0)` [3]

Avrete un'ottima selezione di font TrueType al costo di contaminare il vostro sistema **Free** con font non-Free.

Tutti questi pacchetti di font dovrebbero funzionare in Debian senza sforzi, ed apparire disponibili a tutti i programmi che usano il font system "centrale". Ciò include programmi tipo Xterm, Emacs, e molte altre applicazioni non-KDE e non-GNOME.

Ora, lanciate `xfontsel` e scegliete qualsiasi font TrueType nel menu fndry, dovreste essere in grado di vedere molte voci disponibili nel menu "fmly".

Per KDE2.2 e GNOME1.4 (con libgdkxft0, che è un trucco per costringere GTK 1.2 ad eseguire il rendering dei font anti-aliased), dovete impostare anche Xft1. Xft1 è altamente deprecato ed è in pratica usato solo da GNOME1.4 e da KDE2.2. Modificate `/etc/X11/XftConfig` ed aggiungete una riga tipo

```
dir "/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType"
```

---

[2] Sebbene non disponibili per in Woody, potete installarli da Sarge.

[3] Il pacchetto in Woody non funziona, almeno fino al 8/2002, a causa di modifiche al sito della Microsoft. Usate la versione di Sarge anche in Woody, invece.

prima delle altre righe dir. [4]

Per GNOME2 e KDE3 (dopo il rilascio di Sarge), dovete impostare `fontconfig` che usa Xft2 per trovare i font. [5] Non dovreste aver bisogno di installare alcunchè di extra per questo, pochè tutti i pacchetti che usano `fontconfig` Dipendono già da esso (indirettamente).

Prima gurdate in `/etc/fonts/fonts.conf`. Dovrebbe esserci una riga come quella sotto. Se non c'è, aprite `/etc/fonts/local.conf` ed aggiungete questo

```
        <dir>/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType</dir>
```

subito dopo la riga <fontconfig>.

`Fontconfig` dovrebbe prenderli immediatamente ed "fc-list" dovrebbe elencare i vostri nuovi font. Un'altra valida caratteristica di `fontconfig` è che potete inserire semplicemente i font da `~/.fonts/` e tutti i vostri programmi fontconfigurati avranno accesso a loro immediatamente.

Se installate manualmente un set di font TrueType mentre siete in X senza usare un pacchetto Debian, lanciate

```
   # xset fp rehash
```

per far sì che XFree86 guardi al contenuto di quella directory nuovamente e prenda i nuovi font.

9.4.14 Web Browser (grafici)

In Woody esistono alcuni pacchetti di Web browser con capacità grafiche:

- `mozilla` Il browser Mozilla (nuovo)
- `galeon` Browser basato su Mozilla con interfaccia Gnome (nuovo)
- `konqueror` Browser KDE
- `dillo` Browser GTK
- `amaya-gtk` Browser di riferimento W3C
- `amaya-lesstif` idem come sopra
- `netscape-...` (molti, vecchio)
- `communicator-...` (molti, vecchio)
- ...

La versione di `mozilla` deve corrispondere a quella richiesta da `galeon`. Nonostante differiscano nell'interfaccia, entrambi condividono il motore di rendering HTML Gecko.

I plug-in per i browser tipo `mozilla` e `galeon` possono essere abilitati installando manualmente "`*.so`" nella directory plug-in e riavviando i browser.

Varie risorse per i plug-in:

---

[4] Io non ho più roba xft1 sulla mia macchina, per cui non sono certo se avete bisogno di riavviare X o no prima che le modifiche abbiano effetto. Mi pare di ricordare che "xftcache" aggiorni la cache Xft1, ma sarebbe bello che qualcuno possa confermarlo per me.

[5] `Fontconfig` non esiste in Woody.

- Java plug-in: installate i binari di “J2SE” da `http://java.sun.com`.
- Flash plug-in: installate i binari di “Macromedia Flash Player 5” da `http://www.macromedia.com/software/flashplayer/`.
- `freewrl`: Browser VRML e plugin per Netscape
- …

9.5 SSH

SSH (Secure SHell) è il modo sicuro per connettersi via Internet. Una versione free di SSH, chiamata OpenSSH è disponibile come pacchetto `ssh` in Debian.

9.5.1 Principi basilari di SSH

Installate prima il server ed il client OpenSSH.

```
# apt-get update && apt-get install ssh
```

E′ necessario il riferimento a non-US in `/etc/apt/source.list`. `/etc/ssh/sshd_not_to_be_run` non deve essere presente se si vuole far girare il server OpenSSH.

SSH ha due protocolli di autenticazione:

- Versione 1:
  - Solo Potato supporta questo protocollo.
  - metodi di autenticazione disponibili:
    - RSAAuthentication: autenticazione utente basata su “RSA identity key”
    - RhostsAuthentication: autenticazione host basata su .rhosts (insicura, disabilitata)
    - RhostsRSAAuthentication: autenticazione .rhosts combinata con “RSA host key” (disabilitata)
    - ChallengeResponseAuthentication: autenticazione “RSA challenge-response”
    - PasswordAuthentication: autenticazione basata su password

- Versione 2:
  - tutte le versioni post-Woody la usano come protocollo primario.
  - metodi di autenticazione disponibili:
    - PubkeyAuthentication: autenticazione utente basata su chiave pubblica
    - HostbasedAuthentication: autenticazione basata su `.rhosts` o `/etc/hosts.equiv` combinata con autenticazione dell’host basata su chiave pubblica del client (disabilitata)
    - ChallengeResponseAuthentication: autenticazione challenge-response
    - PasswordAuthentication: autenticazione basata su password

Fate attenzione a queste differenze se state migrando da Woody od usando un sistema non-Debian.

Vedere `/usr/share/doc/ssh/README.Debian.gz`, `ssh(1)`, `sshd(8)`, `ssh-agent(1)`, e `ssh-keygen(1)` per i dettagli.

A seguire i file di configurazione chiave:

- `/etc/ssh/ssh_config`: Il default per il client SSH. Vedere `ssh(1)`. Voci da notare sono:
  - `Host`: Restringe le seguenti dichiarazioni (fino alla Host keyword successiva) ad essere le uniche per i dati host a corrispondere ad uno dei pattern dati dopo la keyword.
  - `Protocol`: Specifica la versione di protocollo SSH. Default è "2,1".
  - `PreferredAuthentications`: Specifica il metodo di autenticazione del client SSH2. Il default è "hostbased,publickey,keyboard-interactive,password".
  - `PasswordAuthentication`: Se volete accedere con una password, dovete assicurarvi chenon sia impostato su `no`.
  - `ForwardX11`: E' disabilitato di default. Si può scavalcare con l'opzione "`-X`".
- `/etc/ssh/sshd_config`: Il default per il server SSH. Vedere `sshd(8)`. Voci da notare sono:
  - `ListenAddress`: Specifica l'indirizzo locale verso cui `sshd` dovrebbe essere in ascolto. Sono permesse opzioni multiple.
  - `AllowTcpForwarding`: Disabilitato di default.
  - `X11Forwarding`: Disabilitato di default.
- `$HOME/.ssh/authorized_keys`: lista delle chiavi pubbliche di default usate dai client per connettersi ad un dato account sul dato host. Vedere `ssh-keygen(1)`.
- `$HOME/.ssh/identity`: Vedere `ssh-add(1)` e `ssh-agent(1)`.

I seguenti comandi lanciano una connessione `ssh` da un client.

```
$ ssh username@hostname.domain.ext
$ ssh -1 username@hostname.domain.ext # Forza la versione 1 di SSH
$ ssh -1 -o RSAAuthentication=no -l username foo.host
    # forza la password su SSH1
$ ssh -o PreferredAuthentications=password -l username foo.host
    # forza la password su SSH2
```

Per l'utente, `ssh` funziona da `telnet` più sicuro ed intelligente (non vi bombarda con ^]).

9.5.2 Port forwarding – per il tunneling SMTP/POP3

Per stabilire una pipe per connettere la porta 25 del *server-remoto* dalla porta 4025 del localhost, e la porta 110 del *server-remoto* dalla porta 4110 del localhost attraverso `ssh`, eseguite sulla macchina locale:

```
# ssh -q -L 4025:server-remoto:25 4110:server-remoto:110 \
           username@server-remoto
```

E' un modo sicuro di eseguire connessioni a server SMTP/POP3 su Internet. impostate la voce `AllowTcpForwarding` a `yes` in `/etc/ssh/sshd_config` sull'host remoto.

9.5.3 Connettersi con meno passwords – RSA

Potete evitare di ricordare una password per ogni sistema remoto usando la RSAAuthentication (protocollo SSH1) o PubkeyAuthentication (protocollo SSH2).

Sul sistema remoto impostate i rispettivi parametri, "RSAAuthentication yes" o "PubkeyAuthentication yes", in `/etc/ssh/sshd_config`.

Infine, generate le chiavi di autenticazione localmente ed installate la chiave pubblica sul sistema remoto:

```
$ ssh-keygen          # RSAAuthentication: chiave RSA per SSH1
$ cat .ssh/identity.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t rsa   # PubkeyAuthentication: chiave RSA per SSH2
$ cat .ssh/id_rsa.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t dsa   # PubkeyAuthentication: chiave DSA per SSH2
$ cat .ssh/id_dsa.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
```

Potrete cambiare la passphrase in seguito con "`ssh-keygen -p`". Verificate le impostazioni provando la connessione. In caso di problemi, usate "`ssh -v`".

Potete aggiungere delle opzioni ai parametri in `authorized_keys` per limitare gli host e per eseguire comandi specifici. Vedere `sshd(8)` per i dettagli.

Notate che SSH2 ha `HostbasedAuthentication`. Affinchè funzioni, dovete aggiustare le impostazioni di `HostbasedAuthentication` a `yes` in both `/etc/ssh/sshd_config` on the server machine and `/etc/ssh/ssh_config` o `$HOME/.ssh/config` sulla macchina client.

9.5.4 Client SSH differenti

Esistono alcuni client SSH disponibili per piattaforme non Unix e simili.

Windows puTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/) (GPL)

Windows (cygwin) SSH in cygwin (http://www.cygwin.com/) (GPL)

Macintosh Classic macSSH (http://www.macssh.com/) (GPL) [Notate che Mac OS X include OpenSSH; usate `ssh` nell'applicazione Terminal]

Vedere anche SourceForge.net, site documentation (http://www.sourceforge.net/docman/?group_id=1), "6. CVS Instructions".

9.5.5 SSH agent

E' più sicuro proteggere la vostra autenticazione SSH con una passphrase. Se non è impostata, usate `ssh-keygen -p` per farlo.

Inserite la vostra chiave pubblica (p. es `~/.ssh/id_rsa.pub`) in `~/.ssh/authorized_keys`, su un host remoto mediante una connessione basata su password come descritto in 'Connettersi con meno passwords – RSA' nella pagina precedente.

```
$ ssh-agent bash # oppure lanciate zsh/tcsh/pdksh.
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/osamu/.ssh/id_rsa:
Identity added: /home/osamu/.ssh/id_rsa (/home/osamu/.ssh/id_rsa)
$ scp foo user@remote.host:foo
 ... da qui in poi più nessuna passphrase richiesta :-)
$^D
 ... terminando la sessione di ssh-agent
```

Per l'X server, i normali script di lancio Debian eseguono `ssh-agent` come processo genitore. Così, dovete eseguire `ssh-add` solo una volta.

Per ulteriori informazioni, leggete `ssh-agent(1)` e `ssh-add(1)`.

9.5.6 Risoluzione dei problemi con SSH

In caso di problemi, controllate i permessi del file di configurazione e lanciate `ssh` con l'opzione "`-v`".

Usate l'opzione "`-P`" se siete root ed avete problemi col firewall, per evitare di usare le porte del server 1–1023.

Se le connessioni `ssh` ad un sito remoto smettono improvvisamente di funzionare, ciò può essere dovuto a maneggiamenti dell'amministratore o, più probabilmente a cambiamenti in `host_key` durante la manutenzione del sistema. Dopo essersi assicurati che è proprio questo il caso e che nessuno sta tentando di falsificare l'host remoto mediante qualche trucco geniale, potete riguadagnare la connessione rimuovendo la voce `host_key` da `$HOME/.ssh/known_hosts` della macchina locale.

9.6 Posta

La configurazione della posta si divide in tre categorie:

- agenti di trasferimento posta (mail transfer agent, MTA): `exim4`, `exim`, `postfix`, `sendmail`, `qmail`, `ssmtp`, `nullmailer`, ...
- utilità per la posta: `procmail`, `fetchmail`, `mailx`, `crm114`, ...
- agenti di gestione posta per l'utente (mail user agent, MUA): `mutt` `emacs+gnus`, ...

9.6.1 Mail transport agent (MTA)

Per un MTA con tutte le funzionalità, usate `exim` in Woody e `exim4` in Sarge. [6] Riferimenti:

- i pacchetti `exim-doc` e `exim-doc-html` per `exim`
- `exim4-doc-info` e `exim4-doc-html` per `exim4`
- `http://www.exim.org/`

L'unica alternativa ragionevole è `postfix` se avete a cuore la sicurezza. `sendmail` e `qmail` sono disponibili come pacchetti Debian, ma non sono raccomandati.

Se non avete bisogno delle capacità specifiche di un MTA, come nel caso di un sistema satellite tipo un portatile, potete prendere in considerazione uno di questi pacchetti leggeri.

- `ssmtp`: ncessita di una connessione SMTP ed supporta gli alias.
- `nullmailer`: può distribuire la posta, ma non supporta gli alias.

Al momento, trovo `exim` più adatto, anche per la mia workstation pesonale, che è un PC portatile.

Potreste dover rimuovere `exim` per installare questi pacchetti conflittuali:

```
# dpkg -P --force-depends exim
# apt-get install nullmailer         # o ssmtp
```

Smarthost

Se avete `exim4` o `exim` su un host connesso tramite servizi commerciali, assicuratevi di inviare la posta in uscita attraverso uno smarthost offerto dal vostro ISP, o da qualcun altro. [7] esistono alcune buone ragioni:

- per assicurarsi più tentativi SMTP, in quanto lo smarthost del vostro ISP ha una connessione più affidabile.
- evitare di inviare la posta direttamente da un **indirizzo IP dinamico**, che verrà con ogni probabilità bloccato dalle liste dial-u dello spam.

---

[6] Le sezioni seguenti usano `exim` negli esempi. Per Sarge sostituitelo con `exim4` dove necessario.

[7] Dovete segure questa regola per qualsiasi host connesso tramite linea analogica, DSL, cavo o LAN attraverso qualche route a larga banda Anche se il vostro host casalingo ha un IP fisso dato dal vostro ISP, è sempre una buona idea seguire questa regola. Granparte delle workstation e dei server ricadono in questa categoria.

- per risparmiare la vostra banda locale per inviare posta a destinatari multipli.

Le uniche eccezioni possibili sono:

- cura di emergenza per i problemi del servizio SMTP del vostro ISP.
- un esperimento a scopi educativi.
- il vostro host è un server hosted professionalmente

Configurazione base di Exim

per usare `exim` e `exim4` come MTA, configurate:

```
/etc/exim/exim.conf     "eximconfig" per crearlo e modificarlo
/etc/exim4/*            "dpkg-reconfigure exim4" per crearlo e modificarlo (exim4
/etc/inetd.conf         decommentate smtp per lanciare exim come demone
/etc/email-addresses    Aggiungete una lista di indirizzi email fasulli
```

controllate i filtri usando `exim4` o `exim` con `-brw, -bf, -bF, -bV, ...` ecc.

Raccolta di tutti gli indirizzi e-mail inesistenti in Exim

In `/etc/exim/exim.conf` (Woody e seguenti), aggiungete nella sezione DIRECTORS alla fine, (dopo localuser: director) un indirizzatore (director) che raccolga tutti gli indirizzi che i directors precedenti non hanno potuto risolvere (Secondo Miquel van Smoorenburg):

```
catchall:
  driver = smartuser
    new_address = webmaster@mydomain.com
```

Se si desidera avere una ricetta più dettagliata per ogni dominio virtuale, ecc., aggiungete la seguente riga alla fine di `/etc/exim/exim.conf` (secondo me, non ben testato):

```
*@yourdomain.com ${lookup{$1}lsearch*{/etc/email-addresses} \
        {$value}fail} T
```

Aggiungete, poi una "*" in `/etc/email-addresses`.

Riscrivere selettivamente l'indirizzo per la posta in uscita (Exim)

La riscrittura selettiva dell'indirizzo per la posta in uscita per avere un giusto header "From:" può essere fatta con `exim` configurando verso la parte finale `/etc/exim/exim.conf`:

```
*@host1.something.dyndns.org \
  "${if eq {${lookup{$1}lsearch{/etc/passwd}{1}{0}}} {1}  \
   {$0}{$1@somethig.dyndns.org}}"  frFs
```

Ciò riscrive tutti gli indirizzi che coincidono con `*@host1.something.dyndns.org`.

1 Cerca dentro `/etc/password` per vedere se la parte locale ($1) è un utente locale o no.

2 Se lo è, riscrive l'indirizzo facendolo coincidere con la stessa cosa trovata al primo posto ($0).

3 Se non lo è, riscrive la parte del dominio.

Configurare l'autenticazione SMTP in Exim

Alcuni servizi SMTP, tipo yahoo.com richiedono l'autenticazione SMTP. Configurate `/etc/exim/exim.conf` come segue:

```
remote_smtp:
  driver = smtp
  authenticate_hosts = smtp.mail.yahoo.com
...

smarthost:
  driver = domainlist
  transport = remote_smtp
  route_list = "* smtp.mail.yahoo.com bydns_a"
...

plain:
  driver = plaintext
  public_name = PLAIN
  client_send = "^cmatheson3^this_is_my_password"
```

Non dimenticate le virgolette doppie sull'ultima riga.

9.6.2 Ritirare la posta – Fetchmail

`fetchmail` viene eseguito in modalità demone per raccogliere la posta dagli account POP3 del vostro provider smistandoli nel sistema locale di posta. Configurate:

```
/etc/init.d/fetchmail
/etc/rc?.d/???fetchmail lancia update-rc.d fetchmail default priority 30
/etc/fetchmailrc        file di configurazione (chown 600, posseduto da fetchmail
```

Le informazioni su come lanciare fetchmail come demone dallo script `init.d` in Potato sono confuse (in Woody è stato risolto). Vedere i file di esempio `/etc/init.d/fetchmail` e `/etc/fetchmailrc` negli script di esempio (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

Se i vostri header di posta sono infestati da ^M per colpa del mailer del vostro provider, aggiungete “stripcr” alle vostre opzioni in `$HOME/.fetchmailrc`:

```
options fetchall no keep stripcr
```

9.6.3 Processare la posta – Procmail

`procmail` è il sistema locale di consegna e filtraggio della posta. Per ogni account che lo usa si deve creare un file `$HOME/.procmailrc`. Per esempio: _procmailrc (`http://www.debian.org/doc/manuals/debian-reference/examples/`)

9.6.4 Processare lo spam con `crm114`

Il pacchetto `crm114` fornisce lo script `/usr/share/crm114/mailfilter.crm`, che è scritto in CRM114. Esso fornisce un filtro per lo spam molto efficace, che può essere instruito fornendogli spam e posta normale.

CRM114 è un piccolo linguaggio disegnato per scrivere dei filtri; consideratelo una versione di grep con superpoteri. Vedere `crm(1)`.

9.6.5 Leggere la posta – Mutt

Usate `mutt` come MUA in combinazione con `vim`. Personalizzatelo tramite `~/.muttrc`; per esempio:

```
# usa il modo visuale e "gq" per riformattare le citazioni
set editor="vim -c ’set tw=72 et ft=mail’"
#
# impostazione degli header, presa direttamente dal manuale ("Sven’s Draconian he
#
ignore *
unignore from: date subject to cc
unignore user-agent x-mailer
hdr_order from subject to cc date user-agent x-mailer
auto_view application/msword
....
```

Aggiungete quanto segue a `/etc/mailcap` o a `$HOME/.mailcap` per visualizzare messaggi in HTML ed allegati MS Word:

```
text/html; lynx -force_html %s; needsterminal;
application/msword; /usr/bin/antiword '%s'; copiousoutput;
description="Microsoft Word Text"; nametemplate=%s.doc
```

9.7 Localizzazione (l10n)

Debian è internazionalizzato, offrendo supporto per un numero crescente di lingue e convenzioni locali. La sottosezione seguente elenca alcune delle diversità che Debian attualmente supporta, mentre la successiva tratta della **localizzazione**, ovvero il processo di personalizzare il vostro ambiente di lavoro per permettere il giusto input ed output del(i) linguaggio(i) scelto(i) e le convenzioni per date, formati numerici e monetari e per tutti gli aspetti che caratterizzano una data regione.

9.7.1 Le basi della localizzazione

Ci sono alcuni aspetti da considerare per la personalizzazione della localizzazione ed il supporto della propria lingua nazionale.

Localizzare la tastiera

Debian è distribuito con le keymap per quasi due dozzine di tastiere. In Woody, riconfigurate la tastiera con:

- `dpkg-reconfigure --priority=low console-data # console`
- `dpkg-reconfigure --priority=low xserver-xfree86 # XF4`
- `dpkg-reconfigure --priority=low xserver-common-v3 # XF3`

Localizzare i file di dati

La gran parte dei pacchetti software Debian supporta il trattamento dei dati formati da caratteri non-US-ASCII attraverso la variabile d'ambiente LC_CTYPE offerta dalla tecnologia **locale** in glibc

- 8-bit clean: praticamente tutti i programmi
- altri set di caratteri "Latin" (p.es. ISO-8859-1 o ISO-8859-2): la maggioranza dei
- linguaggi multibyte tipo Cinese, Giapponese o Coreano: molte applicazioni recenti

Localizzare lo schermo

X è in grado di mostrare qualsiasi coding, compreso UTF-8 e supporta tutti i font. La lista comprende non solo tutti i font a 8-bit, ma anche quelli a 16-bit, come il Cinese, Giapponese o Coreano. Il metodo di inserimento dei caratteri multibite è supportato dal meccanismo XIM.

Vedere 'Esempio per un sistema multilingue per X window system' a pagina 174 e 'Supporto UTF-8 per l'emulatore di terminale in X' a pagina 178.

La visualizzazione del codice Giapponese EUC è anche disponibile nella console grafica (S)VGA, tramite il pacchetto `kon2`. Esiste anche un nuovo display giapponese alternativo, `jfbterm` che usa la console framebuffer. In queste console, il metodo di inserimento dei caratteri giapponesi deve essere fornito dalla applicazione. Usate il pacchetto `egg` per Emacs ed il pacchetto giapponesizzato `jvim` per Vim.

L'installazione di font non Unicode in X aiuta a mostrare i documenti con qualsiasi condifica. Quindi, non preoccupatevi tanto della codifica dei font.

Localizzare i messaggi e la documentazione

Traduzioni esistono per molti dei messaggi di testo e documenti che vengono visualizzati nel sistema Debian, tipo messaggi di errore, output dei programmi, menu e pagine man. Al momento il supporto per le pagine di manuale in Tedesco, Spagnolo, Finlandese, Francese, Ungherese, Italiano, Giapponese, Coreano, Polacco, Portoghese, Cinese e Russo, viene fornito attraverso i pacchetti `manpages-`*`LANG`* (dove *LANG* è una lista separata da virgole dei codici nazione ISO (di due lettere). Usate `apt-cache search manpages-|less` per avere una lista delle pagine di manuale Unix disponibili.)

Per accedere ad una pagina di manuale NLS, l'utente deve impostare la variabile d'ambiente LC_MESSAGES alla stringa appropriata. Per esempio, in caso di pagine di manuale in Italiano, LC_MESSAGES dovrà essere impostata a `it`. Il programma `man` cercherà quindi le pagine in Italiano sotto `/usr/share/man/it/`.

9.7.2 I locale

Debian supporta la tecnologia **locale**. Locale è un meccanismo che permette ai programmi di fornire un valido output e funzionalità in accordo con le convenzioni locali, tipo il set di caratteri, il formato per data e ora, il simbolo di valuta, e così via. Usa delle variabili d'ambiente per determinare il comportamento più appropriato. Per esempio, assumendo che abbiate installati i locale per L'Inglese Americano ed il Tedesco sul vostro sistema, i messaggi di errore di molti programmi potrebbero essere bilingue:

```
$ LANG="en_US" cat foo
cat: foo: No such file or directory
$ LANG="de_DE" cat foo
cat: foo: Datei oder Verzeichnis nicht gefunden
```

Glibc offre il supporto per locale ai programmi sotto forma di libreria. Vedere `locale(7)`.

9.7.3 Introduzione ai locale

Una descrizione completa di locale consiste in 3 parti: `xx_YY.ZZZZ`.

- **`xx`**: ISO 639 codici lingua (minuscolo)
- **`YY`**: ISO 3166 codici nazione (maiuscolo)
- **`ZZZZ`**: codeset, cioè set di caratteri od identificatore di codifica.

Per i codici lingua e nazione, vedere una descrizione pertinente in `info gettext`.

