

I MANUALI di

LINUX
PRO

Server & Networking

Guida completa all'amministrazione
di server Internet
e all'integrazione di reti miste

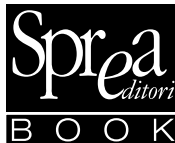
DAL SERVER WEB A SAMBA



Spr
a
ditori
BOOK

Paolo Poli

Server Linux



Speciale Linux Pro
ANNO II - N° 2/2008

Direttore Editoriale: Stefano Spagnolo
Art Director: Silvia Taietti
Coordinamento Editoriale: Giancarlo Calzetta
Coordinamento Redazionale: Mario Bosisio
Responsabile di redazione: Massimiliano Zagaglia
Realizzazione a cura di: Gruppo Orange S.N.C.
Grafica: Zefiro Comunicazione



www.spreabook.it

---Il tuo negozio online---

book@sprea.it

Sprea Book è una divisione di:

Sprea Editori S.p.A.

Via Torino, 51 – 20063 Cernusco sul Naviglio (MI)

Tel (+39) 02.92432.1 Fax (+39) 02.9262.63147

editori@sprea.it - www.sprea.it

CDA

Luca Sprea (Presidente)

Stefano Spagnolo (Vice Presidente)

Gregory Peron (A. D. Edizioni Estere)

Mario Sprea

Stampa: Rotolito Lombarda S.p.A. - Pioltello (MI)

Carta: Valpaco European Paper Trading

Distribuzione: M-Dis Distribuzione S.p.A. Milano

Distributore per le librerie: Consorzio EGAF

Via Tor di Fiorenza 27 – 00199 Roma

Tel (+39) 06.86.10.254 – Fax (+39) 06.86.03.056

www.egafnet.it

Linux Pro

Pubblicazione registrata al Tribunale di Milano il 8/2/2003, con il n.74

Direttore Responsabile

Luca Sprea

Copyright

La Sprea Editori è titolare esclusiva di tutti i diritti di pubblicazione e diffusione. L'utilizzo da parte di terzi di testi, fotografie e disegni, anche parziale, è vietato. L'Editore si dichiara pienamente disponibile a valutare - e se del caso regolare - le eventuali spettanze di terzi per la pubblicazione di immagini di cui non sia stato eventualmente possibile reperire la fonte.

Informativa e Consenso in materia di trattamento dei dati personali (Codice Privacy d.lgs. 196/03)

Nel vigore del D.Lgs 196/03 il Titolare del trattamento dei dati personali, ex art. 28 D.Lgs. 196/03, è Sprea Editori S.p.A. (di seguito anche "Sprea"), con sede in Cernusco sul Naviglio (MI), via Torino, 51. La stessa La informa che i Suoi dati, eventualmente da Lei trasmessi alla Sprea, verranno raccolti, trattati e conservati nel rispetto del decreto legislativo ora enunciato anche per attività connesse all'azienda. La avvisiamo, inoltre, che i Suoi dati potranno essere comunicati e/o trattati (sempre nel rispetto della legge), anche all'estero, da società e/o persone che prestano servizi in favore della Sprea. In ogni momento Lei potrà chiedere la modifica, la correzione e/o la cancellazione dei Suoi dati ovvero esercitare tutti i diritti previsti dagli artt. 7 e ss. del D.Lgs. 196/03 mediante comunicazione scritta alla Sprea e/o direttamente al personale Incaricato preposto al trattamento dei dati. La lettura della presente informativa deve intendersi quale presa visione dell'Informativa ex art. 13 D.Lgs. 196/03 e l'invio dei Suoi dati personali alla Sprea varrà quale consenso espresso al trattamento dei dati personali secondo quanto sopra specificato.

L'invio di materiale (testi, fotografie, disegni, etc.) alla Sprea Editori S.p.A. deve intendersi quale espressa autorizzazione alla loro libera utilizzazione da parte di Sprea Editori S.p.A. per qualsiasi fine e a titolo gratuito, e comunque, a titolo di esempio, alla pubblicazione gratuita su qualsiasi supporto cartaceo e non, su qualsiasi pubblicazione (anche non della Sprea Editori S.p.A.), in qualsiasi canale di vendita e Paese del mondo. Il materiale inviato alla redazione non sarà restituito.

Sommario

Introduzione	IX
La struttura del libro	X
Convenzioni utilizzate.....	XII

Capitolo 1

La scelta della distribuzione Linux.....	1
Che cos'è una distribuzione Linux	1
Il kernel e le distribuzioni.....	2
Le distribuzioni.....	5
Red Hat / Fedora / CentOS.....	7
SUSE / OpenSUSE.....	8
Debian / Ubuntu.....	8
La scelta della distribuzione	9
Server e desktop.....	10
Conclusioni.....	11

Capitolo 2

Il server Web Apache 2.....	13
Apache: le origini, il Web e i protagonisti	13
Installazione di Apache 2.....	15
Configurazione di Apache 2.....	17
La sintassi.....	19
Le direttive globali.....	20
Le direttive per il server.....	22
Host virtuali.....	22
Definire un'interfaccia virtuale.....	23
Connettere Apache 2 alle interfacce virtuali	24
Esecuzione di Apache.....	24
Conclusioni.....	25

Capitolo 3

Il server Web Apache 2: sicurezza e prestazioni	27
Che cosa si intende con “sicurezza”	27
Apache 2 e la sicurezza.....	28
Accesso al sistema da parte degli utenti	29
Configurazione dell’accesso per i gruppi	32
Prestazioni del server Web Apache	34
Un ambiente ideale	34
Conclusioni	35

Capitolo 4

Creazione di un semplice sito Web	37
Come funziona il server Web Apache 2	37
La directory predefinita del sito Web.....	39
Usare i servizi del server Web Apache 2	42
Conclusioni	45

Capitolo 5

Un server di posta elettronica	47
Perché creare un server di posta elettronica?	47
I componenti di un sistema di posta elettronica	48
Inviare e ricevere messaggi email	49
L’agente MTA Sendmail	49
Installiamo Sendmail	50
Configuriamo Sendmail	52
Conclusioni	58

Capitolo 6

Anti-virus e anti-spamming	61
Scansione dei messaggi.....	62
ClamAV	62
Installazione di ClamAV	62
Test post-installazione	64
I file di configurazione	64
Il file clamd.conf.....	65
Il file freshclam.conf	66
Collaudo dell’anti-virus ClamAV	67
SpamAssassin una difesa contro lo spamming.....	69

Perché filtrare con un anti-spammer la posta elettronica	70
Installare SpamAssassin	70
Collaudo dell'installazione	72
Configurazione dei client di posta elettronica.....	73
Conclusioni.....	74

Capitolo 7

EMail + Web = SquirrelMail.....	77
Webmail: che cos'è?.....	77
Vantaggi di SquirrelMail.....	78
Svantaggi di SquirrelMail.....	78
SquirrelMail.....	79
Installiamo SquirrelMail	79
Configurazione di SquirrelMail.....	81
Plug-in per SquirrelMail.....	85
Conclusioni.....	86

Capitolo 8

Un server FTP in Debian	87
Cos'è e a cosa serve il protocollo FTP.....	87
FTP e i rischi	88
Server FTP su Debian.....	88
Configurazione di ProFTPD.....	89
I file di configurazione del server FTP	92
La directory FTP.....	94
Limitare gli accessi anonimi.....	95
Accesso al server FTP	96
Un accesso in scrittura	100
Conclusioni.....	102

Capitolo 9

A ciascuno il suo indirizzo: il protocollo DHCP	103
Il protocollo DHCP	104
Configurazione DHCP su una macchina client Debian.....	106
Un server DHCP Debian.....	108
Installazione del server DHCP.....	109
Configurazione di DHCP Server.....	110
Le opzioni di configurazione	114

Avvio del server DHCP..... 115

Proviamo se funziona? 116

Conclusioni 117

Capitolo 10

Nomi e indirizzi IP: DNS, il grande traduttore 119

Ricerche DNS..... 121

Creare un server DNS con BIND..... 122

Installiamo BIND 124

Configurazione di BIND 125

Un server autorevole sulla zona 126

Proviamo a usare il nuovo server DNS..... 128

Conclusioni..... 131

Capitolo 11

Messaggi istantanei in rete locale: Jabber 133

Il server IM Jabber 134

Installare Jabber..... 134

Configurazione di Jabber..... 134

Installazione di un client per Jabber..... 136

Primo avvio e configurazione di Gabber..... 138

Conclusioni..... 143

Capitolo 12

Creare un server per database MySQL..... 145

MySQL..... 145

Installazione di MySQL 147

Configurazione di MySQL..... 150

Il programma di amministrazione MySQL Monitor 152

Creazione di un database..... 153

Utilizzo in rete del database 159

Gli utenti del database 160

Installiamo un client per Windows..... 160

Conclusioni..... 172

Capitolo 13

E ora Samba! 175

Linux e Windows 176

Samba! 176

Installazione di Samba	177
Configurazione iniziale di Samba.....	181
Utilizzo del client Samba.....	183
Curiosare fra le risorse disponibili in Windows	185
Uso di una stampante Windows	187
Configurazione di un utente per il server Samba	192
Cartelle condivise	194
Accesso da Windows alla macchina Debian Linux	198
Conclusioni.....	201

Capitolo 14

Condividere le stampanti con CUPS	203
Il sistema di stampa CUPS.....	204
Installazione dei driver per una stampante locale.....	204
Stampa dalla macchina locale.....	211
Stampa da una macchina Windows.....	212

Glossario	219
------------------------	------------

Introduzione

Linux è un sistema operativo nato per reti, cresciuto con le reti e che alle reti (e in particolare a Internet) ha dato tantissimo. Non per nulla Linux è il sistema operativo più utilizzato per realizzare server Web, superando di gran lunga tutti i prodotti commerciali disponibili sul mercato.

Linux si differenzia da ogni altro sistema operativo per la sua filosofia completamente aperta, a partire dal suo nucleo centrale: il kernel, di cui è disponibile il codice sorgente.

È stato concepito come un sistema operativo aperto che dovrà sempre rimanere aperto e gratuito per tutti e che tutti potranno utilizzare anche in alternativa ai più avanzati e “alla moda” sistemi operativi proprietari e commerciali.

Dal lato desktop, infatti, riesce a tenere il passo degli ultimi trucchi grafici dei concorrenti (in primis Windows Vista e Macintosh OS X 10.5) grazie alle nuove funzionalità grafiche fornite da Compiz.

Ma Linux rimane comunque soprattutto legato alla creazione di server per ogni tipo di utilizzo.

Si va dai server Web per la realizzazione di siti Internet o di intranet aziendali, fino a sistemi interni per la gestione di database.

Per questo libro si è deciso di adottare la distribuzione Debian, particolarmente aderente alla filosofia aperta di Linux e la base sulla quale sono state costruite molte altre distribuzioni, prima fra tutti la notissima Ubuntu.

Le tecniche esposte potranno essere seguite agevolmente con qualsiasi distribuzione, adattando gli strumenti proposti a quelli effettivamente offerti dal sistema e dal desktop utilizzato (Gnome, KDE o altro).

La distribuzione SuSE adotta invece tutto un altro sistema di installazione e configurazione delle applicazioni, chiamato YaST (Yet another Setup Tool) che segue una filosofia completamente diversa e che, in generale, consente di risolvere ad alto livello e in modo grafico tante piccole operazioni che, su sistemi come Debian, è necessario svolgere manualmente.

Per motivi di spazio, la descrizione dei server presentata in questi capitoli è necessariamente introduttiva. Per quasi tutti gli argomenti presentati, una descrizione approfondita avrebbe richiesto un intero libro!

Dunque, traendo spunto da quanto abbiamo presentato in queste pagine, il lettore può approfondire l'argomento in modo da completare e personalizzare il più possibile la propria installazione dei server.

In particolare dovrebbe essere valutato l'aspetto della sicurezza che, nei capitoli è stato introdotto solo marginalmente e solo in alcuni casi.

Quello che faremo è prendere un sistema Debian nato come un computer desktop e trasformarlo gradualmente in un ricco server in grado di offrire alla propria rete e a Internet un'ampia gamma di servizi.

La struttura del libro

Il Capitolo 1, **La scelta della distribuzione Linux**, ci accompagna nella scelta della distribuzione più adatta. Vengono esaminate le principali distribuzioni, la loro storia e le loro caratteristiche, motivando la scelta della distribuzione Debian. Quindi viene esaminato il concetto di sistema desktop e sistema server. Nel primo caso il computer si comporterà né più e né meno come un personal computer: Linux offre un avanzato desktop grafico, una ricca collezione di applicazioni e tutto il necessario per costruire con poca fatica una macchina perfettamente in grado di rispondere a molte delle esigenze più comuni oggi, dall'utilizzo di Internet (in tutte le sue forme) al tipico lavoro d'ufficio, grazie per esempio ad applicazioni come OpenOffice.org. Completamente diverso è invece l'utilizzo di Linux quale sistema server: su un computer Linux possiamo infatti installare una grande quantità di server fra i più avanzati e diffusi in Internet. Un motivo di questa ampia diffusione è certamente la gratuità degli strumenti software disponibili, ma la realtà è che i server per Linux non hanno in realtà nulla da invidiare agli strumenti commerciali disponibili sul mercato.

Il Capitolo 2, **Il server Web Apache 2**, presenta lo strumento server in assoluto più noto in ambiente Linux e non solo. Dopo aver presentato la storia e l'installazione del software, ne vengono trattate le operazioni di configurazione e le normali attività di utilizzo.

Il Capitolo 3, **Il server Web Apache 2**: sicurezza e prestazioni affronta due argomenti molto sentiti dagli amministratori: innanzitutto la protezione contro gli accessi indesiderati fornendo poi una serie di utili consigli per creare un ambiente ideale per il funzionamento del server Web.

Il Capitolo 4, **Creazione di un semplice sito Web**, presenta i file di esempio di Apache, mostrando dove devono essere collocati i file che costituiscono il sito e come utilizzare i servizi offerti dal server Web Apache 2.

Il Capitolo 5, **Un server di posta elettronica**, presenta l'installazione e la configurazione di un semplice server di posta elettronica con Sendmail. Si vedrà quali sono le motivazioni che spingono a creare un server di posta elettronica locale nella rete e i componenti che formano un server di posta elettronica.

Il Capitolo 6, **Anti-virus e anti-spamming**, presenta due notissimi strumenti software particolarmente legati alla posta elettronica: il software antivirus ClamAV e il filtro anti-spamming SpamAssassin. Questi strumenti consentono di mettere al riparo l'installazione contro due delle minacce più pericolose e fastidiose di Internet.

Il Capitolo 7, **Email + Web = SquirrelMail**, presenta un piccolo strumento software che consente di gestire la posta in modo meno tradizionale utilizzando pagine Web. Nel trattare questi argomenti vengono presentate i vantaggi e gli inevitabili svantaggi. SquirrelMail è dotato di numerosi plug-in che vale la pena di esplorare per estendere le sue funzionalità.

Capitolo 8, **Un server FTP in Debian**, presenta il protocollo FTP che consente di trasformare la macchina in un grande archivio di file che chiunque, internamente all'azienda o da Internet, potrà scaricare o caricare a piacere. Ciò espone l'installazione ad alcuni rischi che possono però essere opportunamente minimizzati.

Il Capitolo 9, **A ciascuno il suo indirizzo: il protocollo DHCP**, introduce questo semplice ma fondamentale servizio che si occupa di assegnare a ciascuna macchina un proprio indirizzo IP privato. Normalmente tale compito viene svolto dal router/gateway che utilizziamo per la connessione Internet, ma definendo un server DHCP in Linux, potremo controllare meglio l'assegnazione degli indirizzi.

Il Capitolo 10, **Nomi e indirizzi IP: DNS, il grande traduttore**, presenta questo importante servizio che esegue senza sosta una continua traduzione dei nomi di siti Internet nei corrispondenti indirizzi IP. Il fatto di creare un proprio server DNS interno offre alla rete una risorsa che consente di accelerare ogni accesso a Internet; le altre macchine, infatti, non dovranno più contattare il server DNS del provider Internet, poiché potranno contare su una risorsa interna che fornirà immediatamente l'indirizzo IP cui devono accedere; ciò garantisce, come vedremo, notevoli incrementi prestazionali nell'accesso ai siti.

Il Capitolo 11, **Messaggi istantanei in rete locale: Jabber**, presenta questo semplice server di messaggistica istantanea che consente di impiantare un sistema di chat interno, utile per piccole o grandi conversazioni testuali, senza costringere gli utenti a utilizzare i grandi servizi Internet come MSN Messenger o Yahoo per comunicare con qualcuno che, magari, si trova solo pochi metri di distanza e che appartiene alla nostra stessa rete.

Il Capitolo 12, **Creare un server per database MySQL**, tratta un altro software fondamentale per Linux: il database MySQL. Si tratta di uno strumento fondamentale, non solo per la creazione di database indipendenti ma anche per la gestione interna di Linux. Nel capitolo vedremo come installare il programma, come configurarlo e poi come creare un database che accedervi sia dalla macchina stessa sia da un client Windows operante su un'altra macchina.

Il Capitolo 13, **E ora Samba!**, tratta l'argomento dell'interoperabilità di rete fra macchine Linux e Windows. Vedremo come installare Samba (che in Debian non viene normalmente installato) e come utilizzare il suo client per consentire alla macchina Linux di accedere alle risorse condivise delle macchine Windows e, viceversa, come utilizzare il suo server per consentire alle macchine Windows di accedere alle risorse presenti nella macchina Linux.

Il Capitolo 14, **Condividere le stampanti con CUPS**, introduce i servizi di stampa che possono essere installati su un server Linux. In questo modo, tutte le macchine connesse alla rete, sia quelle Linux sia quelle Windows, potranno utilizzare stampanti comuni gestite rigorosamente dal server Linux.

Il **Glossario** conclude il libro, introducendo le sigle e i termini utilizzati nei vari capitoli.

Convenzioni utilizzate

L'intero libro si basa su un'installazione standard di Debian Linux in versione desktop, senza alcuna estensione particolare.

Quando necessario, per connetterci, sono stati utilizzati dei sistemi Microsoft Windows, indifferente in versione Vista o XP SP2.

DA SAPERE *Alcune informazioni degne di particolare attenzione o potenzialmente pericolose sono state evidenziate tramite una nota in questo modo.*



Per tutti i listati è stato impiegato un tipo di carattere monospaziato per evidenziare meglio nel testo i comandi, le applicazioni e le direttive da impiegare.

Capitolo 1

La scelta della distribuzione Linux

Linux è un sistema operativo aperto che offre un'ampia libertà di scelta.

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Che cos'è una distribuzione Linux
- ☑ Le distribuzioni
- ☑ Server e desktop

Che cos'è una distribuzione Linux

Quando Linus Torvalds diede alla luce nel 1991 il *kernel*, ovvero il nucleo centrale, di quello che sarebbe diventato il sistema operativo più utilizzato in ambito server e, di gran lunga, il più diffuso sistema operativo aperto del mondo, non immaginava certo che la sua creatura avrebbe avuto questo successo.

Siamo così abituati ad avere a che fare con software a pagamento (e a che prezzo!) che a volte ci sembra impossibile poter utilizzare gratuitamente un intero sistema operativo, ormai più che maturo e alla portata di tutti, il quale vanta, fra l'altro, un corollario di applicazioni assolutamente straordinario quanto ad ampiezza e ricchezza di funzionalità.

Linux, il cui spirito è ben simboleggiato dal piccolo pinguino *Tux* che ne è la mascotte (Figura 1.1) è nato in ambito universitario con lo scopo di superare da un lato le difficoltà di accesso a un sistema Unix e dall'altro i limiti dei sistemi operativi “giocattolo” disponibili in università, in particolare il piccolo sistema operativo Minix.

Il kernel e le distribuzioni

Siamo partiti dal kernel ma sappiamo che al giorno d'oggi Linux è disponibile sotto forma di *distribuzioni*. Qual è la relazione che lega questi due oggetti?

Il kernel di Linux, dopo essere stato sviluppato da Linus Torvalds, è stato messo a disposizione di tutti; questo significa che la piccola creatura di uno studente universitario ha potuto estendersi, raffinarsi e “crescere” grazie al contributo

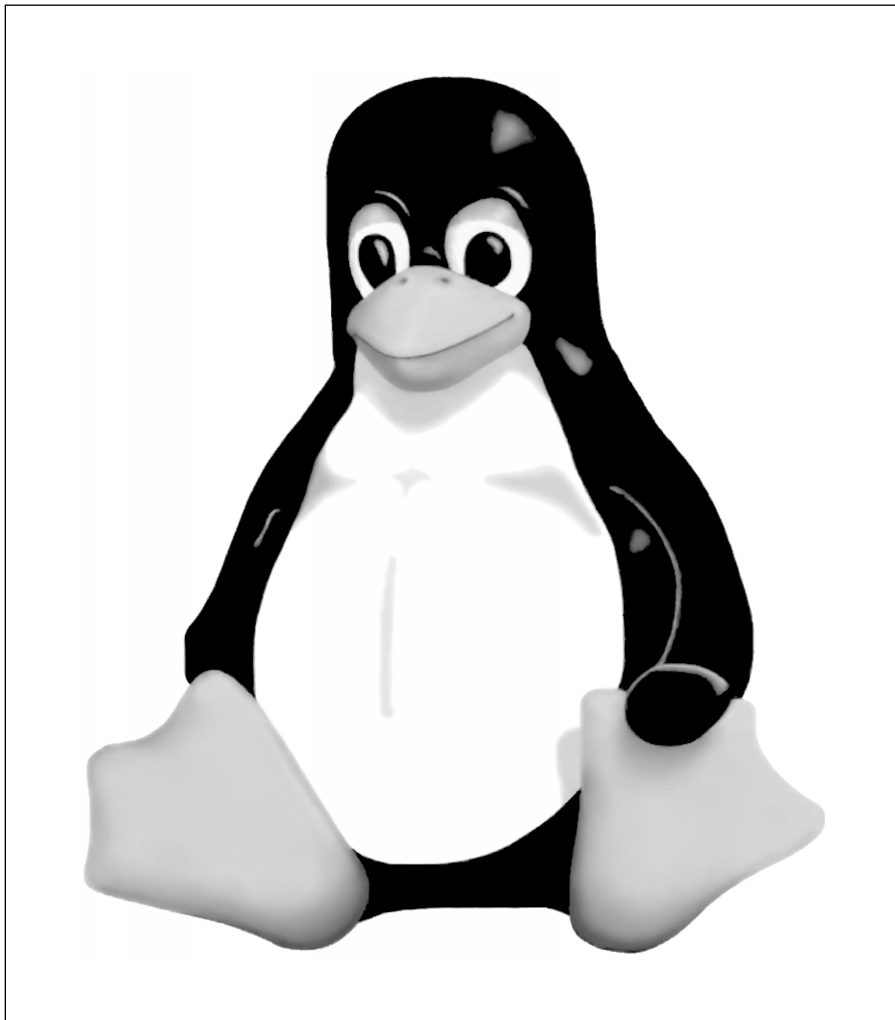


Figura 1.1
Tux, la simpatica mascotte di Linux.

di una comunità di programmatori volontari attiva in tutto il mondo; ognuno di essi aveva la possibilità di correggere, perfezionare e aggiungere ogni singolo elemento software. Questo è il vero punto di forza del *software libero*: quando il software è *proprietario*, segreto e coperto da ogni sorta di brevetti, solo un ristretto, ristrettissimo numero di programmatori può intervenire su di esso, a discrezione del detentore dei diritti di proprietà sul software.