Notate che la parte codeset può venire normalizzata internamente per ottenere una compatibilità cross platform rimuovendo tutti i − e convertendo tutti i caratteri in minuscolo. Codeset tipici sono:

- **UTF-8**: Unicode per tutte le regioni, principalmente in Ottetti 1-3 (il nuovo standard di fatto)
- **ISO-8859-1**: western Europe (di fatto il vecchio standard)
- **ISO-8859-2**: eastern Europe (Bosnian, Croatian, Czech, Hungarian, Polish, Romanian, Serbian, Slovak, Slovenian)
- **ISO-8859-3**: Maltese
- **ISO-8859-5**: Macedonian, Serbian
- **ISO-8859-6**: Arabic
- **ISO-8859-7**: Greek
- **ISO-8859-8**: Hebrew
- **ISO-8859-9**: Turkish
- **ISO-8859-11**: Thai (=TIS-620)
- **ISO-8859-13**: Latvian, Lithuanian, Maori
- **ISO-8859-14**: Welsh
- **ISO-8859-15**: western Europe con euro
- **KOI8-R**: Russian
- **KOI8-U**: Ukrainian
- **CP1250**: Czech, Hungarian, Polish (MS Windows origin)
- **CP1251**: Bulgarian, Byelorussian (MS Windows origin)
- **eucJP**: Unix style Japanese (=ujis)
- **eucKR**: Unix style Korean
- **GB2312**: Unix style Simplified Chinese (=GB, =eucCN) for zh_CN
- **Big5**: Traditional Chinese for zh_TW
- **sjis**: Microsoft style Japanese (Shift-JIS)

Per quanto riguarda il gergo del sistema base di codifica:

- **ASCII**: 7 bit (0-0x7f)
- **ISO-8859-?**: 8 bit (0-0xff)
- **ISO-10646-1**: Universal Character Set (UCS) (31 bit, 0-0x7fffffff)
- **UCS-2**: Primi 16 bit di UCS, 2 Ottetti pieni (Unicode: 0-0xffff)
- **UCS-4**: UCS 4 Ottetti pieni (UCS: 0-0x7fffffff)
- **UTF-8**: UCS codificcato in 1-6 Ottetti (principalmente in 3 Ottetti)
- **ISO-2022**: 7 bit (0-0xff) con la sequenza di escape ISO-2022-JP è la codifica più popolare per la posta elettronica in Giapponese.
- **EUC**: Combinazione 8 bit + 16 bit (0-0xff), Unix style
- **Shift-JIS**: Combinazione 8 bit + 16 bit (0-0xff), Microsoft style.

ISO-8859-?, EUC, ISO-10646-1, UCS-2, UCS-4, ed UTF-8 condividono lo stesso codice con ASCII per i caratteri a 7 bit. EUC o Shift-JIS usa caratteri high-bit (0x80-0xff) per indicare che parte della codifica è a 16 bit. Anche UTF-8 usa caratteri high-bit (0x80-0xff) per indicare byte con sequenze di caratteri non a 7 bit, e questo è il sistema di codifica più sano per gestire i caratteri non-ASCII.

Notate la differenza nell'ordine dei byte della implementazione Unicode:

- **Standard UCS-2, UCS-4**: big endian
- **Microsoft UCS-2, UCS-4**: little endian per ix86 (macchina-dipendente)

Vedere 'Convertire un file di testo con `recode`' a pagina 119 per la conversione fre i vari set di caratteri. Per maggiori informazioni, vedere Introduzione a i18n (`http://www.debian.org/doc/manuals/intro-i18n/`).

9.7.4 Attivare le capacità di supporto locale

Debian **non** arriva con tutti i locale disponibili precompilati. Controllate `/usr/lib/locale` per vedere quali (oltre a quello predefinito, "C") sono compilati sul vostro sistema. Se quello di cui avete bisogno non è presente, avete due opzioni:

- Modificare `/etc/locale.gen` per aggiungere il locale desiderato, poi lanciate `locale-gen` da root per compilarlo. Vedere `locale-gen(8)` e le pagine di manuale elencate nella sua sezione "SEE ALSO".
- Lanciare `dpkg-reconfigure locales` per riconfigurare il pacchetto `locales`. Oppure, se non è stato ancora installato, l'installazione di `locales` invocherà l'interfaccia debconf per farvi scegliere il locale richiesti e compilare il database.

9.7.5 Attivare un locale in particolare

Le seguenti variabili d'ambiente vengono valutate in questo ordine per fornire ai programmi particolari valori di locale:

1 LANGUAGE: Consiste di una lista di nomi di locale, separati da una virgola, in ordine di priorità. Usata solo se il locale POSIX è impostato ad un valore diverso da "C" [in Woody; la versione Potato ha sempre priorità sul locale POSIX]. (Estensione GNU)

2 LC_ALL: Se non è nullo, il valore è usato per tutte le categorie di locale. (POSIX.1) Di solito "" (nullo).

3 LC_*: Se non è nullo, il valore è utilizzato per la categoria corrispondente (POSIX.1). Di solito "C".

Variabili LC_* sono:

- LC_CTYPE: Classificazione e conversione dei caratteri.
- LC_COLLATE: Ordine di collazione.
- LC_TIME: Formati data e ora.
- LC_NUMERIC: Formati numerici non monetari.
- LC_MONETARY: Formati di valuta.

- LC_MESSAGES: Formati dei messaggi informativi e diagnostici, nonchè delle risposte interattive.
- LC_PAPER: Formato carta.
- LC_NAME: Formati dei nomi.
- LC_ADDRESS: Formati degli indirizzi e località.
- LC_TELEPHONE: Formati dei numeri telefonici.
- LC_MEASUREMENT: Unità di misura (Sistema Metrico decimale od Altri).
- LC_IDENTIFICATION: Metadati sulle informazioni di locale.

4 LANG: Se non è nullo e se LC_ALL non è definito, il valore è usato per tutte le categorie di locale LC_* con valori non definiti. (POSIX.1) Di solito "C".

Notate che alcune applicazioni (p.es., Netscape 4) ignorano le impostazioni LC_*.

Il programma `locale` può mostrare le impostazioni attive ed i locale disponibili; vedere `locale(1)`. (NOTA: `locale -a` elenca tutti i locale che il vostro sistema conosce; questo *non* significa che tutti siano compilati! Vedere 'Attivare le capacità di supporto locale' nella pagina precedente.)

9.7.6 Formato data ISO 8601

Il supporto locale per lo standard internazionale di data `aaaa-mm-gg` (formato data ISO 8601) è fornito dal locale chiamato `en_DK`, "English in Denmark" che è una sorta di gioco di parole :-) Sembra funzionare solo in console per `ls`.

9.7.7 Esempio per US (ISO-8859-1)

Aggiungete le righe seguenti in `~/.bash_profile`:

```
LC_CTYPE=en_US.ISO-8859-1
export LC_CTYPE
```

9.7.8 Esempio per l'Italiano con euro (ISO-8859-15)

Aggiungete le righe seguenti in `~/.bash_profile`:

```
LANG=it_IT@euro
export LANG
LC_CTYPE=it_IT@euro
export LC_CTYPE
```

Configurate la tastiera per l'Italiano "QWERTY" come descritto in 'Localizzare la tastiera' a pagina 169. ed aggiungete le pagine di manuale in Italiano installando `manpages-it`. Il tasto

Right-Alt degli USA è Alt-Gr in Europa. Premedolo con altri tasti crea molti caratteri speciali. Per esempio, Alt-Gr+E crea il segno dell'euro.

Molti dei linguaggi dell'Europa occidentale possono essere configurati allo stesso modo.

Vedere Debian Euro HOWTO (`http://www.debian.org/doc/manuals/debian-euro-support/`) per aggiungere il supporto per l'euro e Utiliser et configurer Debian pour le français (`http://www.debian.org/doc/manuals/fr/debian-fr-howto/`) per maggiori dettagli per il Francese (NdT: no, per l'Italiano non mi risulta un equivalente).

9.7.9 Esempio per un sistema multilingue per X window system

Impostiamo un sistema multilingue che supporti simultaneamente Giapponese, Inglese Tedesco e Francese con codifiche EUC, UTF-(ed ISO-8859-1 sulle diverse console.

Vi mostrerò una personalizzazione del Debian menu system. Vedere i dettagli del Debian menu system in `/usr/share/doc/menu/html/index.html`. Creerò anche una scorciatoia al browser `mozilla`. [8]

- aggiungere il supporto locale per il Giapponese ja_JP.eucJP e gli altri locale richiesti usando il metodo descritto in 'Localizzazione (l10n)' a pagina 169. (per tutti)
- installare il sistema di conversione da-Kana-a-Kanji ed il dizionario (per il Giapponese):
  - `canna` — Server locale, (licenza tipo "free beer"), oppure
  - `freewnn-jserver` — server estensibile in network (Dominio Pubblico)
- installare il metodo di input Giapponese:
  - `kinput2-canna` — per X, oppure
  - `kinput2-canna-wnn` — per X,e
  - `egg` — lavora direttamente con Emacsen persino in console (opzionale)
- Installate i terminali compatibili (per tutti)
  - `xterm` – X (per ISO-8859-1 ed UTF-8),
  - `kterm` — X (per Giapponese EUC) e
  - `mlterm` — X (multilingue)
- Aggiungere tutti i pacchetti di font richiesti (per tutti)
- create `~/.xsession` che imposta l'ambiente X specifico per l'utente: come descritto in 'Personalizzare le X session' a pagina 151. (per tutti):

```
#!/bin/sh
# Questo fa funzionare X quando lancio su verso l'account root.
if [ -z "$XAUTHORITY" ]; then
```

---

[8] In questo esempio vengono usate due soluzioni ai bug della versione 2003 di `blackbox`. Io uso `sh -c` nel comando. Nemmeno la voce `~/.menu/*` viene usate, ma avendone bisogno root, si è invece usata `/etc/menu/*`.

```
        XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi

# Set specific environment through debian menu system.
# Reset locale
unset LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
unset LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
unset LC_IDENTIFICATION LC_ALL LANG LANGUAGE PAGER
# set locale default in X
LANG=C
# export locale
export LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
export LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
export LC_IDENTIFICATION LC_ALL LANG LANGUAGE PAGER
###
# activate input method for Japanese with kinput2
kinput2 &
XMODIFIERS="@im=kinput2"
export XMODIFIERS
# How about blackbox window manager (lightweight)
exec blackbox
#exec xfwm
#exec wmaker
```

- impostate locale in `~/.bash_profile` per le console Linux (per tutti).
- rimuovete le impostazioni di locale da `~/.bashrc` se esistenti (per tutti).
- create alcuni file in `/etc/menu/` (per tutti)
  - `/etc/menu/xterm-local`: (aggiungete nuove voci al menu) [9]

```
?package(xterm):\
 needs=x11\
 section=XShells\
 longtitle="XTerm: terminal emulator (en_US.ISO-8859-1)"\
 title="XTerm (en_US.ISO-8859-1)"\
 command="sh -c 'LC_ALL=en_US.ISO-8859-1 xterm'"
?package(xterm):\
 needs=x11\
 section=XShells\
 longtitle="XTerm: terminal emulator (de_DE.ISO-8859-1)"\
 title="XTerm (de_DE.ISO-8859-1)"\
 command="sh -c 'LC_ALL=de_DE.ISO-8859-1 xterm -T xterm-de'"
?package(xterm):\
```

---

[9] Usate nomi dei file che non sono uguali a nessun nome di pacchetto.

```
            needs=x11\
            section=XShells\
            longtitle="XTerm: terminal emulator for X with Unicode support (Japa
            title="UXTerm (ja_JP.UTF-8)"\
            command="sh -c 'LC_ALL=ja_JP.UTF-8 uxterm'"
```

  - `/etc/menu/kterm`: (scavalca le impostazioni predefinite) [10]

```
           ?package(kterm):\
            needs="x11"\
            section="XShells"\
            command="sh -c 'LC_ALL=ja_JP.eucJP PAGER=w3m /usr/X11R6/bin/kterm -x
            title="Kanji Terminal"
           ?package(kterm):\
            needs="x11"\
            section="XShells"\
            command="sh -c 'LANG=ja_JP.eucJP \
                      LC_MESSAGES=en_US.ISO-8859-1 PAGER=w3m /usr/X11R6/bin/kterm
            title="Kanji Terminal (bilingal)"
```

  - `/etc/menu/mozilla-local`: (aggiungete una nuova scorciatoia) [11]

```
           ?package(mozilla-browser):needs="x11" section="/" \
                   title=" Mozilla Navigator" command="mozilla-1.5" hints="Web b
                   icon=/usr/share/pixmaps/mozilla.xpm
```

  - lanciate `update-menus` dall'account di root.

- aggiungete le righe seguenti in `~/.muttrc` (per il Giapponese):

```
# il supporto UTF-8 non è popolare nell'ambiente EMACS Giapponese l'encodi
# 7 bit di iso-2022-jp è più semplice per tutti
# default encoding order = us-ascii --> iso-8859-1 --> iso-2022-jp
set send_charset="us-ascii:iso-8859-1:iso-2022-jp
set allow_8bit=no
```

- attiva XIM `kinput2` per applicazioni X (per il Giapponese)
  - aggiungete `*inputMethod: kinput2` e `KTerm*VT100*OpenIM:true` al vostro file `~/.Xresources` (sembra che Debian se ne prenda carico automaticamente, in qualche modo).
  - Alcune applicazioni (tipo `mlterm`) permettono anche di impostare `*inputMethod:` ed altre informazioni in maniera dinamica all'avvio (premete **Ctrl-MouseButton-3** in `mlterm`).

- lanciate X digitando `startx` o da uno dei display manager (`xdm`, `gdm`, `kdm`, `wdm`, . . .) (per tutti)

---

[10] Usate nomi dei file che corrispondono ai nomi dei pacchetti

[11] La slash in `section="/"` abilita le vosi nel menu iniziale e lo spazio iniziale in `title=" Mozilla Navigator"` pone la voce in cima alla lista.

- lanciate l'applicazione compatibile con il Giapponese, tipo Vim 6, (x)emacs21, mc-4.5, mutt-1.4, . . . in `kterm` (per il Giapponese). (Emacs sembra la piattaforma più popolare, sebbene io non la usi.)
- premete **Maiusc+Spazio** per attivare e disattivare il modo input modo Giapponese.
- legget la pagine di manuale localizzata lanciandone il comando in una console localizzata (per tutti).

Per altro supporto per il linguaggio CJK, vedere le sezioni seguenti ed anche le pagine SuSE per CJK (`http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html`).

9.7.10 Metodi alternativi di input in X

Molti i pacchetti disponibili che supprtano metodi alternativi di input in X:

```
LIngua      LC_CTYPE      XIM server XMODIFIERS                Tasto per il lancio
Japanese    ja_JP*        kinput2    "@im=kinput2"             Maiusc-Spazio
Korean      ko_KR*        ami        "@im=Ami"                 Maiusc-Spazio
Chinese(T)  zh_TW.Big5    xcin       "@im=xcin-zh_TW.big5"     Ctrl-Spazio
Chinese(S)  zh_CN.GB2312  xcin       "@im=xcin-zh_CN.GB2312"   Ctrl-Spazio
```

Il metodo Giapponese, `kinput2` è offerto dai pacchetti `kinput2-canna-wnn`, `kinput2-canna` e `kinput2-wnn`. Il metodo Giapponese necessita di un server per il dizionario, tipo `canna` e `freewnn-jserver` per essere pratico.

9.7.11 Emulatori di terminale in X

Sono molte le console in X che supportano la semplice codifica a 8 bit quando i pacchetti dei font pertinenti sono installati:

- `xterm` – L'X terminal emulator
- `gnome-terminal` – `xterm` per Gnome
- `konsole` – `xterm` per KDE
- `rxvt` – terminale VT102 (leggero)
- `aterm` – VT102 per Afterstep WM
- `eterm` – VT102 per Enlightment WM
- `wterm` – VT102 per WindowMaker WM

Il supporto per la codifica multi-byte vine fornito da `xterm` tramite la codifica UTF-8 ('Supporto UTF-8 per l'emulatore di terminale in X' nella pagina successiva). Altri metodi di supporto tradizionali sono in fase di lavorazione (al 2003). I seguenti pacchetti offrono supporti tradizionali:

- `aterm-ml` – Multi-lingue
- `kterm` – Multi-lingue (Giapponese, . . .)

- `rxvt-ml` – Multi-lingue
- `wterm-ml` – Multi-lingue
- `cxterm-big5` – Cinese (Trad., Big5)
- `cxterm-gb` – Cinese (Simp., GB)
- `cxterm-ks` – Cinese (KS)
- `cxterm-jis` – Giapponese
- `hanterm-classic` – Coreano (Hangul)
- `hanterm-xf` – Coreano (Hangul)
- `hztty` – Cinese (GB, Big5, zW/HZ)

Per `kterm` (e possibilmente altri), potreste voler attivare XIM da menu dopo aver premuto Ctrl-tasto-centrale del mouse.

9.7.12 Supporto UTF-8 per l'emulatore di terminale in X

Il supporto UTF-8 per l'emulatore di terminale viene fornito dal programma `uxterm` del pacchetto `xterm` per XFree86 4.x. Abilita il supporto per tutte le lingue. E' un wrapper di `xterm(1)` che invoca quest'ultimo con impostata la X resource class "UXTerm".

Per esempio, per abilitare la bella visione ingrandita dei caratteri Inglese, Russo, Giapponese, Cinese e Coreano, aggiungete quanto segue al vostro `~/.Xresources` dopo aver installato i font pertinenti:

```
! set large font
UXTerm*font: -misc-fixed-medium-r-normal-*-18-120-100-100-c-90-iso10646-1
! Use XIM for Japanese
*inputMethod: kinput2
```

Poi eseguite `xrdb -merge ~/.Xresources` per aggiornare le X resources come descritto in 'X resources' a pagina 155.

Sebbene gran parte dei pacchetti dei programmi di console popolari, come `vim`, `mutt`, e `emacs` sono stati resi compatibili con UTF-8 di recente (Woody-Sarge), un programma come `mc` ancora non lo è, ma semplicemente è 8-bit clean. Se state lavorando sulla parte a 7 bit ASCII di un file dalla codifica sconosciuta o mista, è più sicuro usare un editor locale ignaro dell'8-bit clean.

Vedere The Unicode HOWTO (`http://www.tldp.org/HOWTO/Unicode-HOWTO.html`).

9.7.13 Esempio per UTF-8 in console framebuffer

Il supporto UTF-8 sulla console FB è fornito da `bterm` usato nel `debian-installer`.

9.7.14 Oltre i locale

Quando state impostando il sistema per la prima volta per un **linguaggio nazionale**, prendete in considerazione l'uso di `tasksel` o di `aptitude` per scoprire quali pacchetti vengono selezionati scegliendo il task corrispondente al linguaggio prescelto. La scelta dei pacchetti fatta in tal modo è utile persino per un setup multilingua. Se vi imbattete in qualche conflitto di dipendenze durante l'installazione del vostro raffinato sistema, evitate di installare i programmi che sono in conflitto col sistema esistente. Potreste dover utilizzare `update-alternative` per riguadagnare lo stato originale per alcuni comandi, dato che quelli puù recentemente installati potrebbero avere priorità più elevate rispetto a quelli preesistenti.

I programmi principali più recenti usano glibc 2.2 e sono molto internazionalizzati. Quindi, versioni localizzate in maniera particolare, come `jvim` per VIM possono non essere necessarie e la sua funzionalità è offerta da `vim` versione 6.0 in X. In realtà, è ancora da raffinare. Dato che `jvim` ha una versione compilata con il supporto diretto per il Giapponese IM (`canna`) ha il supporto persino in console ed è indirizzato verso molte altre caratteristiche giapponesi, ed è maturo, potreste ancora volerlo :-)

I programmi potrebbero avere la necessità di essere configurati oltre la configurazione `locale`, per fornire un ambiente di lavoro confortevole. Il pacchetto `language-env` ed il suo comando `set-language-env` facilitano di molto questo processo.

Vedere anche il documento sulla internazionalizzazione, Introduction to i18n (`http://www.debian.org/doc/manuals/intro-i18n/`). E' indirizzato agli sviluppatori, ma è anche utile per gli amministratori di sistema.

9.8 Multilingualizzazione (m17n)

'Localizzazione (l10n)' a pagina 169 abilitato dal pacchetto `language-env` e simili, è mirato ad ottenere una localizzazione monolingua. Questi pacchetti usano anche codifiche tradizionali come scelta per la codifica del testo. Non potete mescolare in un tale ambiente testo in Francese e Giapponese, poiche utilizzano codifiche incompatibili ISO-8859-1 ed EUC-JP, rispettivamente.

Potete ottenere un Desktop multilingua UTF-8 usando GNOME e KDE eseguiti sotto uno dei locale UTF-8 disponibili. (Sarge) In tale ambiente potrete mescolare caratteri inglesi, cinesi, russi e giapponesi sotto programmi complianti con UTF-8.

`m17n-env` è uno script di aiuto per l'impostazione di un tale ambiente. L'ambiente multilingua UTF-8 viene configurato eseguendo il comando `set-m17n-env` dagli account di root ed utente. [12]

In un tale ambiente, il nuovo metodo di input multilingualizzato (IM) che usa `scim` è da preferirsi. IM offerto da `scim` viene attivato e disattivato premendo Ctrl-spazio insieme. Il motore di conversione dell'input può essere cambiato cliccando il piccolo pannello SCIM.

Potete ancora avere un facile accesso all'ambiente di codifica tradizionale attraverso le console locale personalizzate, create da `m17n-env`. Ciò torna utile in caso di modifiche a vecchi file con codifica EUC-JP o ISO-8859-1.

---

[12] Il pacchetto `language-env` non è molto utile in un ambiente multilingualizzato.

Capitolo 10

Configurare la rete

Questo capitolo è focalizzato sull'amministrazione di rete in Debian. per un'introduzione generale alla rete in GNU/Linux, leggete il Net-HOWTO (`http://www.tldp.org/HOWTO/Net-HOWTO/index.html`).

Per permettere ad un host Debian di accedere ad Internet, le sue interfacce di rete devono essere supportate dal kernel e propriamente configurate.

Il primo requisito è il supporto del kernel per i dispositivi di rete, tipo schede Ethernet, Wi-Fi e modem. Per ottenere detto supporto potreste dover ricompilare il kernel, o aggiungere moduli ad esso, come descritto in 'Il kernel Linux su Debian' a pagina 99.

La configurazione dei dispositivi di rete è spiegata sotto. Le informazioni di questo capitolo sono state aggiornate per Sarge. Molte di esse non si applicano a distribuzioni precedenti.

10.1 Le basi dell'IP networking

Un host Debian può avere parecchie interfacce, ciscuna con il proprio indirizzo IP (Internet Protocol). Le interfacce possono essere di tipi differenti, inclusi:

- Loopback: `lo`
- Ethernet: `eth0,eth1,...`
- Wi-Fi: `wlan0,wlan1,...` [1]
- Token Ring: `tr0,tr1,...`
- PPP: `ppp0,ppp1,...`

Esiste una vasta gamma di altri dispositivi di rete disponibili, inclusi SLIP, PLIP (linea seriale e parallela IP), dispositivi "shaper" per controllare il traffico su certe interfacce, frame relay, AX.25, X.25, ARCnet, e LocalTalk.

---

[1] Notate che un'interfaccia Wi-Fi è in realtà uno pseudonimo per un'interfaccia Ethernet che fornisce l'accesso ai parametri di configurazione specifici per Wi-Fi. Questi parametri vengono controllati mediante il programma `iwconfig`.

Ciascuna interfaccia connessa direttamente ad Internet (o a qualsiasi rete basata su IP) viene identificata da un indirizzo unico a 32 bit. [2] L'indirizzo Ip può essere diviso in una parte relativa alla rete e ad una parte relativa all'host. Se prendete un indirizzo IP, impostate ad 1 i bit relativi alla rete e a 0 quelli relativi all'host, ottenete il net mask della rete.

tradizionalmente, le reti IP sono raggruppate in classi, le cui parti di indirizzo sono lunghe 8, 16 o 24 bit. [3]

	Indirizzi IP	net mask	lunghezza
Classe A	1.0.0.0 - 126.255.255.255	255.0.0.0	= /8
Classe B	128.0.0.0 - 191.255.255.255	255.255.0.0	= /16
Classe C	192.0.0.0 - 223.255.255.255	255.255.255.0	= /24

Gli indirizzi che non sono in questi intervalli vengono usati per scopi speciali.

Esistono intervalli in ciascuna classe che sono riservati per l'uso nelle local area network (LAN). Questi indirizzi sono garantiti non entrare in conflitto con qualsiasi indirizzo di Internet propriamente detto. (Per lo stesso ragionamento, se uno di essi viene assegnato ad un host, allora detto host non deve accedere direttamente ad Internet, ma attraverso un gateway che agisca come proxy per i singoli servizi od altrimenti esegua Network Address Translation.) Questi intervalli vengono forniti nella tabella seguente, insieme al numero di intervalli per ciascuna classe.

	indirizzi	lunghezza	quantità
Class A	10.x.x.x	/8	1
Class B	172.16.x.x - 172.31.x.x	/16	16
Class C	192.168.0.x - 192.168.255.x	/24	256

Il primo indirizzo di una rete IP è l'indirizzo della rete stessa. L'ultimo è l'indirizzo di broadcast per la rete. [4] Tutti gli altri indirizzi della rete possono essere allocati agli host. Di questi, il primo o l'ultimo indirizzo vengono in genere allocati al gateway Internet per la rete.

La tablla d'instradamento contiene le informazioni del kernel su come inviare i pacchetti IP alle loro destinazioni. Ecco un esempio di tabella per un host Debian su una LAN con indirizzo IP 192.168.50.x/24. L'host 192.168.50.1 (anch'esso sulla LAN) è un router per la rete aziendale 172.20.x.x/16 e l'host 192.168.50.254 (anch'esso sulla LAN) è un router per Internet.

```
# route
Kernel IP routing table
Destination    Gateway          Genmask        Flags Metric Ref Use Iface
```

---

[2] Ciò è vero se si usa la versione 4 di IP. In IPv6 gli indirizzi sono a 128 bit. Vedere `http://www.ipv6.org/`.

[3] Questo sistema era rigido e portava alla perdita di molti indirizzi IP, così oggi le reti IPv4 sono allocate con indirizzi di lunghezza variabile.

[4] L'indirizzo della rete si può ottenere mediante la funzione AND bit a bit eseguita su un indirizzo della rete e la net mask. Quello di broadcast, invece, mediante la funzione OR bit a bit di un indirizzo di rete con il complemento a 1 della net mask.

```
127.0.0.0      *              255.0.0.0     U     0      0     2 lo
192.168.50.0   *              255.255.255.0 U     0      0   137 eth0
172.20.0.0     192.168.50.1   255.255.0.0   UG    1      0     7 eth0
default        192.168.50.254 0.0.0.0       UG    1      0    36 eth0
```

- La prima riga dopo l'intestazione dice che il traffico destinato alla rete `127.x.x.x` verrà reindirizzato attraverso `lo`, l'interfacia di loopback.
- La seconda riga dice che il traffico destinato agli host sulla LAN verrà reindirizzato attraverso `eth0`.
- La terza dice che il traffico destinato alla ret aziendale verrà reindirizzato verso il gateway `192.168.50.1`, sempre tramite `eth0`.
- La quarta dice che il traffico destinato ad Internet verrà reindirizzato verso il gateway `192.168.50.254`, sempre tramite `eth0`.

Gli indirizzi IP nella tabella possono apparire anche come nomi, ottenuti guardando gli indirizzi in `/etc/networks` o usando il resolver della Libreria C.

In aggiunta al reindirizzamento, il kernel può eseguire network address translation, traffic shaping e filtering.

Vedere il Net-HOWTO (`http://www.tldp.org/HOWTO/Net-HOWTO/index.html`) ed altri HOWTO sul networking (`http://www.tldp.org/HOWTO/Networking-Overview-HOWTO.html`) per maggiori informazioni di base.

10.2 Configurazione della rete a basso livello

Gli strumenti tradizionali per la configurazione a basso livello dei sistemi GNU/Linux, sono `ifconfig` e `route`, che sono forniti dal pacchetto `net-tools`. Essi sono stati ufficialmente soppiantati da `ip`, fornito dal pacchetto `iproute`. Il programma `ip` funziona con Linux 2.2 ed oltre ed è più valido degli altri strumenti. Comunque, questi ultimi ancora funzionano e sono più familiari a molti utenti.

10.2.1 Configurazione a basso livello – `ifconfig` and `route`

Ecco un'illustrazione su come modificare l'indirizzo IP dell'interfaccia `eth0` da `192.168.0.3` a `192.168.0.111` e per rendere `eth0` il route alla rete `10.0.0.0` via `192.168.0.1`. Iniziamo lanciando `ifconfig` e `route` senza argomenti, per mostrare lo stato attuale di tutte le interfacce di rete e del reindirizzamento.

```
# ifconfig
eth0 Link encap:Ethernet  HWaddr 08:00:46:7A:02:B0
     inet addr:192.168.0.3  Bcast:192.168.255.255  Mask:255.255.0.0
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:23363 errors:0 dropped:0 overruns:0 frame:0
     TX packets:21798 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:100
          RX bytes:13479541 (12.8 MiB)  TX bytes:20262643 (19.3 MiB)
          Interrupt:9

     lo   Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:230172 errors:0 dropped:0 overruns:0 frame:0
          TX packets:230172 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:22685256 (21.6 MiB)  TX bytes:22685256 (21.6 MiB)
     # route
     Kernel IP routing table
     Destination  Gateway      Genmask          Flags Metric Ref Use Iface
     192.168.0.0  *            255.255.0.0      U     0      0     0 eth0
     default      192.168.0.1  255.255.255.255  UG    0      0     0 eth0
```

Prima disattiviamo l’interfaccia.

```
# ifconfig eth0 inet down
# ifconfig
lo   Link encap:Local Loopback
  ... (la voce eth0 è sparita)
# route
  ... (le voci della tabella di reindirizzamento sono scomparse)
```

Poi, riattiviamola con il nuovo indirizzo IP e nuovo reindirizzamento.

```
# ifconfig eth0 inet up 192.168.0.111 \
           netmask 255.255.255.0 broadcast 192.168.0.255
# route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.1 dev eth0
```

Il risultato:

```
# ifconfig
eth0 Link encap:Ethernet  HWaddr 08:00:46:7A:02:B0
     inet addr:192.168.0.111  Bcast:192.168.0.255  Mask:255.255.255.0
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     ...

lo   Link encap:Local Loopback
     inet addr:127.0.0.1  Mask:255.0.0.0
     ...
# route
```

```
Kernel IP routing table
Destination  Gateway      Genmask        Flags Metric Ref Use Iface
192.168.0.0  *            255.255.255.0  U     0      0     0 eth0
10.0.0.0     192.168.0.1  255.0.0.0      UG    0      0     0 eth0
```

Per maggiori informazioni, vedere `ifconfig(8)` e `route(8)`.

10.2.2 Configurazione a basso livello – `ip`

I comandi equivalenti per `ip` dei precedenti per `ifconfig` e `route` sono:

- `ip link show`
- `ip route list`
- `ip link set eth0 down`
- `ip addr del dev eth0 local 192.168.0.3`
- `ip addr add dev eth0 local 192.168.0.111/24 broadcast 192.168.0.255`
- `ip link set eth0 up`
- `ip route add dev eth0 to 10.0.0.0/8 src 192.168.0.111 via 192.168.0.1`

`Ip` stampa la sintassi dei suoi comandi se lanciato con l'argomento `help`. Per esempio, `ip link help` mostra:

```
Usage: ip link set DEVICE { up | down | arp { on | off } |
                     dynamic { on | off } |
                     multicast { on | off } | txqueuelen PACKETS |
                     name NEWNAME |
                     address LLADDR | broadcast LLADDR |
                     mtu MTU }
       ip link show [ DEVICE ]
```

Vedere anche `ip(8)`.

10.2.3 Configrazione di un'interfaccia Wi-Fi.

Per le interfacce Wi-Fi, si usa il programma `iwconfig`, che è fornito dal pacchetto `wireless-tools` in aggiunta a `ifconfig` o `ip`.

Vedere `iwconfig(8)`.

10.2.4 Configurare un'interfaccia PPP

Se accedete ad Internet attraverso un modem connesso alla linea telefonica analogica, allora la connessione viene negoziata mediante il Protocollo Punto-a-Punto (PPP). Si accede a tali connessioni come interfacce di rete `ppp0`, `ppp1` e così via.

L'interfaccia PPP viene gestita dal demone `pppd`, fornito dal pacchetto `ppp`. Ergo, per l'utente, configurare l'interfaccia PPP significa configurare `pppd`.

Configurare `pppd` manualmente

Affinchè una connessione di rete venga stabilita, si deve aprire una porta di comunicazione (in genere seriale), devono essere dati dei comandi ad un dispositivo di comunicazione (in gener un modem), si deve fare il numero telefonico, ci si deve autenticare presso un demone PPP esterno, si deve creare un'interfacia PPP e le tabelle di reindirizzamento devono essere modificate così che il traffico venga inviato sulla connessione. `pppd` fa tutto ciò e, di conseguenza, ha un elenco molto lungo di opzioni operative. Esse sono descritte in `pppd(8)`.

Su un sistema Debian, le opzioni globali vengono impostate in `/etc/ppp/options`. Quelle specifiche per l'utente in `~/.ppprc`. Quelle che dipendono dalla porta utilizzata vengono immagazzinate in `/etc/ppp/options.`*`nomeporta`*. Per esempio, ammettiamo che abbiate due modem—uno interno, Lucent LT modem, a cui si accede tramite /dev/LT-modem ed uno esterno, a cui si accede tramite /dev/ttyS0. Create i seguenti file di opzioni.

```
# cat > /etc/ppp/options.LT-modem <<EOF
115200
init "/usr/sbin/chat -f /etc/chatscripts/setup-LT-modem"
EOF
# cat > /etc/ppp/options.ttyS0 <<EOF
115200
init "/usr/sbin/chat -f /etc/chatscripts/setup-ttyS0"
EOF
```

Essi fanno riferimento ai seguenti script chat. Il primo, `/etc/chatscripts/setup-LT-modem`.

```
ABORT ERROR
'' ATZ
OK 'ATW2X2 S7=70 S11=55'
OK AT
```

Il secondo `/etc/chatscripts/setup-ttyS0`.

```
ABORT ERROR
'' ATZ
OK 'ATL1M1Q0V1W2X4&C1&D2 S6=4 S7=70 S11=55 S95=63 S109=1 +FCLASS=0'
OK AT
```

Il contenuto di questi file dipende dal vostro hardware, naturalmente.

Le opzioni possono essere anche date come argomenti di `pppd`.

In Debian `pppd` vien lanciato in genere dal comando `pon`. Quando si usa `pon`, il primo argomento chiama un file in `/etc/ppp/peers/` che viene letto anche da `pppd`. [5] Qui è dove impostate le opzioni specifiche per un particolare peer—per esempio un particolare Internet Service Provider (ISP).

Supponiamo, per esempio, che viaggiate tra Amstrdam e Den Haag. In ciascuna città avete accesso a due ISP—Planet e KPN. Per prima cosa, create un file di opzioni base per ciascun ISP.

```
# cat > /etc/ppp/peers/KPN <<EOF
remotename KPN
noauth
user kpn
noipdefault
ipparam KPN
EOF
# cat > /etc/ppp/peers/Planet <<EOF
remotename Planet
auth
user user3579@planet.nl
noipdefault
mru 1000
mtu 1000
ipparam Planet
EOF
```

Questi file impostano le opzioni che differiscono tra i due ISP. Quelle comuni ai due ISP possono essere messe in `/etc/ppp/options` o in uno dei file specifici per interfaccia, se appropriato.

Ora create i file per ciascun ISP, in ciascuna città. Nel nostro esempio l'unica differenza fra connettersi ad un ISP in un luogo e in un altro sta nel chatscript richiesto. (Essi sono diversi, poichè i numeri di accesso telefonico sono diverso.)

```
# cat > /etc/ppp/peers/KPN-Amsterdam <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/KPN-Amsterdam"
file /etc/ppp/peers/KPN
EOF
# cat > /etc/ppp/peers/KPN-DenHaag <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/KPN-DenHaag"
file /etc/ppp/peers/KPN
EOF
# cat > /etc/ppp/peers/Planet-Amsterdam <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/Planet-Amsterdam"
```

---

[5] Questo file viene incluso usando l'opzione `call`.

```
file /etc/ppp/peers/Planet
EOF
# cat > /etc/ppp/peers/Planet-DenHaag <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/Planet-DenHaag"
file /etc/ppp/peers/Planet
EOF
```

Le direttive `file` includono ciascuno dei file mostrati prima. La direttiva `connect` specifica il comando che usa `pppd` per eseguire la connessione. In genere si usa il programma `chat` per ciò, adattando il chatscript all'ISP. Ecco quelli per Den Haag; quelli per Amsterdam possono essere simili, tranne che per il numero telefonico, oppure possono differire se l'ISP offre lì i servizi attraverso un'altra compagnia.

```
# cat > /etc/chatscripts/KPN-DenHaag <<EOF
ABORT BUSY
ABORT 'NO CARRIER'
ABORT VOICE
ABORT 'NO DIALTONE'
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
ABORT ERROR
OK-AT-OK ATDT 0676012321
CONNECT \d\c
EOF
# cat > /etc/chatscripts/Planet-DenHaag <<EOF
ABORT BUSY
ABORT 'NO CARRIER'
ABORT VOICE
ABORT 'NO DIALTONE'
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
ABORT ERROR
OK-AT-OK ATDT 0676002505
CONNECT \d\c
EOF
```

Per potervi connettere a questi ISP, avete bisogno dei nomi del client e delle password che `pppd` possa fornire al peer a richiesta. Questa informazione è contenuta o in `/etc/ppp/pap-secrets` (se si usa il protocollo PAP) o in `/etc/ppp/chap-secrets` (se è CHAP). Sebbene CHAP sia più sicuro, PAP è ancora più largamente usato. Siccome questi file contengono dei dati segreti, gruppo e resto del mondo non dovrebbero avere permessi in lettura o scrittura. Il formato di questi file è spiegato in `pppd(8)`. Un "secret" (terzo campo) viene cercato nel file trovando il nome del client (primo campo) e/o il nome del server (secondo campo). Quando ci si collega ad un ISP, in genere non si conosce il nome del server, così si fornisce invece il nome del client; ciò viene fatto nelle righe `user` in `peers/KPN` e `peers/Planet` sopra.

```
# client name       server name  secret
kpn                 *            kpn
user3579@planet.nl  *            myfavoritepet
```

Vedere `/usr/share/doc/ppp/README.Debian.gz` per maggiori informazioni.

Configurare `pppd` con `pppconfig`

Un modo veloce per configurare `pppd` è tramite il programma `pppconfig`, fornito dal pacchetto omonimo. `pppconfig` imposta i file come quelli sopra, dopo aver chiesto all'utente alcune domande, tramite un'interfaccia a menu.

Configurare un'interfaccia PPP con `wvdial`

Un approccio diverso all'uso di `pppd` è di lanciarlo da `wvdial`, fornito dal pacchetto `wvdial`. Invece di far lanciare a `pppd` `chat` per aprire e negoziare la connessione, `wvdial` esegue il numero e la negoziazione iniziale, poi lancia `pppd` per fare il resto. Dando solo numero telefonico, username e password a `wvdial` si ha successo nel compiere la connessione nella maggior parte dei casi.

10.3 Dare un nome al computer

10.3.1 Hostname

Un sistema Debian ha talvolta bisogno di identificarsi tramite un nome. All'uopo, il kernel mantiene un **hostname**.

Lo script di inizializzazione `/etc/init.d/hostname.sh` imposta l' hostname al boot (usando il comando `hostname`) al nome contenuto in `/etc/hostname`. Questo file deve contenere **soslo** l'hostname, non un nome di dominio pienamente qualificato.

Per mostrare l'hostname attuale, lanciate `hostname` senza argomenti.

10.3.2 Mailname

Il **mailname** di un host è il nome che i programmi che hanno a che fare con la posta usano per identificare l'host. Il file `/etc/mailname` contiene questo nome, seguito da una riga vuota. Il mailname è in genere uno dei nomi dominio pienamente qualificati dell'host. Vedere `mailname(5)`.

Quello che il destinatario di una mail vede nel campo `From:` di un messaggio inviato dal vostro host Debian dipende da come sono configurati i vostri Mail User Agent (MUA) e Mail Transfer Agents (MTA). Supponiamo che un utente locale *`foo`* mandi un messaggio da un hos con mailname *`myhost.dom`*. Il campo `From:` della mail in uscita sarà:

- “`From: foo@myhost.dom`” se il MUA non ha il campo `From:` impostato;
- “`From: bar@myhost.dom`” se il MUA ha il campo “`From: bar`” impostato;
- “`From: bar@bogus.dom`” se il MUA ha il campo “`From: bar@bogus.dom`” impostato;

Persino quando il MUA ha un campo `From:` impostato, il MTA può aggiungere un campo “`Sender:foo@herman.dom`” per indicarne la sua vera origine.

Naturalmente, quando tutti i MTA coinvolti eseguono la riscrittura dell’indirizzo, come discusso in ‘Raccolta di tutti gli indirizzi e-mail inesistenti in Exim’ a pagina 166 e ‘Riscrivere selettivamente l’indirizzo per la posta in uscita (Exim)’ a pagina 166, l’indirizzo visto dal destinatario può essere cambiato in qualsiasi cosa.

10.4 Domain Name Service (DNS)

Gli host possono venire definiti per nome dominio e per indirizzo IP. Il DNS è un sistema client-server in cui i risolutori del nome consultano i nameserver per associare i nomi dominio con gli indirizzi IP ed altre proprietà degli host. La libreria GNU C `resolver(3)` è anch’essa in grado di cercare gli indirizzi Ip nei file o di consultare i Network Information Services (NIS).

Per vedere quale nome dominio è associato con l’host locale, usate il comando `hostname --fqdn`. Quasto mostra il primo nome dominio pienamente qualificato che il risolutore trova per l’hostname locale. [6]

10.4.1 Il risolutore

Il lavoro di scoprire quali indirizzi IP sono associati con un dominio particolare, è a carico del risolutore. Il più comunemente usato è il gruppo di funzioni che va sotto il nome di resolver (`resolver(3)`) nella libreria GNU C. Un altro è il risolutore FireDNS, fornito da pacchetto `libfiredns`.

Come il risolutore LIBC risolva i nomi dipende dalla riga `hosts` nel file di configurazione `/etc/nsswitch.conf`. Questa riga elenca i servizi che dovrebbero essere usati per risolvere un nome: p. es., `dns`, `files`, `nis`, `nisplus`. [7] Vedere `nsswitch.conf(5)`. Fintanto che si usa il servizio `files`, il comportamento del risolutore viene anche gestito dal file di configurazione `/etc/hosts`. Vedere `hosts(5)`.

Tutti i file sopra descritti sono statici e possono essere modificati con il vostro editor preferito.

Fintanto che si usa il servizio `dns`, il comportamento delb risolutore viene gestito anche dal file di configurazione `/etc/resolv.conf`. Vedere `resolv.conf(5)`. Una delle funzioni principali di `resolv.conf` è di elencare gli indirizzi IP dei nameserver che verranno contattati

---

[6] Tecnicamente è il FQDN dato da `gethostbyname(2)` per l’hostname dato da `gethostname(2)`.

[7] Il controllo della risoluzione dei nomi è anche affidato al file di configurazione `/etc/host.conf`. La riga `order` in questo file elenca i metodi che devono essere usati per risolvere in nome: p. es., `bind`, `hosts`, `nis`. Vedere `host.conf(5)`. Penso che questa riga sia stata soppiantata da quella `hosts` in `nsswitch.conf`, ma non ne sono certo.

per risolvere il nome. Tale lista spesso dipende dal tipo di rete a cui ci si connette e queto tipo può cambiare nel tempo mentre il vostro computer sta funzionando. Programmi tipo `pppd` e `dhclient` sono in grado di manipolare `resolv.conf` per aggiungere e rimuovere righe, ma quaste funzionalità non funzionano sempre in maniera appropriata, entrando in conflitto una con l'altra. Il pacchetto `resolvconf` risolve il problema in maniera migliore, fornendo un'infrastruttura standard per aggiornare questo file. Vedere 'Gestire le informazioni del nameserver – `resolvconf`' in questa pagina.

10.4.2 Gestire le informazioni del nameserver – `resolvconf`

Il pacchetto `resolvconf` fornisce un'infrastruttura per la gestione dinamica delle informazioni sui nameserver disponibili. Risolve l'annoso problema di come mantenere una lista dinamica di nameserver utilizzabile dal risolutore a dalle cache DNS. Resolvconf si pone come intermediario fra i programmi che controllano le interfacce di rete e forniscono le informazioni sul nameserver e le applicazioni che necessitano di dette informazioni.

`resolvconf` è disegnato per lavorare senza alcuna necessità di configurazione manuale. Comunque, il pacchetto è abbastanza nuovo e può richiedere un intervento manuale per farlo funzionare in maniera appropriata. Ciò è sicuramente vero se avete personalizzato dei pacchetti in maniera da aggiornare `/etc/resolv.conf`: dovrete disabilitare le vostre modifiche. Vedere `/usr/share/doc/resolvconf/README.gz` for details.

10.4.3 Salvare i nomi cercati – `nscd`, `dnsmasq`, `pdnsd`, `bind9`

Se il vostro nameserver è lento a rispondere, allora potete usare `nscd` per mettere in cache i risultati delle ricerche ottenuti usando il risolutore `libc6`.

Se volete salvare i risultati per altri host sulla vostra rete locale, allora potreste voler utilizzare un caching forwarding nameserver tipo `dnsmasq` o `pdnsd`.

Se volete, potete anche usare `named` di `bind9`, per lo stesso scopo. E', però, un programma pesante per cui a meno che abbiate bisogno delle sue funzionalità avanzate, vi troverete meglio con uno dei pacchetti menzionati prima.

Tutti questi pacchetti funzionano bene con `resolvconf`.

10.4.4 Fornire un Domain Name Service – `bind`

Se dovete fornire un name service autoritativo per un dominio, allora avete bisogno di un nameserver con tutti i crismi, come `named`, fornito dal pacchetto `bind9`.

Se installate `bind9`, dovreste installare anche `dnsutils`. Potreste voler installare anche questi pacchetti di utilità: `bind9-host`; `dns-browse`; `dnscvsutil`; `nslint`. Ed anche il pacchetto con la documentazione: `bind9-doc`. Nonchè questi pacchetti per lo sviluppo: `libbind-dev`; `libnet-dns-perl`. Se configurate le interfacce con DHCP, allora troverete questo pacchetto utile: `dhcp-dns`.

Installate `bind9` o `dpkg-reconfigure` – atelo per le impostazioni di base. La configurazione consiste nella modifica di `named.conf`. In Debian esso si trova in `/etc/bind/` ed è usato principalmente per definire le zone DNS di base; `include` altri due file: `named.conf.local`, usato per definire le zone locali e `named.conf.options`, usato per impostare le opzioni. (Quest'ultimo viene processato da `resolvconf` per produrre `/var/run/bind/named.options`, che è identico all'originale tranne che per la specificazione `forwarders`, che è una lista dei nameserver correntemente non disponibili. Per usarla modificate la riga `include` in `named.conf` in modo che comprenda `/var/run/bind/named.options`. Vedere 'Gestire le informazioni del nameserver – `resolvconf`' nella pagina precedente.)

I file di database nominati in `named.conf*` senza un percorso completo vengono salvati in `/var/cache/bind/`. Questo è il posto giusto per salvare i file generati da `named`: per esempio, i database per le zone per le quali il demone è secondario. I file statici dei database `/etc/bind/` sono e devono essere riferiti in `named.conf` con il loro percorso completo. Vedere `/usr/share/doc/bind9/README.Debian.gz` for details.

10.5 Configurare le interfacce di rete con DHCP

La configurazione a basso livello delle inetrfacce può essere automatizzata mediante il Dynamic Host Configuration Protocol (DHCP). Il vostro firewall, o router o il vostro ISP a larga banda possono fornire gli indirizzi IP ed altri parametri in questo modo.

Per falro funzionare dovete installare uno dei seguenti pacchetti:
- `dhcp3-client` (versione 3, Internet Software Consortium)
- `dhcpcd` (Yoichi Hariguchi e Sergei Viznyuk)
- `pump` (Red Hat)

`pump` è semplice e largamente usato. `dhcp3-client` è complesso, ma ampiamente configurabile. [8]

10.6 Configurazione di rete ad alto livello in Debian

Per semplificare la configurazione di rete, Debian fornisce uno strumento di configurazione standard, che consiste nei programmi `ifup` ed `ifdown` e nel file `/etc/network/interfaces`. [9] Se scegliete `ifupdown` per la vostra configurazione di rete, allora **non** do-

---

[8] Da Aprile 2004 esiste anche un pacchetto `dhcp-client` disponibile. Questo contiene la versione 2 del client ISC DHCP. Questo è stato soppiantato dalla versione 3, attualmente sotto forma di pacchetto `dhcp3-client`. I manutentori pensano di rinominare `dhcp3-client` in `dhcp-client` dopo il rilascio di Sarge. Accertatevi di non avere installate le versioni sperimentali di `dhcp-client`. `ifupdown` non funziona con esse.

[9] Il formato di `/etc/network/interfaces` per le versioni attuali di `ifupdown` è legermente incompatibile con quello delle versioni precedenti di Potato. Lo script post-installazione di `ifupdown` dovrebbe aggiornare il file automaticamente, se necessario. Comunque è una buona idea controllare il file convertito.

vreste usare, di norma, anche i comandi a basso livello. [10] Il programma `ifupdown` è stato scritto con l'intento di utilizzarlo da solo per configurare e deconfigurare le interfacce di rete.

Per aggiornare la configurazione di un'interfaccia, fate così:

```
# ifdown eth0
# editor /etc/network/interfaces  # modificatelo a piacimento
# ifup eth0
```

Per maggiori informazioni, vedere `interfaces(5)`, `/usr/share/doc/ifupdown/examples/network-interfaces.gz` e `ifup(8)`.