Ma quando il software è “di tutti”, le menti che possono operare su di esso sono molte di più e ciò garantisce il fatto che ogni problema venga immediatamente risolto, ogni necessità venga immediatamente considerata e ciò sotto il controllo di tutti gli altri programmatori.

In pratica Linux è un po' la “Wikipedia” dei sistemi operativi: in Wikipedia chiunque ha la possibilità di aggiungere quanto desidera, ma sempre sotto il controllo di tutti gli altri utenti, in uno sforzo collettivo che, di fatto, ha dato origine alla più ampia, completa e variegata enciclopedia prodotta dal genere umano. Analogamente, il kernel di Linux, ha seguito varie fasi di sviluppo: nel 1994 veniva rilasciata la versione 1.0 del kernel e attualmente, a inizio 2008, quello stesso kernel è cresciuto fino a raggiungere la versione 2.6.22-14, ma periodicamente vengono rilasciate sempre nuove versioni che, normalmente, vengono applicate ai sistemi grazie ai meccanismi di aggiornamento automatico del software di cui è dotato Linux.

Dunque da un lato abbiamo il kernel, il nucleo centrale, che sovrintende le funzionalità di base del computer.

Ma sarebbe troppo complicato chiedere a un utente medio, anche se dotato di una certa esperienza d'uso dei sistemi operativi, di predisporre il necessario per trasformare un computer e un kernel in un sistema funzionante.

- Prima di tutto è necessario prevedere una *procedura di installazione e configurazione* del sistema operativo sul computer.
- In un sistema operativo moderno, è irrinunciabile la presenza di un'*interfaccia utente grafica* (Figura 1.2) con il suo corollario di finestre, controlli, pulsanti, icone e così via.
- Un sistema sarebbe pressoché inutile se non avesse in dotazione una serie di *strumenti* in grado di consentire la configurazione di base del computer: editor di testo, strumenti per la configurazione del video, delle periferiche, dei sistemi di connessione di rete cablata e wireless.
- La natura aperta di Linux ha favorito lo sviluppo di *applicazioni di produttività individuale* che spesso non hanno nulla da invidiare alle più blasonate (ma costose) applicazioni commerciali; primi fra tutti il pacchetto di applicazioni per l'ufficio OpenOffice.org e l'editor grafico the Gimp (Figura 1.3).

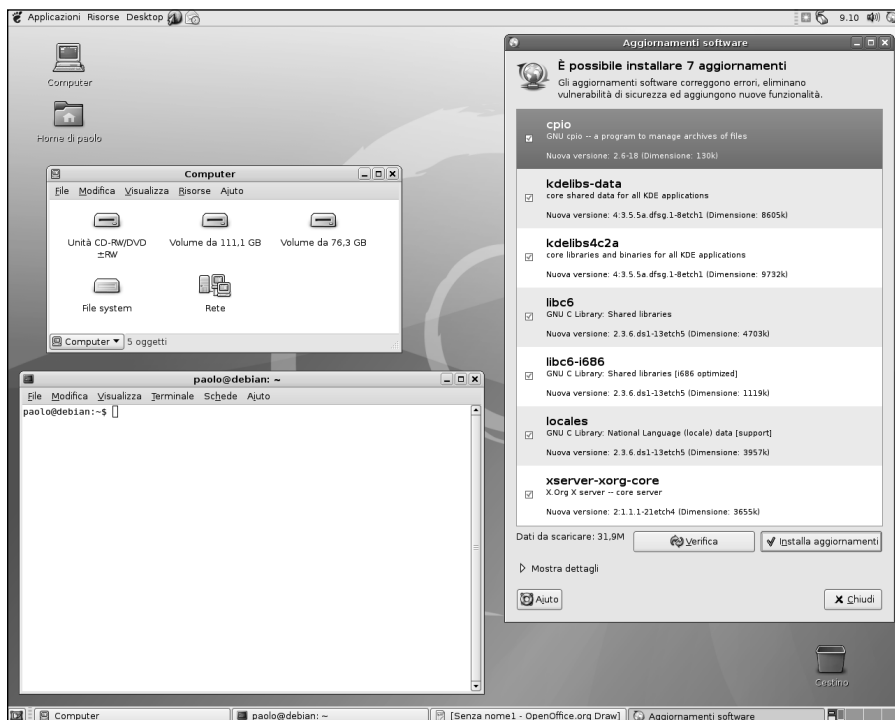


Figura 1.2

Il desktop di Linux. Qui è rappresentato l'aspetto standard della distribuzione Debian, su cui si basa questa guida.

- Dato che Linux ha da sempre manifestato una solidità e una sicurezza ben superiore ai sistemi operativi commerciali concorrenti, è diventato il sistema operativo più diffuso per la realizzazione di *server*; uno fra tutti, il server Web Apache (Figura 1.4).

Le distribuzioni sono dunque costituite dal kernel affiancato da una raccolta selezionata e accuratamente personalizzata di questi pacchetti software, per così dire “di contorno”.

Ogni distribuzione fa capo a un'organizzazione, a volte privata, a volte legata a gruppi di interesse; talvolta è a pagamento (specialmente nel caso delle distribuzioni professionali, che offrono un servizio di supporto) ma nella maggior parte dei casi è gratuita.

In pratica, dunque, la maggior parte di noi è in grado di procurarsi una distribuzione Linux in modo gratuito o quasi (pagando eventualmente solo i costi

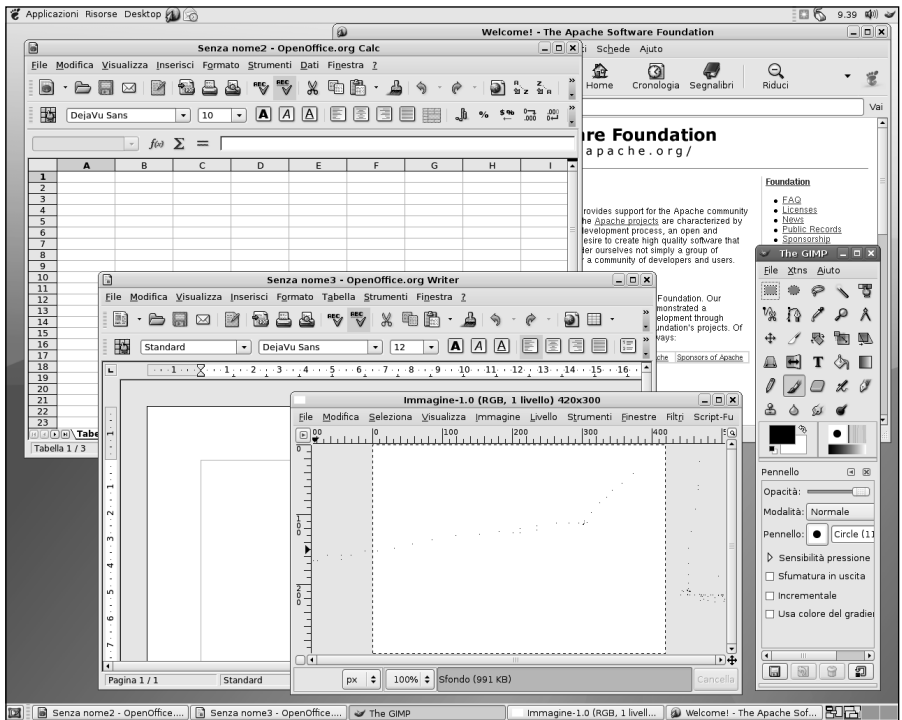


Figura 1.3

L'usabilità di Linux è costantemente cresciuta, merito soprattutto di ottime applicazioni per la produttività individuale e per l'ufficio.

effettivi del supporto utilizzato per l'installazione, CD o DVD) e di installare sul proprio sistema la distribuzione che preferisce.

Le distribuzioni

Nel corso degli anni sono state sviluppate innumerevoli distribuzioni di Linux. Alcune di esse sono dedicate prevalentemente all'utente occasionale e altre sono decisamente orientate a un'utenza professionale.

Si differenziano naturalmente per il livello di assistenza fornito e anche per il fatto che le prime tendono a essere completamente gratuite, mentre le seconde, a fronte di un servizio più completo e utile per gli utenti professionali, prevedono una qualche forma di pagamento.

La tabella seguente elenca le distribuzioni più note di Linux, mentre la Figura 1.5 mostra le linee evolutive delle principali distribuzioni di Linux.

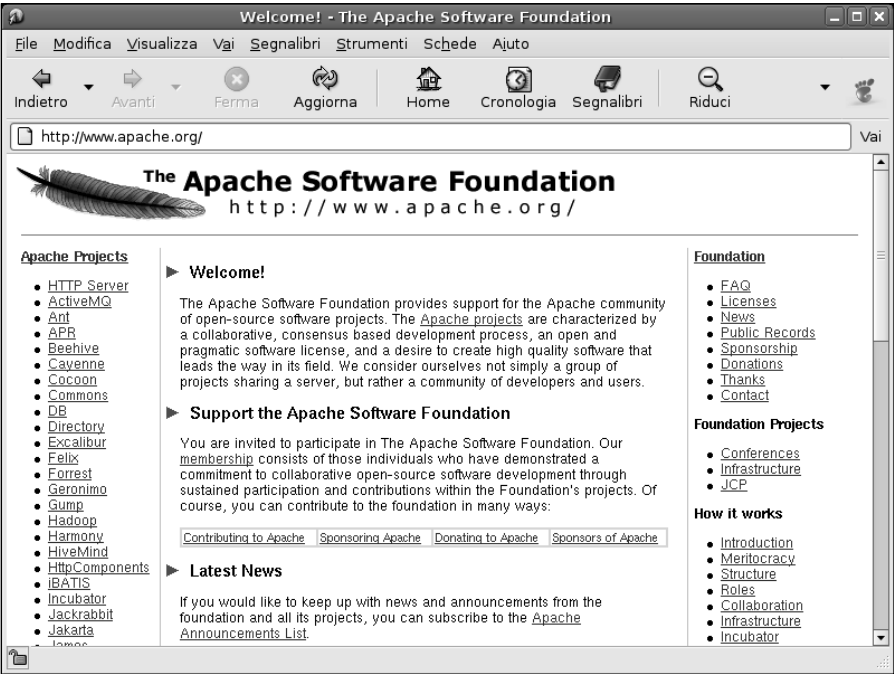


Figura 1.4
La fondazione Apache Software cura lo sviluppo del server di gran lunga più famoso di Linux

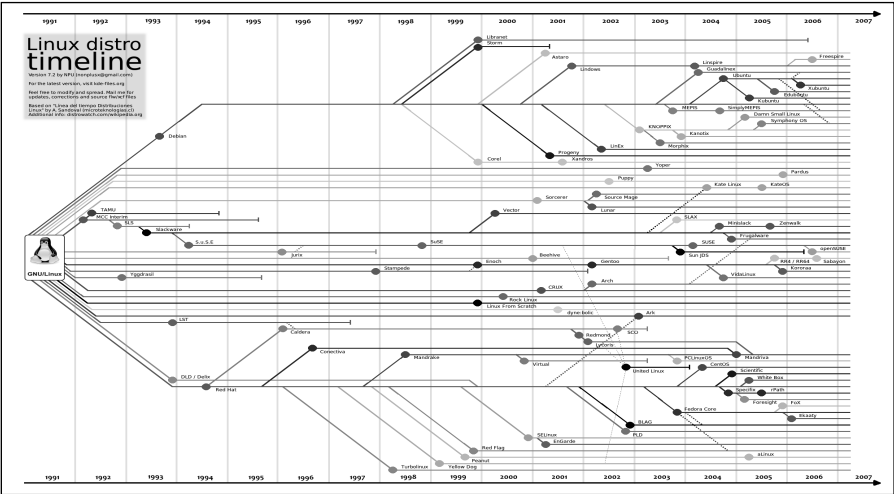


Figura 1.5
La complessa linea evolutiva delle varie distribuzioni di Linux (vedere anche <http://futurist.se/gldt>).

Distribuzione Linux	Descrizione	Sito Web
Red Hat Enterprise Linux	Distribuzione commerciale di Linux, particolarmente diffusa negli Stati Uniti.	http://www.redhat.it/
Fedora	La versione non commerciale e "comunitaria" di Red Hat Linux.	http://fedoraproject.org/
CentOS	La versione completamente gratuita di Red Hat Enterprise Linux.	http://www.centos.org/
SUSE Linux Enterprise	Distribuzione commerciale professionale, particolarmente apprezzata in Europa.	http://www.novell.com/it-it/linux/
openSUSE	La versione gratuita di SUSE Linux Enterprise	http://it.opensuse.org/Benvenuto_su_openSUSE.org
Debian	Distribuzione non commerciale.	http://www.debian.org/
Ubuntu	Distribuzione non commerciale derivata dalla Debian	http://www.ubuntu.com/
Mandriva	Distribuzione che si caratterizza per un'estrema facilità di installazione e utilizzo. In precedenza era nota con il nome di Mandrake Linux.	http://www.mandriva.com/
Xandros	Distribuzione commerciale caratterizzata dalla possibilità di essere impiegata su sistemi dotati di scarse risorse. Si basa sulla distribuzione Corel Linux, a sua volta derivata dalla distribuzione Debian.	http://www.xandros.com/index.html.en

A loro volta, queste distribuzioni si suddividono in più versioni, in base al tipo di utilizzo a cui sono destinate, al tipo di desktop impiegato (tipicamente Gnome o KDE) e/o a livello di supporto offerto.

Red Hat / Fedora / CentOS



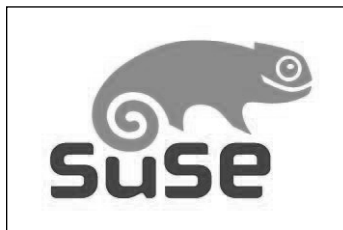
Red Hat è la distribuzione commerciale più nota, diffusa soprattutto nel mercato nordamericano. Si distingue per un ottimo livello di supporto tecnico e ciò può far propendere la scelta su questa distribuzione per tutti coloro che non

intendono utilizzare Linux come un semplice giocattolo ma che desiderano basare la propria attività professionale su macchine Linux.

Fedora è un progetto parallelo ma aperto. Rientra quindi nell'ambito delle distribuzioni gratuite, di potenzialità paragonabili a quelle della distribuzione Red Hat ma, naturalmente, senza supporto tecnico.

CentOS è la versione completamente gratuita della distribuzione Red Hat, dotata di strumenti simili alla versione professionale; pertanto può rappresentare una buona soluzione per installare sistemi professionali qualora si decidesse di sposare la distribuzione Red Hat senza però affrontarne la spesa.

SUSE / OpenSUSE



Anche Novell, produttore della distribuzione *SUSE*, ha seguito per certi versi l'esempio di Red Hat predisponendo una distribuzione a pagamento dotata di un supporto tecnico utile per gli utenti professionali e, parallelamente, una versione aperta, gratuita e disponibile a tutti.

Debian / Ubuntu



La distribuzione *Debian* è forse quella che segue in modo più fedele lo spirito di Linux, un sistema operativo aperto, disponibile e gratuito. Questo è il motivo per cui da questa distribuzione derivano molte altre distribuzioni che ne personalizzano in vario modo le caratteristiche. Debian pertanto è sostenuta dall'intera comunità di sviluppatori di software libero.

Particolarmente interessante, fra le distribuzioni Linux derivate da Debian è il caso della distribuzione *Ubuntu*, sostenuta finanziariamente dal filantropo sudafricano Mark Shuttleworth, che spinge la gratuità della propria distribuzione fino all'estremo: è infatti possibile richiedere i CD/DVD ufficiali di installazione di Ubuntu Linux senza alcuna spesa, neppure le spese di spedizione. Più gratuito di così!

La scelta della distribuzione

Naturalmente ognuno è libero di scegliere la distribuzione più adatta alle proprie esigenze.

- Le aziende dotate di risorse economiche non indifferenti e che svolgono nell'ambito delle proprie funzioni intense attività di commercio elettronico, non possono correre certo il rischio di rimanere inattive a causa di un problema tecnico del software. Pertanto orienteranno naturalmente la propria scelta su una distribuzione dotata di supporto tecnico, che le aiuterà a mantenere sempre in perfette condizioni il sistema operativo, a personalizzare le applicazioni in base alle proprie specifiche esigenze e a risolvere gli inevitabili problemi che potranno presentarsi, reagendo tempestivamente a ogni richiesta di aiuto.
- All'estremità opposta si possono collocare le distribuzioni come Ubuntu, estremamente popolare e diffusissima, probabilmente la prima scelta per il neofita ma anche per l'utente professionale che intenda dotarsi di un sistema di utilizzo non complesso. Ubuntu è da molto tempo in testa alle classifiche di popolarità delle distribuzioni Linux curate dal noto sito <http://distrowatch.com> (Figura 1.6). Da sempre è apprezzata la sua facilità di installazione e la sua semplicità, che consentono la predisposizione di un perfetto sistema Linux a chiunque sia in possesso di una macchina "di questo millennio" e delle conoscenze necessarie per installare un qualsiasi sistema operativo commerciale, come Windows XP o Windows Vista.
- In posizione intermedia, si colloca la distribuzione Debian, molto popolare ma anche completa, gratuita e configurabile senza difficoltà per offrire il tipo di servizio desiderato. In altre parole, predisporre una macchina Debian è facile quanto installare una macchina Ubuntu; la dotazione software della macchina potrà poi essere facilmente integrata in modo da fornire il tipo di servizi desiderato: server Web, server di posta elettronica e così via.

Questo è il motivo per cui abbiamo scelto di descrivere l'installazione e l'utilizzo di strumenti server proprio sulla distribuzione Debian.

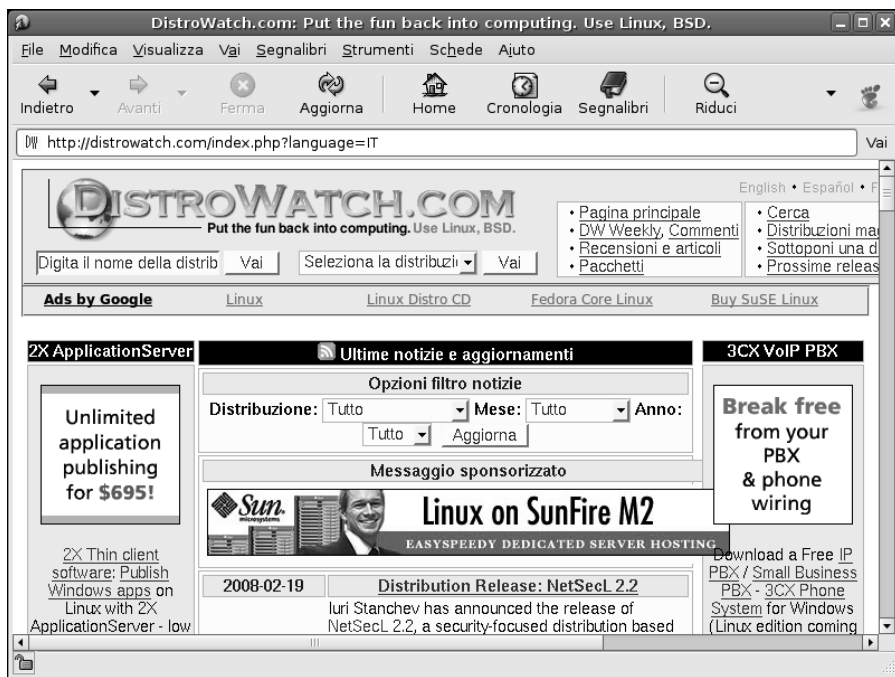


Figura 1.6

Il sito DistroWatch offre una panoramica aggiornata di tutte le novità riguardanti le tante distribuzioni Linux disponibili.

Sostanzialmente, le stesse tecniche descritte per la distribuzione Debian potranno essere impiegate, con alcune variazioni facilmente reperibili nel web, anche su quelle da essa derivate, in primo luogo la distribuzione Ubuntu.

In ogni caso, escludendo la procedura di installazione, che potrà prevedere l'impiego di strumenti di installazione e aggiornamento differenti da distribuzione a distribuzione, quanto descritto in questi capitoli potrà essere applicato a qualsiasi distribuzione.

Server e desktop

Le macchine Linux possono rispondere a due diverse esigenze, per certi versi antitetiche: se abbiamo bisogno di predisporre una macchina sulla quale utilizzare i tanti pacchetti applicativi, sempre più avanzati, disponibili per Linux o se la nostra macchina deve essere connessa a una rete preesistente e funzionare come sistema personale di un utente, dobbiamo installare una versione *desktop* del sistema operativo Linux.

Se invece la macchina deve rappresentare uno degli elementi centrali della rete, in grado di offrire connettività e servizi alle altre macchine che compongono la rete locale o anche di fornire i tipici servizi Internet, come ospitare un sito Web, creare un server FTP e così via, dovremo installare una versione *server* del sistema operativo Linux.

Ma in realtà non vi è nulla di particolare che distingua una macchina desktop da una macchina server. Quello che cambia non è il sistema operativo: non avremo bisogno di installare un sistema operativo “differente” per creare una macchina desktop o una macchina server. Quello che cambia è solo la collezione di pacchetti software che devono essere installati sulla macchina.

Se sceglieremo di eseguire un’installazione “desktop” del nostro sistema Linux, sul computer verrà installata una raccolta di software di produttività personale: strumenti per l’ufficio, strumenti gestionali, strumenti di connessione a Internet, applicazioni grafiche e giochi.

Se invece sceglieremo di eseguire un’installazione “server”, sul sistema verranno installate molte delle applicazioni disponibili in questo ambito e che, in realtà, hanno fatto la vera fortuna di Linux.

Naturalmente l’installazione e la configurazione di un sistema server richiede competenze e conoscenze non comuni, principalmente perché dovremo conoscere esattamente ciò di cui abbiamo bisogno e disporre di alcune informazioni avanzate riguardanti la connessione a Internet.

Nei prossimi capitoli supporremo di aver installato una macchina desktop, che dovremo progressivamente “vestire” di strumenti software in grado di trasformarla in un perfetto server sia per la rete interna sia per Internet.

Conclusioni

Questo capitolo ha presentato una breve introduzione al sistema operativo Linux, evidenziandone le caratteristiche principali.

Abbiamo parlato dei concetti di kernel e distribuzione e abbiamo presentato le principali distribuzioni disponibili, esaminandone sommariamente le principali caratteristiche. Dopo aver introdotto le varie distribuzioni, abbiamo motivato la scelta della distribuzione Debian, che meglio incarna lo spirito libero e aperto di Linux.

Nel prossimo capitolo cominceremo a trasformare la nostra macchina desktop in un server, presentando l’applicazione che, da sola, ha decretato il successo mondiale di Linux come sistema operativo di rete: il server Web Apache.

Capitolo 2

Il server Web Apache 2

**Predisponiamo la nostra presenza Web
con il “principe” dei server Internet di Linux.**

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Apache: le origini, il Web e i protagonisti
- ☑ Installazione di Apache 2
- ☑ Configurazione di Apache 2
- ☑ La sintassi
- ☑ Le direttive globali
- ☑ Le direttive per il server

Apache: le origini, il Web e i protagonisti

Il Web, così come lo conosciamo oggi, trae le sue origini dall'impegno di Tim Berners-Lee che negli anni '90 definì tutti gli elementi che lo caratterizzano. Successivamente, nei laboratori dell'NCSA (National Center for Supercomputing Applications) dell'Università dell'Illinois venne realizzato il server Web più utilizzato al mondo: Apache.

Apache trae le proprie origini dal server Web HTTPd, sempre dell'NCSA. Riconosciuti i limiti di tale server, Brian Behlendorf iniziò a sviluppare gli elementi di base di un nuovo server Web, che nacque nell'aprile del 1995.

Entro la fine del 1995 venne resa disponibile la prima versione stabile del server Web Apache e un anno dopo Apache aveva già superato HTTPd in termini di diffusione in Internet.

A questo punto, Apache cominciò a essere impiegato da società sempre più importanti, come Yahoo! e Amazon. Successivamente anche i grandi pilastri dell'hardware, come IBM, hanno sviluppato sempre più prodotti e servizi basati su Apache, tanto che oggi sembra straordinario che un prodotto gratuito e open-source come Apache sia in grado di sostenere attività commerciali per milioni di euro. Oggi, come si può vedere nella Figura 2.1 (http://news.netcraft.com/archives/web_server_survey.html), Apache vanta una presenza preponderante nel panorama dei server Web. Un successo senza paragoni nell'ambito del software libero!

Dal 1995 Apache ha subito varie revisioni; la versione più popolare (e tuttora in uso) è Apache 1.3, tuttavia il suo destino è decisamente segnato.

Apache 2.0, nato nel 2000, è dovuto a una radicale riscrittura del codice, che si è affrancato quindi dal codice inizialmente sviluppato alla NCSA.

La nuova versione di Apache migliora molti degli aspetti chiave e prestazionali del server, privilegiando la portabilità, la scalabilità, la facilità di configurazione e gestione dell'I/O. Lo sviluppo del server Web Apache non si è mai fermato e dopo l'uscita di Apache 2.1 gli sviluppatori hanno iniziato ad adottare la stessa convenzione normalmente impiegata per lo stesso kernel di Linux: alle release stabile è stata attribuita una numerazione pari (2.0, 2.2, 2.4 e così via), mentre alle versioni di sviluppo è stata attribuita una numerazione dispari (2.1, 2.3, 2.5 e così via).

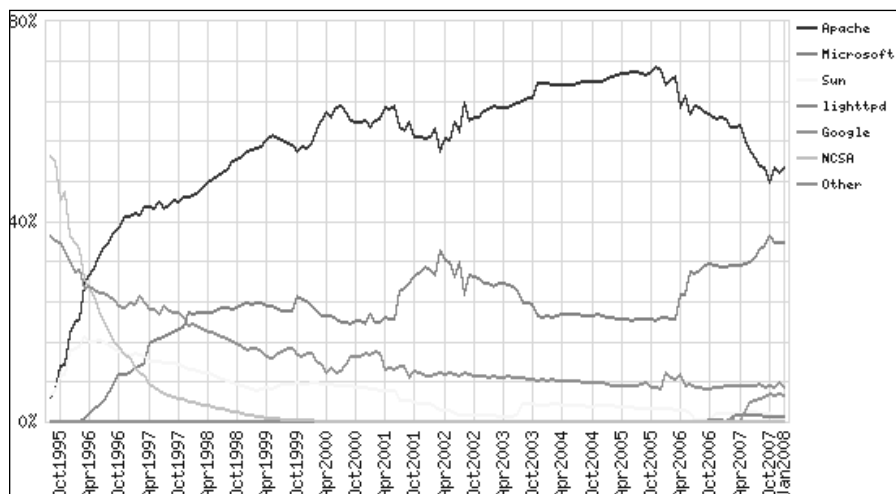


Figura 2.1

Fin quasi dalla nascita, Apache ha manifestato una presenza preponderante fra i server Web.

Al momento attuale Apache è uno strumento universale nelle mani dei professionisti di reti e degli amministratori di sistemi. Oltre a fungere da server Web per aziende, enti e semplici utenti, può essere impiegato come server proxy, di posta elettronica, MP3 e di applicazioni, grazie alle sue numerose estensioni. Nelle prossime pagine impareremo a installare e configurare Apache come un server Web, esplorando la configurazione e l'utilizzo di questo software per ospitare uno o più siti Web su un server Debian.

Installazione di Apache 2

Il sistema decisamente più semplice per installare un server Web sul proprio sistema consiste nell'utilizzare gli strumenti messi a disposizione dalla distribuzione. Se avete configurato fin dall'inizio il sistema Linux come un server, è molto probabile che il software sia già installato; per sincerarvene, provate a osservare il contenuto della cartella `/etc/apache2` del sistema. Se non esiste, dovrete procedere all'installazione.

Naturalmente è possibile scaricare e compilare il codice sorgente direttamente dal sito Web di Apache all'indirizzo www.apache.org, ma nel caso della distribuzione Debian e della maggior parte delle distribuzioni che offrono uno strumento per la gestione dei pacchetti applicativi, è di gran lunga più facile procedere direttamente dal desktop.

Selezionando direttamente dal desktop il comando **Desktop > Gestore pacchetti Synaptic**, si aprirà la finestra del Gestore di pacchetti Synaptic, rappresentata nella Figura 2.2.

Facendo clic sul pulsante **Cerca** nella barra degli strumenti si aprirà la finestra di dialogo **Trova**; qui, nella casella di testo **Cerca** possiamo digitare la parola **"apache"** e poi fare clic sul pulsante **Cerca** (Figura 2.3).

Verrà visualizzato l'elenco dei pacchetti che contengono la parola **"apache"** nel nome o nella descrizione. Nell'elenco possiamo immediatamente notare la presenza del vecchio server Web Apache in versione 1.3.X di cui abbiamo parlato in precedenza e del nuovo server Web Apache 2, contraddistinto dalla voce **apache2**. Facciamo clic sulla casella a sinistra e selezioniamo l'opzione **Marca per l'installazione**. Verrà presentata una finestra che presenta alcuni altri pacchetti che devono essere installati obbligatoriamente per consentire il funzionamento del server Web Apache (Figura 2.4).

Facciamo clic sul pulsante **Marca** e verranno automaticamente selezionati tutti i pacchetti necessari per l'utilizzo di Apache 2. Tali pacchetti saranno contrassegnati da una freccia e da un'evidenziazione verde, come si può vedere nella Figura 2.5.

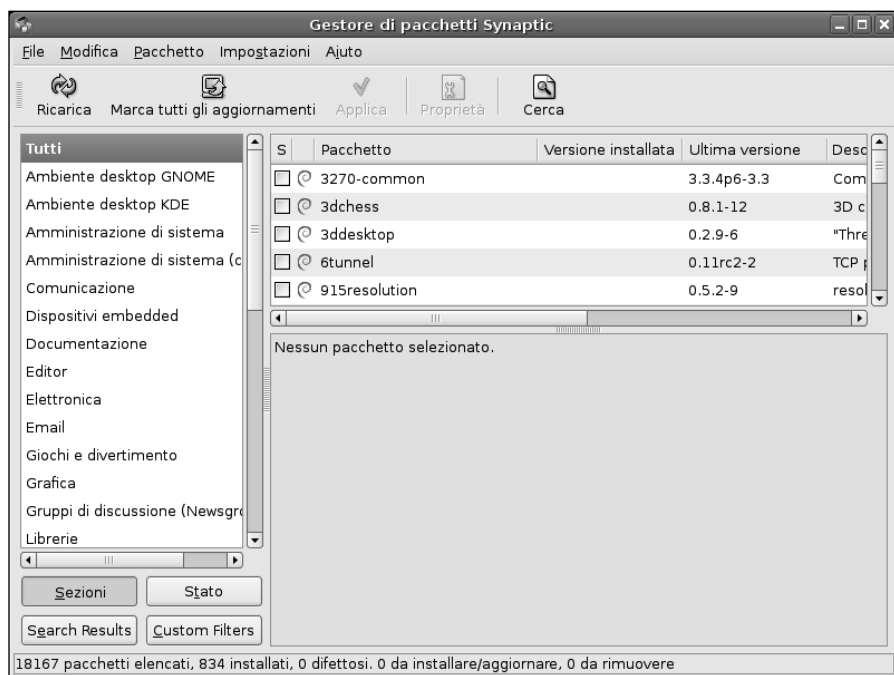


Figura 2.2

L'aspetto iniziale del Gestore di pacchetti Synaptic, dalla quale possiamo controllare l'installazione di qualsiasi applicazione disponibile per Debian Linux.

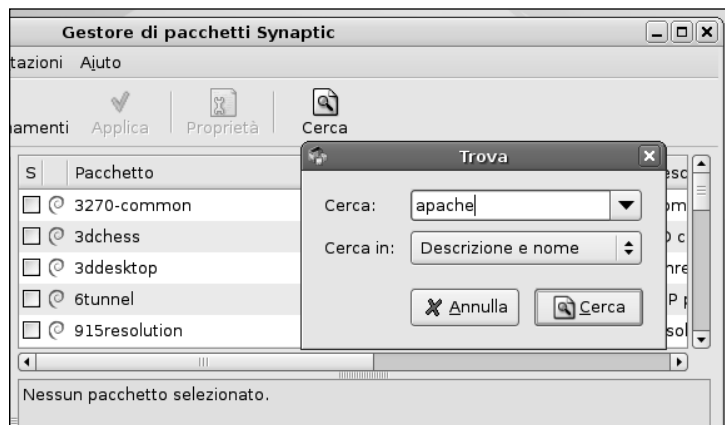


Figura 2.3

L'installazione di un'applicazione come Apache con il Gestore a pacchetti è estremamente semplice: basta orientare la ricerca sull'applicazione desiderata.

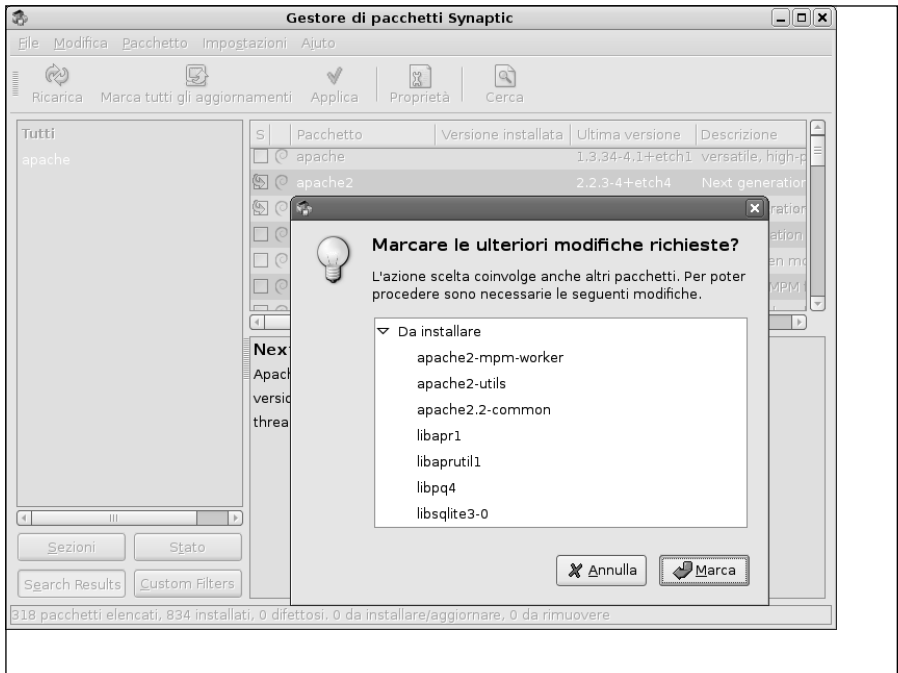


Figura 2.4

La collezione di pacchetti ausiliari che devono essere obbligatoriamente installati insieme al server Web Apache 2.

I pacchetti selezionati come impostazione predefinita per l'estrazione insieme ad Apache sono più che sufficienti per l'utilizzo dell'applicazione.

Facendo clic sul pulsante **Applica** nella barra degli strumenti del Gestore di pacchetti Synaptic, potremo avviare la procedura di installazione (Figura 2.6) che si occuperà di scaricare e installare il pacchetto in modo assolutamente trasparente, senza costringerci a ricorrere a intricati comandi da introdurre nel Terminale di Linux. L'intera procedura non richiede che pochi secondi e al termine verrà visualizzata la finestra di conferma **Modifiche applicate** che possiamo chiudere facendo clic sul pulsante **Chiudi**. Ora possiamo anche chiudere il Gestore di pacchetti Synaptic con la combinazione di tasti **CTRL+Q**.

Configurazione di Apache 2

Ora che il pacchetto del server Web Apache è installato, occorre configurare il suo ambiente di lavoro. Apache può essere configurato dalla riga di comando

(ovvero da una finestra Terminale di Linux) intervenendo direttamente sui file con un editor di testi.

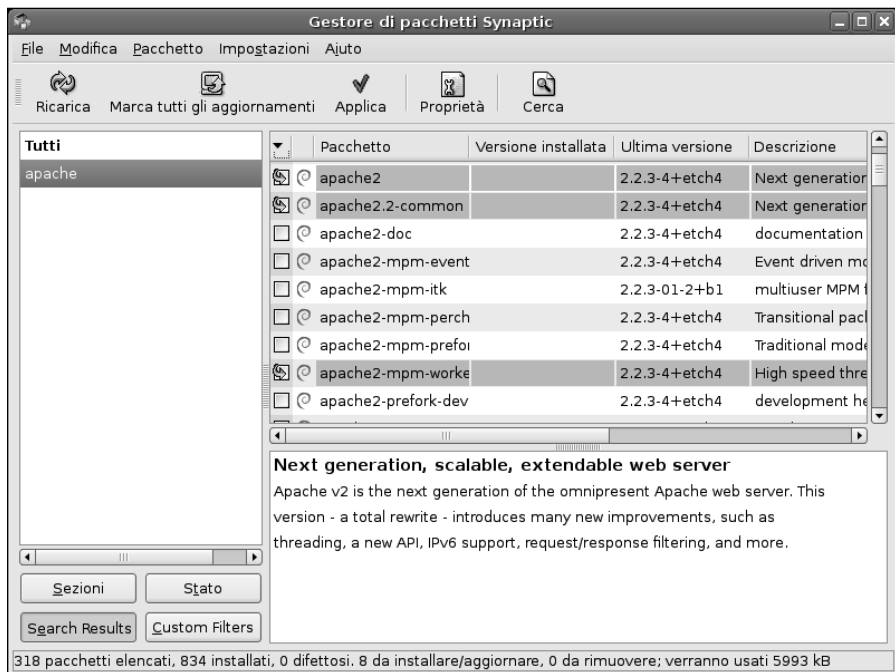


Figura 2.5

Tutti i pacchetti selezionati per l'installazione di Apache 2 sono evidenziati in verde. Nella figura non è possibile vedere né i colori né tutti i pacchetti selezionati.



Figura 2.6

È in corso l'installazione di Apache2. L'intera operazione dura solo pochi secondi: si tratta di un pacchetto decisamente "leggero"

Per poter configurare o avviare Apache 2 è necessario essere connessi come utenti root.

Il file di configurazione di Apache 2, `apache2.conf`, si trova nella directory `/etc/apache2/`; si tratta di un comune file di testo che contiene le direttive per il controllo del server Web, dei siti Web e dei file. Le modifiche che eseguiremo sui file di configurazione vengono riconosciute solo al successivo riavvio del server.

Il file `apache2.conf` è normalmente suddiviso in tre sezioni.

- **Global Environment** Contiene direttive che controllano il funzionamento generale di Apache.
- **Server** Contiene i parametri di configurazione del sito Web principale o predefinito.
- **Virtual hosts** Lo stesso processo server può esaudire le richieste relative a più server Web.

La sintassi

I comandi del file di configurazione adottano una sintassi piuttosto semplice: ogni riga contiene una direttiva; se la direttiva procede sulla riga successiva, basta specificare al termine della riga il segno di backslash (“\”).

DA SAPERE *Le direttive non distinguono fra lettere maiuscole e minuscole; al contrario i loro argomenti attribuiscono significati differenti alle lettere maiuscole e minuscole.*



Il carattere hash (#) contrassegna l’inizio di un commento; tuttavia non sempre tali commenti vengono specificati.

Le righe vuote che precedono una direttiva vengono sempre ignorate; ciò ci offre la possibilità di suddividere in gruppi le varie direttive, con lo scopo di migliorarne la leggibilità.

Per verificare la correttezza del file di configurazione si può utilizzare lo strumento `apache2ctl configtest`.

Ecco per esempio, come possiamo verificare il file di configurazione di Apache 2. Se la sintassi di configurazione è valida, `apache2ctl` restituisce `Syntax OK`.

```
apache2ctl configtest
Syntax OK
```


Se invece il file contiene un errore sintattico, il test di configurazione restituisce un messaggio d'errore appropriato.

```
apache2ctl configtest
Syntax error on line 2 of /etc/apache2/apache2.conf:
Invalid command 'Based'
```

Le direttive globali

Le direttive contenute in questa sezione del file `apache2.conf` controllano il funzionamento generale di Apache; per esempio controllano il numero di richieste che il server è in grado di gestire contemporaneamente e il modo in cui elaborare tali richieste.

Apache opera in base a un metodo chiamato *Prefork*: all'avvio, Apache crea una serie di processi figli; il loro numero e l'utente per il quale vengono richiamati devono essere definiti nel file di configurazione. Ognuno di questi processi figli gestisce le richieste sotto il controllo del server Web: un processo figlio per ogni richiesta in arrivo. Qualora le richieste in arrivo superassero il numero dei processi figli in esecuzione, il processo genitore di Apache creerà ulteriori processi figli, sempre nei limiti delle risorse disponibili sul sistema.



DA SAPERE *Il metodo Prefork ha lo scopo di anticipare le richieste, in modo che ogni nuova richiesta trovi un processo figlio pronto per esaudirla.*

Il seguente elenco presenta le principali direttive che controllano il metodo Prefork di Apache 2.

- **StartServers** Il numero dei processi figli da lanciare all'avvio di Apache.
- **MinSpareServers** Il numero *minimo* di processi figli che devono essere disponibili.
- **MaxSpareServers** Il numero *massimo* di processi figli che devono essere disponibili.
- **MaxClients** Il numero massimo di connessioni accettate.
- **MaxRequestsPerChild** Il numero massimo di richieste gestibili durante la "vita" di un processo figlio.