10.6.1 Configurare un'interfaccia con un indirizzo IP statico

Supponiamo vogliate configurare un'interfaccia Ethernet in maniera che abbia un indirizzo IP fisso di `192.168.0.111`. Esso inizia con `192.168.0`, quindi deve essere di una LAN. Supponiamo, inoltre, che `192.168.0.1` sia l'indirizzo del gateway ad Internet della LAN. Modificate `/etc/network/interfaces` in maniera che abbia un blocco di righe come questo:

```
iface eth0 inet static
        address 192.168.0.111
        netmask 255.255.255.0
        gateway 192.168.0.1
```

Potete configurare altri aspetti dell'interfaccia od eseguire altre azioni dopo averla attivata o prima di disattivarla, specificando i comandi appropriati sulle righe "up" e "down".

```
iface eth0 inet static
        address 192.168.0.111
        netmask 255.255.255.0
        gateway 192.168.0.1
        up route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.2 dev $IFACE
        down route del -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.2 dev $IFACE
        up echo Interface $IFACE going up | /usr/bin/logger -t ifup
        down echo Interface $IFACE Going down | /usr/bin/logger -t ifdown
```

In alternativa i comandi possono essere inseriti in script nelle directory `/etc/network/if-up.d` e `/etc/network/if-down.d`. Essi possono anche implementare opzioni più estese. Vedere `interfaces(5)` for details. Per esempio, il pacchetto `resolvconf` comprende degli script che vi permettono di aggiungere delle opzioni che specificano che le informazioni sul DNS vengano incluse in `/etc/resolv.conf` mentre l'interfaccia è attiva:

---

[10] Ciò vuol dire nojn utlizzare nemmeno altri strumenti di configurazione ad alto livello, tipo `whereami`, che chiamano a loro volta strumenti a basso livello.

```
iface eth0 inet static
        address 192.168.0.111
        netmask 255.255.255.0
        gateway 192.168.0.1
        dns-search somedomain.org
        dns-nameservers 195.238.2.21 195.238.2.22
```

L'argomento `somedomain.org` dell'opzione `dns-search` corrisponde all'argomento dell'opzione `search` in `resolv.conf(5)`. Gli argomenti `195.238.2.21` e `195.238.2.22` dell'opzione `dns-nameservers` corrisponde agli argomenti delle opzioni `nameserver`. Altre opzioni riconosciute sono `dns-domain` e `dns-sortlist`. Vedere 'Gestire le informazioni del nameserver – `resolvconf`' a pagina 191.

10.6.2 Configurare un'interfaccia con DHCP

per configurare un'interfaccia con DHCP, modificate `/etc/network/interfaces` in maniera che comprenda una serie di righe come questa:

```
iface eth0 inet dhcp
```

Affinchè funzioni doverete avere installato uno dei client DHCP menzionati in 'Configurare le interfacce di rete con DHCP' a pagina 192.

10.6.3 Configurare un'interfaccia Wi-Fi

Il pacchetto `wireless-tools` comprende uno script di aggancio `/etc/network/if-pre-up.d/wireless-tools` che gli rende possibile configurarare l'harware Wi-Fi (801.11a/b/g) prima che l'interfaccia venga attivata. La configurazione viene fatta con il programma `iwconfig`; vedere `iwconfig(8)`. Per tutti i parametri possibili di `iwconfig` potete includere un'opzione in `/etc/network/interfaces` con il nome del parametro preceduto da un prefisso "wireless-". Per esempio, per impostare l'ESSID di `eth0` a `myessid` e la chiave di cifratura a `123456789e` prima di attivare `eth0` mediante DHCP, modificate `/etc/network/interfaces` in maniera che comprenda una serie di righe tipo questa:

```
iface eth0 inet dhcp
        wireless-essid myessid
        wireless-key 123456789e
```

Notate che non dovreste usare questo metodo per impostare ESSID e chiave se state usando `waproamd` per questa interfaccia. Nel momento in cui `ifup` viene eseguito `waproamd` ha già impostato sia ESSID che chiave. Vedere 'Iniziare la configurazione di rete – `waproamd`' a pagina 203.

10.6.4 Configurare un'interfaccia PPP

I programmi `ifup` e `ifdown` usano `pon` e `poff` per aggiungere e rimuovere le interfacce PPP, per cui leggete prima 'Configurare un'interfaccia PPP' a pagina 185.

Supponiamo che abbiate impostato PPP per funzionare con il peer `myisp`. Modificate `/etc/network/interfaces` in maniera che includa una serie di righe come questa:

```
iface ppp0 inet ppp
        provider myisp
```

In questo modo, `ifup ppp0` esegue

```
pon myisp
```

Sfortunatamente, non è al momento possibile fornire opzioni aggiuntive di `pppd` alle righe `ppp` in `/etc/network/interfaces`. [11]

Non è attualmente possibile usare `ifupdown` per eseguire una configurazione ausiliaria delle interfacce PPP. Siccome `pon` termina prima che `pppd` abbia finito di stabilire la connessione, `ifup` eseguegli script `up` prima che l'interfaccia PPP sia pronta all'uso. Finchè questo baco [12] non verrà risolto, resta la necessità di fare una configurazione ausiliaria in `/etc/ppp/ip-up` o in `/etc/ppp/ip-up.d/`.

10.6.5 Configurare un'interfaccia PPPoE

Molti Internet Service Provider (ISP) con la larga banda usano PPP per negoziare le connessioni, anche se le macchine dei loro clienti sono connesse tramite reti Ethernet e/o ATM. Ciò si ottiene mediante PPP over Ethernet (PPPoE), che è una metodica per l'incapsulazione di flussi PPP all'interno di blocchi Ethernet. Supponiamo che il vostro ISP si chiami *`myisp`*. Per prima cosa configurate PPP e PPPoE per il peer *`myisp`*. Il modo più semplice è di installare il pacchetto `pppoeconf` e di lanciare `pppoeconf` da console. Poi modificate `/etc/network/interfaces` in maniera che includa un serie di righe come questa:

```
iface eth0 inet ppp
        provider myisp
```

Talvolta ci possono essere problemi con le Maximum Transmit Unit (MTU) con PPPoE su Digital Subscriber Line (DSL). Vedere DSL-HOWTO (`http://www.tldp.org/HOWTO/DSL-HOWTO/`) for details.

Notate che se il vostro modem a larga banda contiene un router, allora sarà il modem/router a gestire la connessione PPPoE e ad apparire sul versante LAN come un semplice gateway Ethernet per Internet.

---

[11] Vedere bug #196877 (`http://bugs.debian.org/196877`).

[12] Vedere bug #127786 (`http://bugs.debian.org/127786`).

10.6.6 Configurare più interfacce Ethernet per un gateway

Supponiamo che `eth0` sia connesso ad Internet con un indirizzo IP configurato mediante DHCP e che `eth1` sia connesso alla LAN con un indirizzo IP statico `192.168.1.1`. Modificate `/etc/network/interfaces` in maniera da comprendere una serie di righe come questa:

```
iface eth0 inet dhcp

iface eth1 inet static
        address 192.168.1.1
        netmask 255.255.255.0
```

Se attivale il NAT su questo host come descritto in 'Costruire un gateway router' a pagina 207, allora potete condividere la connessione ad Internet con tutti gli altri host dell LAN.

10.6.7 Configurare le interfacce virtuali

Mediante le interfacce virtuali potete configurare una singola scheda Ethernet come interfaccia a parecchie sottoreti IP. Per esempio, ammettiamo che il vostro host sia sulla LAN della rete 192.168.0.x/24. Volete conneterlo ad Internet mediante un indirizzo IP pubblico fornito via DHCP usando la vostra scheda Ethernet già esistente. Modificate `/etc/network/interfaces` in maniera che comprenda una serie di righe come questa:

```
iface eth0 inet static
        address 192.168.0.1
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255

iface eth0:0 inet dhcp
```

L'interfaccia `eth0:0` è un'interfaccia virtuale. Quando viene attivata, verrà attivata consensualmente `eth0`.

10.7 Configurazione di rete mediante definizione delle interfacce logiche

E' importante per il lettore comprendere la differenza fra un' **interfaccia fisica** ed un' **interfaccia logica**. [13] Un'interfaccia **fisica** è quella che abbiamo finora chiamato "l'interfaccia", la cosa che il kernel definisce `eth0`, `eth1`, `ppp0`, o quel che avete. Un'interfaccia **logica** è rappresentata da un gruppo di valori assegnati ai parametri variabili di un' interfacia fisica. Se lo

---

[13] Questa terminologia è usata nella documentazione di `ifupdown`.

trovate ostico, Sostituite l'espressione "configurato come interfaccia logica *X*" con l'espressione "configurato con il profilo dell'interfaccia *X*" quando leggete.

Le definizioni `iface` in `/etc/network/interfaces` sono in realtà definizioni di interfacce logiche, non fisiche. [14] Se non vorrete mai riconfigurare le vostre interfacce, allora potete ignorare questo fatto, pocihè l'interfaccia fisica *foo* verrà configurata in partenza come interfaccia logica *foo*.

Tuttavia, ammettiamo che il vostro computer sia un portatile che trasportate fra casa e lavoro. Quando lo connettete alla rete aziendale o alla LAN casalinga, dovete configurare `eth0` di conseguenza.

Definite prima due interfacce logiche, `home` e `work` (invece che `eth0`, come fatto prima), che descrivono come l'interfaccia deve essere configurata per la rete casalinga ed aziendale, rispettivamente.

```
iface home inet static
        address 192.168.0.123
        netmask 255.255.255.0
        gateway 192.168.0.1

iface work inet static
        address 81.201.3.123
        netmask 255.255.0.0
        gateway 81.201.1.1
```

Allora l'interfaccia fisica `eth0` può essere attivata per casa con la configurazione appropriata, specificando sulla riga di comando:

```
# ifup eth0=home
```

Per riconfigurare `eth0` per il lavoro, date i comandi:

```
# ifdown eth0
# ifup eth0=work
```

Notate che con il file `interfaces` scritto nel modo precedente non sarà più possibile attivare `eth0` con il solo `ifup eth0`. La ragione sta nel fatto che `ifup` usa il nome dell'interfaccia fisica come nome predefinito di quella logica ed ora nel nostro esempio nessuna interfaccia logica `eth0` è definita.

---

[14] Notate che la interfacce chiamate sulle righe `auto` devono essere fisiche, non logiche.

10.8 Magie della configurazione di rete

I nomi delle interfacce possono essere “mappati” come altri nomi quando gira `ifup`. Come ciò accada può essere fatto dipendere dalle circostanze. Perciò, `ifup` può essere configurato in maniera tale da attivare una data interfaccia fisica come interfaccia logica appropriata, scegliendo fra una serie di alternative predefinite.

La mappatura dei nomi delle interfacce avviene così:

- Se nessun nome di interfaccia logica è dato ad `ifup` da riga di comando, allora il nome della interfaccia fisica viene usato come nome iniziale di quella logica.
- Se il nome dell’interfaccia logica corrisponde ad uno di un blocco generale di righe `mapping`, allora quella mappatura verrà applicata per generare un nuovo nome per l’interfaccia logica. Ciò avviene a turno per ogni blocco.
- Se il nome finale dell’interfaccia logica è l’etichetta di una definizione di interfaccia logica in `/etc/network/interfaces`, allora l’interfaccia fisica corrispondente verrà attivata come logica. Altrimenti `ifup` stampa un messaggio che è “Ignoring unknown interface” ed esce.

La sintassi del gruppo di righe `mapping` è:

```
mapping glob-pattern
        script script-name
        [map script input]
```

Lo script chiamato dal gruppo `mapping` viene sempre eseguito con il nome dell’interfaccia **fisica** come argomento e con il contenuto di tutte le righe seguenti “map” nel gruppo (senza la parola “map”) fornitogli come standard input. Lo script stampa il risultato della mappatura sul proprio standard output prima di uscire.

Per esempio, il blocco `mapping` seguente dice a `ifup` di attivare l’interfaccia `eth0` come interfaccia logica `home`.

```
mapping eth0
        script /usr/local/sbin/echo-home
```

dove `/usr/local/sbin/echo-home` è:

```
#!/bin/sh
echo home
```

Poichè la mappatura è eseguita con uno script, è possibile scegliere automaticamente l’interfaccia logica — basandosi su una sorta di test. Vedere ‘Selezione delle interfacce logice con guessnet’ a fronte per un esempio di ciò.

10.8.1 Selezione delle interfacce logice con guessnet

Installate `guessnet` e poi aggiungete un blocco di righe come il seguente a `/etc/network/interfaces`:

```
mapping eth0
        script guessnet-ifupdown
        map home
        map work
```

Ora, quando date `ifup eth0`, `guessnet` controlla se `eth0` può essere attivata come `home` o `work`. Per far ciò usa le informazioni contenute nelle definizioni delle interfacce logiche.

10.8.2 Configurazione di rete automatica con laptop-net

Il pacchetto `laptop-net` usa un approccio diverso per la riconfigurazione automatica della rete. Laptop-net non usa le interfacce logiche di `ifupdown`, ma ha il proprio sistema di configurazione, basato su "schemes" e "profiles". Laptop-net, però, usa `ifup` e `ifdown` per configurare le interfacce fisiche. Per maggiori informazioni consultate l'ottima documentazione in `laptop-net-doc`.

10.9 Gestire nomi inconsistenti delle interfacce dati dal kernel

I nomi `eth0`, `eth1`, ecc. vengono assegnati dal kernel in maniera che possa creare le interfacce con il medesimo nome. mentre gli adattatori riconosciuti al boot sono in genere riconosciuti ogni volta nello stesso ordine e quindi ricevono gli stessi nomi ogni volta, lo stesso no vale per quelli che vengono inseriti a caldo. Questi possono venire riconosciuti in qualsiasi ordine e finire con nomi diversi assegnati dal kernel in occasioni differenti.

A causa di ciò, su un sistema in cui gli adattatori di rete vengono inseriti a caldo, non sempre vale la pena di definire delle interfacce logiche in `/etc/network/interfaces` con i nomi `eth0`, `eth1`, ecc. e di fidarsi della mappatura predefinita. Invece, dovrete dare nomi distinti alle interfacce logiche ed usare uno dei metodi seguenti per decidere quale interfaccia può essere assegnata a quale adattatore.

Un metodo è di usare o l'utilità `nameif` (del pacchetto `net-tools`) o la più flessibile `ifrename` (del pacchetto `ifrename`) per far assegnare al kernel i nomi alle interfacce in accorso con le proprietà degli adattori soggiacenti. Con tale schema, in effetti, il nome dell'interfaccia fisica può essere utilizzato per risalire a quale adattatore esso corrisponda.

Un altro metodo consiste nell'usare il meccanismo di mappatura di `ifup` in maniera tale che un'interfaccia logica venga scelta per attivarne una fisica sulla base di alcune caratteristiche tipiche del dato adattatore a cui corrisponde.

Ammettiamo, per esempio che avete due adattatori di rete diversi che usate con le reti `net1` e `net2`, rispettivamente. La directory `/usr/share/doc/ifupdown/examples/` contiene uno script di mappatura che può essere utilizzato per scegliere un'interfaccia logica basandosi sull'indirizzo del Media Access Controller (indirizzo MAC) dell'adattatore. Per prima cosa installate lo script nella directory appropriata.

```
# install -m770 /usr/share/doc/ifupdown/examples/match-mac-address.sh \
   /usr/local/sbin/
```

Aggiungete,poi, un gruppo di righe come il seguente a `/etc/network/interfaces`:

```
mapping eth0
        script /usr/local/sbin/match-mac-address.sh
        map 02:23:45:3C:45:3C net1
        map 00:A3:03:63:26:93 net2
```

Vedere 'Mappatura multilivello' a pagina 204 per un esempio più complesso.

In entrambi i metodi applicati, la proprietà che viene usata più comunemente per identificare l'adattatore è l'indirizzo MAC.

10.10 Iniziare la configurazione di rete

Abbiamo visto come le interfacce possano essere configurate o riconfigurate. Ciò deve avvenire al momento giusto.

Tradizionalmente, la rete veniva configurata durante la sequenza di avvio tramite lo script di inizio `/etc/rcS.d/S40networking` e veniva raramente riconfigurata. I servizi che dipendevano dalla rete venivano lanciati dopo nella sequenza di avvio. Allo spegnimento o al riavvio gli script venivano eseguiti nell'ordine inverso.

Ora, invece, c'è un trend in GNU e Linux verso il supporto di hardware e circostanze che cambiano in maniera dinamica. All'inizio venne aggiunto il supporto per le schede PCMCIA inseribili a caldo; più di recente è stato aggiunto il meccanismo `hotplug`, così che molte altre periferiche possono essere messe e tolte mentre il computer sta andando. Ciò comprende l'hardware di rete. Notate che i servizi che dipendono da hardware che viene inserito a caldo devono essere lanciati solo dopo il suo inserimento e fermati dopo la sua rimozione. Ciò significa che tali servizi devono essere tolti al controllo del sistema di avvio System V e messi sotto il controllo di `ifupdown`.

Per esempio, supponiamo che il servizio `foo`, controllato dallo script `/etc/init.d/foo` dipenda dalla interfaccia di rete riconfigurata dinamicamente `eth0`.

- Prima rimuovete `foo` dal controllo del sistema di avvio. Se usate il sistema `sysv-rc`, allora fate quanto segue. [15]

  ```
  # rm /etc/rc[2345].d/S??foo
  ```

---

[15] Notate che ciò lascia il collegamento "stop". (`/etc/rc?.d/K??foo`) intatto. Vedere 'I Runlevel' a pagina 21 per maggiori informazioni.

- Poi, mettete `foo` sotto il controllo di `ifupdown` aggiungendo le opzioni `up` e `down` al gruppo `eth0` in `/etc/network/interfaces`, che contiene le chiamate allo script `foo`:

```
     iface eth0 inet dhcp
             up /etc/init.d/foo start
             down /etc/init.d/foo stop
```

10.10.1 Iniziare la configurazione di rete all'avvio

All'avvio lo script di inizio `/etc/rcS.d/S40networking` lancia il comando `ifup -a`. Ciò attiva tutte le interfacce fisiche elencate nei gruppi `auto` di `/etc/network/interfaces`.

Oggigiorno è spesso meglio gestire la configurazione di rete con metodi dinamici. Una volta che i meccanismi che supportano hardware dinamico sono in posizione, diventa più semplice anche gestire l'hardware statico come fosse dinamico. L'avvio può essere trattato, allora, come un altro evento hotplug. (Vedere 'Iniziare la configurazione di rete – `hotplug`' in questa pagina.)

Comunque in quasi tutti i casi uno vuole che al meno l'interfaccia di loopback `lo` venga attivata all'avvio. Perciò assicuratevi che `/etc/network/interfaces` contenga i seguenti gruppi.

```
auto lo

iface lo inet loopback
```

Potete elencare i nomi delle interfacce fisiche addizionali nei gruppi `auto` se volete attiavarli anche essi all'avvio. **Mai** elencare le interfacce PCMCIA nei gruppi `auto`. `Cardmgr` per PC;CIA viene lanciato più tardi nella sequenza d'avvio rispetto a `/etc/rcS.d/S40networking`.

10.10.2 Iniziare la configurazione di rete – `hotplug`

Per il supporto hot-plug, installlate il pacchetto `hotplug`.

L'hardware di rete può essere inserito a caldo all'avvio o dopo che una scheda (p. es. PCMCIA) è stata inserita nel computer oppure dopo che un' utilità tipo `discover` è stata eseguita ed ha caricato i moduli necessari.

Quando il kernel riconosce del niovo hardware, inizializza il driver per esso e poi lancia il programma `hotplug` per configurarlo. Se poi l'hardware viene rimosso, il kernel lancia di nuovo `hotplug` con variabili d'ambiente diversamente impostate. In Debian, quando `hotplug` viene chiamato, esegue degli script in `/etc/hotplug/` e `/etc/hotplug.d/`. Vedere `hotplug(8)` per i dettagli.

Hardware di rete appena inserito viene configurato dallo script `/etc/hotplug/net.agent`.

[16] Supponiamo che la vostra scheda di rete PCMCIA sia stata inserita, risultando l'interfaccia `eth0` disponibile per l'uso. `/etc/hotplug/net.agent` fa quanto segue [17] :

```
ifup eth0=hotplug
```

A meno di aver aggiunto una definizione di una interfaccia logica, o una mappatura definita `hotplug` a `/etc/network/interfaces`, questo comando non produrrà nulla. Per fare sì che configuri `eth0`, aggiungete il gruppo seguente a `/etc/network/interfaces`:

```
mapping hotplug
        script echo
```

Come spiegato in 'Configurazione di rete mediante definizione delle interfacce logiche' a pagina 196, ciò mapperà il comando mostrato sopra in maniera che sia equivalente al seguente:

```
ifup eth0=eth0
```

(**Non** includete un gruppo come questo se state usando delle istanze di `ifplugd` o `waproamd` lanciate da `hotplug` per controllare l'interfaccia.)

Se volete attivare solo `eth0` e nient'altro all'inserimento a caldo, allora usate `grep` al posto di `echo` come segue:

```
mapping hotplug
        script grep
        map eth0
```

Vedere 'Magie della configurazione di rete' a pagina 198 e `/usr/share/doc/hotplug/README.Debian` per altri consigli.

10.10.3 Iniziare la configurazione di rete – `ifplugd`

Il demone `ifplugd` attiva o disattiva un'interfaccia a seconda se l'hardware soggiacente sia connesso o meno alla rete. Il programma è in grado di riconoscere un cavo attivo connesso ad una interfaccia Ethernet o un punto di accesso associato ad una interfaccia Wi-Fi (sebbene `waproamd` sia probabilmente quello che volete usare in quest'ultimo caso). Quando `ifplugd` vede che lo stato del collegamento è cambiato, lancia uno script proxy che da predefinito chiama `ifup` o `ifdown`.

---

[16] Può anche essere configurato da script di aggancio che sono stati installati in `/etc/hotplug.d/net/`. I pacchetti `ifplugd` e `waproamd`, per esempio, installano i loro script qui.

[17] Dalla versione 0.0.20040329-4 o giù di lì, `hotplug` può essere messo in modalità in cui si comporta in maniera diversa da come è descritto qui. Uno di tali modi è il così detto "all", dove `hotplug` attiva tutte le interfacce inserite a caldo. L'altra modalità è la così detta "auto", in cui `hotplug` attiva le interfacce solo se sono elencate nelle righe `auto` in `/etc/network/interfaces`. In queste modalità alternative `ifup` viene invocato senza il suffisso `=hotplug`.

10.10.4 Iniziare la configurazione di rete – `waproamd`

Il demone `waproamd` è come `ifplugd`, solo che è stato pensato per le schede Wi-Fi. Cerca attivamente i punti di accesso a cui l'hardware Wi-Fi è in grado di connettersi. Una volta che la connessione è stabilita, `waproamd` lancia `ifup`.

Se usate `waproamd`, allora in via generale configurerete la scheda Wi-Fi tramite `waproamd` e non tramite le opzioni `wireless-*` in `/etc/network/interfaces`.

10.10.5 Configurazione di rete e PCMCIA

Esistono parecchi approcci possibili alla configurazione delle interfacce di rete PCMCIA (per i kernel 2.4 e 2.6).

- Per le schede a 32 bit PCI (CardBus):
  - `ifupdown` controllato da `hotplug`
    - In Woody e Sarge dovete abilitare localmente il controllo da parte di `hotplug` di `ifupdown`, aggiungendo una serie di righe di mappatura a `/etc/network/interfaces`, come descritto in 'Iniziare la configurazione di rete – `hotplug`' a pagina 201.
- Per schede ISA a 16 bit:
  - `ifupdown` controllato da `hotplug` con `pcmcia-cs`
    - **raccomandato**
    - In Woody e Sarge dovete disabilitare localmente il comportamento predefinito di `pcmcia-cs` di prendere il controllo di `ifupdown` aggiungendo la riga `exit 0` all'inizio di `/etc/pcmcia/network` Inoltre, dovete abilitare localmente il controllo da parte di `hotplug` su `ifupdown` aggiungendo un gruppo di righe di mappatura a `/etc/network/interfaces`, come descritto in 'Iniziare la configurazione di rete – `hotplug`' a pagina 201.
  - `ifupdown` controllato da `pcmcia-cs` mediante il predefinito `/etc/pcmcia/network`
    - **Deprecato**, ma ancora il predefinito per Woody e Sarge
  - strumenti a basso livello controllati da `pcmcia-cs` mediante codice speciale in `/etc/pcmcia/network`
    - **Deprecato**
    - In Woody e Sarge il codice speciale viene abilitato modificando `/etc/pcmcia/network.opts`

L'approccio raccomandato per le schede a 16 bit si avvantaggia del fatto che il sottosistema hotplug Linux 2.4 ora supporta PCMCIA. [18]

---

[18] Nei rilasci precedenti di Debian il modo standard di configurare le schede di rete PCMCIA era tramite gli script di aggancio di `cardmgr` `/etc/pcmcia/network` e `/etc/pcmcia/network.opts`. Essi erano stati sviluppati nell'era prima che Linux acquisisse una maggiore capacità di gestire l'hotplug più generale. Alcuni ancora usano gli script di Debian Woody nel loro stato predefinito in cui chiamano semplicemente `ifup` dopo che l'interfaccia è stata aggiunta e `ifdown` quando è stata rimossa. Come notato prima, è ora raccomandato l'uso di `hotplug` per fare ciò. Altri ancora usano il metodo speciale di chiamare i comandi di configurazione a basso livello, che vengono attivati quando determinate variabili in `/etc/pcmcia/network.opts` vengono impostate a "y". Ciò

Le schede di rete PCMCIA sono inseribili a caldo. Di conseguenza, qualsiasi che richiede l'accesso alla rete tramite loro, dovrebbe essere configurato in maniera da partire all'inserimento della scheda e terminare alla sua rimozione. In genere ciò si ottiene facendo sì che il servizio parta ad `ifup` e termini ad `ifdown`. Alcuni, tuttavia, scelgono di limitarsi all'inserzione a freddo della loro scheda PCMCIA: inseriscono la scheda prima di avviare il sistema e poi lanciano i servizi che richiedono l'accesso alla rete tramite la scheda nella sequenza di avvio. Se siete uno di questi, allora per assicurarvi che la scheda sia pienamente configurata prima dell'avvio dei servizi dovete fare quanto segue:

- Impostare `CARDMGR_OPTS="-f"` in `/etc/default/pcmcia` per forzare `cardmgr` a girare in foreground.
- Rinominare `/etc/rc?.d/S20pcmcia` a qualcosa tipo `/etc/rc?.d/S12pcmcia`.