Il file `apache2.conf` (Figura 2.7) contiene, come impostazione predefinita le seguenti impostazioni:

<code>StartServers</code>	5
<code>MinSpareServers</code>	5
<code>MaxSpareServers</code>	10
<code>MaxClients</code>	150
<code>MaxRequestsPerChild</code>	0

Dunque il processo genitore di Apache 2 genera 5 processi figli (`StartServers`) per gestire le richieste per il server Web. Poi il processo genitore controlla i processi figli e quando uno di essi è impegnato a esaudire una richiesta, aggiunge nuovi processi a quelli in attesa, mantenendo sempre attivi almeno 5 processi (`MinSpareServers`) e al massimo 10 processi (`MaxSpareServers`). In ogni caso il processo genitore non creerà mai più di 150 processi (`MaxClients`) per rispondere ai picchi di richieste sul sito Web.

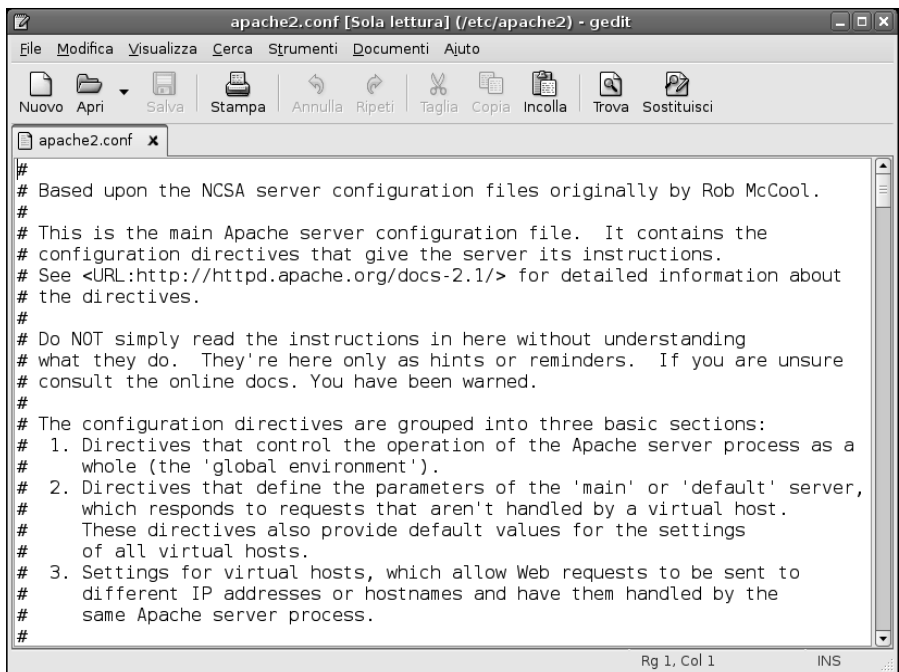


Figura 2.7

L'aspetto del file `apache2.conf` di configurazione del server Web Apache, qui aperto in una finestra dell'editor grafico gedit.

Ogni processo figlio può gestire nel corso della sua vita un numero illimitato di richieste (`MaxRequestsPerChild = 0`). Questa direttiva indica il numero totale di richieste che possono essere progressivamente gestite dal processo figlio: se fosse uguale a 100, il processo figlio potrà esaudire fino a 100 richieste e poi morirà, mentre il processo genitore creerà un nuovo processo figlio, il quale potrà esaudire altre 100 richieste. Questa vita “a termine” dei processi figli ha lo scopo di limitare la quantità di memoria consumata da un processo e di contenere il numero di processi e il carico di lavoro per il server.

Apache 2 introduce anche una nuova soluzione per la gestione dei processi, denominata MPM (Multi-Processing Modules). Tuttavia la configurazione standard di Apache 2 rimane il metodo Prefork.

Le direttive per il server

Nella sezione `Main Server` del file di configurazione devono essere collocate le direttive utilizzate per gestire il funzionamento del sito Web principale o di default gestito da Apache. I valori impiegati dalle direttive di questa sezione vengono poi impiegati come valori predefiniti anche per le direttive presenti nella sezione `Virtual Host`, dedicata alla creazione di host virtuali (ovvero più siti sulla stessa macchina).

Le direttive normalmente utilizzate nelle sezioni `Main Server` e `Virtual Host` riguardano la configurazione delle risorse del filesystem o dello spazio Web.

Le direttive `<Directory>` e `<Files>` sono in realtà gruppi di direttive che si applicano alle risorse così come sono viste dal filesystem. Le direttive racchiuse in `<Directory>` riguardano sia la directory specificata sia le sue subdirectory. Le direttive racchiuse in `<Files>` riguardano solo il file nominato, in qualunque directory si trovi.

Nella sezione `Webspace` del file `apache2.conf`, la direttiva `<Location>` opera sulla configurazione delle risorse. A differenza di `<Directory>` e `<Files>`, la direttiva `<Location>` non individua una risorsa situata sul filesystem e dunque è utile per i contenuti dinamici che non hanno alcuna vera posizione fisica individuabile nel filesystem.

Host virtuali

Inizialmente, una macchina si trovava, al più, a gestire un unico sito Web (per esempio www.mariorossi.it). Oggi le cose sono molto cambiate: sempre più persone possono avere la necessità di definire una propria, sia pur minima, presenza Web.

Per questo motivo sono nate aziende il cui unico scopo è quello di offrire spazio Web a chi non ha la possibilità.

In altre parole, questo significa che un'unica macchina risponderà alle richieste relative a più indirizzi (quindi non solo `www.mariorossi.it` ma anche, per esempio `www.luigiverdi.it`, `www.pietrobruni.it` e così via). Il meccanismo con il quale il server Web Apache può gestire questo “condominio” Web prevede l'impiego di host virtuali che fanno riferimento ognuno a un'interfaccia virtuale distinta della stessa macchina.

Dunque quando la nostra macchina verrà raggiunta da una richiesta Web (identificata normalmente dalla porta 80 per il protocollo HTTP), andrà a osservare anche l'indirizzo IP richiesto. In questo modo potrà esaudire la richiesta rinviandola al sito Web corrispondente.

Il trucco utilizzato per rendere possibile questo meccanismo consiste nel fare in modo che la macchina risponda a più indirizzi IP rispetto al numero di interfacce di rete presenti fisicamente sulla macchina stessa. Si creano pertanto delle interfacce di rete virtuali che fanno riferimento da un lato tutte alla stessa porta fisica presente sulla macchina e dall'altro ai vari domini definiti sulla macchina, in un rapporto “uno-a-molti”. In questo modo una singola macchina può ospitare più siti Web, anche decine o centinaia.

Definire un'interfaccia virtuale

Come possiamo definire sul nostro sistema Debian un'interfaccia virtuale? Dobbiamo innanzitutto creare tale interfaccia a livello del protocollo TCP/IP. Poi bisogna informare Apache 2 sulle interfacce virtuali disponibili, associandole ognuna a un sito differente.

Su sistemi Debian, occorre predisporre le definizioni dell'interfaccia virtuale all'interno del file `/etc/network/interfaces`. La sintassi del comando necessario per definire tale interfaccia virtuale è la seguente:

```
iface <interfaccia>:<interfaccia virtuale> inet static
    address <indirizzo IP>
    netmask <maschera di rete>
    broadcast <indirizzo broadcast>
```

Ecco per esempio come si può configurare un'interfaccia virtuale relativa all'indirizzo `138.128.150.243` come prima (0) interfaccia virtuale dell'interfaccia fisica `eth0`:


```
iface eth0:0 inet static
    address 138.128.150.243
    netmask 255.255.255.192
    broadcast 138.128.150.191
```

Connettere Apache 2 alle interfacce virtuali

A questo punto non rimane che informare Apache della disponibilità di queste interfacce virtuali. In questo modo il server Web potrà “smistare” le richieste, rinviandole al sito appropriato.

Per fare ciò occorre inserire una direttiva `VirtualHost` all’interno del file di configurazione `apache2.conf`. Occorre predisporre una direttiva per ciascun host virtuale (e interfaccia virtuale) definito in precedenza.

Ecco per esempio quale potrebbe essere l’aspetto di una direttiva `VirtualHost`:

```
# Host virtuale per il sottodominio www.mariorossi.it
<VirtualHost 138.128.150.243>
    ServerName www.mariorossi.it
    ServerAdmin webmaster@www.mariorossi.it
    DocumentRoot /var/www/htdocs/mariorossi
    ErrorLog logs/www.mariorossi.it-error_log
    CustomLog logs/www.mariorossi.it-access_log combined
    ScriptAlias /cgi-bin/ /var/www/cgi-bin/mariorossi
</VirtualHost>
```

In pratica, tutti coloro che cercheranno di collegarsi all’host virtuale 138.128.150.243 verranno servite le pagine Web contenute nella directory `/var/www/htdocs/mariorossi`. Occorrerà dunque predisporre una clausola di questo tipo per ogni host virtuale definito sulla macchina.

Esecuzione di Apache

Per lanciare il server Web non rimane che eseguire alla riga di comando, come utenti root, il seguente comando, che sfrutta il link presente nella directory dei servizi avviati automaticamente al boot del sistema:

```
/etc/init.d/apache2 start
```


Per fermare il server Web basta invece usare:

```
/etc/init.d/apache2 stop
```

Per fermare e riavviare immediatamente il server Web si usa il comando:

```
/etc/init.d/apache2 restart
```

Conclusioni

Il presente capitolo ha introdotto l'installazione e le basi di configurazione del server Web Apache 2 su un sistema Debian. Per prima cosa abbiamo introdotto la storia di questo elemento importante della storia di Linux, che, da solo, ha la responsabilità maggiore nel successo di questo sistema operativo.

Successivamente abbiamo visto quanto sia facile installare Apache 2 utilizzando il gestore di pacchetti Synaptic fornito in dotazione con la distribuzione Debian e tutte le distribuzioni da essa derivate.

Abbiamo poi parlato della configurazione di Apache 2 e delle direttive globali di impostazione del server.

Infine abbiamo affrontato l'argomento degli host virtuali, che consentono di ospitare su un'unica macchina un numero arbitrario di siti Web distinti.

Capitolo 3

Il server Web Apache 2: sicurezza e prestazioni

Se il server Web è la nostra “finestra sul mondo”, cerchiamo di tenere fuori i malintenzionati

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Che cosa si intende con “sicurezza”
- ☑ Apache 2 e la sicurezza
- ☑ Prestazioni del server Web Apache

Che cosa si intende con “sicurezza”

La sicurezza è un problema fondamentale, soprattutto al giorno d’oggi, e possiamo dire che ognuno di noi potrebbe probabilmente dare a questa parola uno o più significati differenti.

Un amministratore di rete si preoccuperà soprattutto di *proteggere il software* e di *tenerlo costantemente aggiornato*, per eliminare tutti i punti deboli che vengono scoperti di volta in volta.

Uno sviluppatore di siti Web e di applicazioni dovrà verificare che *ogni utente sia stato adeguatamente verificato* e che i dati forniti dagli utenti *vengano conservati in un luogo sicuro*.

Per chi visita un sito Web la sicurezza significa poter contare sulla *riservatezza dei dati personali* forniti: dovranno rimanere inaccessibili a ogni estraneo e dovranno essere crittografati per renderli inutilizzabili ad altri.

Con riferimento all'utilizzo e alla gestione di un server Web Apache, tali problemi riguardano tre concetti fondamentali: l'autenticazione dell'accesso, l'autorizzazione all'accesso e il controllo degli accessi:

- **autenticazione** prima di consentire un accesso, il sito Web deve verificare l'identità di un utente ovvero assicurarsi che sia proprio chi sostiene di essere;
- **autorizzazione** occorre poi verificare che l'utente abbia i diritti di accesso necessari per ottenere le informazioni che desidera;
- **controllo degli accessi** ogni utente ha accesso solo a determinate informazioni; bisogna pertanto impedire loro di accedere a dati cui non dovrebbero avere accesso.

Il server Web Apache 2, gestisce questi processi di sicurezza tramite i moduli `mod_auth` e `mod_access`. Possiamo inserire speciali direttive di configurazione di questi due moduli direttamente nel file di configurazione principale del server, `apache2.conf`, oppure nei file di configurazione specifici previsti per l'accesso alle directory, `.htaccess`.

Apache 2 e la sicurezza

In realtà, Apache vanta una storia di alto profilo rispetto alla sicurezza, proprio in virtù del numero di sviluppatori che collaborano a questo progetto e che sono in grado di individuare e correggere tempestivamente ogni eventuale difetto che può essere rilevato e ogni varco che potrebbe essere utilizzato da un hacker per violare il sistema.

Ciononostante, è naturale che uno strumento software così complesso possa manifestare dei difetti e pertanto, come sempre, il migliore accorgimento per garantire la sicurezza del sistema consiste nell'eseguire un costante aggiornamento del software installato sul sistema.

Ancora oggi capita, non di rado, di trovare installazioni software che si basano ancora sulla versione 1.3 di Apache. Ottima versione, ma ormai un po' datata. Occorre anche dire che, solo rarissimamente la violazione di un sistema è effettivamente dovuta al server Web Apache 2. Molto più spesso si tratta di difetti presenti in altri elementi software che compongono il sistema Linux o, tutt'al più, nelle estensioni aggiunte al server o, ancora, nel codice impiegato per la gestione del sito Web (o dei siti Web) gestiti dal server. Per questo motivo è importante anche curare l'aggiornamento di tutti gli altri elementi software presenti nel sistema.

Accesso al sistema da parte degli utenti

Innanzitutto un sito Web non deve essere modificabile da estranei e dunque è necessario impostare una password di accesso alle directory e ai file.

Pertanto occorre innanzitutto creare un file delle password, il quale deve essere collocato fuori dalle directory interessate dal sito Web. Per esempio, se i documenti del sito Web si trovano nella directory `/srv/www/mariorossi/directory`, il file delle password deve trovarsi all'esterno di questo spazio, per esempio in `/etc/http-passwd`.

Creare il file delle password è semplice; basta usare il comando `htpasswd`, sempre presente nel pacchetto di Apache. Dovremo specificare la password per il nome-utente che abbiamo specificato sulla riga di comando e, per sicurezza, dovremo, come di consueto, ripetere la password. Nel file delle password verrà così creata una nuova voce.

```
debian:/# htpasswd -c /etc/http-passwd paolo
New password:
Re-type new password:
Adding password for user paolo
debian:/#
```

Con questo semplice comando creiamo (-c) nella directory `/etc/` il file di password `http-passwd` (Figura 3.1) nel quale viene inserita la voce relativa all'utente `paolo` con la rispettiva password.

Naturalmente è bene che ai file del sito possano accedere anche altri utenti che hanno la necessità di manipolarne i contenuti. Ora che abbiamo creato il file delle password, possiamo aggiungere con facilità nuovi utenti, più o meno come abbiamo fatto per creare il primo utente.

La creazione del file delle password è stata determinata dall'uso del parametro di creazione `-c`. Per aggiungere nuovi utenti al file senza cancellarlo ogni volta basta non specificare tale parametro. L'operazione è eseguibile anche dal comando Applicazioni > Accessori > Terminale root del desktop Gnome di Debian (vedere la Figura 3.2):

```
debian:/# htpasswd /etc/http-passwd alice
New password:
Re-type new password:
Adding password for user alice
debian:/#
```

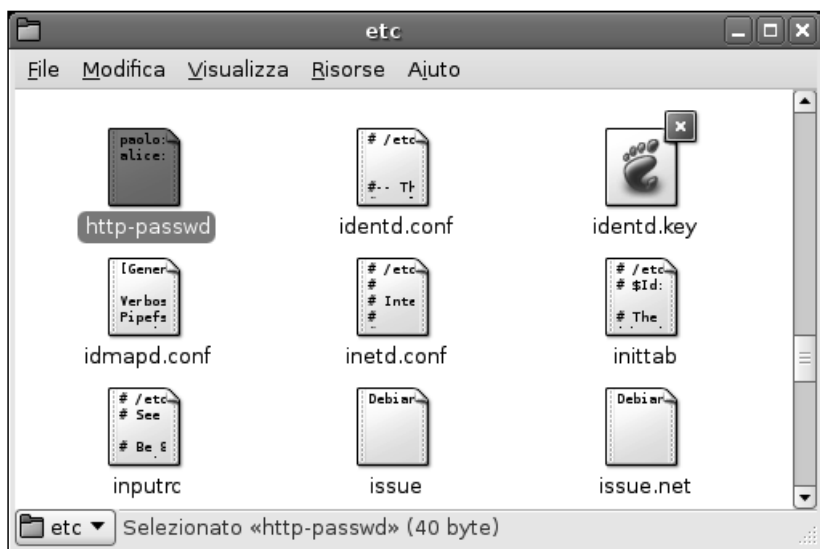



Figura 3.1

Abbiamo inserito il file delle password `http-passwd` al di fuori delle directory che compongono il sito, per metterlo al riparo da occhi indiscreti.

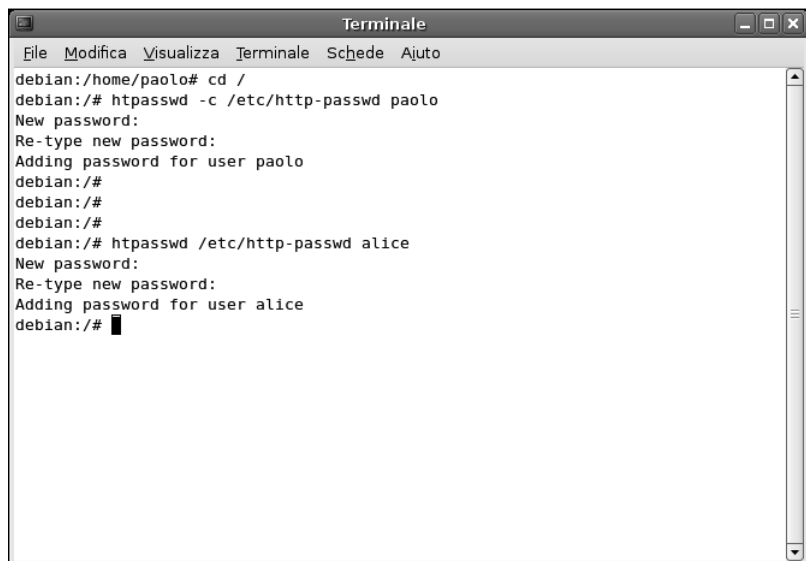


Figura 3.2

I semplici comandi necessari per compilare il file delle password di accesso ai file che compongono il sito Web.

Ora che il file delle password contiene il nome-utente e la password di un paio di utenti (Figura 3.3), dobbiamo configurare il server in modo che richieda una password di accesso e comunicargli gli utenti ai quali intendiamo consentire l'accesso. Per esempio, se dobbiamo proteggere con una password una directory privata, possiamo inserire nel file `apache2.conf` una voce `<Directory ...>`.

```
<Directory /srv/www/mariorossi/directory>  
    AuthType Basic  
    AuthName "Directory ad accesso limitato"  
    AuthUserFile /etc/http-passwd  
    Require user paolo  
</Directory>
```

Le quattro direttive di controllo degli accessi che abbiamo inserito in questa voce `<Directory ...>` hanno il seguente significato.

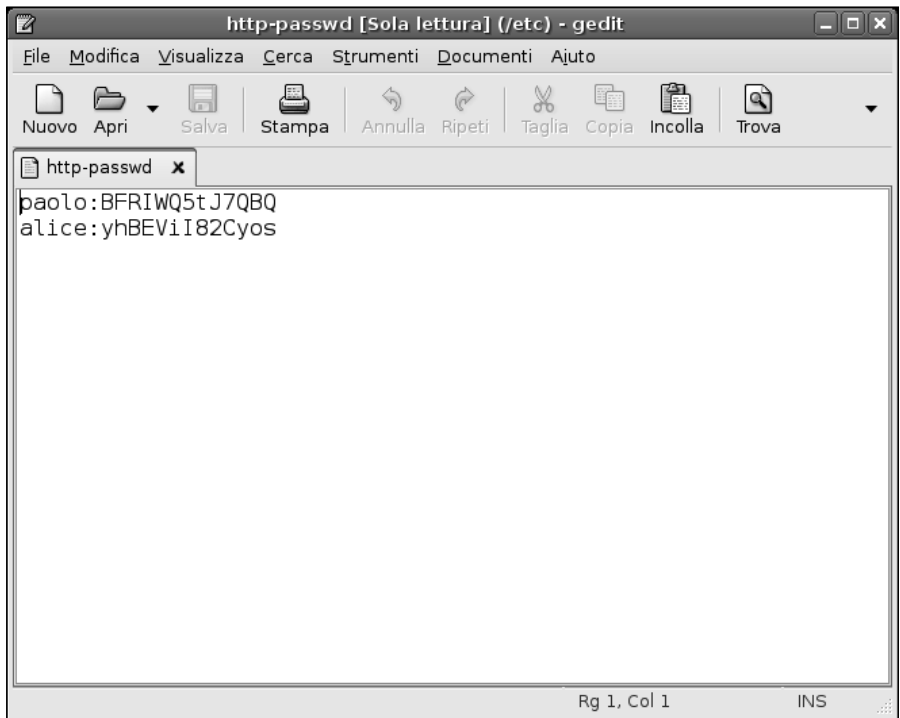


Figura 3.3

L'aspetto del file delle password: solo il nome-utente è conservato in chiaro, la password è codificata.

- **AuthType** Indica il metodo HTTP tramite il quale verrà *autenticato* l'utente; il metodo più utilizzato è **Basic**, implementato dal modulo **mod_auth** di Apache.
- **AuthName** Definisce le informazioni di autorizzazione all'accesso. Tali informazioni vengono impiegate principalmente per due scopi:
 - il programma che gli utenti impiegheranno per accedere alla directory indicata presenterà all'utente queste informazioni all'interno di un'apposita finestra di dialogo di richiesta della password.
 - viene impiegata dal client per determinare la password da inviare a una determinata area di autenticazione del sito, qualora sullo stesso sito Web siano state definite più aree protette.
- **AuthUserFile** Qui dobbiamo specificare il percorso completo del file delle password, che abbiamo appena creato con **htpasswd**.
- **Require** Quali utenti possono accedere all'area indicata del server Web? Questa direttiva rappresenta la parte di *autorizzazione* del processo: dopo che la fase di autenticazione si è conclusa con successo, occorre infatti indicare a quali utenti è consentito l'accesso all'area così definita.

PASSWORD IN CHIARO

Il metodo di autenticazione **Basic** che abbiamo appena impiegato e che è implementato dal modulo **mod_auth** di Apache 2 invia in chiaro il nome-utente e la password attraverso la rete che collega il client al server. Apache 2 offre anche altri metodi di autenticazione, fra i quali il metodo crittografato **AuthType Digest**, implementato dal modulo **mod_auth_digest** di Apache 2. Questo metodo di autenticazione rappresenta un sistema decisamente più sicuro per la gestione della password, in quanto trasmette in rete una versione codificata della password. Questo tipo di autenticazione funziona solo con le versioni aggiornate dei browser Web.

Configurazione dell'accesso per i gruppi

La voce `<Directory ...>` che abbiamo visto all'opera nel paragrafo precedente concede l'accesso solo a un utente, **paolo**. Si tratta di una condizione molto limitante, in quanto non ha molto senso che un solo utente abbia la possibilità di lavorare sui file che costituiscono il sito. È molto più comodo consentire l'accesso a un intero gruppo di utenti. Le opzioni possibili per ottenere questo risultato sono due.

Al posto della direttiva `Require user paolo`, che limita l'accesso al solo utente `paolo`, la direttiva `Require valid-user` limita l'accesso alla directory in questione a tutti gli utenti che abbiamo elencato nel file delle password.

In alternativa possiamo creare un *file dei gruppi* che ci consentirà di associare a un nome di gruppo un elenco di utenti. Anche in questo caso si tratta di un semplice file di testo che può essere creato con un qualsiasi editor di testi:

TeamSito: paolo alice isabella

Ora dobbiamo indicare alla voce `<Directory ...>` dove può trovare il file delle password e il file dei gruppi. Poiché un file dei gruppi può contenere le specifiche relative a più gruppi, dobbiamo anche specificare il gruppo al quale intendiamo consentire l'accesso.

```
<Directory /srv/www/mariorossi/directory>
  AuthType Basic
  AuthName "Accesso limitato al gruppo"
  AuthUserFile /etc/http-passwd
  AuthGroupFile /etc/http-gruppi
  Require group TeamSito
</Directory>
```

A questo punto avranno accesso all'area "Accesso limitato al gruppo" tutti (e solo) gli utenti che sono stati preventivamente autenticati dal sistema (con il metodo `Basic`) e che abbiamo elencato nel gruppo `TeamSito`, definito nel file `/etc/http-gruppi`.

Il metodo di autenticazione non crittografato `Basic` presenta un altro grave difetto: il nome-utente e la password dell'utente devono essere verificati a ogni richiesta di risorse sottoposta al server; può per esempio trattarsi di una pagina HTML, di un'immagine o di qualsiasi altra risorsa contenuta nella directory che intendiamo proteggere. Ciò può allungare drasticamente i tempi di risposta del server Web.

DA SAPERE *A dire la verità, l'entità di questo rallentamento dipende dalle dimensioni del file delle password.*



Il server Web Apache 2 è costretto ad aprire il file e leggerlo tutto o almeno finché non trova l'utente in questione; deve svolgere questa operazione ogni volta che viene richiesto il caricamento di una pagina. Dunque esiste un limite pratico al numero di utenti che possiamo elencare in un file delle password.

Tuttavia, fino a qualche centinaio di utenti contenuti nel file delle password, le prestazioni del server Web non dovrebbero soffrire al punto da costringerci ad adottare un metodo di autenticazione più efficace.

Un'alternativa può essere rappresentata dal modulo `mod_auth_dbm`, il quale offre la direttiva `AuthDBMUserFile`, che consente di usare dei file di password in formato *DBM*, gestibili tramite lo script Perl `dbmmanage` fornito con Apache. In alternativa si può utilizzare il modulo `mod_auth_mysql` che consente al server Web Apache di gestire le coppie nome-utente/password tramite un database MySQL, all'interno del quale le informazioni possono essere consultate e manipolate in modo molto più efficiente.

Tuttavia l'argomento dell'autenticazione avanzata degli utenti sul server Web Apache 2 non rientra negli scopi di questo breve volume.

Prestazioni del server Web Apache

Apache 2 è stato sottoposto, nel corso del tempo, a svariati raffinamenti, volti soprattutto a migliorare la sua flessibilità e la portabilità in altri ambienti, senza ignorare i problemi prestazionali. Sebbene le priorità non riguardassero tanto le prestazioni assolute del server rispetto ad altri prodotti analoghi, quanto piuttosto la sua solidità e utilizzabilità, il software offre ottime prestazioni. Questo è uno dei motivi per cui viene scelto come implementazione di riferimento dalla maggior parte dei nuovi siti Web.

Nel passaggio dalla versione 1.3 alla versione 2, molti miglioramenti apportati tenevano in considerazione proprio i problemi prestazionali. Queste nuove funzionalità vengono normalmente attivate nell'installazione standard di Apache e dunque, in realtà, quello che abbiamo fra le mani è già un prodotto ottimizzato in senso prestazionale, che non richiede ulteriori interventi, se non in casi estremamente particolari.

Tuttavia vi sono alcuni importanti fattori che possono incidere, anche notevolmente, sulle prestazioni del nostro server. Data la velocità sempre più frenetica con cui gli utenti navigano nelle pagine Web di un browser e data la velocità sempre crescente con cui è possibile scaricare le pagine, l'ottimizzazione delle prestazioni del server Web Apache 2 sembra un imperativo per garantire un'esperienza ottimale da parte degli utenti.

Un ambiente ideale

Per prima cosa bisogna assicurarsi che il server abbia a disposizione tutte le risorse di cui ha bisogno. Un requisito che non riguarda necessariamente solo il server Web ma, in generale, tutte le applicazioni in esecuzione sul sistema,

specialmente i servizi che sono sottoposti a un utilizzo continuativo caratterizzato da picchi di lavoro più o meno frequenti.

- Innanzitutto occorre dotare il sistema di una quantità di memoria RAM sufficiente alle esigenze. La quantità di memoria RAM dipende in particolare dall'impostazione del parametro `MaxClients` che determina il livello di picco nel numero di richieste simultanee al server. Si tratta di una procedura empirica che va collaudata caso per caso e considerando il livello di popolarità del sito o dei siti Web ospitati.
- Bisogna curare la velocità dei principali organi del sistema da cui dipende la rapidità della risposta del server Web: CPU, scheda di rete e dischi fissi. Anche in questo caso si deve procedere in modo empirico, cercando di armonizzare i livelli prestazionali di tutti questi componenti. Non avrebbe senso avere una CPU estremamente veloce come una connessione di rete tragicamente lenta o dischi fissi troppo vecchi e lenti. Analogamente, una connessione di rete estremamente veloce non potrà fare nulla se la CPU è troppo lenta (magari perché il server è situato su una macchina piuttosto datata) oppure è costantemente impegnata a eseguire operazioni di swap su disco fisso, a causa di una scarsità di memoria RAM.
- Occorre curare sempre l'aggiornamento del sistema operativo Linux. Spesso gli aggiornamenti riguardano proprio problemi prestazionali, come la gestione degli stack TCP e dei processi. In particolare, a partire dalla versione 2.4 di Linux, nel sistema operativo sono state introdotte delle chiamate di sistema particolarmente efficaci nell'accelerare l'invio di contenuti statici e nel ridurre il consumo del tempo di CPU.

Conclusioni

Il server Web Apache 2 spesso rappresenta un elemento fondamentale nella strategia di accesso a Internet di un'azienda. Per questo motivo è importante che sia sempre in grado di funzionare correttamente e senza rischi di intrusioni.

In questo capitolo abbiamo imparato a limitare l'accesso ai file di un sito solo agli utenti e ai gruppi di utenti che ne hanno l'autorizzazione.

Molti dei problemi prestazionali delle versioni precedenti di Apache sono stati risolti con le nuove versioni. Presumibilmente, sul nostro sistema installeremo una versione 2.x, che apporta notevoli miglioramenti prestazionali, specialmente per quanto riguarda la generazione di nuovi processi "figli", ognuno dei quali ha lo scopo di esaurire una richiesta in arrivo.

Nelle versioni precedenti di Apache, il tempo di creazione di un nuovo processo poteva essere talmente lento da essere incompatibile con una normale fruizione dei contenuti del sito.

Dunque per privilegiare la velocità del sistema è sempre opportuno dotarsi della versione più aggiornata possibile sia del sistema operativo sia del software stesso del server Web.

In ogni caso, se il nostro sito è molto frequentato, è importante ottimizzare e potenziare la velocità dei componenti più stressati dalle attività del server Web, ovvero la CPU, la scheda di rete e i dischi fissi; inoltre occorre curare che la dotazione di memoria RAM sia adeguata al tipo di carico che dovrà essere sopportato dal sistema.

Nel prossimo capitolo esamineremo alcune attività di gestione di un piccolo sito Web.

Capitolo 4

Creazione di un semplice sito Web

Ora che il server Web Apache è attivo, possiamo iniziare a creare il nucleo delle pagine che formeranno il sito Web

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Come funziona il server Web Apache 2
- ☑ La directory predefinita del sito Web

Finora abbiamo predisposto e messo in sicurezza l'accesso al sito Web. Ma, come impostazione predefinita, dove si trovano i file che compongono il sito? Come fare per raggiungerli sulla macchina utilizzata? Come possiamo accedere alle pagine dalla macchina locale e dalle altre eventuali macchine connesse a formare la rete locale o da una macchina connessa dall'esterno via Internet?

Come funziona il server Web Apache 2

Apache è un software particolarmente silenzioso: lavora dietro le quinte senza mai manifestare troppo la propria presenza.

Nel Capitolo 2 abbiamo imparato a utilizzare i comandi di attivazione, ma nulla, sul sistema, ci dice che in realtà sia cambiato qualcosa. Sembra tutto come prima, tuttavia il nostro server Web è vivo e vegeto, pronto a rispondere alle richieste in arrivo dalla macchina, da eventuali altre macchine connesse alla rete locale in cui si trova questa macchina Linux o anche dall'esterno, da Internet. Poiché ormai Linux ci gratifica di un'interfaccia grafica di alto livello con la quale possiamo controllare e conoscere molte delle sue caratteristiche, utilizziamo degli strumenti grafici per confermare che il nostro server Web sia effettivamente in funzione.

In particolare, il comando Desktop > Amministrazione > Servizi apre la finestra Impostazioni servizi che consente di verificare e controllare l'attivazione dei servizi forniti dalla nostra macchina.

La Figura 4.1 mostra per esempio i servizi attualmente attivi sul sistema. Tra questi individuiamo il servizio Server web (apache2).



Figura 4.1

Con il comando Desktop > Amministrazione > Servizi scopriamo quali servizi sono attualmente in funzione sul sistema. Fra questi si trova proprio Apache 2.

OK, il nostro server è attivo. Questo significa che risponderà alle richieste di pagine Web che giungeranno alla macchina sia dall'interno, ovvero dalla macchina stessa, sia dall'esterno, ovvero da una macchina connessa in rete o da Internet. La richiesta giungerà al server il quale risponderà con una pagina Web. Ma dove si trova la pagina Web che verrà restituita?

La directory predefinita del sito Web

Come abbiamo visto nel Capitolo 2, Apache 2 è stato installato in Debian nella directory `/etc/apache2/`. In questa directory si trova la subdirectory `sites-available` (vedere la Figura 4.2), nella quale troviamo il file di testo default, il quale contiene le indicazioni utilizzate dal server Web Apache per rispondere alle richieste di pagine Web.

Il file di testo default contiene una direttiva `<Directory ...>` (vedere la Figura 4.3) che rimanda proprio al luogo in cui Apache conserva come impostazione

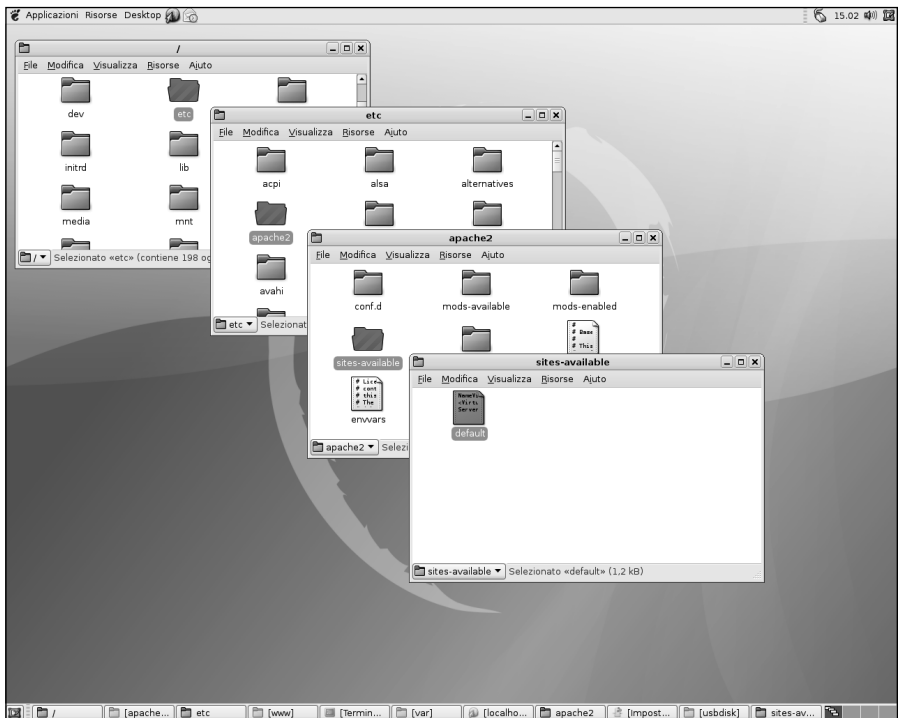


Figura 4.2

Nella directory `sites-available` troviamo il file `default`, che contiene le indicazioni utilizzate dal server per andare a trovare le pagine Web del sito.

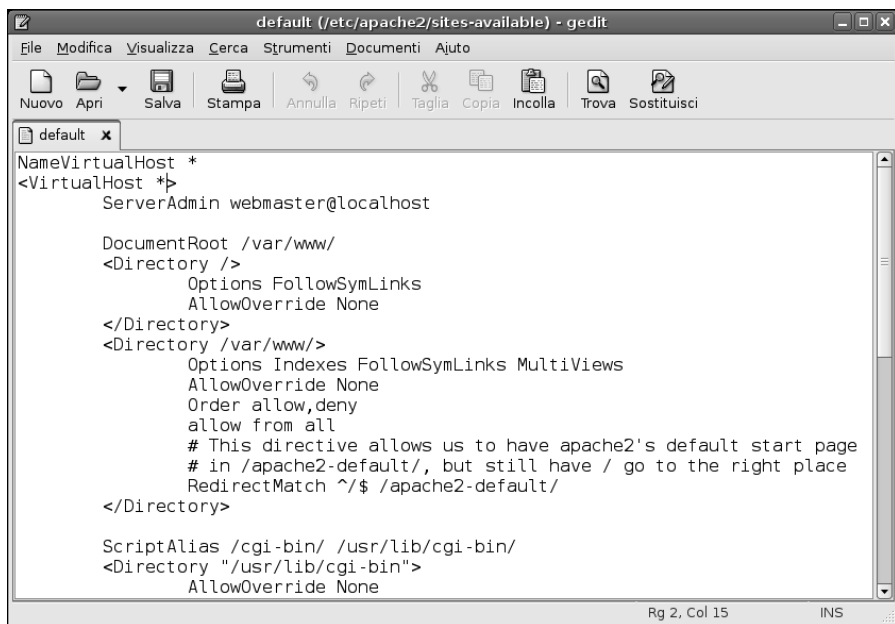


Figura 4.3

Con l'editor grafico gedit osserviamo il contenuto del file default contenuto nella subdirectory sites-available di Apache 2.

predefinita i file che compongono il sito Web. Scopriamo così che tali file si trovano nella directory `/var/www/`. In particolare, la sotto-direttiva `RedirectMatch` indica la subdirectory nella quale verranno presi i file che comporranno il sito Web: `apache2-default`. Quindi il percorso di directory completo per raggiungere i file che formeranno il sito Web sarà proprio `/var/www/apache2-default/`.

Andiamo dunque a vedere che cosa troviamo all'interno di questa directory: come si può vedere nella Figura 4.4, la directory contiene alcuni file grafici che riportano il logo di Apache, più lo "scheletro" del sito Web: la pagina HTML `index.html`. Inizialmente questa pagina contiene solo un titolo: questo ha lo scopo di confermare il funzionamento del server Web sulla macchina.

In pratica, se, dalla macchina locale o da un'altra macchina chiediamo l'intervento del server Web, ci verrà restituito il contenuto di questa pagina.

Un file HTML è un semplice file di testo. Per aprirlo, dall'interfaccia grafica del desktop di Gnome basta fare clic sul suo nome con il pulsante destro del mouse e selezionare dal menu rapido l'opzione `Apri con Editor di testo`. Il file verrà così aperto sempre con l'editor grafico installato sul sistema, normalmente gedit. La Figura 4.5 mostra il documento standard leggermente personalizzato

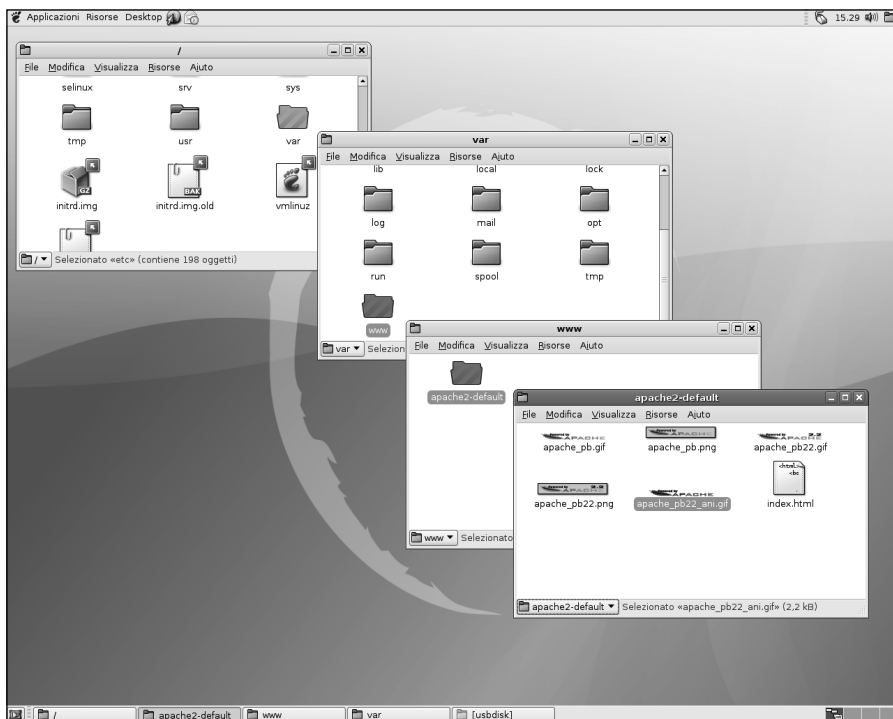


Figura 4.4

La ramificazione di directory all'interno della quale dobbiamo collocare i file che compongono il sito Web.

per mostrare un messaggio in italiano e formattato in modo da renderlo più leggibile. Al momento, in pratica, tale file mostra solamente una conferma di funzionamento del server Web Apache 2.

Con un doppio clic possiamo aprire il file con il browser fornito in dotazione con il sistema (in questo caso si tratta del browser Web Epiphany, in dotazione con il desktop Gnome).

Ma così facendo, non faremo altro che dare in pasto al browser il file, senza passare dal server Web: vogliamo sapere come verrà visualizzato il file e il browser, obbediente, apre il documento HTML e lo riproduce sullo schermo. Niente di più (vedere la Figura 4.6).

Proviamo ora a vedere se il nostro server Web è attivo ed è effettivamente in grado di inviare tale file. In questo caso potremo inserire in questa directory tutte le pagine e le immagini che compongono il sito e che verranno pertanto visualizzate in risposta a ogni richiesta Web. Naturalmente non vi sono molti

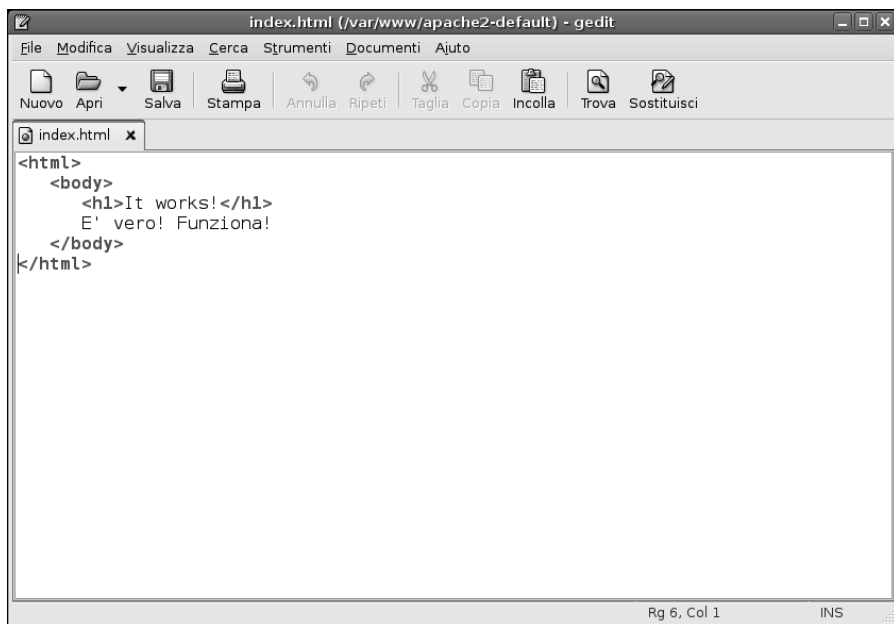


Figura 4.5

Abbiamo voluto personalizzare un po' il file standard presentato da Apache 2 in risposta alle richieste Web.

limiti pratici alla complessità di questo sito: potrà trattarsi di una sola pagina o di un'intera ramificazione di pagine, articolata a piacere.