Questo trucco funziona solo per le schede a 16 bit.

Notate che, se usate le schede a 16 bit, `pcmcia-cs` è ancora necessario. Il demone `cardmgr` che il pacchetto contiene è responsabile della gestione dei socket e del caricamento dei moduli. Non vogliamo solo che chiami i programmi di configurazione di rete via `/etc/pcmcia/network`.

Affinchè `cardmgr` funzioni correttamente, potreste dover modificare `/etc/pcmcia/config.opts` per configurare le risorse assegnate alle schede a 16 bit. Vedere 'PCMCIA' a pagina 102 ed il Linux PCMCIA HOWTO (`http://www.tldp.org/HOWTO/PCMCIA-HOWTO.html`) per maggiori informazioni.

10.11 Mappatura multilivello

Supponiamo che i vostri adattatori di rete siano dinamici e che abbiate abilitato la configurazione automatica come descritto in 'Iniziare la configurazione di rete – `hotplug`' a pagina 201. Supponiamo anche che abbiate bisogno di mappare le interfacce logiche in "fisiche" a seconda sia dell'adattatore soggiacente all'interfaccia fisica (come descritto in 'Gestire nomi inconsistenti delle interfacce dati dal kernel' a pagina 199) che della rete connessa all'interfaccia (come descritto, per esempio, in 'Selezione delle interfacce logice con guessnet' a pagina 199). Potete ottenere ciò con la mappatura multilivello.

Il primo livello prende il nome del gruppo `hotplug` e restituisce il nome dell'interfaccia assegnato dal kernel, se questa deve essere inserita dinamicamente. Il secondo livello prende un nome dell'interfaccia assegnato dal kernel e restituisce il nome di un adattatore. Il terzo livello mappa i nomi degli adattattori a nomi di interfacce logiche, basandosi sul tipo di rete.

```
# Allow hotplug to bring up interfaces
mapping hotplug
 script echo

# Determine whether interface is wired or Wi-Fi
```

---

ha parecchi problemi. E' afflitto da race conditions; funziona solo per schede a 16 bit; quello che fa è meglio lasciarlo fare a `ifupdown`. Di conseguenza, è deprecato.

```
mapping eth?
        script /usr/local/sbin/match-mac-address.sh
        map 02:23:45:3C:45:3C wired
        map 00:A3:03:63:26:93 wifi

# Detect which wired network is available
mapping wired
        script guessnet-ifupdown
        map work-wired
        map home

# Detect which Wi-Fi network is available
mapping wifi
        script ifscout
        map starbucks
        map work-wireless

iface work-wired inet static
 ...
```

10.12 Configurazione dei servizi di rete

Una configurazione tipica dei servizi di rete di un desktop o di un server casalingo comprende:

- Il *super-server* Internet ed il wrapper demone TCP/IP, vedere 'Controllare l'accesso ai servizi' a pagina 140.
  - `/etc/inetd.conf`
- `ssh`: La shell sicura OpenSSH, vedere 'SSH' a pagina 161.
  - `/etc/ssh/ssh_config`
  - `/etc/ssh/sshd_config`
- `exim`: mail transport agent, vedere 'Mailname' a pagina 189 e 'Mail transport agent (MTA)' a pagina 165.
  - `/etc/exim/exim.conf`
  - `/etc/mailname`
  - `/etc/aliases`
  - `/etc/email-addresses`
- `fetchmail`: demone per ritirare la posta da un account POP3, vedere 'Ritirare la posta – Fetchmail' a pagina 167.
  - `/etc/fetchmailrc`
- `procmail`: programma di distribuzione e filtro locale della posta, vedere 'Processare la posta – Procmail' a pagina 168.
  - `~/.procmailrc`
- Hostname e DNS (proxy, cache, ...), vedere 'Hostname' a pagina 189 e 'Domain Name Service (DNS)' a pagina 190.
  - `/etc/host.conf`

  - `/etc/hostname`
  - `/etc/hosts`
  - `/etc/hosts.allow`
  - `/etc/hosts.deny`
  - `/etc/resolv.conf`
  - `/etc/bind/named.conf` (modificare)
  - `/etc/bind/db.lan` (aggiungere per gli host della LAN)
  - `/etc/bind/db.`*`192.168.0`* (aggiungere per il LAN reverse)
- DHCP, vedere 'Configurare le interfacce di rete con DHCP' a pagina 192.
  - `/etc/dhcp3/dhclient.conf` (DHCP client)
  - `/etc/default/dhcp3-server` (DHCP server)
  - `/etc/dhcp3/dhcpd.conf` (DHCP server)
- `cvs`: concurrent versions system, vedere 'Concurrent Version System (CVS)' a pagina 217.
  - `/etc/cvs-cron.conf`
  - `/etc/cvs-pserver.conf`
- `nfs-kernel-server`: network file system, vedere 'Configurare NFS' a pagina 40. (per sistemi simil-unix)
  - `/etc/exports`
- `samba`: condivisione file e stampanti per Windows, vedere 'Configurare Samba' a pagina 40 and 'Samba' a pagina 133.
  - `/etc/samba/smb.conf`
- Printer daemon system, vedere 'Configurare la stampante' a pagina 41.
  - `/etc/printcap` (per lpr)
- `apache` e `apache2`: web server.
  - `/etc/apache/*`
  - `/etc/apache2/*`
- `squid`: web proxy cache server.
  - `/etc/squid/*`

10.13 Risoluzione dei problemi di rete

Se incontrate dei problemi, controllate l'output dei seguenti come prima presa di contatto:

```
# ifconfig
# cat /proc/pci
# cat /proc/interrupts
# dmesg | more
```

Vedere anche le sezioni seguenti 'Testare la rete: le basi' a pagina 128.

Se avete problemi con certi siti web, vedete 'Strani problemi di accesso con alcuni siti web' a pagina 45.

10.14 Costruire un gateway router

Un host Debian può diventare una macchina gateway per tutti gli scopi che esegue Network Address Translation (NAT, anche noto come masquerading), trasferimento posta, DHCP, DNS caching, HTTP proxy caching, servizio CVS, servizio NFS e Samba. Vedere 'Quale host ed IP usare per una LAN' a pagina 31 per l'esempio di tale impostazione.

10.14.1 Configurare Netfilter

Il progetto netfilter/iptables è un sottosistema di firewalling per Linux 2.4 e successivi. Vedere Netfilter (`http://www.netfilter.org/`), dove vengono spiegati molti argomenti relativi alla configurazione di rete.

Le basi di netfilter

Netfilter processa i pacchetti usando una catena di 5 tabelle: PREROUTING, INPUT, FORWARD, OUTPUT, and POSTROUTING.

```
                 decisione sul
                 routing
IN ------>  PRE ---> ------> FORWARD -----> ----> POST -----> OUT
interfaccia ROUTING  \       tracciamento /       ROUTING     interfaccia
            DNAT     |       filtro       ^       SNAT
            REDIRECT |                    |      MASQUERADE
                     v                    |
                   INPUT                OUTPUT
                     | filtro             ^ filtro,DNAT
                     v                    |
                     \--> Processo Locale --/
                             programmi nello user-space
```

10.14.2 La tabella di netfilter

I pacchetti vengono processati ad ogni catena, usando le seguenti tabelle.

- filter (filtro pacchetti, default)
  - INPUT (per i pacchetti che vanno all'interno della macchina stessa)
  - FORWARD (per i pacchetti che vengono instradati attraverso la macchina)
  - OUTPUT (per pacchetti generati localmente).
- nat (network address translation)
  - PREROUTING (per alterare i pacchetti non appena in entrata)
  - OUTPUT (per alterare i pacchetti generati localmente prima dell'instradamento)
  - POSTROUTING (per alterare i pacchetti non appena vengono messi in uscita)
- mangle (mascherare il network address, valido solo dopo 2.4.18)
  - tutte e 5 le catene.

I target di netfilter

Le regole di firewall hanno parecchi target:

- 4 target di base:
    - ACCEPT significa lasciar passare il pacchetto.
    - DROP significa lasciar cadere il pacchetto.
    - QUEUE significa far passare il pacchetto nello userspace (se è supportato dal kernel).
    - RETURN significa fermare l'attraversamento di questa catena e ricominciare alla prossima regola nella catena precedente (in chiamata).
- target estesi:
    - LOG attiva il kernel logging.
    - REJECT ritorna un pacchetto di errore e lascia cadere il pacchetto.
    - SNAT altera l'indirizzo di partenza del pacchetto, ed è usato solo nella catena POSTROUTING. (solo per la tabella nat)
      `--to-source  ipaddr[-ipaddr][:port-port]`
    - MASQUERADE è equivalente a SNAT, ma solo per le connessioni con IP dinamico (dialup). (solo per la tabella nat)
      `--to-ports port[-port]`
    - DNAT modifica l'indirizzo di destinazione del pacchetto ed è usato nelle catene PREROUTING ed OUTPUT ed in quelle definite dall'utente che vengono chiamate solo da quelle catene (PREROUTING ed OUTPUT). (solo per la tabella nat)
      `--to-destination ipaddr[-ipaddr][:port-port]`
    - REDIRECT modifica l'indirizzo IP della destinazione per inviare il pacchetto alla macchina stessa.
      `--to-ports port[-port]`

I comandi di netfilter

I comandi base di `iptables` sono:

```
iptables -N catena       # crea una catena

iptables -A catena \     # aggiunge una regola alla catena
         -t table \ # usa una table (filter, nat, mangle)
  -p protocollo \ # tcp, udp, icmp, o tutti,
  -s indirizzo-sorgente[/mask] \
  --sport porta[:porta] \ # porta sorgente se -p è tcp o udp
  -d destination-address[/mask] \
         --dport porta[:porta] \ # porta di destinazione se -p è tcp o udp
         -j target \ # cosa fare se corrisponde
  -i nome-interfaccia-ingresso \ # per INPUT, FORWARD, PREROUTING
  -o nome-interfaccia-uscita # per FORWARD, OUTPUT, POSTROUTING
```

Network Address Translation

Le macchine in una LAN possono accedere all'esterno attraverso un gateway che traduce gli indirizzi IP sulle LAN in indirizzi IP utilizzabili su Internet.

```
# apt-get install ipmasq
```

Applicate le regole come da esempio per rafforzare la protezione di `ipmasq`. Vedere `/usr/share/doc/ipmasq/examples/stronger/README`. Per la kernel-image-2.4 Debian di woody, assicuratevi di caricare i moduli appropriati. La versione di ipmasq di Sarge ha risolto questo problema. Vedere 'Funzioni di rete' a pagina 102 per le configurazioni necessarie.

Per la kernel-image-2.2 Debian, modificate `Z92timeouts.rul` in `/etc/masq/rules` come segue per assicurare una connessione più lunga ai siti remoti (buona per mail di grosse dimensioni, ecc.):

```
# tcp, tcp-fin, udp
# 2hr, 10 sec, 160 sec - default
# 1 day, 10 min, 10 min - più lungo, per esempio
$IPCHAINS -M -S 86400 600 600
```

In più, se la rete è raggiunta tramite una NIC PCMCIA, `ipmasq` deve essere lanciato da `/etc/pcmcia/network.opts`. (leggere: `/usr/share/doc/ipmasq/ipmasq.txt.gz`) o da `/etc/network/interfaces` (leggere: 'Configurazione di rete e PCMCIA' a pagina 203 e 'Iniziare la configurazione di rete' a pagina 200).

Redirigere una connessione SMTP (2.4)

Supponiamo che abbiate un portatile che è configurato per utilizzare altre LAN e che vogliate usare il vostro agente di posta sul portatile senza doverlo riconfigurare.

Aggiungendo le regole seguenti tramite il comando `iptables` alla macchina gateway redirigerà la connessione SMTP verso di essa.

```
# iptables -t nat -A PREROUTING -s 192.168.1.0/24 -j REDIRECT \
           -p tcp --dport smtp --to-port 25 # smtp=25, INPUT is open
```

Per una serie di regole di redirezione più completa, considerate l'installazione del pacchetto `ipmasq` e l'aggiunta di M30redirect.def (`http://www.debian.org/doc/manuals/debian-reference/examples/`) alla directory `/etc/ipmasq/rules/`.

10.14.3 Gestione di connessioni multiple di rete

[DA CORREGGERE] Regole di routing: (da Phil Brutsche `<pbrutsch@tux.creighton.edu>`) Vedere il manuale iproute (`http://lartc.org/`) per i dettagli. Il traffic control (tc) potrebbe anche essere interessante.

L'ambiente:

```
eth0: 192.168.1.2/24; gateway 192.168.1.1
eth1: 10.0.0.2/24; gateway 10.0.0.1
Nessun masquerading su questa macchina.
```

Alcune magie:

```
ip rule add from 192.168.1.2 lookup 1
ip rule add from 10.0.0.2 lookup 2
ip route add to default via 10.0.0.1 metric 0
ip route add to default via 192.168.1.1 metric 1
ip route add table 1 to 192.168.1.0/24 via eth0
ip route add table 1 to 10.0.0.2/24 via eth1
ip route add table 1 to default via 192.168.1.1
ip route add table 2 to 192.168.1.0/24 via eth0
ip route add table 2 to 10.0.0.2/24 via eth1
ip route add table 2 to default via 10.0.0.2
```

[DA CORREGGERE] Non l'ho mai fatto personalmente. Come impostare una connessione dialup come riserva di una connessione veloce con la caratteristica di autodial? Patch sono ben accolte :)

Capitolo 11

Gli Editor

11.1 Editor più popolari

Linux molte scelte fra gli editor di testo per console. Fra questi elenchiamo:

- `vim`: Editor di eredità BSD potente e leggero. VI iMproved.
- `emacs`: L'editor definitivo di eredità GNU, pesante. Originale da RMS (Richard M. Stallman).
- `xemacs`: "Emacs: The Next Generation", in origine da Lucid.
- `mcedit`: Editor GNU per principianti. Identico all'editor interno di `mc`. Vedere 'L'editor in MC' a pagina 52.
- `ae`: Piccolo editor di default (Potato). Evitatelo.
- `nano`: Piccolo editor GNU di default (Woody). Emula `pico`.
- `joe`: Per i nostalgici di WordStar o TurboPascal.
- `jed`: Editor veloce, ricco di dotazioni, completamente gestito da menu, con comandi da tastiera Emacs.
- `jove`: Editor molto piccolo, con comandi da tastiera Emacs.
- `nvi`: Il nuovo vi. Compatibile baco per baco con il vi originale.

Usate `update-alternatives --config editor` per cambiare l'editor predefinito. In aggiunta, alcuni programmi usano variabili d'ambiente `EDITOR` o `VISUAL` per decidere quale usare. Vedere 'L'editor in MC' a pagina 52.

Degni di nota anche alcuni editor basati su X:

- `gvim`: Vim con GUI (pacchetti `vim` e `vim-gtk`)
- `emacs`: L'Unico Vero Emacs (riconosce automaticamente X).
- `xemacs`: Emacs di nuova generazione (riconosce automaticamente X).

Questi comandi xclient accettano opzioni standard, tipo `-fn` *`a24`*, cosa che rende la vita facile per gente più vecchia come me :) Vedere 'X client' a pagina 150.

11.2 Editor di salvataggio

Esistono alcuni editor che risiedono in `/bin`. Uno di questi dovrebbe essere installato, per facilitare la modifica dei file quando `/usr` non è accessibile.

- `elvis-tiny`: Editor minimo vi (`vi` per lanciarlo)
- `nano-tiny`: Editor minimo non-vi (`nano-tiny` per lanciarlo)
- `nano`: Editor minimo non-vi (`nano` per lanciarlo (Sarge)
- `ed`: Editor minimo (sempre lì, ma difficile da usare)

11.3 Emacs e Vim

11.3.1 Tracce per Vim

Leggete il documento "VIM - file di aiuto principale" premendo **<F1>** mentre usate il programma.

```
<F1>          Aiuto
<esc>         Ritorno al modo normale
V              Modo Visuale
i              Modo Inserimento
:              Comandi da console
:set tw=72     Imposta larghezza testo a 72
<F11>         Modo copia testo
:r! date -R    Inserisce la data come da RFC-822
qa      Registra i tasti premuti nel registro a
q              Ferma la registrazione dei tasti premuti
@a      Esegue i comandi registrati dal registro a
:edit foo.txt  Modifica un altro file caricando foo.txt
:wnext         Scrive il file corrente e modifica il file successivo
```

`q` e `@` possono essere utilizzati per la registrazione e la riproduzione delle macro. Per esempio, per creare una macro che inserisca delle tag HTML in corsivo attorno ad una parola nella sede del cursore, si possono dare i seguenti comandi: `qii<i>^[ea</i>^[q` (dove `^[` è il tasto ESC). Poi, digitando `@i` all'inizio di una parola aggiungerà le tag **<i>** e**</i>**.

Vedere anche 'Uso con Vim' a pagina 247.

11.3.2 Tracce per Emacs

```
<F1>          Aiuto
<F10>         Menu
C-u M-! date -R     Inserisce la data come da RFC-822
```

11.3.3 Lanciare l’editor

lancia l’editor:	emacs nomefile	vim nomefile
lancia un editor vi-compatibile:		vim -C
lancia un editor non vi-compatibile:		vim -N
lancia l’editor in modo compilatore di default:	emacs -q	vim -N -u N

11.3.4 Riassunto dei comandi (Emacs e Vim)

esce:	C-x C-c	:qa /:wq /:xa /:q!
Ritorno/modo comando:	C-g	<esc>
Indietro(sinistra):	C-b	h
Avanti(destra):	C-f	l
Successivo(giù):	C-n	j
Precedente(su):	C-p	k
Inizio riga(^):	C-a	0
Fine riga($):	C-e	$
Comandi multipli:	C-u nnn cmd	:count cmd
Comandi multipli:	M-numero cmd	
Salva File:	C-x C-s	:w file
Inizio buffer:	M-<	1G
Fine Buffer:	M->	G
avanti di una pagina:	C-v	^F
avanti di 1/2 pagina:		^D
avanti di una riga:		^E
indietro di una pagina:	M-v	^B
indietro di 1/2 pagina:		^U
indietro di una riga:		^Y
Scorri alla pagina superiore:	M-C-v	
cancella nel cursore:	C-d	x
cancella dal cursore alla fine della riga:	C-k	D
iCerca in avanti:	C-s	
iCerca indietro:	C-r	
cerca in avanti:	C-s enter	/
cerca indietro:	C-r enter	?
icerca regexp:	M-C-s	
icerca indietro regexp:	M-x isearch-backward-regexp	
cerca regexp:	M-C-s enter	/
cerca indietro regexp:	M-x isearch-backward-regexp enter	?
Aiuto:	C-h C-h	:help
Aiuto Apropos:	C-h a	
Aiuto comandi:	C-h b	:help [key]
Aiuto Info:	C-h i	

```
Aiuto modo Major:               C-h m
Aiuto tutorial:                 C-h t           :help howto
Annulla:                        C-_             u
Ripeti:                         C-f             ^R
Marca la posizione del cursore: C-@             m{a-zA-Z}
Scambia Marcatura e posizione:  C-x C-x
vai al segno nel file corrente:                 ’{a-z}
vai al segno in un altro file:                  ’{A-Z}
copia regione (di testo):       M-w             {visual}y
elimina regione:                C-w             {visual}d
Copia e tieni il buffer:        C-y             p y
Copia e cancella il buffer:     M-y             p
Cambia una regione in maiuscolo:C-x C-u         {visual}U
Cambia una regione in minuscolo:C-x C-l         {visual}u
Inserisci carattere speciale:   C-q ottale/keystroke
                                                ^V decimale/keystroke
sostituisci:                    M-x replace-string      :%s/aaa/bbb/g
sostituisci regexp:             M-x replace-regexp      :%s/aaa/bbb/g
trova e sostituisci:            M-%                     :%s/aaa/bbb/gc
trova e sostituisci:            M-x query-replace
trova e sostituisci regexp:     M-x query-replace-regexp
Apri file:                      C-x C-f         :r file
Salva file:                     C-x C-s         :w
Salva tutti i buffers:          C-x s           :wa
Salva come...:                  C-x C-w file    :w file
Chiede un buffer specifico:     C-x b
Elenca i buffer:                C-x C-b         :buffers
Commuta in sola lettura:        C-x C-q         :set ro
Chiede ed elimina un buffer:    C-x k
Divide lo schermo in verticale: C-x 2           :split
Divide in orizzontale:          C-x 3           :vsplit (ver. 6)
Muove all’altra finestra:       C-x o           ^Wp
Cancella la finestra corrente:  C-x 0           :q
Cancella le/l’altra finestra(e):C-x 1           ^Wo
lancia una shell sullo sfondo:  M-x compile
uccide la shell sullo sfondo:   M-x kill-compilation
lancia make                                     :make Makefile
controlla i msg di errore:      C-x`            :echo errmsg
lancia una shell e registra:    M-x shell       :!script -a tmp
...pulisce BS, ...                                :!col -b <tmp >record
...salva/richiama una
registrazione:                  C-x C-w record  :r record
lancia una shell:               M-! sh          :sh
lancia un comando:              M-! cmd         :!cmd
lancia un comando ed inserisce: C-u M-! cmd     :r!cmd
lancia un filtro:               M-| file        {visual}:w file
```

```
lancia un filtro ed inserisce:  C-u M-| filter  {visual}:!filter
mostra le opzioni:                              :se[t] {option}?
riporta l'opzione al default                    :se[t] {option}&
resetta un'opzione booleana                     :se[t] no{option}
commuta un'opzioe booleana                      :se[t] inv{option}
a capo alla colonna 72                          :se tw=72
niente a capo                                   :se tw=0
autoindentazione                                :se ai
estende tabulazione                             :se et
specifica un commento (posta)                   :se comments=n:>,n:\|

Lancia GDB                      M-x gdb
descrivi il modo GDB            C-h m
salta una riga                  M-s
riga successiva                 M-n
salta una istruzione (stepi)    M-i
Finisci lo stack frame corrente C-c C-f
continua                        M-c
up arg frames                   M-u
down arg frames                 M-d
copia un numero a partire dal punto, inseriscilo alla fine
                                C-x &
Imposta una cesura              C-x SPC
```

11.3.5 Configurare Vim

Per utilizzare tutte le caratteristiche di vim e la sintassi evidenziata, includete le linee seguenti in `~/.vimrc` oppure `/etc/vimrc`:

```
set nocompatible
set nopaste
set pastetoggle=<f11>
syn on
```

Il modo incolla evita che l'autoindentazione interferisca con le operazioni di taglia-e-incolla, se eseguite in terminale. E' molto di più di un semplice ":set noai".

Vedere 'Uso con Vim' a pagina 247 per l'integrazione con GnuPG.

11.3.6 Ctags

`apt-get install exuberant-ctags` e lanciate `ctags` sul file sorgente. Digitando `:tag` *`function_name`* in Vim arrivate alla riga dove inizia *function_name*. Funziona per C, C++, Java, Python e moltri altri linguaggi.

Emacs possiede la medesima funzionalità.

11.3.7 Convertire una schermata da sintassi-evidenziata in sorgente HTML

`so \$VIMRUNTIME/syntax/2html.vim` dal modo comando in Vim converte il testo evidenziato in testo HTML. Salvate con `:w file.html` e `:q`. Utile per codice in C, ecc.

11.3.8 Dividere lo schermo con `vim`

`vim` può aprire più file in un ambiente con uno schermo diviso in più finestre. Digitate `:help usr_08.txt` per i dettagli.

Per dividere lo schermo e mostrare file differenti, digitate al prompt di vi:

```
:split altro-file
:vsplit altro-file
```

o al prompt della shell:

```
$ vi -o file1.txt file2.txt   # Divide orizzontalmente
$ vi -O file1.txt file2.txt   # Divide verticalmente
```

risulterà un vi a molte finestre.

```
$ vimdiff file.txt~ file.txt         # controlla i cambiamenti recenti di file.tx
$ vimdiff file.en.sgml file.fr.sgml  # controlla i cambiamenti fra le traduzioni
$ gvimdiff file.txt~ file.txt        # in X
```

fornisce una buona visione con un file di backup. In SGML accoppia le tag, così fare una comparazione fra traduzioni è molto facile.

Movimenti speciali del cursore con i comandi CTRL-W:

```
CTRL-W +      ingrandisce una finestra
CTRL-W -      rimpicciolisce una finestra
CTRL-W h      va alla finestra di sinistra
CTRL-W j      va alla finestra inferiore
CTRL-W k      va alla finestra superiore
CTRL-W l      va alla finestra di destra
...
```

I controlli seguenti modificano lo scorrimento dello schermo:

```
:set scrollbind
:set noscrollbind
```

Capitolo 12

Sistemi per il controllo versione

12.1 Concurrent Version System (CVS)

Leggete `/usr/share/doc/cvs/html-cvsclient`, `/usr/share/doc/cvs/html-info`, `/usr/share/doc/cvsbook` con `lynx`, od eseguite un `info cvs` e `man cvs` per informazioni dettagliate.

12.1.1 Installare un server CVS

Le impostazioni seguenti permettono il commit solo da parte di un membro del gruppo “src”, e l’amministrazione del cvs solo da parte di un membro del gruppo “staff”, riducendo così le possibilità di mettersi nei guai.

```
# cd /var/lib; umask 002 ; mkdir cvs # [Woody] FSH
# apt-get install cvs cvs-doc cvsbook
# export CVSROOT=/var/lib/cvs
# cd $CVSROOT
# chown root:src .         # mettete "staff" per aumentare le restrizioni.
# chmod 3775 .             # Se si usa "staff", utilizzate 2775
# cvs -d /var/lib/cvs init # è più sicuro specificare l’opzione -d in maniera esp
# cd CVSROOT
# chown -R root:staff .
# chmod 2775 .
# touch val-tags
# chmod 664 history val-tags
# chown root:src history val-tags
```

12.1.2 Esempi di una sessione CVS

Quanto segue imposta l’ambiente di shell per l’accesso al deposito CVS.

CVS anonimo (solo download)

Accesso remoto in sola lettura

```
$ export CVSROOT=:pserver:anonymous@cvs.sf.net:/cvsroot/qref
$ cvs login
$ cvs -z3 co qref
```

Uso del server CVS locale

Accesso locale da una shell sulla stessa macchina:

```
$ export CVSROOT=/var/lib/cvs
```

Uso di un pserver CVS remoto

Accesso remoto senza SSH (usate le capacità del protocollo RSH in `cvs`)

```
$ export CVSROOT=:pserver:account@cvs.foobar.com:/var/lib/cvs
$ cvs login
```

E' vulnerabile ad attacchi tipo eavesdropping.

CVS remoto tramite `ssh`

Accesso remoto con SSH:

```
$ export CVSROOT=:ext:account@cvs.foobar.com:/var/lib/cvs
```

oppure, per SourceForge:

```
$ export CVSROOT=:ext:account@cvs.sf.net:/cvsroot/qref
```

Potete anche utilizzare l'autenticazione RSA ('Connettersi con meno passwords – RSA' a pagina 163), che elimina la necessità della richiesta di password.

Creare un nuovo archivio CVS

Per,

OGGETTO	VALORE	SIGNIFICATO
albero sorgente	~/*progetto-x*	Tutti i codici sorgente
Nome progetto	*progetto-x*	Nome per questo progetto
Vendor Tag:	*Main-branch*	Tag per la branca intera
Release Tag:	*Versione-iniziale*	Tag per una versione specifica

Quindi,

```
$ cd ~/progetto-x             # entra nella directory sorgente
 ... crea un albero sorgente ...
$ cvs import -m "Start progetto-x" progetto-x Main-branch Versione-iniziale
$ cd ..; rm -R ~/progetto-x
```

Lavorare con CVS

Per richiamare e lavorare con le versioni locali del *progetto-x* con l'archivio CVS:

```
$ cd              # muove all'area di lavoro.
$ cvs co progetto-x # copia i sorgenti CVS all'archivio locale
$ cd progetto-x
 ... esegui cambiamenti al contenuto ...
$ cvs diff -u       # simile a diff -u repository/ local/
$ cvs up -C file_modificato      # elimina le modifiche ad un file
$ cvs ci -m Descrivi i cambiamenti   # salva i sorgenti locali nel CVS
$ vi nuovofile_aggiunto
$ cvs add nuovofile_aggiunto
$ cvs ci -m Aggiunto nuovofile_aggiunto
$ cvs up                        # fonde l'ultima versione da CVS
 ... per creare tutte le sottodirectory appena create da CVS, usate
 .... "cvs up -d -P", invece
 ... cercate le righe che iniziano per "C nomefile"
 ... il codice immodificato viene spostato a `.#nomefile.version'.
 ... Cerca "<<<<<<<" e ">>>>>>>" in nomefile.
$ cvs tag Release-1             # aggiunge la tag di versione
... esegui ulteriori modifiche ...
$ cvs tag -d Release-1          # rimuove la tag di versione
$ cvs ci -m altri commenti
$ cvs tag Release-1             # ri-aggiunge la tag di versione
$ cd                            # ritorna all'area di lavoro.
$ cvs co -r Release-initial -d old progetto-x
```

```
 ... riporta la versione originale alla directory old
$ cd old
$ cvs tag -b Release-initial-bugfixes # crea la tag di branca (-b)
 ... Ora si può lavorare sulla vecchia versione (Tag=sticky)
$ cvs update -d -P             # non crea directory vuote
 ... L'albero sorgente ha ora una tag fissa "Release-initial-bugfixes"
 ... Lavorate su questa branca
$ cvs up -d -P # si sincronizza con i file modificati da altri su questa branca
$ cvs ci -m "controllate questa branca"
$ cvs update -kk -A -d -P
 ... Rimuovete la tag fissa e dimenticate il contenuto
 ... Aggiornate la linea principale senza espansione per parola chiave
$ cvs update -kk -d -P -j Release-initial-bugfixes
 ... Fonde la branca Release-initial-bugfixes nella linea
 ... principale senza espansione per parola chiave.  Risolvete i conflitti con l'
$ cvs ci -m fusa Release-initial-bugfixes
$ cd
$ tar -cvzf old-progetto-x.tar.gz old # produce un archivio, -j per bz2
$ cvs release -d old     # rimuove i sorgenti locali (opzionale)
```

Alcune opzioni utili da ricordare (da usare come primi argomenti per `cvs`):

```
-n      esecuzione secca, nessun effetto
-t      mostra messaggi sui passi dell'attività di cvs
```

Esportare i file da CVS

Per ottenere l'ultimissima versione da CVS, usate "tomorrow":

```
$ cvs ex -D tomorrow nome_modulo
```

Amministrare CVS

Aggiungere un alias ad un progetto(server locale):

```
$ su - admin           # un membro del team
$ export CVSROOT=/var/lib/cvs
$ cvs co CVSROOT/modules
$ cd CVSROOT
$ echo "px -a progetto-x" >>modules
$ cvs ci -m Ora px è un alias per progetto-x
$ cvs release -d .
$ exit                 # o Ctrl-D per uscire da su
```

```
$ cvs co -d progetto px
 ... check out progetto-x (alias:px) da CVS alla directory del progetto
$ cd project
 ... modifica il contenuto ...
```

12.1.3 Risoluzione dei problemi

Permessi dei file nel deposito

CVS non sovrascriverà il file corrente, ma lo rimpiazzerà con un altro. Quindi, *il permesso di scrittura nella directory deposito* è critico. Ogni volta che si crea un nuovo deposito, è bene lanciare i seguenti comandi per assicurarsi che detta condizione venga soddisfatta se necessario.

```
# cd /var/lib/cvs
# chown -R root:src deposito
# chmod -R ug+rwX   deposito
# chmod    2775     deposito
                     # se necessario, anche per le sottodirectory
```

Il bit d'esecuzione

Il bit d'esecuzione di un file viene mantenuto alla chiusura della sessione (check-out). Tutte le volte che appaiono problemi di permessi di esecuzione dei file, cambiate i permessi nel deposito CVS con il comando seguente.

```
# chmod ugo-x filename
```

12.1.4 Comandi CVS

Ecco i comandi CVS con le abbreviazioni corrispondenti.

```
{add|ad|new} [-k kflag] [-m 'message'] files...
{admin|adm|rcs} [rcs-options] files...
{annotate|ann} [options] [files...]
{checkout|co|get} [options] modules...
{commit|ci|com}   [-lnR]  [-m  'log_message'  |  -f  file] \
        [-r revision] [files...]
{diff|di|dif} [-kl] [rcsdiff_options] [[-r rev1 | -D date1] \
        [-r rev2 |  -D date2]] [files...]
{export|ex|exp} [-flNn] -r rev|-D date [-d dir] [-k kflag] module...
{history|hi|his} [-report] [-flags] [-options args] [files...]
```

```
{import|im|imp} [-options] repository vendortag releasetag...
{login|logon|lgn}
{log|lo|rlog} [-l] rlog-options [files...]
{rdiff|patch|pa} [-flags] [-V vn] [-r t|-D d [-r t2|-D d2]] modules...
{release|re|rel} [-d] directories...
{remove|rm|delete} [-lR] [files...]
{rtag|rt|rfreeze} [-falnR]  [-b]  [-d]  [-r  tag  |  -D  date] \
        sym_bolic_tag modules...
{status|st|stat} [-lR] [-v] [files...]
{tag|ta|freeze} [-lR] [-F] [-b] [-d] [-r tag | -D date]  [-f] \
        sym_bolic_tag [files...]
{update|up|upd} [-AdflPpR] [-d] [-r tag|-D date] files...
```

12.2 Subversion

Subversion è un sistema di controllo versione di prossima generazione, che è inteso come rimpiazzo di CVS. Gli sviluppatori lo considerano al momento in stadio “alfa”, ma è probabilmente sufficientemente stabile per gran parte degli usi. Al momento della scrittura di questa guida, Subversion è disponibile solo in Debian `unstable`.

12.2.1 Installazione di un server Subversion

Il metapacchetto `subversion-server` dipende dai pacchetti necessari per impiantare un server (`libapache2-dav-svn` e `subversion-tools`).

Impostare un deposito

Attualmente il pacchetto `subversion` non imposta un deposito, per cui bisogna farlo manualmente. Una locazione possibile per un deposito è in `/var/local/repos`.

Create la directory:

```
# mkdir -p /var/local/repos
```

Create il database del deposito:

```
# svnadmin create /var/local/repos
```

Rendete il deposito scrivibile dal server WWW:

```
# chown -R www-data:www-data /var/local/repos
```

Configurare Apache2

Per permettere l'accesso al deposito tramite autenticazione degli utenti, aggiungete quanto segue a `/etc/apache2/mods-available/dav_svn.conf`:

```
<Location /repos>
  DAV svn
  SVNPath /var/local/repos
  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile /etc/subversion/passwd
  <LimitExcept GET PROPFIND OPTIONS REPORT>
    Require valid-user
  </LimitExcept>
</Location>
```

Poi create un file per l'autenticazione degli utenti con il comando:

```
htpasswd2 -c /etc/subversion/passwd username-di-qualcuno
```

Riavviate Apache2 ed il vostro nuovo deposito Subversion sarà accessibile all'URL http://*hostname*/repos.

12.2.2 Muovere un deposito CVS a Subversion

12.2.3 Subversion usage examples

Le sezioni seguenti vi insegnano l'utilizzo dei diversi comandi in subversion.

Creare un nuovo archivio Subversion

Per creare un nuovo archivio Subversion, digitate quanto segue:

```
$ cd ~/il-tuo-progetto         # vai alla directory sorgente
$ svn import http://localhost/repos il-tuo-progetto \
  nome-progetto -m "importazione iniziale progetto"
```

Ciò crea una directory denominata *nome-progetto* nel vostro deposito Subversion che contiene i file del progetto. Guardate in http://localhost/repos/ per vedere se è lì.

Lavorare con Subversion

Lavorare con il *progetto-y* usando Subversion:

```
$ cd                            # muove all’area di lavoro
$ svn co http://localhost/repos/progetto-y  # Copia (checkout) i sorgenti
$ cd progetto-y
 ... lavorateci ...
$ svn diff                      # simile a diff -u repository/ local/
$ svn revert file_modificato      # cancella le modifiche di un file
$ svn ci -m "Descrivi le modifiche"  # copia le vostre modifiche nel deposito
$ vi nuovofile_aggiunto
$ svn add nuovofile_aggiunto
$ svn add nuova_dir          # aggiunge tutti i file ricorsivamente in nuova_dir
$ svn add -N nuova_dir2      # aggiunge la directory non ricorsivamente
$ svn ci -m "Aggiunto nuovofile_aggiunto, nuova_dir,nuova_dir2"
$ svn up                        # aggiorna l’ultima versione dal deposito
$ svn log                       # mostra tutte le modifiche effettuate
$ svn copy http://localhost/repos/progetto-y \
      http://localhost/repos/progetto-y-branca \
      -m "sto creando la mia branca del progetto-y"
          # creazione di una branca del progetto-y
$ svn copy http://localhost/repos/progetto-y \
      http://localhost/repos/prog-y_rilascio1.0 \
      -m "progetto-y rilascio 1.0"    # aggiunta tag di release
 ... notate che il processo di branching e di tagging sono la stessa cosa. L’unic
 ... è che le branche subiscono il commit mentre le tag no.

 ... modificate la branca ...

$ # fa confluire la branca copiata nel tronco principale
$ svn merge http://localhost/repos/progetto-y \
   http://localhost/repos/progetto-y-branca
$ svn co -r 4 http://localhost/repos/progetto-y # prende la revisione 4
```

Capitolo 13

Programmazione

Non usate “test” come nome di un file eseguibile di prova. `test` è un comando interno di shell.

13.1 Dove iniziare

Riferimenti:

- Documenti ed esempi sotto `/usr/share/doc/`*`pacchetti`*
- Unix / Programming Information (`http://arioch.unomaha.edu/~jclark/#info`)
- *Linux Programming Bible* (John Goerzen/IDG books)

Molti documenti info più lunghi possono essere ottenuti rilegati da GNU (`http://www.gnu.org/`).

Le quattro sezioni a seguire contengono dei semplici script, in linguaggi differenti, per creare un file di testo contenente le informazioni di account, da aggiungere a `/etc/passwd`, utilizzando un processore batch come il programma `newusers`. Ogni script richiede come input un file con le righe sotto forma di `nome cognome password`. (Le directory home reali di ciascun utente non vengono create con questi script.)

13.2 Shell

Leggere gli script della shell è il **miglior** modo per comprendere il funzionamento di un sistema tipo Unix. Qui fornisco alcuni indirizzi per la programmazione nella shell. Vedere Errori di Shell (`http://www.greenend.org.uk/rjk/2001/04/shell.html`) per imparare dagli errori.

13.2.1 Bash – la shell interattiva standard GNU

Riferimenti per Bash:

- `bash(1)`
- `info bash`
- l' LDP BASH Programming - Introduction HOWTO (`http://www.tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html`) per iniziare.
- `mc /usr/share/doc/bash/examples/ /usr/share/doc/bash/` (Installate il pacchetto `bash-doc` per vedere dei file di esempio.)
- *Learning the bash Shell*, Seconda edizione (O'Reilly)

Esempio di programma breve (crea delle voci di account per `newusers` a partire da un input standard):

```
#!/bin/bash
# (C) Osmu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
pid=1000;
while read n1 n2 n3 ; do
if [ ${n1:0:1} != "#" ]; then
let pid=$pid+1
echo ${n1}_${n2}:password:${pid}:${pid}:,,,/home/${n1}_${n2}:/bin/bash
fi
done
```

13.2.2 Le shell POSIX

Parecchi pacchetti forniscono una shell POSIX shell in Debian:

- `dash` (Sarge)
  - Priorità: opzionale
  - Dimensioni da installato: 176
  - Il più piccolo ed estremamente veloce — ottimo per il boot d'inizio
- `ash` (Woody)
  - Priorità: opzionale
  - Dimensioni da installato: 180
  - Piccolo e veloce — buono per il boot d'inizio
- `bash`
  - Essenziale: si
  - Priorità: richiesto
  - Dimensioni da installato: 580
  - Più grosso e ricco di funzioni — molte le estensioni implementate
- `pdksh`
  - Priorità: opzionale
  - Dimensioni da installato: 408
  - AT&T completo, somiglia a ksh

Se scrivete uno script che sia portatile, è meglio scriverlo come script POSIX. Usate `/bin/sh` collegato a `ash` o (`dash`) per testarne la complianza con POSIX. Evitate di scrivere gli script con espressioni tipiche di bash o zsh. Evitate, per esempio:

- `if [ foo == bar ] ; then ...`
- `diff -u file.c{.orig,}`
- `mkdir /foo{bar,baz}`

La descrizione per la shell in questo documenti si applica solo per le shell POSIX e perciò non si applica alle shell di tipo `csh`, inclusa `tcsh`.

13.2.3 Parametri della shell

Ecco parecchi **parametri speciali** da ricordare:

```
$0      = nome della shell o dello script
$1      = primo(1) argomento della shell
 ...
$9      = nono(9) argomento della shell
$#      = numero dei parametri posizionali
"$*"    = "$1 $2 $3 $4 ... $n"
"$@"    = "$1" "$2" "$3" "$4" ... "$n"
$?      = stato di uscita del comando più recente
$$      = PID di questo script
$!      = PID del processo più recente lanciato in background
```

Espansioni dei parametri basilari da ricordare:

```
    Forma        Se var è impostata(*)    Se var non è impostata(*)
${var:-stringa}  $var                    stringa
${var:+stringa}  stringa                  null
${var:=stringa}  $var                    stringa
                                        (e lancia var=stringa)
${var:?stringa}  $var                     (echo stringa ed esce)
```

I due punti, qui, ':' in tutti questi operatori sono opzionali.

- Con ':' = l'operatore fa il test per "esiste" e per "non zero".
- Senza ':' = l'operatore fa il test solo per "esiste".

Sostituzioni dei parametri basilari da ricordare:

```
    Forma        Risultato
${var%suffisso}   Rimuove il più piccolo pattern suffisso
${var%%suffisso}  Rimuove il più largo pattern suffisso
${var#prefisso}   Rimuove il più piccolo pattern prefisso
${var##prefisso}  Rimuove il più largo pattern prefisso
```

13.2.4 Redirezione

Redirezione basilare da ricordare (dove [*n*] è un numero opzionale per specificare il descrittore del file):

`[`*n*`]>` *file*	Redirige l'output standard (o n) a *file*.
`[`*n*`]>>` *file*	Aggiunge l'output standard (o n) a *file*.
`[`*n*`]<` *file*	Redirige l'input standard (o n) da *file*.
`[`*n1*`]>&n2`	Redirige l'output standard (o n1) a n2.
`>` *file* `>&2`	Redirige l'output standard e di errore a *file*.
| *comando*	Con una pipe invia l'output standard (o n) a *comando*.
`>&2` | *comando*	Coun una pipe invia l'output standard e di errore a *comando*.
`2>&1` | *command*	Pipe stderr and stdout to *command*.

Dove,
- stdin: standard input (descrittore di file = 0)
- stdout: standard output (descrittore di file = 1)
- stderr: standard error (descrittore di file = 2)

La shell vi permette di aprire i file mediante `exec` con un descrittore di file arbitrario.

```
$ echo Hello >foo
$ exec 3<foo 4>bar  # apre i file
$ cat <&3 >&4       # redirige stdin a 3, stdout a 4
$ exec 3<&- 4>&-    # chiude i file
$ cat bar
Hello
```

Qui *n*`<&-` e *n*`>&-` significano chiudere il descrittore del file *n*.

13.2.5 Condizionali

Ciascun comando ritorna uno **stato di uscita** che può essere utilizzato per le espressioni condizionali:
- Successo: 0 (Vero)
- Errore: 1 - 255 (Falso)

Notate che l'utilizzo del valore 0 per "vero" differisce dalle convenzioni usuali in altre aree di programmazione. In aggiunta, '[' è l'equivalente del comando `test`, che valuta i suoi argomenti fino a ']' come espressione condizionale.

Idiomi condizionali basilari da ricordare sono:

```
comando && se_successo_lancia_anche_questo_comando
comando || se_non_ha_successo_lancia_anche_questo_comando

if [ espressione_condizionale ]; then
```

```
 se_ha_successo_lancia_anche_questo_comando
else
 se_non_ha_successo_lancia_anche_questo_comando
fi
```

Qui, `||` `true` era necessario per assicurarsi che lo script non uscisse a questa riga accidentalmente, quando la shell era invocata con l’opzione `-e`.

Gli operatori per la comparazione dei **file** nelle espressioni condizionali sono:

```
-e file         Vero se file esiste.
-d file         Vero se file esiste ed è una directory.
-f file         Vero se file esiste ed è un file regolare.
-w file         Vero se file esiste ed è scrivibile.
-x file         Vero se file esiste ed è eseguibile.
file1 -nt file2 Vero se file1 è più recente di file2. (modificato)
file1 -ot file2 Vero se file1 è più vecchio di file2. (modificato)
file1 -ef file2 Vero se se sono gli stessi numeri di inode e device.
```

Gli operatori di comparazione delle **stringhe** nelle espressioni condizionali sono:

```
     -z str    Vero se la lunghezza di str è zero.
     -n str    Vero se la lunghezza di str è non-zero.
str1 == str2   Vero se str sono uguali.
str1 = str2    Vero se str sono uguali.
               ("=" dovrebbe essere usato al posto di "==", per una stretta
        complianza POSIX )
str1 != str2   Vero se str non sono uguali.
str1 <  str2   Vero se str1 viene prima di str2 (dipende da locale).
str1 >  str2   Vero se str1 viene dopo str2 (dipende da locale).
```

Gli operatori **aritmetici** di comparazione degli interi nelle espressioni condizionali sono `-eq`, `-ne`, `-lt`, `-le`, `-gt`, o `-ge`.

13.2.6 Processamento delle righe di comando

La shell processa uno script come segue:

- divide in **tokens** sulla base dei metacaratteri: SPACE, TAB, NEWLINE, ;, (,), <, >, |, &
- controlla le **keyword** se non sono contenute fra “. . . ” o ’. . . ’ (loop)
- espande gli **alias** se non sono contenuti fra “. . . ” o ’. . . ’ (loop)
- espande le **parentesi graffe**, A{b|c} -> Ab Ac, se non sono contenute fra “. . . ” o ’. . . ’
- espande le **tilde**, *~user* -> $HOME/$USER, se non sono contenute fra “. . . ” o ’. . . ’
- espande **parametro**, $*PARAMETRO*, se non è contenuto fra ’. . . ’
- espande la **sostituzione del comando**, $(*comando*), se non è contenuta fra ’. . . ’
- divide in **parole** con $IFS se non sono contenute fra “. . . ” o ’. . . ’
- espande il **pathname** *?[] se non è contenuto fra “. . . ” o ’. . . ’
- cerca **comando**

  - funzione
  - built-in
  - file in $PATH
- loop

Le virgolette semplici all'interno delle doppie non hanno effetto.

Eseguire `set -x` nella shell, o invocarla con l'opzione `-x` fa sì che stampi tutti i comandi eseguiti. Molto utile per il debugging.

13.3 Awk

Riferimenti per Awk:

- *Effective awk Programming*, Terza edizione (O'Reilly)
- *Sed & awk*, Seconda edizione (O'Reilly)
- `mawk(1)` e `gawk(1)`
- `info gawk`

Esempio di programma breve (crea delle voci di account per `newusers`):

```
#!/usr/bin/awk -f
# Script per creare un file utilizzabile con il comando 'newusers',
# a partire da un file che contiene user IDs e passwords sotto forma di:
# Nome Cognome password
# Copyright (c) KMSelf Sat Aug 25 20:47:38 PDT 2001
# Distributed under GNU GPL v 2, or at your option, any later version.
# This program is distributed WITHOUT ANY WARRANTY.

BEGIN {
    # Assign starting UID, GID
    if ( ARGC > 2 ) {
        startuid = ARGV[1]
        delete ARGV[1]
    }
    else {
        printf( "Usage:  newusers startUID file\n" \
          "  where:\n" \
          "    startUID is the starting userid to add, and\n" \
          "    file is an input file in form:\n" \
          "      first_name last_name password\n" \
        )
        exit
    }

    infile = ARGV[1]
    printf( "Starting UID: %s\n\n", startuid )
```

```
}

/^#/ { next }

{
    ++record
    first = $1
    last = $2
    passwd = $3
    user= substr( tolower( first ), 1, 1 ) tolower( last )
    uid = startuid + record - 1
    gid = uid
    printf( "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n",  \
        user, passwd, uid, gid, first, last, user \
        )
}
```

Due sono i pacchetti che forniscono il POSIX `awk` in Debian:

- `mawk`
  - Priorità: richiesto
  - Dimensioni da installato: 228
  - Più piccolo è molto più veloce — ottimo per l'installazione base
  - Esiste un limite di tempo di compilazione
    - NF = 32767
    - sprintf buffer = 1020
- `gawk`
  - Priorità: opzionale
  - Dimensioni da installato: 1708
  - Più grande e ricco di funzioni — molte le estensioni implementate
    - versione UNIX System V Release 4
    - awk dei Bell Labs
    - GNU-specifico

13.4 Perl

Questo è **l'** interprete su un sistema simil-Unix.

Riferimenti per Perl:

- `perl(1)`
- *Programming Perl*, Terza edizione (O'Reilly)
- La directory Perl (`http://www.perl.org/`)

Esempio di programma breve (crea delle voci di account per `newusers`):

```
#!/usr/bin/perl
```

```
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
$pid=1000;
while (<STDIN>) {
        if (/^#/) { next;}
        chop;
        $pid++;
        ($n1, $n2, $n3) = split / /;
        print $n1,"_",$n2,":", $n3, ":",$pid,
                  ":",$pid,",,,/home/",$n1,"_",$n2,":/bin/bash\n"
}
```

Installate il modulo Perl *module_name*:

```
# perl -MCPAN -e 'install module_name'
```

13.5 Python

E' un valido interprete object-oriented.