Usare i servizi del server Web Apache 2

Proviamo ora ad aprire il browser Web Epiphany (o comunque il browser Web che stiamo impiegando sul sistema) e a richiamare il nostro "sito" che attualmente è formato da un'unica semplice pagina HTML. All'apertura, il browser mostrerà la Pagina iniziale che abbiamo precedentemente predisposto nelle Preferenze del browser, in questo caso la pagina di Google Italia (Figura 4.7). Nella barra dell'indirizzo, possiamo digitare l'identificatore che individua la macchina stessa, ovvero, semplicemente, localhost. Il sistema riconoscerà questa richiesta del browser, la indirizzerà al server Web Apache il quale risponderà visualizzando proprio il documento HTML che rappresenta, in nuce, il nostro sito Web (Figura 4.8). Nella barra dell'indirizzo verrà indicato il percorso completo della pagina, ovvero l'indirizzo che avremmo dovuto in realtà scrivere per raggiungere il nostro sito: `http://localhost/apache2-default/`. All'interno di questa directory verrà dunque visualizzato il file `index.html`.

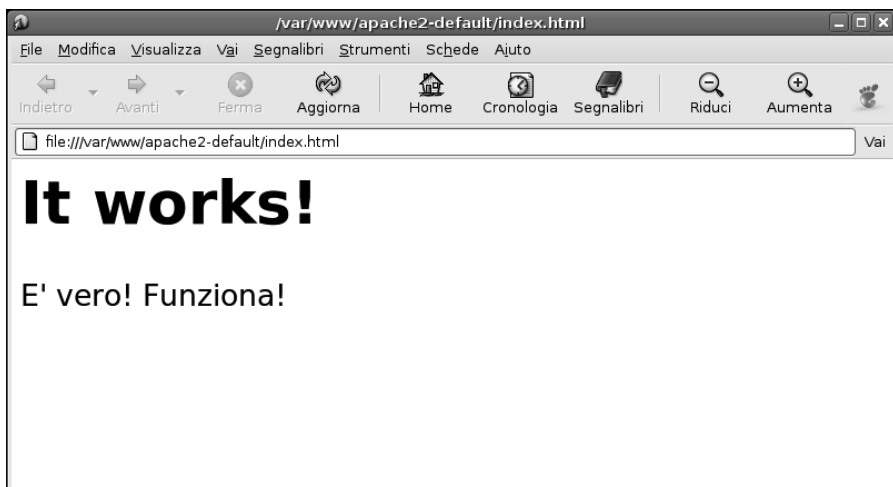


Figura 4.6

Il file aperto dal browser Web Epiphany: in questo caso abbiamo fatto clic qualche volta sul pulsante Ingrandisci per rendere più evidente il testo.



Figura 4.7

La macchina è connessa a Internet e pertanto andrà a richiamare la Pagina iniziale impostata nelle Preferenze.

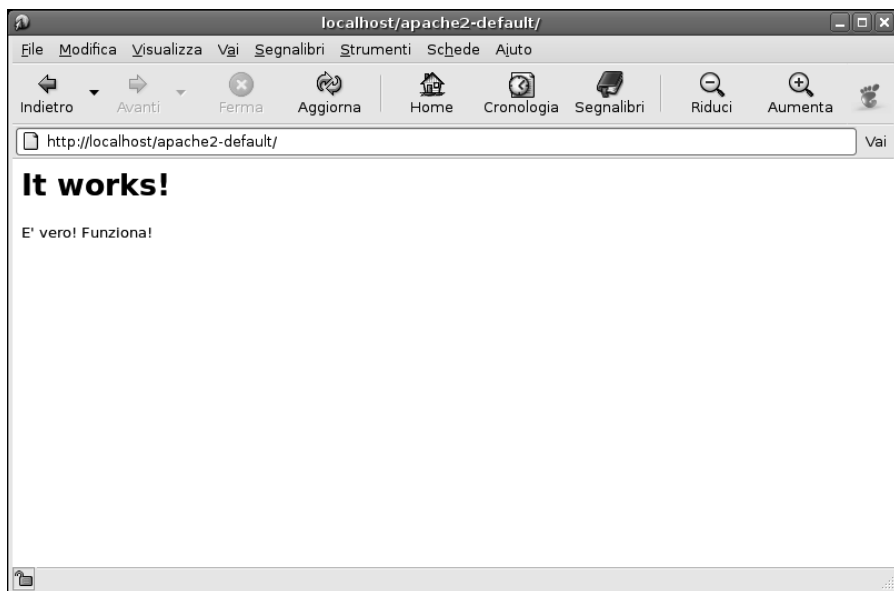


Figura 4.8

Questa volta abbiamo aperto la pagina Web passando attraverso i servizi del server Web Apache 2.

Che cosa è accaduto?

- Sul sistema è arrivata una richiesta di una pagina Web, in particolare una richiesta HTTP, in arrivo sulla porta TCP 80.
- Il server Web Apache 2 è in ascolto proprio su quella porta, riconosce che si tratta di una richiesta HTTP ben formattata, ovvero di una richiesta di una pagina Web.
- Risponde pertanto restituendo il contenuto della pagina index.html situata nella directory del sito. Naturalmente avrebbe potuto trattarsi della home page di un complesso sito costituito da molte più pagine.

Bella forza! Tutto si svolgeva su una sola macchina. È naturale che funzioni! Caso vuole però che la macchina sia connessa a una rete locale, in questo caso di macchine miste Linux e Windows. In particolare, la macchina in questione ha un determinato indirizzo IP interno della rete locale. Vediamo se la macchina è anche in grado di rispondere alle richieste provenienti dall'esterno. Per esempio, la Figura 4.9 mostra una richiesta eseguita con il browser Internet Explorer di una macchina Windows XP e la risposta fornita dalla macchina, ovvero dal suo server Web Apache 2: come possiamo notare, la risposta è esat-

tamente identica a quella fornita dal server Web sulla macchina locale: in altre parole, Apache 2 sta rispondendo anche alle richieste provenienti dall'esterno della macchina, potenzialmente anche da Internet, se abbiamo registrato un dominio le cui richieste vengono pertanto indirizzate dal nostro provider alla nostra rete locale e, in ultima analisi, alla macchina sulla quale è in funzione il server Web.

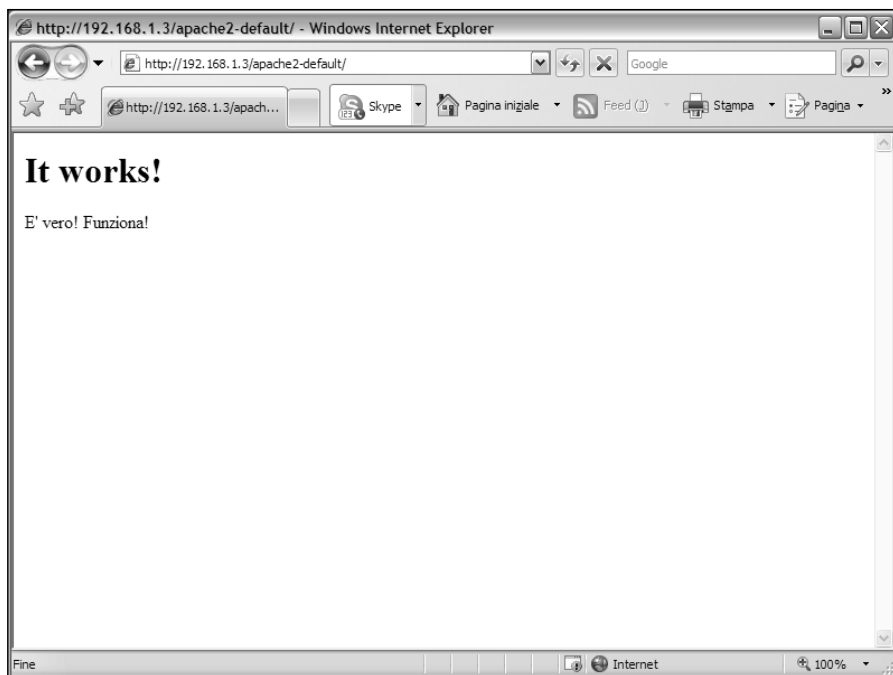


Figura 4.9

Verifichiamo il funzionamento del server Web Apache sulla macchina Linux utilizzando un'altra macchina (con Windows XP) connessa nella rete locale o, potenzialmente, da Internet.

Conclusioni

In questo capitolo abbiamo verificato il funzionamento del server Web Apache. In particolare abbiamo innanzitutto verificato che i servizi del server Web Apache fossero attivi, utilizzando le funzionalità offerte dal menu principale del desktop di Gnome.

Quindi abbiamo individuato nei file di configurazione la posizione in cui devono essere collocati i documenti che andranno a costituire il server Web.

Poi abbiamo personalizzato la semplice e scarna pagina che rappresenta il nostro “sito Web”.

Dopo aver verificato l’aspetto di tale pagina, abbiamo provato a richiamarla, sulla stessa macchina, sfruttando i servizi offerti dal server Web Apache.

Infine abbiamo verificato che il nostro “sito Web” fosse raggiungibile anche dall’esterno, contattando la macchina e, in particolare, il suo server Web da un’altra macchina connessa in rete o, potenzialmente, da Internet.

Nel prossimo capitolo esamineremo le operazioni necessarie per installare e configurare un server di posta elettronica, in grado di smistare la posta elettronica in ingresso e in uscita dal sistema e dalla rete locale.

Capitolo 5

Un server di posta elettronica

C'era una volta... un mondo senza posta elettronica.

C'è da chiedersi come siamo sopravvissuti!

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Perché creare un server di posta elettronica?
- ☑ I componenti di un sistema di posta elettronica
- ☑ Inviare e ricevere messaggi email
- ☑ L'agente MTA Sendmail

La posta elettronica è indubbiamente uno degli strumenti più utilizzati di Internet. La sua popolarità è certamente dovuta al fatto che è estremamente intuitiva, tanto da poter essere utilizzata da tutti, non solo per motivi di lavoro ma anche per studio o anche per tenersi in contatto con amici o parenti lontani (o anche vicini, naturalmente).

La gestione della posta elettronica è forse il secondo utilizzo per cui viene normalmente utilizzata una macchina Linux (il primo è il server Web Apache, che abbiamo esaminato nei capitoli precedenti).

Perché creare un server di posta elettronica?

Per quale motivo dovremmo definire e gestire un server di posta elettronica? In fin dei conti la gestione della posta elettronica da parte del provider è più che adatta alle nostre esigenze. Il vantaggio più importante che otterremo nel gestire un nostro server di posta elettronica è il *controllo*. In ambito professionale, la gestione della posta elettronica può essere un elemento chiave della strategia aziendale. Il controllo della posta elettronica ci offre parecchi vantaggi.

- Mantenendo la gestione dei messaggi all'interno dell'azienda, potremo attivare e gestire uno scambio interno di messaggi, senza passare da Internet e dal provider.
- I messaggi ricevuti e inviati saranno sempre conservati all'interno delle strutture dell'azienda, senza dipendere dalle strutture del provider e dalle sue politiche di backup.
- Non avremo più alcun limite in termini di spazio occupato dai messaggi sui dischi del provider.
- Per le attività più complesse e specifiche di gestione della posta non dipenderemo più dal provider, che si limiterà a una gestione standard dei messaggi in ingresso e uscita.
- Potremo implementare una nostra politica di gestione dei messaggi di spamming e di protezione contro i virus (argomento che affronteremo nel prossimo capitolo).

Per contro, una gestione interna della posta elettronica esige una competenza e una responsabilità superiori.

I componenti di un sistema di posta elettronica

Nel nostro computer Linux dovremo prevedere tre diversi componenti o “agenti”, ognuno dei quali si occupa di un diverso aspetto della gestione della posta elettronica.

- **Agente MTA (Mail Transfer Agent):** è in pratica “l'ufficio postale” locale del nostro sistema o della nostra rete locale. Il suo compito è quello di gestire la ricezione e l'invio dei messaggi, utilizzando il protocollo SMTP (Simple Mail Transfer Protocol). Il principale di questi strumenti è Sendmail.
- **Agente MDA (Mail Delivery Agent):** questo è invece il “postino”, che gestisce la consegna del messaggio nella casella postale di ciascun utente. Per svolgere tale funzionalità può impiegare i protocolli POP3 (Post Office Protocol versione 3) o IMAP (Internet Message Access Protocol). I più noti agenti MDA sono Qpopper e Cyrus.
- **Agente MUA (Mail User Agent):** questo è, più comunemente, il client utilizzato dagli utenti per leggere la posta. La distribuzione Debian utilizza il client Evolution, ma può essere impiegato Mozilla Thunderbird, senza dimenticare le varie versioni di Outlook per machine Windows o i programmi testuali.

Inviare e ricevere messaggi email

Inviare e ricevere la posta elettronica sono operazioni decisamente alla portata di tutti: basta scrivere un messaggio, fare clic sul pulsante di invio e qualcosa nella macchina si preoccuperà di lanciare in Rete il messaggio, il quale uscirà dalla “nube misteriosa” di Internet, proprio nella casella di posta elettronica del nostro corrispondente. Ma, in realtà, che cosa accade dietro le quinte? In base a quali meccanismi il messaggio raggiunge il nostro destinatario?

Il processo è, in realtà, più semplice di quello che si potrebbe immaginare.

1. Luca crea un messaggio per `paolo@mariorossi.it`. Il suo server di posta elettronica SMTP (che si occupa della posta in uscita) è `mail.nomeprovider.it` e dunque il suo client di posta elettronica (poniamo il caso Mozilla Thunderbird) contatta questo sistema “bussando” alla porta 25 (SMTP): l’utente lo ha incaricato di inviare un messaggio a `paolo@mariorossi.it`.
2. Il server di posta elettronica di luca deve verificare la correttezza sia della connessione sia del messaggio. Se il messaggio viene accettato, tale server di posta elettronica prova a inviare il messaggio al destinatario finale, `paolo`.
3. Il server di posta del mittente, luca, sa che la posta indirizzata al dominio `mariorossi.it` non può essere gestita internamente e dunque attiva una ricerca MX (Mail Exchange) di `mariorossi.it`. Un record MX indica il server che gestisce la posta elettronica di `mariorossi.it` e pertanto rimanda a un sistema in grado di gestire la sua posta. Per esempio, nel caso dei destinatari di tipo `indirizzo@mariorossi.it`, il record MX potrebbe avere il seguente aspetto: MX 10 `mail.mariorossi.it`.
4. Ora che il server di posta di luca ha trovato un record MX adatto, tenta di connettersi al relativo server, `mail.mariorossi.it`, sempre sulla porta 25, per recapitare a `paolo` il messaggio inviato da Luca. Anche il server di posta di `paolo` deve assicurarsi di poter accettare la connessione e il destinatario specificato.
5. Poiché il messaggio è indirizzato a `paolo@mariorossi.it`, il server di posta elettronica di `paolo` accetta la connessione e quindi depone il messaggio nella casella di posta locale di `paolo`.
6. Qui la posta verrà conservata finché `paolo` non la preleverà. L’agente MDA presenterà la corrispondenza a `paolo` perché possa prelevarla.

L’agente MTA Sendmail

Sendmail è l’agente MTA più utilizzato in ambiente Linux. Scritto da Eric Allman (l’autore anche di `delivermail`, lo storico agente MTA di ARPANET, la

“nonna” di Internet) nel 1983, Sendmail è stato sviluppato in modo da poter essere configurato dinamicamente intervenendo su un file. Da allora sono stati sviluppati altre soluzioni alternative, ma Sendmail rimane a tutt’oggi l’agente MTA più diffuso in ambiente Linux. Attualmente lo sviluppo di Sendmail viene svolto dalla comunità open-source (www.sendmail.org) e dalla società Sendmail fondata da Eric Allman (www.sendmail.com).

Vedremo quindi come installare, configurare e attivare sendmail su un sistema Linux Debian.

Installiamo Sendmail

Sendmail non è normalmente installato su un sistema Linux Debian. Per installare Sendmail si può utilizzare il Gestore di pacchetti Synaptic, che abbiamo già visto all’opera nel Capitolo 2 per l’installazione di Apache.

1. Avviamo il programma Gestore di pacchetti Synaptic utilizzando il comando Desktop > Amministrazione > Gestore pacchetti Synaptic del desktop Gnome di Debian.
2. Poi dobbiamo fare clic sul pulsante Cerca nella barra degli strumenti, specificare `sendmail` nella casella di testo Cerca e poi fare clic sul pulsante Cerca (Figura 5.1). Troveremo il pacchetto di Sendmail nell’elenco alfabetico dei pacchetti individuati da questa ricerca.
3. Dobbiamo fare clic sulla casella che richiede l’installazione del pacchetto `sendmail` e selezionare dal menu rapido l’opzione Marca per l’installazione. Insieme al pacchetto principale verranno installati altri pacchetti obbligatori, da cui Sendmail dipende (Figura 5.2).



DA SAPERE *Se il sistema si lamenta del fatto che esistono altri problemi relativi alle dipendenze, selezioniamo il comando Impostazioni > Preferenze di Synaptic e attiviamo l’opzione Considerare i pacchetti raccomandati come dipendenze. Se vi sono ulteriori problemi di dipendenze, dovremo contrassegnare manualmente gli altri pacchetti di cui è richiesta l’installazione. La Figura 5.3 mostra un elenco parziale dei pacchetti da installare, contrassegnati in verde. Scorrendo l’elenco si troveranno anche alcuni pacchetti che devono essere disinstallati. Si tratta dell’agente MTA Exim, normalmente installato sui sistemi Debian.*

4. Per avviare il processo di installazione (e rimozione), facciamo clic sul pulsante Applica nella barra degli strumenti di Synaptic. Nella finestra di dialo-

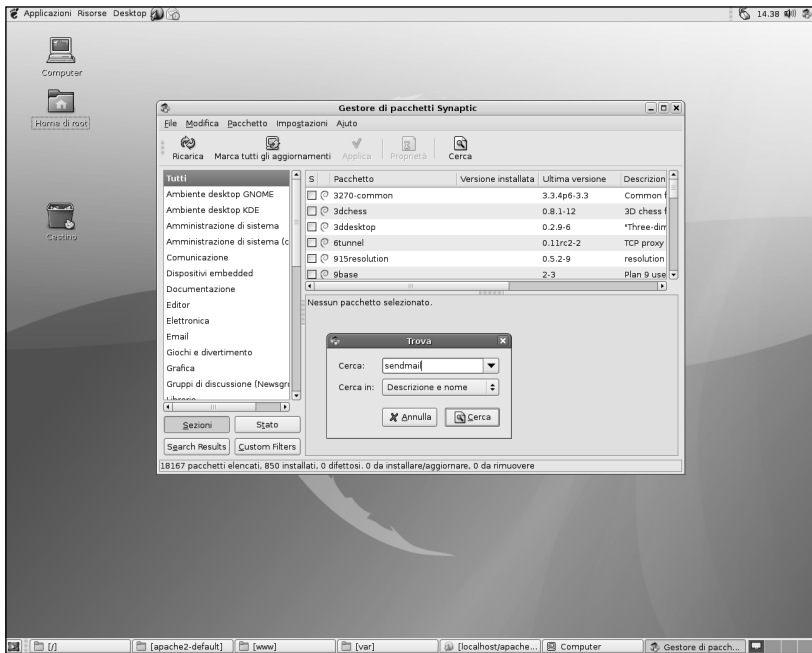


Figura 5.1

Cerchiamo il pacchetto Sendmail nel gestore dei pacchetti Synaptic.

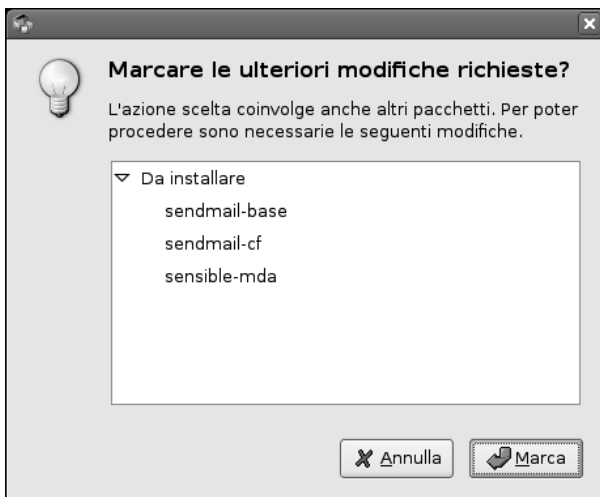


Figura 5.2

Alcuni dei pacchetti che devono essere obbligatoriamente installati insieme a Sendmail.