Riferimenti per Python:

- `python(1)`
- *Learning Python* (O'Reilly).
- Python Programming Language (`http://www.python.org/`)

Esempio di programma breve (crea delle voci di account per `newusers`):

```
#! /usr/bin/env python
import sys, string

# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
# Ported from awk script by KMSelf Sat Aug 25 20:47:38 PDT 2001
# This program is distributed WITHOUT ANY WARRANTY.

def usages():
    print \
"Usage:  ", sys.argv[0], " start_UID [filename]\n" \
"\tstartUID is the starting userid to add.\n" \
"\tfilename is input filename. If not specified, standard input.\n\n" \
"Input file format:\n"\
"\tfirst_name lastname password\n"
                return 1

def parsefile(startuid):
    #
```

```
        # main filtering
        #
        uid = startuid
        while 1:
            line = infile.readline()
            if not line:
                break
            if line[0] == '#':
                continue
            (first, last, passwd) = string.split(string.lower(line))
            # above crash with wrong # of parameters :-)
            user = first[0] + last
            gid = uid
            lineout = "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n" %  \
                (user, passwd, uid, gid, first, last, user)
            sys.stdout.write(lineout)
            +uid

    if __name__ == '__main__':
        if len(sys.argv) == 1:
            usages()
        else:
            uid = int(sys.argv[1])
            #print "# UID start from: %d\n" % uid
            if len(sys.argv) > 1:
                infilename   = string.join(sys.argv[2:])
                infile = open(infilename, 'r')
                #print "# Read file from: %s\n\n" % infilename
            else:
                infile = sys.stdin
            parsefile(uid)
```

13.6 Make

Riferimenti per Make:

- `info make`
- `make(1)`
- *Managing Projects with make*, Seconda edizione (O'Reilly)

Semplici variabili automatiche:

Regole di sintassi:

```
    target: [ prerequisites ... ]
     [TAB]  command1
```

```
 [TAB]  -command2 # ignora gli errori
 [TAB]  @command3 # sopprime l'eco
```

Qui `[TAB]` è un codice TAB. Ogni riga viene interpretata dalla shell dopo la sostituzione della variabile da parte di make. Usate `\` alla fine della riga per continuare a capo lo script. Usate `$$` per `$` per le variabili d'ambiente dello shell script.

Le **regole implicite** per *target* e *prerequisites* può essere scritta, per esempio come:

```
%: %.c header.h
```

oppure,

```
%.o: %.c header.h
```

Qui, *target* contiene il carattere `%` (esattamente uno di loro). Il `%` può corrispondere con qualsiasi sottostringa non vuota nei file target attuali. Similmente *prerequisites* usa `%` come i loro nomi si correlano al nome del target attuale.

Le **regole di suffisso** sono il **vecchio** modo di definire le regole implicite per `make`. Sono ancora supportate in GNU `make` per compatibilità, ma usate quando possibile le regole equivalenti del modello:

```
vecchia regola  --> nuova regola
.c:             --> %  : %.c
.c.o:           --> %.o: %.c
```

Variabili automatiche per le summenzionate regole:

```
foo.o: new1.c new2.c old1.c new3.c
$@ == foo.o                         (obiettivo)
$< == new1.c                        (il primo)
$? == new1.c new2.c new3.c          (i più recenti)
$^ == new1.c new2.c.c old1.c new3.c (tutti)
$* == `%' motivo corrispondente al motivo obiettivo.
```

Riferimenti delle variabili:

```
foo1 := bar    # Espansione unica
foo2  = bar    # Espansione ricorsiva
foo3 += bar    # Appendi
SRCS := $(wildcard *.c)
OBJS := $(foo:c=o)
OBJS := $(foo:%.c=%.o)
OBJS := $(patsubst %.c,%.o,$(foo))
DIRS  = $(dir directory/filename.ext) # Estrae la "directory"
$(notdir NAMES...), $(basename NAMES...), $(suffix NAMES...) ...
```

Per vedere le regole interne automatiche, lanciate `make -p -f/dev/null`.

13.7 C

Preparazione:

```
# apt-get install glibc-doc manpages-dev libc6-dev gcc
```

Riferimenti per C:
- `info libc` (riferimento della libreria funzioni in C)
- `gcc(1)`
- `<var>qualsiasi_nome_funzione_libreria_C</var>(3)`
- Kernighan & Ritchie, *The C Programming Language*, Seconda edizione (Prentice Hall).

13.7.1 Un semplice programma in C (gcc)

Un semplice esempio di compilazione di `example.c`, con una libreria `libm` in un eseguibile `run_example`:

```
$ 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
```

Qui, `-l`**`m`** è necessario per il collegamento della libreria `lib`**`m`** per `sqrt()`. La libreria vera è in `/lib` con il nome `libm.so.6`, collegamento simbolico a `libm-2.1.3.so`.

Guardate l'ultimo parametro nel testo risultante. Ci sono più di 10 caratteri, anche se viene specificato `%10s`.

L'uso di puntatori di funzioni che richiedano operazioni in memoria senza controlli sui loro "confini", tipo `sprintf` e `strcpy`, non è considerato sufficiente a prevenire gli effetti di exploit tipo buffer overflow, che annullano gli effetti di overrun. Utilizzate, invece, `snprintf` e `strncpy`.

13.7.2 Debugging

Debugging con `gdb`

Preparazione:

```
# apt-get install gdb
```

Riferimenti per `gdb`:

- `info gdb` (tutorial)
- `gdb(1)`
- `http://www.unknownroad.com/rtfm/gdbtut/gdbtoc.html`

Per il debugging di un programma compilato con l'opzione `-g`, usate `gdb`. Molti comandi possono essere abbreviati. L'espansione del comando mediante tab funziona come per la shell.

```
$ gdb program
(gdb) b 1                # imposta il punto di interruzione alla riga 1
(gdb) run arg1 arg2 arg3 # lancia il programma
(gdb) next               # riga successiva
...
(gdb) step               # un passo avanti
...
(gdb) p parm             # stampa parm
...
(gdb) p parm=12          # imposta il valore di parm a 12
```

Per il debugging da emacs, fate riferimento a 'Riassunto dei comandi (Emacs e Vim)' a pagina 213.

Dato che tutti i binari installati devono essere adattati al sistema Debian in maniera predefinita, gran parte dei simboli di debug sono stati rimossi. Per rendere `gdb` utile per il debug dei pacchetti Debian, i pacchetti pertinenti devono essere ricompilati con la cura seguente:

- Modificate `debian/control` per eliminare il pacchetto version (`http://www.debian.org/doc/debian-policy/ch-controlfields#s-f-Version`).
- Controllate gli script di compilazione ed assicuratevi che usino `CFLAGS=-g -Wall` per compilare i binari.
- Esportate `DEB_BUILD_OPTIONS=nostrip,noopt` per compilare il pacchetto Debian.

Vedere Policy 10.1 (`http://www.debian.org/doc/debian-policy/ch-files#s10.1`) per maggiori informazioni.

Controllo delle dipendenze dalle librerie

Usate `ldd` per scoprire da quali librerie dipende un programma:

```
$ ldd /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)
```

Affinchè `ls` funzioni in un ambiente `chroot` le librerie di cui sopra devono essere disponibili all'interno di esso.

Utili anche i seguenti comandi:
- `strace`: traccia le chiamate di sistema ed i segnali
- `ltrace`: traccia le chiamate alle librerie

Debugging con gli strumenti per scoprire i memory leak

Parecchi sono gli strumenti per il riconoscimento dei memory leak in Debian.
- `njamd`
- `valgrind`
- `dmalloc`
- `electric-fence`
- `memprof`
- `memwatch` (non è impacchettato, prendetelo da memwatch (`http://directory.fsf.org/devel/debug/memwatch.html`).)
- `mpatrol`
- `leaktracer`
- `libgc6`
- `Insure++` da Parasoft (`http://www.parasoft.com`). (non-free, commerciale, a pagamento)

Vedere anche Debugging Tools for Dynamic Storage Allocation and Memory Management (`http://www.cs.colorado.edu/homes/zorn/public_html/MallocDebug.html`).

13.7.3 Flex – un miglior Lex

`flex` è un veloce generatore ed analizzatore lessicale.

Riferimenti per `flex`:
- `info flex` (tutorial)
- `flex(1)`

Al vostro programma dovete fornire una propria `main()` e `yywrap()`, altrimenti `program.l` apparirà come nell'esempio qui sotto, tentando la compilazione senza librerie (`yywrap` è una macro; `%option main` diventa implicitamente `%option noyywrap`):

```
%option main
%%
.|\n    ECHO ;
%%
```

In alternativa, potete compilarlo con l'opzione `-lfl` al termine della riga di comando `cc` (tipo AT&T-Lex con `-ll`). In questo caso nessuna `%option` è richiesta.

13.7.4 Bison – un Yacc migliore

Alcuni pacchetti forniscono un LALR parser generator Yacc-compatibile in Debian:

- `bison`: GNU LALR parser generator
- `byacc`: Il Berkeley LALR parser generator
- `btyacc`: Backtracking parser generator basato su byacc

Riferimenti per `bison`:

- `info bison` (tutorial)
- `bison(1)`

Dovete fornire la vostra `main()` e `yyerror()`. `main()` chiama `yyparse()` che chiama `yylex()`, normalmente creata con FleX.

```
%%

%%
```

13.7.5 Autoconf

`autoconf` è uno strumento per produrre degli shell script in grado di configurare automaticamente il codice sorgente dei programmi usando l'intero sistema di compilazione GNU, adattandoli a molti tipi di sistema simil-Unix.

`autoconf` produce lo script di configurazione `configure`. `configure` crea automaticamente un `Makefile` personalizzato e `Makefile.am`.

Compilare ed installare un programma

Debian non tocca i file in `/usr/local` (vedere 'Supportare le differenze' a pagina 22). Quindi, se compilate un programma dai sorgenti, installatelo in `/usr/local`, così non interferirà con Debian.

```
$ cd src
$ ./configure --prefix=/usr/local
$ make
$ make install # questo comando mette i file nel sistema
```

Disinstallare un programma

Se avete ancora il sorgente e Se utilizza `autoconf/automake` e se ricordate come l'avete configurato:

```
$ ./configure tutte-le-opzioni-che-avevate-dato
# make uninstall
```

In alternativa, se siete assolutamente sicuri che il processo di installazione pone i file solo sotto `/usr/local` e che non c'è nulla di importante lì, potete cancellarne utto il contenuto con:

```
# find /usr/local -type f -print0 | xargs -0 rm -f
```

Se non siete sicuri di dove i file siano installati, dovreste prendere in considerazione l'uso di `checkinstall`, che fornisce un percorso pulito per la disinstallazione.

13.8 Web

Pagine web interattive dinamiche basilari possono essere create come segue:

- Le query vengono presentate al browser mediante form HTML.
- Riempire e cliccare le voci del form invia una URL con i parametri codificati. [1] dal browser al server web. Per esempio:
  - `http://www.foo.dom/cgi-bin/program.pl?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`
  - `http://www.foo.dom/cgi-bin/program.py?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`
  - `http://www.foo.dom/program.php?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`
- Il programma CGI (uno qualsiasi di `program.*`) sul server web riceverà i parametri codificati "`VAR1=VAL1 VAR2=VAL2 VAR3=VAL3`" come contenuto della variabile di ambiente "`QUERY_STRING`" ed esegue sè stesso.
- Lo `stdout` del programma viene inviato al browser e presentato come pagina web dinamica interattiva.

Per motivi di sicurezza è meglio non fare modifiche a mano dei parametri per il processamento CGI. per loro esistono dei moduli già stabiliti in Perl (vedere 'Perl' a pagina 231) e Python (vedere 'Python' a pagina 232). PHP (`http://www.php.net/`) arriva con queste funzionalità. Quando si richiede l'immagazzinamento dei dati del client, si usano i cookie.. Quando ne è richiesto il processamento, su usa frequentemente javascript.

Per altro, vedere The Common Gateway Interface (`http://hoohoo.ncsa.uiuc.edu/cgi/`), The Apache Software Foundation (`http://www.apache.org/`), e JavaScript (`http://www.mozilla.org/js/`).

Cercare su Google "CGI tutorial" digitando l'URL codificata `http://www.google.com/search?hl=en&ie=UTF-8&q=CGI+tutorial` direttamente nella finestra dell'indirizzo del browser, è un buon modo per vedere lo script CGI in azione sul server.

---

[1] Qui, `%nn` è usato per il carattere codificato dell'esadecimale `nn`.

13.9 Preparazione di documenti

13.9.1 `roff`

Tradizionalmente, `roff` è il principale sistema di scrittura testo in Unix.

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)`,ed "`info groff`".

Esiste un buon tutorial sulle macro `-me`. Se avete groff (1.18 o più recente), trovate `/usr/share/doc/groff/meintro.me.gz` e fate quanto segue:

```
$ zcat /usr/share/doc/groff/meintro.me.gz | \
     groff -Tascii -me - | less -R
```

Quanto segue produrrà un file totalmente in formato testo:

```
$ zcat /usr/share/doc/groff/meintro.me.gz | \
    GROFF_NO_SGR=1 groff -Tascii -me - | col -b -x > meintro.txt
```

Per la stampa, usate l'output PostScript.

```
$ groff -Tps meintro.txt | lpr
$ groff -Tps meintro.txt | mpage -2 | lpr
```

13.9.2 SGML

Preparazione:

```
# apt-get install debiandoc-sgml debiandoc-sgml-doc
```

Riferimenti per `debiandoc-sgml`:

- `/usr/share/doc/debiandoc-sgml-doc`
- `debiandoc-sgml(1)`
- *DocBook: The Definitive Guide* (*`/usr/share/doc/docbook-defguide/html/docbook.html`*), di Walsh and Muellner (O'Reilly) (pacchetto `docbook-defguide`)

SGML permette la gestione dei formati multipli dei documenti. Un sistema SGML semplice è Debiandoc, utilizzato qui. Richiede delle conversioni minori dai file di testo originali per i seguenti

- "<" –> `<`
- ">" –> `>`
- " " –> ` ` (spazio non divisibile)
- "&" –> `&`

- “%” –> `%`
- “©” –> `©`
- “–” –> `–`
- “—” –> `—`

Per marcare una sezione come commento non stampabile, date:

```
<!-- Il commento va qui ... -->
```

Per marcare una sezione con un commento modificabile, date:

```
<![ %FIXME [ Il commento va qui ... ]]>
```

In SGML, la *prima definizione* di un’entità vince. Per esempio:

```
<!entity % qref "INCLUDE">
<![ %qref; [ <!entity param "Data 1"> ]]>
<!entity param "Data 2">
&param;
```

Questa termina come “Data 1”. Se la prima riga è, invece, “IGNORE”, questa terminerà come “Data 2” (La seconda riga è un’affermazione condizionale). Anche le frasi ripetute possono essere definite a priori, separatamente dal contesto.

```
<!entity dichièquesto "mio">
Ciao amico &dichièquesto;.
Questo è il &dichièquesto; libro.
```

Che dà ciò come risultato:

```
Ciao amico mio.
Questo è il mio libro.
```

Vedere il breve esempio in SGML `sample.sgml` in esempi (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

Quando i documenti SGML diventano voluminosi, talvolta TeX può dare degli errori. Vedere ‘TeX/LaTeX’ nella pagina seguente.

13.9.3 TeX/LaTeX

Preparazione:

```
# tasksel # scegliete Miscellaneous  --> TeX/LaTeX environment
```

Riferimenti per LaTeX:

- The teTeX HOWTO: The Linux-teTeX Local Guide (`http://www.tldp.org/HOWTO/TeTeX-HOWTO.html`)
- `tex(1)`
- `latex(1)`
- *The TeXbook*, di Donald E. Knuth, (Addison-Wesley) [2]
- *LaTeX - A Document Preparation System*, di Leslie Lamport, (Addison-Wesley)
- *The LaTeX Companion*, di Goossens, Mittelbach, Samarin, (Addison-Wesley)

Qusto è l'ambiente per la scrittura più potente. Molti processori SGML lo usano come loro processore back end. Lyx, fornito dal pacchetto `lyx`, `lyx-xforms`, oppure `lyx-qt` e GNU TeXmacs fornito da `texmacs` offrono dei bei ambienti di scrittura WYSIWYYG per LaTeX, mentre molti usano Emacs e Vim come come edito dei sorgenti.

Molte sono le risorse disponibili in rete:

- teTeX - A Documentation Guide (`/usr/share/doc/texmf/newhelpindex.html`) (il pacchetto `tetex-doc`)
- A Quick Introduction to LaTeX (`http://www.msu.edu/user/pfaffben/writings/`)
- A Simple Guide to Latex/Lyx (`http://www.stat.rice.edu/~helpdesk/howto/lyxguide.html`)
- Word Processing Using LaTeX (`http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex_basic/latex_basic.html`)
- Local User Guide to teTeX/LaTeX (`http://supportweb.cs.bham.ac.uk/documentation/LaTeX/lguide/local-guide/local-guide.html`)

Quando i documenti si ingrandiscono, talvolta TeX può causare degli errori. Dovete aumentare le dimensioni del pool in `/etc/texmf/texmf.cnf` (o, più appropriatamente, modificate `/etc/texmf/texmf.d/95NonPath` e lanciate `update-texmf`) per risolvere il problema.

13.9.4 Literate Programming

Invece di scrivere codice contenente documentazione, il programmatore letterato scrive documentazione contenente codice. Questo approccio assicura una buona documentazione per un programma.

Per saperne di più sul literate-programming, vedere Literate Programming (`http://www.literateprogramming.com/`).

---

[2] I sorgenti TeX di questo libro sono reperibili presso `ftp://ftp.dante.de/pub/tex/systems/knuth/tex/texbook.tex`. `ftp://ftp.dante.de/pub/tex/systems/knuth/lib/manmac.tex` contiene gran parte delle macro richieste. Potete processare questo documento con `tex` dopo aver commentato le righa da 7 a 10, aggiungendo `\input manmac \proofmodefalse`. E' caldamente raccomandato l'acquisto di questo libro (e degli altri di Donald E. Knuth) invece di usare la versione on line, ma il sorgente è un grande esempio di TeX!

Noweb

Preparazione:

```
# apt-get install nowebm
```

Riferimenti per Noweb:

- Noweb — A Simple, Extensible Tool for Literate Programming (http://www.eecs.harvard.edu/~nr/noweb/)
- `noweb(1)`

E' uno strumento per il literate-programming simil-WEB, che è più semplice, e fornisce allo stesso tempo estensibilità e indipendenza dal tipo di linguaggio. [3] Quando si invoca `noweb` scrive il codice sorgente del programma sui file di output menzionati nel file noweb, ed un file TeX per la documentazione stampabile.

Il pacchetto Debian `ifupdown` ne è un raffinato esempio.

```
$ apt-get source ifupdown
$ cd ifupdown*
$ make ifupdown.pdf ifupdown.ps
```

Doxygen

Preparazione:

```
# apt-get install doxygen doxygen-doc doxygen-gui
```

Riferimenti per Doxygen (creato da `doxygen`!):

- Homepage (http://www.doxygen.org/)
- `/usr/share/doc/doxygen-doc/html/index.html`

Può generare documentazione HTML, RTF, pagine di manuale Unix, PostScript, PDF (usando LaTeX) per C++, C, Java, IDL, e programmi PHP e C# in qualche modo. Doxygen è compatibile con JavaDoc (1.1), Qt-Doc, KDOC e è stato disegnato specificatamente per essere usato per i progetti che usano il toolkit Troll Tech Qt (http://www.trolltech.no/qt/). Crea i grafici delle dipendenze di include, diagrammi collaborativi, e grafici di gerarchie di classi grafiche persino per i programmi non documentati. L'output è simile alla documentazione di QT.

---

[3] Questo WEB non ha **niente** a che fare con il World Wide Web. WEB (per PASCAL) e CWEB (per C/C++) sono strumenti tradizionali per il literate-programming.

13.10 Creare pacchetti debian

Preparazione:

```
# apt-get install debian-policy developers-reference \
        maint-guide dh-make debhelper
# apt-get install packaging-manual # se su Potato
```

Riferimenti per il packaging:
- 'Il sistema di gestione dei pacchetti Debian' a pagina 11 (basi)
- Debian New Maintainers' Guide (tutorial)
- `dh-make(1)`
- Debian Developer's Reference (migliore dal punto di vista pratico)
- Debian Policy Manual (autorevole)
- Packaging Manual (Potato)

13.10.1 Impacchettare un singolo binario

Metodo spiccio per impacchettare un singolo binario, da Joey Hess.

```
# mkdir -p mypkg/usr/bin mypkg/DEBIAN
# cp binary mypkg/usr/bin
# cat > mypkg/DEBIAN/control
Package: miopacchetto
Version: 1
Architecture: i386
Maintainer: Joey Hess <joeyh@debian.org>
Description: il mio piccolo pacchetto
Non vi aspettate granchè.
^D
# dpkg-deb -b mypkg
```

13.10.2 Impacchettare con gli strumenti

Usate `dh_make` dal pacchetto `dh-make` per creare un pacchetto base. Poi, procedete secondo le istruzioni contenute in `dh-make(1)`. Queste usano `debhelper` in `debian/rules`.

Un approccio più datato è quello di usare `deb-make` dal pacchetto `debmake`. Non usa nessuno script `debhelper` e dipende esclusivamente dalla shell.

Per degli esempi di pacchetti con sorgenti multipli vedete "mc" (`dpkg-source -x mc_4.5.54.dsc`), che usa "sys-build.mk" di Adam Heath (<`doogie@debian.org`>) e "glibc" (`dpkg-source -x glibc_2.2.4-1.dsc`), che usa un altro sistema di Joel Klecker (<`espy@debian.org`>).

Capitolo 14

GnuPG

Riferimenti:

- `gpg(1)`.
- `/usr/share/doc/gnupg/README.gz`
- *GNU privacy handbook* in `/usr/share/doc/gnupg-doc/GNU_Privacy_Handbook/` (installate il pacchetto `gnupg-doc`)

14.1 Installazione di GnuPG

```
# gpg --gen-key                         # genera una nuova chiave
# gpg --gen-revoke mio_user_ID          # genera una chiave di revoca
per mio_user_ID
# host -l pgp.net | grep www|less       # cerca i pgp keyserver
```

Un buon keyserver predefinito impostato in `$HOME/.gnupg/gpg.conf` (o la vechia locazione `$HOME/.gnupg/options` contiene:

```
keyserver hkp://subkeys.pgpg.net
```

Bisogna fare attenzione a **non** creare più di due sottochiavi, altrimenti i keyservers su pgp.net **corromperanno** la vostra chiave. Usate il `gnupg` più recente (>1.2.1-2) per maneggiare queste sottochiavi corrotte. Vedere `http://fortytwo.ch/gpg/subkeys`.

14.2 Uso di GnuPG

Gestione dei file:

```
$ gpg  [opzioni]  comando  [argomenti]
```

```
$ gpg {--armor|-a} {--sign|-s} file # firma un file in file testo file.asc
$ gpg --clearsign file              # firma un messaggio
$ gpg --clearsign --not-dash-escaped patchfile  # firma in chiaro una patch
$ gpg --verify file                 # verifica un file firmato in chiaro
$ gpg -o file.sig {-b|--detach-sign} file    # crea una firma staccata
$ gpg --verify file.sig file # verifica file con file.sig
$ gpg -o crypt_file {--recipient|-r} nome {--encrypt|-e} file
        # Criptatura mediante chiave pubblica destinata a nome
$ gpg -o crypt_file {--symmetric|-c} file # criptatura simmetrica
$ gpg -o file --decrypt crypt_file  # decrittatura
```

14.3 Gestione di GnuPG

Gestione delle chiavi:

```
$ gpg --edit-key ID_utente  # "help" per aiuto, interattivo
$ gpg -o file --exports     # esporta tutte le chiavi in file
$ gpg --imports file        # importa tutte le chiavi da file
$ gpg --send-keys ID_utente # invia le chiavi di ID_utente al keyserver
$ gpg --recv-keys ID_utente # recupera la chiave di ID_utente dal keyserver
$ gpg --list-keys ID_utente # elenca le chiavi dello ID_utente
$ gpg --list-sigs ID_utente # elenca le firme dello ID_utente
$ gpg --check-sigsID_utente # verifica le firme dello ID_utente
$ gpg --fingerprintID_utente # verifica il fingerprint dello ID_utente
$ gpg --list-sigs | grep '^sig' | grep '[User id not found]' \
  | awk '{print $2}' | sort -u | xargs gpg --recv-keys
  # prende le chiavi sconosciute
  # aggiorna le chiavi per tutte le firme sconosciute.
$ gpg --refresh-keys                # aggiorna il keyring locale
```

Codice di confidenza:

```
-         Nessun codice assegnato / non ancora calcolato.
e         Il calcolo della confidenza è fallito.
q         Informazioni insufficienti per il calcolo.
n         Non confidare mai su detta chiave.
m         Confidenza parziale.
f         Confidenza totale.
u         Confidenza finale.
```

Quanto segue carica la mia chiave *“A8061F32”* al keyserver popolare `hkp://subkeys.pgp.net`:

```
$ gpg --keyserver hkp://subkeys.pgp.net --send-keys A8061F32
```

14.4 Usare GnuPG con altre applicazioni

14.5 Uso con Mutt

Aggiungete le righe seguenti a `~/muttrc` per evitare che uno GnuPG troppo lento venga lanciato automaticamente, ma solo quando necessario, premendo ‘`S`’ al menu.

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

14.5.1 Uso con Vim

Aggiungete il contenuto di `_vimrc`, ottenuto dalla sottodirectory degli esempi (`http://www.debian.org/doc/manuals/debian-reference/examples/`) in `~/.vimrc` per far girare GnuPG in modo trasparente.

Capitolo 15

Supporto per Debian

Per ottenere aiuto, consigli e supporto per Debian si possono seguire i riferimenti seguenti. Prima di chiedere aiuto (magari gridando) in lista, sarebbe bene fare del proprio meglio per utilizzare le fonti che uno ha a disposizione :)

Notate che potete accedere ad una buona fetta della documentazione presente sul vostro sistema tramite un banale browser WWW, tramite i comandi “dwww” o “dhelp”, che trovate nei rispettivi pacchetti.

15.1 Riferimenti

I seguenti riferimenti sono disponibili per Debian e Linux in generale. Qualora trovaste informazioni contrastanti, fidatevi sempre della fonte primaria di informazioni, e non delle fonti secondarie come, ad esempio, questo documento.