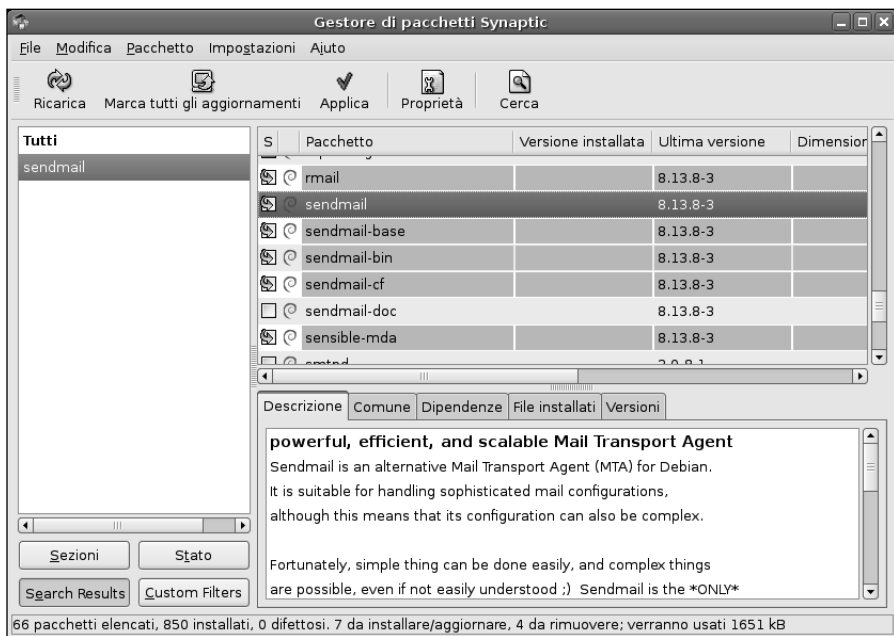


Figura 5.3

Insieme a Sendmail devono essere installati molti altri pacchetti. Non tutti sono elencati in questa figura. Altri invece devono essere rimossi.

go Riepilogo verranno elencati tutti i pacchetti da rimuovere e da installare (Figura 5.4). Facciamo clic sul pulsante **Applica**.

5. Prima verranno scaricati i nuovi pacchetti da installare, poi verranno rimossi i pacchetti previsti e infine verranno installati i nuovi pacchetti di Sendmail. Al termine possiamo fare clic su **Chiudi** e infine chiudere il Gestore di pacchetti Synaptic.

Ora l'agente MTA Sendmail è installato sul sistema.

Configuriamo Sendmail

I file di configurazione di Sendmail si trovano tutti nella directory `/etc/mail` (Figura 5.5). Il file di configurazione principale, `/etc/mail/sendmail.cf`, è un comune file di testo che contiene coppie nome/valore.

Anche Debian, come la maggior parte dei sistemi che utilizza Sendmail, crea il file `/etc/mail/sendmail.cf` a partire da un altro file, `sendmail.mc` utilizzando il processore di macro `m4`.

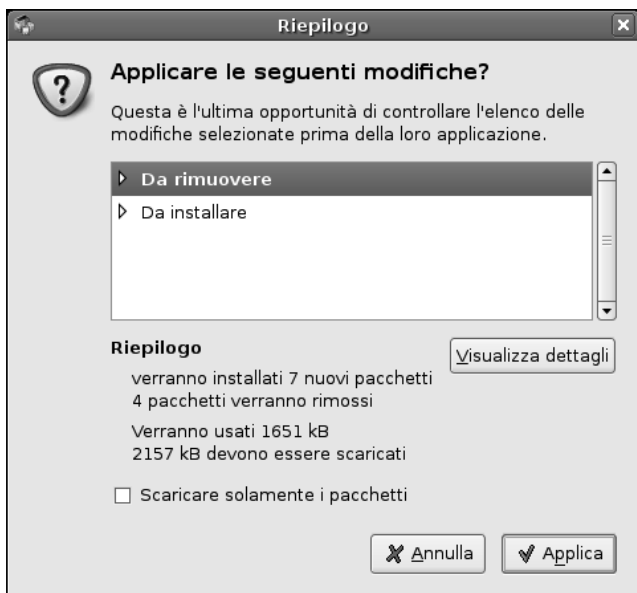


Figura 5.4
Riepilogo dei pacchetti da installare e rimuovere.

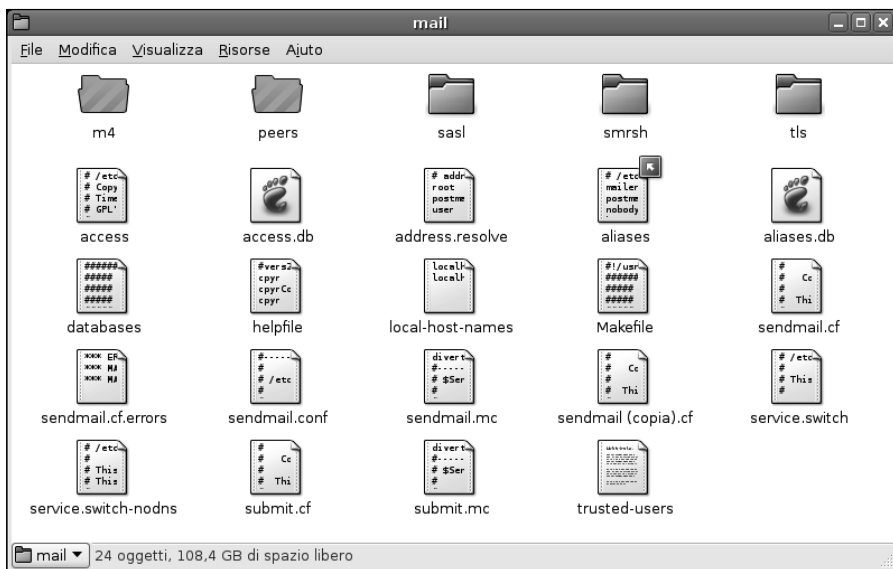


Figura 5.5
Il contenuto della directory di configurazione di Sendmail.

Questi file di configurazione sono esempi straordinari e leggendari di complicazione, astrusione, dipendenze interne e la loro compilazione manuale è assolutamente sconsigliabile. Provate a chiedere in giro a un esperto di Linux; se vi risponderà “Si può fare” vi sono due possibilità: o mente o avete il permesso di prostrarvi di fronte a una vera autorità in materia. Tanto per farsene un’idea, la Figura 5.6 mostra un frammento del file di configurazione `sendmail.mc`.

Per prima cosa dobbiamo intervenire (appena un po’) sul file di configurazione `/etc/mail/sendmail.mc`. Possiamo aprire il file con un doppio clic direttamente dal desktop. Il file si aprirà nell’editor grafico Gedit. Raggiungiamo la parte del file raffigurata nella Figura 5.7 e trasformiamo in commenti le righe evidenziate, facendole precedere dal prefisso `dn1` seguito da uno spazio.



DA SAPERE *Per poter intervenire sui file di sistema dobbiamo naturalmente essere connessi come utenti root. Il modo più semplice (e anche un po’ pericoloso) consiste nell’attivare la connessione grafica per l’utente root: dobbiamo richiamare dal desktop Gnome il comando Desktop > Amministrazione > Finestra login per aprire la finestra di dialogo Preferenza finestra di login. Qui, nella scheda Sicurezza dobbiamo selezionare l’opzione Consentire il login locale dell’amministratore di sistema. A questo punto possiamo disconnetterci come utenti “comuni” e ricollegarci come utenti root, “plenipotenziari” sul sistema. Naturalmente è sconsigliabile lavorare sempre come utenti root: da una grande potenza derivano grandi responsabilità ed è facile commettere errori non (facilmente) recuperabili.*

Ora dobbiamo raggiungere la coda del file e inserirvi le seguenti righe:

```
MAILER(`local')dn1
MAILER(`smtp')dn1
Cw debian.cognome.it
Cw cognome.it
```



DA SAPERE *Attenzione: i due caratteri di apice che precedono e seguono ``local'` e ``smtp'` non sono uguali. La Figura 5.8 mostra graficamente il loro aspetto. Possiamo anche provare a produrli con la tastiera, ma è molto più comodo copiarli e incollarli dalle righe circostanti.*

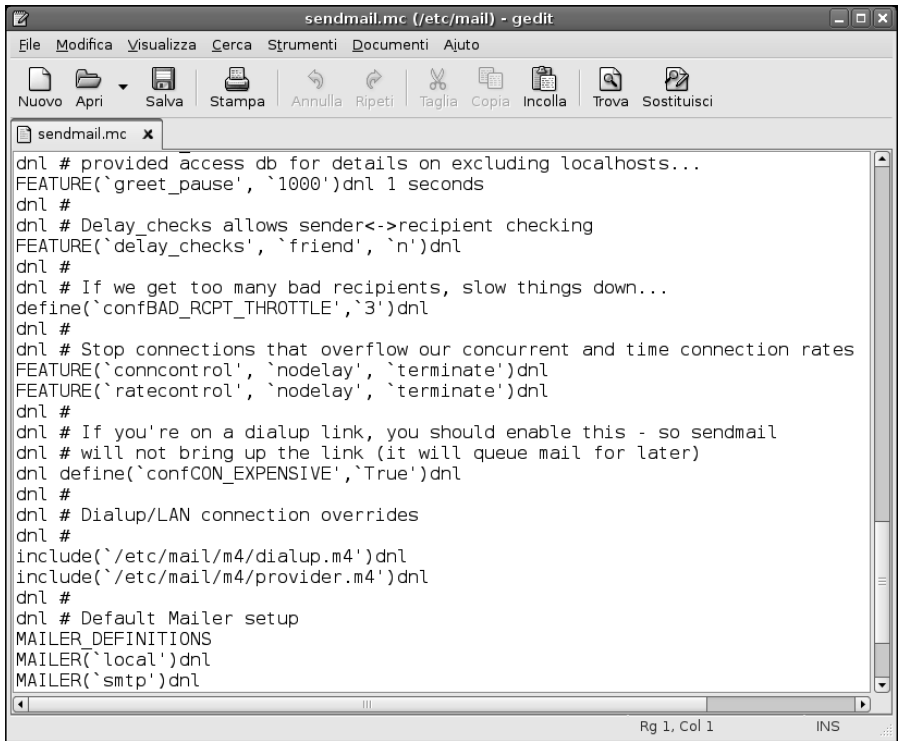


Figura 5.6

Non penseremo certo di mettere troppo le mani su un file di questo tipo? La risposta è “nì”.

A questo punto possiamo configurare sendmail dal Terminale introducendo alla tastiera il seguente comando che semplifica la configurazione degli aspetti più complessi di sendmail.

`sendmailconfig`

Il programma di configurazione `sendmailconfig` porrà alcune domande di base:

- innanzitutto chiederà se vogliamo usare il file `sendmail.conf`; confermiamo premendo Y e poi Invio;
- poi chiederà se vogliamo usare il file `sendmail.mc`; confermiamo ancora premendo Y e poi Invio;
- dopo qualche istante, `sendmailconfig` ci chiederà di riavviare sendmail; confermiamo un'ultima volta premendo Y e poi Invio.

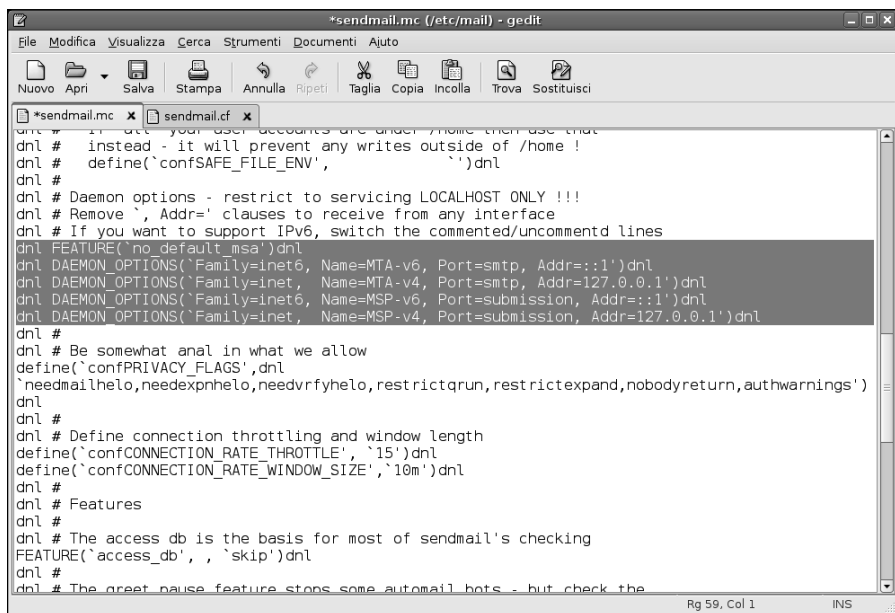


Figura 5.7

Dobbiamo trasformare in commenti le righe evidenziate. Nella figura è visibile il risultato finale.

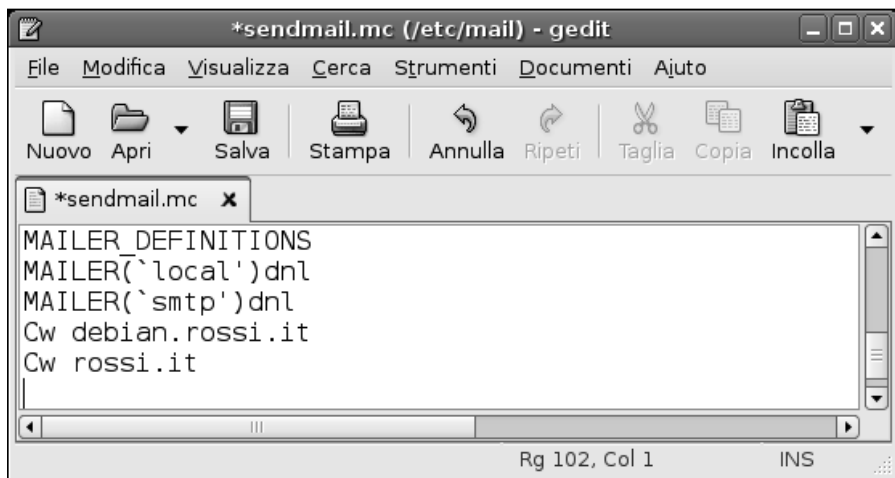


Figura 5.8

Gli apici aperto e chiuso utilizzati in queste righe del file sendmail.mc.

Terminata la configurazione di sendmail dovremo intervenire con l'editor di testi sul file `/etc/hosts`. Per aprire il file basta, come di consueto, fare doppio clic su di esso.

Dobbiamo assicurarci che la riga relativa al sistema locale contenga i seguenti elementi:

- l'indirizzo IP del nostro sistema, seguito da un segno di tabulazione;
- il nome-host del nostro sistema, seguito da un segno di tabulazione;
- il nome FQDN (Fully Qualified Domain Name) del nostro sistema, seguito da un segno di tabulazione;
- le due parole `localhost` e `mailhost` separate da uno spazio.

In pratica le righe del file dovranno avere un aspetto simile al seguente (specificando naturalmente nella seconda riga l'indirizzo IP corretto):

```
127.0.0.1      localhost.localdomain      localhost
192.168.10.10  debian.cognome.it             debian    loghost mailhost
```

L'aspetto del file è rappresentato nella Figura 5.9.

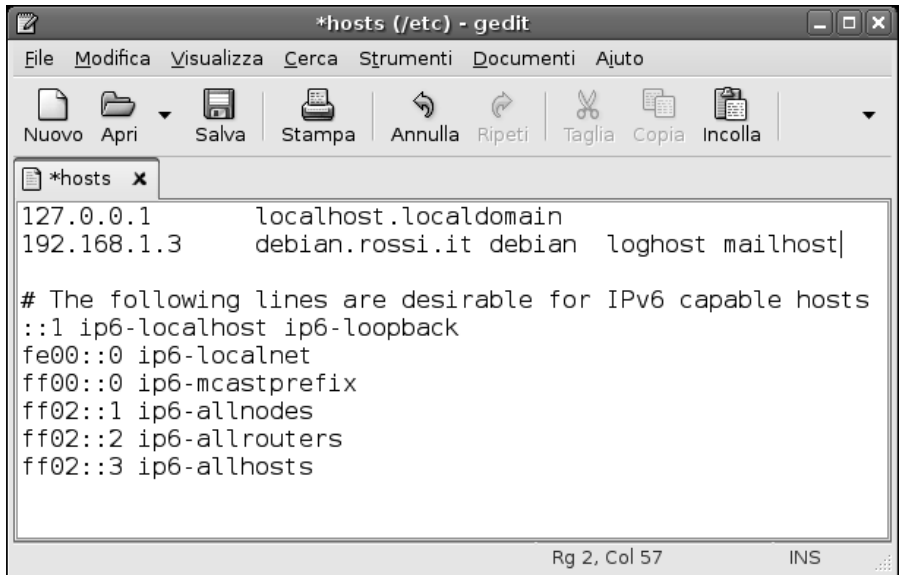


Figura 5.9

Il nostro file `/etc/hosts` dopo le modifiche.

Questo è tutto: il server è configurato. Ora dovremo riavviare Sendmail, poiché legge i file di configurazione solo all'avvio. Visto però il tipo di interventi che abbiamo eseguito, in genere in questa fase è preferibile riavviare il sistema.

Quello che abbiamo attivato adesso è un server che funziona solo nel sistema o nella rete locale. Per farlo funzionare davvero dobbiamo chiedere al nostro provider di cambiare il record MX (ne abbiamo parlato all'inizio del capitolo) in modo che punti proprio alla nostra connessione Internet o al firewall che esegue la traduzione degli indirizzi per il nostro sistema di posta.

I file di Sendmail in un sistema Debian si trovano nelle seguenti directory.

- `/etc/mail` Directory dei file di configurazione.
- `/var/spool/mail` Coda dei messaggi in arrivo. I messaggi sono conservati all'interno di file corrispondenti a ciascun nome-utente. Si tratta proprio delle "caselle postali" degli utenti, conservate sul server di posta elettronica. La casella postale è un unico file contenente più messaggi. Questo file viene "ripulito" ogni volta che gli utenti scaricano i messaggi sul proprio account locale (a meno che il client sia impostato in modo da lasciare i messaggi sul server).
- `/var/spool/mqueue` Coda dei messaggi in uscita. Se disponiamo di una connessione full-time a Internet, questa directory dovrebbe essere vuota, poiché i messaggi verranno inviati immediatamente dopo aver riconosciuto il destinatario via DNS.
- `/usr/sbin` Il file eseguibile di Sendmail.

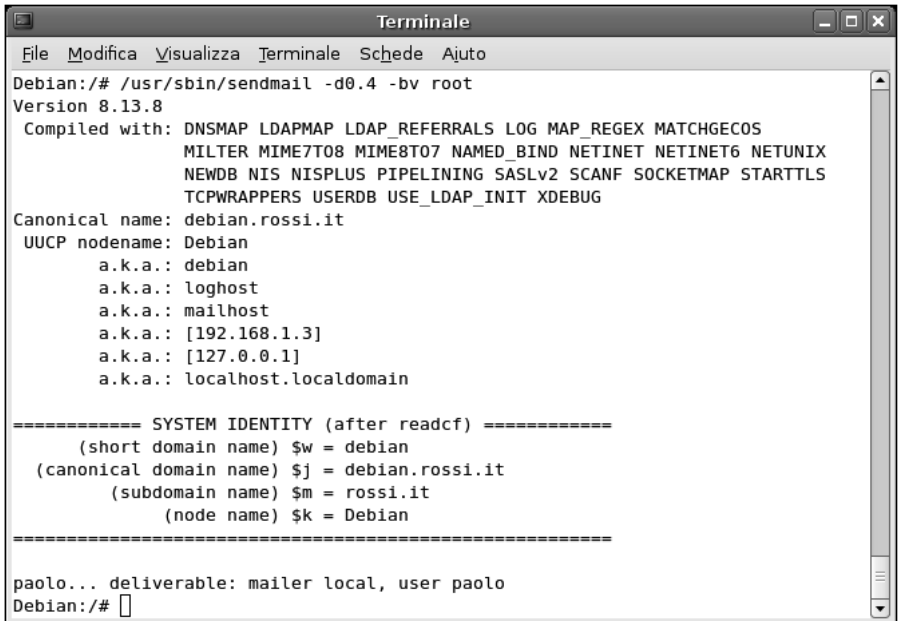
Per conoscere la versione di Sendmail impiegata e altre informazioni utili per la configurazione della posta elettronica e di Sendmail, si può digitare il seguente comando:

```
/usr/sbin/sendmail -d0.4 -bv root
```

Verrà visualizzato il risultato rappresentato nella Figura 5.10, dove troviamo il numero di versione e altre informazioni utili per configurare i client di posta elettronica e per i rapporti con il provider Internet.

Conclusioni

Configurare un sistema di posta elettronica su un sistema Linux Debian non è molto complesso, specialmente se si sfruttano gli strumenti messi a disposizione dalla distribuzione stessa, che agevolano gli interventi sui file di configu-



```

Terminale
File Modifica Visualizza Terminale Schede Ajuto
Debian:/# /usr/sbin/sendmail -d0.4 -bv root
Version 8.13.8
Compiled with: DNSMAP LDAPMAP LDAP_REFERRALS LOG MAP_REGEX MATCHGECOS
               MILTER MIME7TO8 MIME8TO7 NAMED_BIND NETINET NETINET6 NETUNIX
               NEWDB NIS NISPLUS PIPELINING SASLv2 SCANF SOCKETMAP STARTTLS
               TCPWRAPPERS USERDB USE_LDAP_INIT XDEBUG
Canonical name: debian.rossi.it
UUCP nodename: Debian
a.k.a.: debian
a.k.a.: loghost
a.k.a.: mailhost
a.k.a.: [192.168.1.3]
a.k.a.: [127.0.0.1]
a.k.a.: localhost.localdomain

===== SYSTEM IDENTITY (after readcf) =====
(short domain name) $w = debian
(canonical domain name) $j = debian.rossi.it
(subdomain name) $m = rossi.it
(node name) $k = Debian
=====

paolo... deliverable: mailer local, user paolo
Debian:/# 

```

Figura 5.10

L'output prodotto dal comando informativo `/usr/sbin/sendmail -d0.4 -bv root`.

razione di Sendmail, notoriamente uno degli argomenti più ostici e complessi di Linux

Nel prossimo capitolo ci occuperemo della protezione del nostro nuovo server email.