- Manuale di installazione (fonte primaria)
  - Da leggere prima di installare ed aggiornare.
  - Web: `http://www.debian.org/releases/stable/installmanual`
  - Web: `http://www.debian.org/releases/testing/installmanual` (lavoro in corso, talvolta non esiste)
  - Pacchetto: `Not available in install-doc: Bug#155374`
  - File: `DebianCDunder/doc/`
- Note di rilascio (primaria)
  - Da leggere obbligatoriamente prima di installare ed aggiornare, anche se pensate di sapere tutto.
  - Web: `http://www.debian.org/releases/stable/releasenotes`
  - Web: `http://www.debian.org/releases/testing/releasenotes` (lavoro in corso, talvolta non esiste)
  - Pacchetto: `Not available in install-doc: Bug#155374`
  - File: `DebianCDunder/doc/`

- FAQ (secondaria)
    - Domande più frequenti (non aggiornatissime)
    - Web: `http://www.debian.org/doc/manuals/debian-faq/`
    - Pacchetto: `doc-debian`
    - File: `/usr/share/doc/debian/FAQ/index.html`

- La Guida Debian (secondaria)
    - Manuale utente post-installazione molto completo
    - Web: `http://www.debian.org/doc/manuals/debian-reference/`
    - Pacchetto: `debian-reference-it`
    - File: `/usr/share/doc/Debian/reference/`

- APT HOWTO (secondaria)
    - Guida utente dettagliata per il sistema di gestione dei pacchetti in Debian (woody)
    - Web: `http://www.debian.org/doc/manuals/apt-howto/`
    - Pacchetto: `apt-howto`
    - File: `/usr/share/doc/Debian/apt-howto/`

- Securing Debian Manual (secondaria)
    - Guita utente per rafforzare e rendere sicura l'installazione Debian predefinita. (Woody)
    - Web: `http://www.debian.org/doc/manuals/securing-debian-howto/`
    - Pacchetto: `harden-doc`
    - File: `/usr/share/doc/harden-doc/html/securing-debian-howto/`

- `dselect` Documentazione per principianti (secondaria)
    - Corso per `dselect`
    - Web: `http://www.debian.org/releases/woody/i386/dselect-beginner`
    - Pacchetto: `Not available in install-doc: Bug#155374`
    - File: `DebianCDunder/doc/`

- Debian Policy Manual (primaria)
    - Testo di riferimento Debian.
    - Web: `http://www.debian.org/doc/debian-policy/`
    - Pacchetto: `debian-policy`
    - File: `/usr/share/doc/debian-policy/`

- Debian Developer's Reference (primaria)
    - Informazioni base per sviluppatori.
    - Il resto di noi dovrebbe dargli una scorsa almeno una volta.
    - Web: `http://www.debian.org/doc/manuals/developers-reference/`
    - Pacchetto: `developers-reference`
    - File: `/usr/share/doc/developers-reference/`

- Nuova guida del manutentore Debian (primaria)
    - Guida pratica per sviluppatori.

  - Corso di “pacchettizzazione” per comuni mortali.
  - http://www.debian.org/doc/manuals/maint-guide/
  - Pacchetto: `maint-guide`
  - File: `/usr/share/doc/maint-guide/`

- Manuale per la produzione dei pacchetti (Potato)
  - Pacchetto: `packaging-manual` in Potato. (Mosso in appendice a *Developer's Reference*

- Pagine di manuale stile Unix (primaria)
  - `dlocate -man` *`package-name`* (elenco disponibili)
  - `man` *`section command-name`*

- GNU-style info pages (primary)
  - `info` (access top level)
  - `info` *`nome-comando`*

- Documenti specifici dei pacchetti (primaria)
  - Li trovate in `/usr/share/doc/<package-name>`

- LDP: Linux Documentation Project (secondaria)

  - HOWTO e mini-HOWTO per Linux in genere
  - Web: http://www.tldp.org/
  - Pacchetto: `doc-linux-text` e `doc-linux-html`
  - File: `/usr/share/doc/HOWTO/`

- Linux Gazette (secondaria) – uscite mensili
  - The Linux Gazette
  - Web: http://www.linuxgazette.com/
  - Pacchetto: `lg-all` o `lg-latest-two`
  - File: `/usr/share/doc/lg/`

- DDP: Debian Documentation Project (secondaria)
  - Manuali specifici per Debian
  - Web: http://www.debian.org/doc/

- Debian Developers' Corner (secondaria)
  - Informazioni chiave per gli sviluppatori Debian
  - Approfondimento per l'utente finale
  - Web: http://www.debian.org/devel/

- Codice sorgente (assolutamente primaria)

  - Nessuno può contestarlo :-)
  - Scaricare il codice seguendo ‘Il codice sorgente’ a pagina 11

- Internet Assigned Numbers Authority (primaria)
  - Web: `http://www.iana.org/`
  - Pacchetto: `doc-iana`
  - File: `/usr/share/doc/doc-iana/`

- Internet requests for comments (IETF standard) (primaria)
  - Web: `http://www.ietf.org/rfc.html`
  - Pacchetto: `doc-rfc`
  - File: `/usr/share/doc/RFC/`

I riferimenti seguenti sono disponibili per Unix in generale. Notate che esistono alcune minime differenze fra i diversi sistemi Unix. I nomi dei dispositivi ed i metodi di avvio richiedone attenzione supplementare.

- *The UNIX Programming Environment*
  - Il libro da leggere per imparare il funzionamento di UNIX
  - Di B. W. Kernighan e R. Pike
  - Pubblicato da Princeton Hall Software Series

- *The C Programming Language* (seconda edizione)
  - Il libro da leggere per imparare ANSI C
  - Di B. W. Kernighan e D. M. Ritchie
  - Pubblicato da Princeton Hall Software Series

- *UNIX Power Tools*
  - Il libro da leggere per imparare i trucchi per Unix
  - Di Jerry Peek, Tim O'Reilly, e Mike Loukides
  - Pubblicato da O'Reilly and Associates

- *Essential System Administration* (seconda edizione)
  - Il libro da leggere per imparare l'amministrazione di sistema per varie vesioni di Unix.
  - Di Aeleen Frisch
  - Pubblicato da O'Reilly and Associates

- *Linux: Rute User's Tutorial and Exposition*
  - Un bel libro e versione on line che copre l'amministrazione dei sistemi GNU/Linux.
  - Di Paul Sheer
  - Pubblicato da Prentice Hall
  - Web: `http://www.icon.co.za/~psheer/book/index.html.gz`
  - Pacchetto: `rutebook` (da `non-free`)
  - File: `/usr/share/doc/rutebook/`

- Bell Labs: Computing Sciences Research
  - Ricco archivio di storia Unix
  - Principale: `http://cm.bell-labs.com/cm/cs/`

  - Rapporti tecnici selezionati: `http://cm.bell-labs.com/cm/cs/cstr.html`
  - Alcuni articoli: `http://cm.bell-labs.com/cm/cs/papers.html`

- Risorse generali per Linux in rete
  - Debian Planet (`http://www.debianplanet.org/`)
  - debianHELP (`http://www.debianhelp.org/`)
  - Linux.com (`http://linux.com/`)
  - La home page Linux presso Linux Online (`http://www.linux.org/`)
  - Red Hat (venditore commerciale di Linux) (`http://www.redhat.com/`) (RPM, Sys-V init)
  - SuSE, Inc. (venditore commerciale di Linux) (`http://www.suse.de/`) (RPM, Sys-V init)
  - Slackware (`http://www.slackware.com/`) (TGZ, BSD-style init)

- Risorse generali per Unix in rete
  - The UNIX System by The Open Group (`http://www.unix.org/`)
  - A UNIX Introductory Course from Ohio State University (`http://www-wks.acs.ohio-state.edu/unix_course/unix.html`)
  - UNIXhelp from The University of Edinburgh (`http://unixhelp.ed.ac.uk/`)
  - Unix / Programming Information (`http://arioch.unomaha.edu/~jclark/#info`)
  - comp.unix.questions FAQ (`http://www.faqs.org/faqs/unix-faq/faq/`)
  - comp.unix.user-friendly FAQ (`http://www.camelcity.com/~noel/usenet/cuuf-FAQ.htm`)
  - FreeBSD Documentation (`http://www.freebsd.org/docs.html`)
  - The FreeBSD Handbook (`http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/index.html`)
  - UNIX GUIDE (`http://ieee.uow.edu.au/documents/`)
  - The Unix Heritage Society (`http://www.tuhs.org/`)

- Home page dei progetti di software libero
  - GNU Project (`http://www.gnu.org/`)
  - The Linux Documentation Project (`http://www.tldp.org/`)
  - The Linux Kernel Archives (`http://www.linux.org/`)
  - The XFree86 Project, Inc (`http://www.xfree86.org/`)
  - GNOME (`http://www.gnome.org/`)
  - K Desktop Environment (`http://www.kde.org/`)
  - GNU software at Red Hat (`http://sources.redhat.com/`)
  - Mozilla (`http://www.mozilla.org`)
  - FreeBSD (`http://www.freebsd.org/`)
  - OpenBSD (`http://www.openbsd.org/`)
  - NetBSD (`http://www.netbsd.org/`)

15.2 Trovare il significato di una parola

Molti termini usati in Debian sono criptici. In più vengono utilizzati anche molti acronimi. Il seguente comando risolverà molti dubbi:

```
$ dict <metti-qui-un-termine-sconosciuto>
```

15.3 Scoprire la popolarità di un pacchetto Debian

I pacchetti in Debian sono numerosi ed è talvolta difficile sapere quale provare per primo. Vedere Debian Popularity Contest Results (`http://www.debian.org/~apenwarr/popcon/`) per uno sguardo su cosa usano gli altri. Installate anche il pacchetto `popularity-contest` per contribuire.

15.4 Il sistema Debian di tracciamento dei bachi

La distribuzione Debian ha un bug tracking system (BTS) (`http://bugs.debian.org/`) che archivia i dettagli dei bachi riportati dagli utenti e dagli sviluppatori. Ad ogni baco viene assegnato un numero, e viene mantenuto archiviato finchè non è segnato come risolto.

Prima di segnalare un baco, dovreste controllare se qualcun altro non lo ha già fatto. La lista dei bachi più importanti è disponibile in World Wide Web (`http://bugs.debian.org/`) and altrove (`http://www.debian.org/Bugs/Access`). Vedere anche 'Controllo dei bachi in Debian e ricerca di aiuto' a pagina 85.

Possono esserci molte segnalazioni di bachi release-critical marcate con **FTBFS**. Ciò significa "Fails To Build From Source".

Il metodo per riportare i bachi è descritto in `http://www.debian.org/Bugs/Reporting`

15.5 Liste di messaggi

Per tenervi aggiornati, leggete almeno debian-devel-announce (Inglese, sola lettura, basso traffico)

Le liste di principale interesse per l'utente Debian sono la debian-user (Inglese, aperta, alto traffico) e la "debian-user-*language*" (per le altre lingue)

Per informazioni e dettagli sulla iscrizione vedere `http://lists.debian.org/`. Controllate sempre gli archivi prima di porre domande, ed attenetevi agli standard di etichetta.

Se non volete ricevere copie delle risposte che inviate alla vostra lista, usate l'intestazione `Mail-Followup-To:`, che è una misura molto efficace. Questa è la convenzione informale delle liste di messaggi, come spiegato in `http://cr.yp.to/proto/replyto.html`.

15.6 Internet Relay Chat (IRC)

IRC (Internet Relay Chat) è un modo per parlare con le persone in tutto il mondo in tempo reale. I canali IRC dedicati a Debian possono essere trovati sul network IRC freenode (`http://www.freenode.info/`). Per collegarvi avete bisogno di un client IRC. Tra i più popolari trovate XChat, BitchX, ircII, irssi, epic4 e KSirc, tutti pachetti Debian. Una volta istallato, dovete dire al client di connettersi al server. Con molti client fate questo digitando:

```
/server irc.debian.org
```

Una volta connessi, unitevi al canale #debian digitando

```
/join #debian
```

Per lasciare il canale, digitate

```
/part #debian
```

Potete chiudere il client digitando

```
/quit
```

Per inviare un messaggio privato "Hello Mr. Foo" a *foo*, digitate

```
/msg foo Hello Mr. Foo
```

Notate che qualsiasi cosa digitiate senza essere preceduta da / viene inviata al canale come messaggio

Nota: client come XChat hanno spesso un'interfaccia grafica differente per collegarsi ai server/canali.

15.7 Motori di ricerca

Molti sono i motori di ricerca che cercano anche tra la documentazione relativa a Debian:

- La pagina Debian per la ricerca sul WWW (`http://search.debian.org/`).
- Google (`http://www.google.com/`): includere "site:debian.org" come termine di ricerca.

- Google Groups (`http://groups.google.com/`): motore di ricerca per newsgroup. Includere “linux.debian.user” come termine di ricerca.
- AltaVista (`http://www.altavista.com/`)

Per esempio, la ricerca su una stringa tipo “cgi-perl” fornisce una spiegazione molto più dettagliata su questo pacchetto rispetto alla breve descrizione del suo file di controllo. Vedere per esempio ‘Controllo dei bachi in Debian e ricerca di aiuto’ a pagina 85

15.8 Siti Web

Di seguito alcuni indirizzi casuali da me raccolti per argomenti specifici.

- IBM developerWorks: Linux (`http://www.ibm.com/developerworks/linux/`)
- I pacchetti di Adrian Bunk più recenti (back port a stable) (`http://www.fs.tum.de/~bunk/`)
- Linux on Laptops (`http://www.linux-laptop.net/`)
- Xterm FAQ (`http://dickey.his.com/xterm/xterm.faq.html`)
- EXT3 File System mini-HOWTO (`http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html`)
- Large File Support in Linux (`http://www.suse.de/~aj/linux_lfs.html`)
- Window Manager per X (`http://www.xwinman.org`)
- Linux USB Project (`http://www.linux-usb.org/`)
- Pagine SuSE per CJK (`http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html`)
- LNX-BBC (Business-card-sized boot CD project) (`http://www.lnx-bbc.org/`)
- Linux info di Karsten Self (Partitionamento, backup, browser. . .) (`http://kmself.home.netcom.com/Linux/index.html`)
- Backup info HOWTO di Alvin Oga (`http://www.Linux-Backup.net/`)
- Security info HOWTO di Alvin Oga (`http://www.Linux-Sec.net/`)
- Varie fonti NON-UFFICIALI per APT (`http://www.apt-get.org/`)
- Laptop Ethernet Configuration (`http://www.orthogony.com/gjw/lap/lap-ether-intro.html`)

Appendice A

Appendice

A.1 Autori

La guida Debian è stata iniziata da Osamu Aoki `<osamu@debian.org>` come memo personale di installazione e successivamente è stata chiamata "Quick Reference . . . ". Buona parte del contenuto proviene dagli archivi della mailing list debian-user. Anche la "Debian – Installation Manual" e le "Debian – Release Notes" sono state utilizzate.

Seguendo il consiglio di Josip Rodin, che è molto attivo al Debian Documentation Project (`http://www.debian.org/doc/ddp`) (DDP) ed è anche l'attuale responsabile della "Debian FAQ", questo documento è stato rinominato "La guida Debian" ed è stato fuso con parecchi capitoli provenienti dalla "Debian FAQ" con un contenuto simil-bibliografico. Da qui "La guida Debian rapida" è stato formato come un estratto.

Questo documento stato scritto, tradotto ed espanso dai seguenti membri del gruppo QREF:

- Per il documento originale inglese "Quick Reference. . . "
    - Osamu Aoki `<osamu@debian.org>` (responsabile per tutto il progetto)
- Per la correzione delle bozze e la riscrittura inglese
    - David Sewell `<dsewell@virginia.edu>` (responsabile per la sezione en)
    - Thomas Hood `<jdthood@yahoo.co.uk>` (relativo alla rete)
    - Brian Nelson `<nelson@bignachos.com>`
    - Jan Michael C Alonzo `<jmalonzo@spaceants.net>`
    - Daniel Webb `<webb@robust.colorado.edu>`
    - Suggerimenti da tutti i traduttori
- Traduzione francese
    - Guillaume Erbs `<gerbs@free.fr>` (responsabile sez. fr)
    - Rénald Casagraude `<rcasagraude@interfaces.fr>`
    - Jean-Pierre Delange `<delange@imaginet.fr>`
    - Daniel Desages `<daniel@desages.com>`
- Traduzione italiana
    - Davide Di Lazzaro `<mc0315@mclink.it>` (responsabile sez. it)
- Traduzione Portoghese
    - Paulo Rogério Ormenese `<pormenese@uol.com.br>` (responsabile sez. pt-br)

  - Andre Luis Lopes <`andrelop@ig.com.br`>
  - Marcio Roberto Teixeira <`marciotex@pop.com.br`>
  - Rildo Taveira de Oliveira <`to_rei@yahoo.com`>
  - Raphael Bittencourt Simoes Costa <`raphael-bsc@bol.com.br`>
  - Gustavo Noronha Silva <`kov@debian.org`> (coordinator)
- Traduzione Spagnola
  - Walter Echarri <`wecharri@infovia.com.ar`> (responsabile sez. es)
  - José Carreiro <`ffx@urbanet.ch`>
- Traduzione Tedesca
  - Jens Seidel <`tux-master@web.de`> (responsabile sez. de)
  - Willi Dyck <`wdyck@gmx.net`>
  - Stefan Schröder <`stefan@fkp.uni-hannover.de`>
  - Agon S. Buchholz <`asb@kefk.net`>
- Traduzione Polacca—i seguenti membri del PDDP (`http://debian.linux.org.pl`):
  - Marcin Andruszkiewicz
  - Mariusz Centka <`mariusz.centka@debian.linux.org.pl`>
  - Bartosz Fenski <`fenio@debian.linux.org.pl`> (resp. sez. pl)
  - Radoslaw Grzanka <`radekg@debian.linux.org.pl`>
  - Bartosz 'Xebord' Janowski
  - Jacek Lachowicz
  - Rafal Michaluk
  - Leonard Milcin, Jr.
  - Tomasz Z. Napierala <`zen@debian.linux.org.pl`>
  - Oskar Ostafin <`cx@debian.linux.org.pl`>
  - Tomasz Piekos
  - Jacek Politowski
  - Mateusz Prichacz <`mateusz@debian.linux.org.pl`>
  - Marcin Rogowski
  - Pawel Rozanski
  - Mariusz Strzelecki
  - Krzysztof Scierski
  - Przemyslaw Adam Smiejek <`tristan@debian.linux.org.pl`>
  - Mateusz Tryka <`uszek@debian.linux.org.pl`>
  - Cezary Uchto
  - Krzysztof Witkowski <`tjup@debian.linux.org.pl`>
  - Bartosz Zapalowski <`zapal@debian.linux.org.pl`>
- Traduzione Cinese (semplificato)
  - Hao "Lyoo" LIU <`iamlyoo@163.net`> (resp.: zh-cn)
  - Ming Hua <`minghua@rice.edu`>
- Traduzione Cinese (tradizionale)
  - Asho Yeh <`asho@debian.org.tw`> (resp.: zh-tw)
  - Tang Wei Ching <`wctang@csie.nctu.edu.tw`> (ex resp.: zh-tw)
- Traduzione Giapponese
  - Shinichi Tsunoda <`tsuno@ngy.1st.ne.jp`> (resp.: ja)
  - Osamu Aoki <`osamu@debian.org`>

QREF è l'abbreviazione del titolo originale del documento, "Quick Reference. . . " ed è anche il nome del progetto presso qref.sourceforge.net.

Per scrivere questo documento sono state utilizzate molte pagine di manuale e pagine info del sistema Debian. Nella misura in cui Osamu Aoki le ha considerate delle semplici citazioni, di molte parti, in particolare le definizioni dei comandi, ne sono state utilizzate intere frasi, dopo accurati sforzi editoriali per inserirle nel contesto di stile ed obiettivi di questo documento.

Gran parte del contenuto del capitolo 'Nozioni fondamentali della Debian' a pagina 5 proviene dalla "Debian FAQ" (Marzo 2002):

- 5. Gli archivi Debian FTP: `ftparchives.sgml` (tutto il capitolo)
- 6. Le basi del Sistema di Gestione dei Pacchetti Debian: `pkg_basics.sgml` (tutto il capitolo)
- 7. Gli strumenti del Sistema di Gestione dei Pacchetti Debian: `pkgtools.sgml` (tutto il capitolo)
- 8. Mantenere aggiornato il vostro sistema Debian: `uptodate.sgml` (tutto il capitolo)
- 9. Debian ed il kernel: `kernel.sgml` (tutto il capitolo)
- 10. Personalizzare l'installazione del vostro Debian GNU/Linux: `customizing.sgml` (parte del capitolo)

Queste sezioni della "Debian FAQ" sono state incluse nel documento dopo una importante riorganizzazione, per riflettere i recenti cambiamenti del sistema Debian. Il contenuto di questo documento è più recente.

La "Debian FAQ" originale era stata creata e mantenuta da J.H.M. Dassen (Ray) e Chuck Stickelman. Gli autori della "Debian FAQ" riveduta e corretta sono Susan G. Kleinmann e Sven Rudolph. Dopo di loro, la "Debian FAQ" è stata mantenuta da Santiago Vila. Il responsabile attuale è Josip Rodin.

Parte delle informazioni per la "Debian FAQ" proviene:

- L'annuncio del rilascio della Debian-1.1, di Bruce Perens (`http://www.perens.com/`).
- La Linux FAQ, di Ian Jackson (`http://www.chiark.greenend.org.uk/~ijackson/`).
- Gli archivi della lista di posta Debian (`http://lists.debian.org/`),
- il dpkg programmers' manual ed il Debian Policy manual (vedere 'Riferimenti' a pagina 249)
- tanti sviluppatori, volontari, beta testers, e la
- memoria traballante dei suoi autori. :-)

Alcune parti della sezione "Lezioni" sono derivate da

- "Debian Tutorial" di Havoc Pennington, Oliver Elphick, Ole Tetlie, James Treacy, Craig Sawyer, e Ivan E. Moore II. (Il documento è derivato a sua volta da "Linux User's Guide" di Larry Greenfield.)
- "Debian GNU/Linux: Guide to Installation and Usage" di John Goerzen e Ossama Othman.

Gli autori desiderano ringraziare tutti coloro che hanno dato il loro contributo per rendere questo documento possibile.

A.2 Garanzie

Dato che non sono un esperto, non pretendo di possedere la verità assoluta su Debian o Linux in generale. Le considerazioni sulla sicurezza che faccio sono solamente applicabili per un uso casalingo.

Questo documento non rimpiazza alcuna guida di riferimento.

Non esistono garanzie. Tutti i marchi registrati appartengono ai rispettivi proprietari.

A.3 Feedback

Commenti ed aggiunte a questo documento sono sempre i benvenuti. Mandate una mail al Sistema Debian di tracciamento dei bachi (`http://bugs.debian.org/`) relativa al pacchetto `debian-reference`, o ai pacchetti delle rispettive trauzioni. L'uso di `reportbug` semplifica l'invio di un rapporto di un baco. Potete sempre mandare una email a Osamu Aoki (`http://people.debian.org/~osamu/`) at `<osamu@debian.org>` in inglese, oppure ai traduttori delle rispettive lingue.

Dato che non sono di lingua madre inglese, qualsiasi correzione della grammatica è la benvenuta.

La cosa migliore è un diff per la versione SGML, ma anche per la versione testo è accettabile. Vedere 'Il documento ufficiale' a pagina 1 per il sito ufficiale del documento.

I file originali in SGML utilizzati per la creazione di questo documento sono disponibili anche in CVS presso: `:pserver:anonymous@cvs.sf.net/cvsroot/qref` oppure `http://qref.sourceforge.net/Debian/qref.tar.gz`.

A.4 Formato del documento

Il documento è stato scritto utilizzando il DebianDoc SGML DTD (riscritto da LinuxDoc SGML). Il sistema DebianDoc SGML ci permette di creare dei files in formati diversi, a partire da un unico sorgente, ovvero potete leggere questo documento come HTML, testo, TeX DVI, PostScript, PDF, o GNU info.

Le utilità per la conversione da DebianDoc SGML sono disponibili nel pacchetto Debian `debiandoc-sgml`.

A.5 Il labirinto di Debian

Il sistema Linux è una piattaforma molto potente per un PC in rete. Tuttavia, imparare a padroneggiare tutti i suoi strumenti non è così semplice. Pensate, ad esempio, alla configurazione della stampante.

Esiste una mappa di questo sistema, completa e dettagliata, chiamata “CODICE SORGENTE”. E’ una mappa molto accurata, ma altrettanto difficile da comprendere. Esistono anche fonti di informazione chiamate HOWTO e mini-HOWTO, più facili da comprendere, che però tendono a dare forse troppi dettagli, perdendo di vista la visione d’insieme. Mi capita talvolta, quando ho bisogno di trovare dei comandi da usare, di avere dei problemi a trovare la sezione giusta in un lungo HOWTO.

Proprio per viaggiare attraverso i meandri della configurazione di un sistema Linux, iniziai a scrivere delle semplici note in formato testo, da utilizzare come riferimenti rapidi. L’elenco delle note crebbe sempre più con il passare del tempo, e nel frattempo imparai debiandoc. Il risultato finale è questa *La guida Debian*.

A.6 Citazioni Debian

Ecco alcune citazioni interessanti, prese dalla mailing list Debian

- “Questo è Unix. Ti dà abbastanza corda per impiccarti da solo.” — Miquel van Smoorenburg `<miquels@cistron.nl>`
- “Unix **E’** user friendly… E’ solo selettivo su chi può essergli amico.” — Tollef Fog Heen `<tollef@add.no>`