Capitolo 6

Anti-virus e anti-spamming

La posta elettronica è il veicolo principale con il quale esplodono le epidemie virali. Impariamo a proteggerci dai virus e dalla piaga dello spamming.

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Scansione dei messaggi
- ☑ ClamAV
- ☑ Installazione di ClamAV
- ☑ Collaudo dell'anti-virus ClamAV
- ☑ SpamAssassin una difesa contro lo spamming
- ☑ Perché filtrare con un anti-spammer la posta elettronica
- ☑ Installare SpamAssassin
- ☑ Configurazione dei client di posta elettronica

Se da un lato è vero che i sistemi Linux sono decisamente meno sensibili all'attacco dei virus rispetto alle macchine Windows, è anche vero che non si può lasciare sguarnito il sistema, in balia di un potenziale, improbabile e per questo ancora più pericoloso virus per Linux.

Ma c'è di più: se la macchina Linux funge da server di posta elettronica per le macchine di una rete locale mista Linux/Windows, potrebbe trasformarsi in un veicolo inconsapevole dei virus.

In altre parole non possiamo permetterci di ignorare su una macchina Linux il problema della protezione contro i virus.

Nella seconda parte del capitolo vedremo inoltre come proteggerci contro un'altra delle grandi piaghe di Internet: i messaggi spamming.

Scansione dei messaggi

Potremmo lasciar eseguire la scansione dei messaggi di posta elettronica in arrivo sulle macchine client, ma la realtà è che è pericoloso contare sul fatto che in ogni macchina sia stato correttamente installato e aggiornato un software anti-virus. Possiamo quindi prevedere una scansione centralizzata dei messaggi sul server di posta elettronica, il quale avrà il compito di garantire che tutti i messaggi inviati o ricevuti siano esenti da virus.

Linux offre varie soluzioni anti-virus. In questo capitolo tratteremo l'installazione e l'uso del noto software open-source ClamAV, per il quale viene frequentemente aggiornato il database dei virus.

ClamAV

Clam AntiVirus (la cui home page www.clamav.org è visibile nella Figura 6.1) è un software anti-virus open-source che si integra con facilità con i server di posta elettronica, eseguendo la scansione degli allegati per eliminare i virus noti. Il pacchetto è costituito da un servizio di scansione flessibile e stabile, uno scanner a riga di comando e uno strumento di aggiornamento automatico.

ClamAV offre un database di virus fra i più efficaci e aggiornati, in grado di rilevare virus, worm e trojan per tutti i sistemi e le applicazioni, compresi i virus a macro per Microsoft Office. Ecco alcuni fra i tipi di documenti contro i quali ClamAV è in grado di proteggere.

- File eseguibili compressi. Questo è il tipico formato degli eseguibili Windows e uno dei metodi più comuni per la diffusione di virus.
- Documenti Windows contenenti script o eseguibili, in particolare i file Microsoft OLE2, i file cabinet Microsoft, i file Microsoft CHM (Compressed HTML) e i file Microsoft SZDD.
- Altri formati di archiviazione/compressione che ClamAV è in grado di analizzare sono: RAR (2.0), ZIP, Gzip, Bzip2 e Tar.

Installazione di ClamAV

Come di consueto, per installare un nuovo software sul nostro sistema Debian possiamo ricorrere al Gestore di pacchetti Synaptic, che semplifica la risoluzione delle dipendenze da altri pacchetti.

Facciamo quindi clic su Desktop > Amministrazione > Gestore pacchetti Synaptic e, nella finestra del programma, facciamo clic sull'icona Cerca nella barra degli strumenti. Nella finestra di dialogo Trova scriviamo clamav e poi facciamo

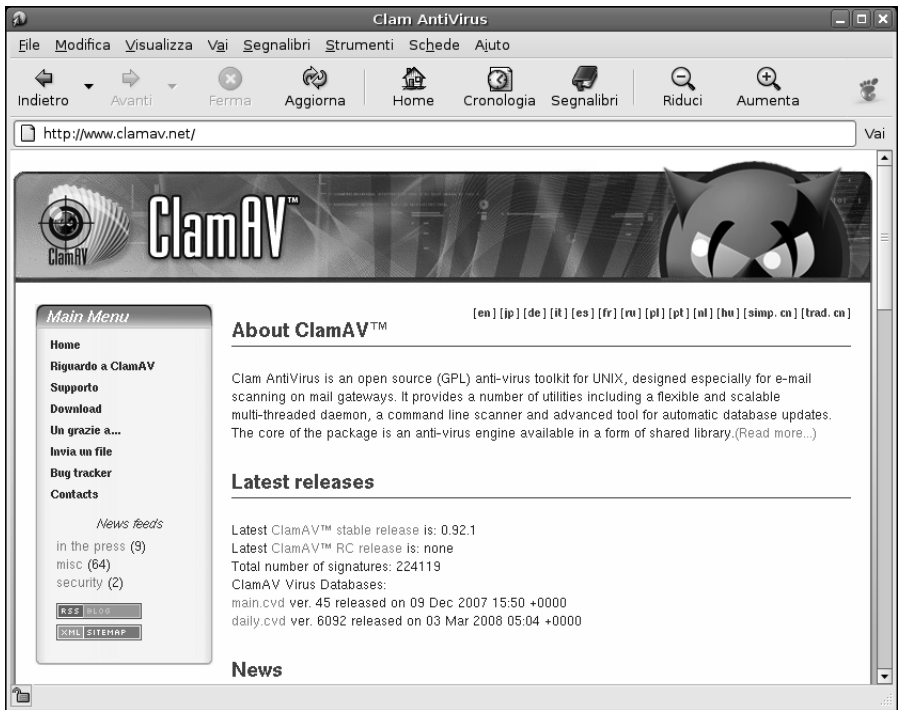


Figura 6.1
La home page del progetto antivirus ClamAV.

clic sul pulsante Cerca. Verranno visualizzati tutti i programmi che hanno a che fare con lo scanner anti-virus ClamAV.

Ora facciamo clic sulla casella quadrata che si trova a lato della voce `clamav` e selezioniamo l'opzione `Marca` per l'installazione. Anche in questo caso, come nei casi precedenti, verranno segnalati tutti gli altri pacchetti che devono essere installati insieme al pacchetto principale. A questo punto la situazione dovrebbe essere quella rappresentata nella Figura 6.2.

Oltre ai pacchetti da cui dipende il pacchetto principale di ClamAV occorre installare anche i pacchetti `clamav-daemon` (il daemon che esegue il controllo anti-virus), `clamav-milter` (lo scanner anti-virus per Sendmail) e `clamav-testfiles` (file utili per verificare il corretto funzionamento di ClamAV).

Per avviare l'installazione, possiamo fare clic sul pulsante `Applica` nella barra degli strumenti e confermare facendo clic sul pulsante `Applica` nella finestra `Riepilogo`. In pochi istanti, i file verranno scaricati da Internet e poi inizierà l'installazione del software (Figura 6.3).

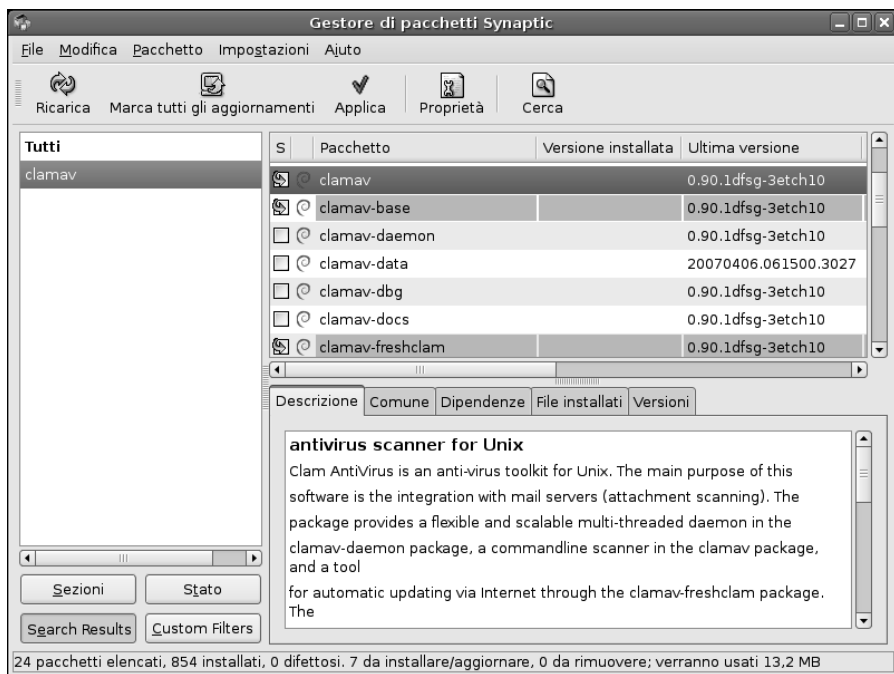


Figura 6.2

Abbiamo richiesto l'installazione di ClamAV e dei pacchetti da cui dipende.

Test post-installazione

Possiamo assicurarci che il software sia correttamente installato, provando il seguente test, che esegue una scansione della directory corrente:

```
Debian:~# clamscan
```

Il risultato sarà analogo a quanto visualizzato nella Figura 6.4. Per eseguire una scansione ricorsiva, possiamo utilizzare l'opzione `-r`. Per eseguire una scansione completa del sistema usiamo il comando:

```
Debian:~# clamscan -r /
```

I file di configurazione

L'installazione ha già impostato l'ambiente adatto per l'esecuzione dell'anti-virus. In particolare avrà già avviato il daemon anti-virus `clamd` che esegue la scansione continua dei messaggi.

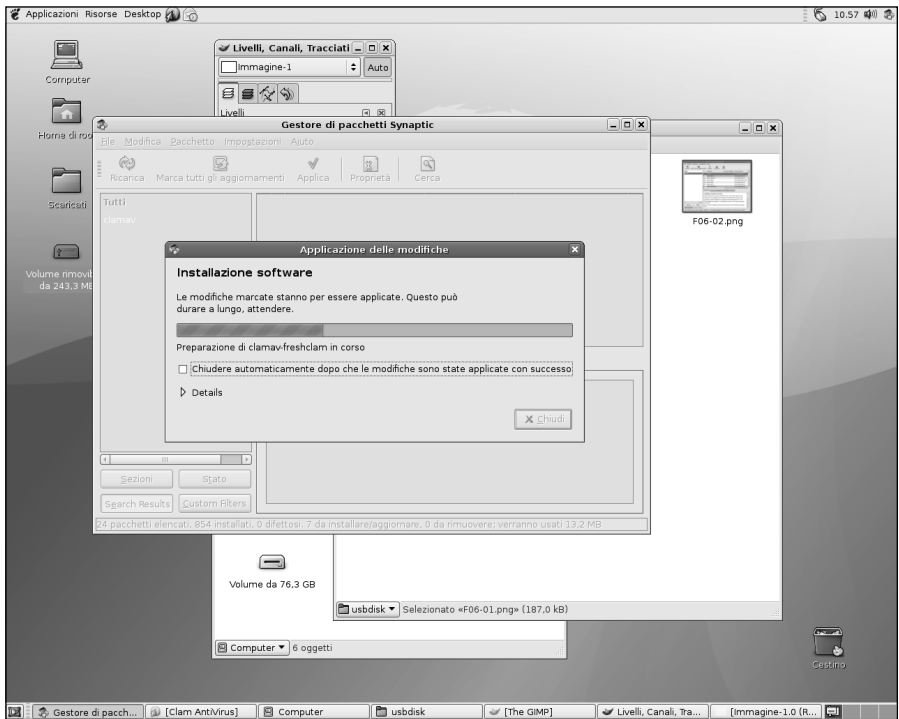


Figura 6.3
È in corso l'installazione del software ClamAV.

I due file di configurazione di ClamAV sono `/etc/clamav/clamd.conf`, che configura il software di scansione dei virus, e `/etc/clamav/freshclam.conf`, dove si configurano gli aggiornamenti automatici al database dei virus.

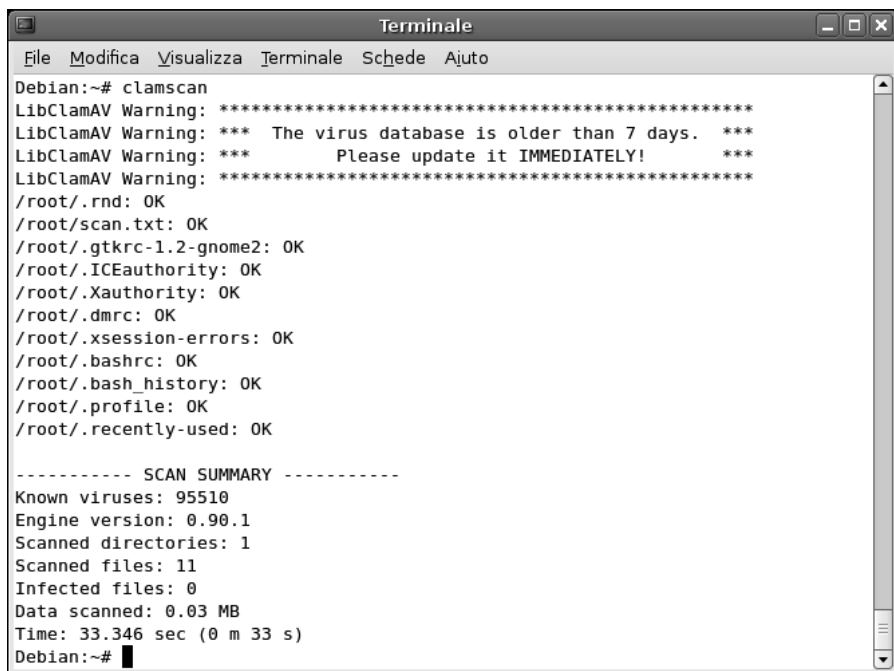
Il file `clamd.conf`

Come si può facilmente verificare, il daemon di scansione `clamd` è già attivo e funzionante:

```
Debian:~# clamd
Running as user clamav (UID 110, GID 114)
```

Il messaggio mostra anche che l'installazione ha creato un nuovo utente `clamav` dedicato alla configurazione di ClamAV.

La Figura 6.5 mostra l'aspetto del file `/etc/clamav/clamd.conf`, che in generale non deve essere modificato dopo l'installazione.



```

Terminale
File Modifica Visualizza Terminale Schede Ajuto
Debian:~# clamscan
LibClamAV Warning: *****
LibClamAV Warning: *** The virus database is older than 7 days. ***
LibClamAV Warning: *** Please update it IMMEDIATELY! ***
LibClamAV Warning: *****
/root/.rnd: OK
/root/.scan.txt: OK
/root/.gtkrc-1.2-gnome2: OK
/root/.ICEauthority: OK
/root/.Xauthority: OK
/root/.dmrc: OK
/root/.xsession-errors: OK
/root/.bashrc: OK
/root/.bash_history: OK
/root/.profile: OK
/root/.recently-used: OK

----- SCAN SUMMARY -----
Known viruses: 95510
Engine version: 0.90.1
Scanned directories: 1
Scanned files: 11
Infected files: 0
Data scanned: 0.03 MB
Time: 33.346 sec (0 m 33 s)
Debian:~# █

```

Figura 6.4
Test della scansione di ClamAV.

Il file freshclam.conf

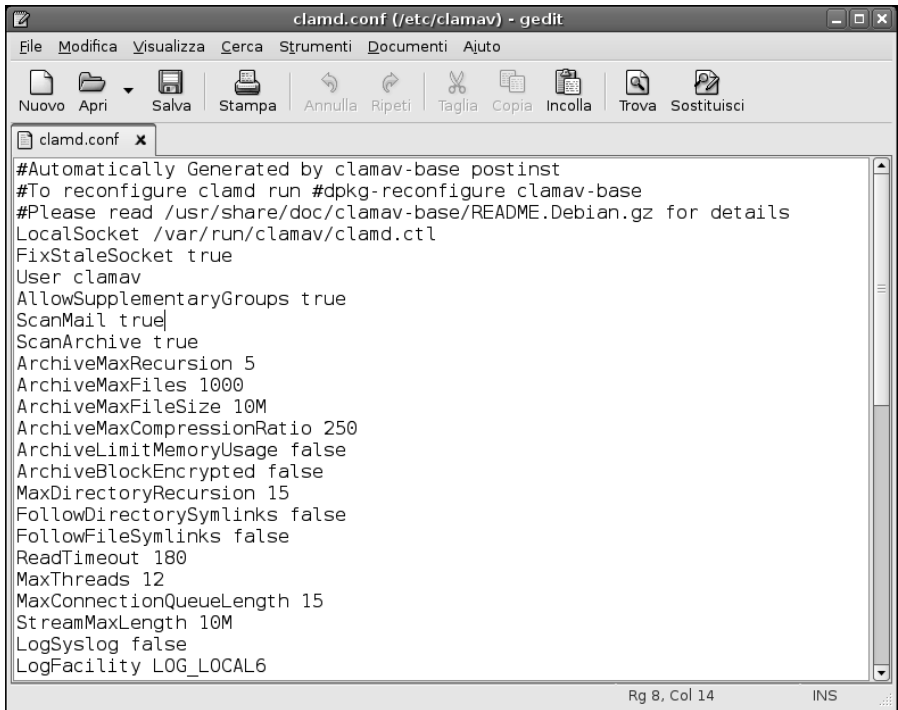
Qui invece si trovano alcune informazioni utili per l'aggiornamento del database dei virus.

Possiamo aggiornare il database dei virus introducendo il comando:

```
Debian:~# freshclam
```

Il comando ricerca nel file `freshclam.conf` (Figura 6.6) le informazioni di configurazione degli aggiornamenti. Particolarmente interessanti sono i parametri `DatabaseDirectory /var/lib/clamav` (che indica la directory contenente il database aggiornato dell'antivirus) e `Checks 24` che esegue un controllo e un eventuale aggiornamento dei file ogni ora (per l'appunto, 24 volte al giorno).

Un'altra voce del file di configurazione su cui si può intervenire è `DatabaseMirror`. Il mirror predefinito per scaricare il database dei virus è `database.clamav.net`, ma nel file `freshclam.conf` si possono specificare più voci. Per esempio si

**Figura 6.5**

Il file di configurazione del daemon di ClamAV.

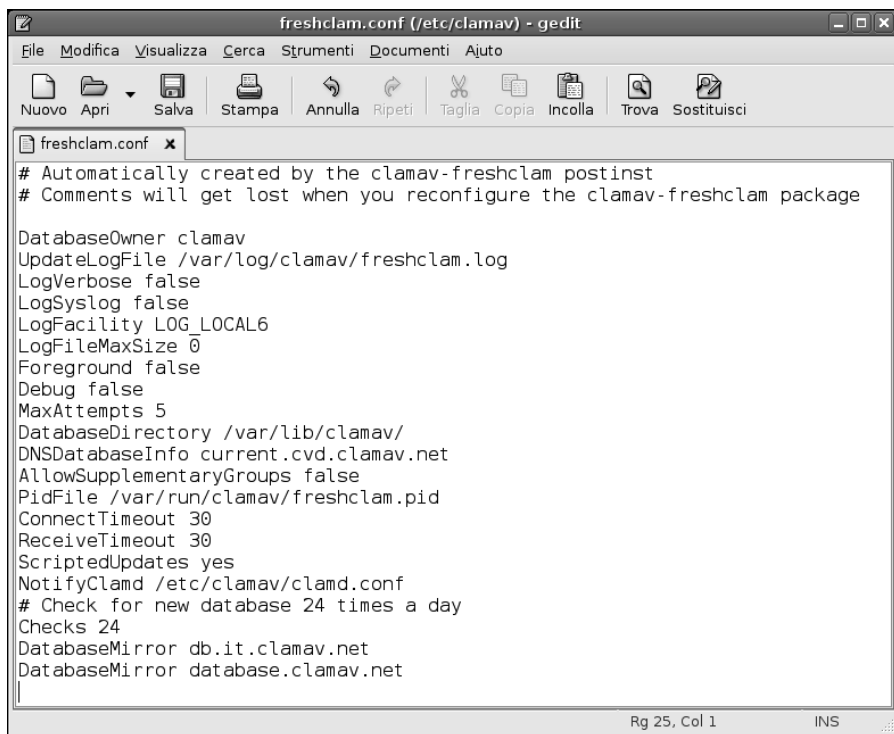
può specificare la voce `db.it.clamav.net`, specifica per l'Italia. Dunque alla fine del file `freshclam.conf` dovrebbero trovarsi le due righe seguenti:

```
DatabaseMirror db.it.clamav.net
DatabaseMirror database.clamav.net
```

Se l'aggiornamento non dovesse essere in grado di utilizzare la prima voce, verrà effettuato un tentativo di eseguire il download dalla seconda voce mirror.

Collaudo dell'anti-virus ClamAV

In rete esistono file che pur non essendo virali devono essere riconosciuti dai software antivirus. Se ne può scaricare uno dal sito dell'EICAR (European Institute for Computer Anti-virus Research) all'indirizzo <http://www.eicar.org/>

**Figura 6.6**

Il file `freshclam.conf` dopo le modifiche apportate alle direttive di aggiornamento del database dei virus.

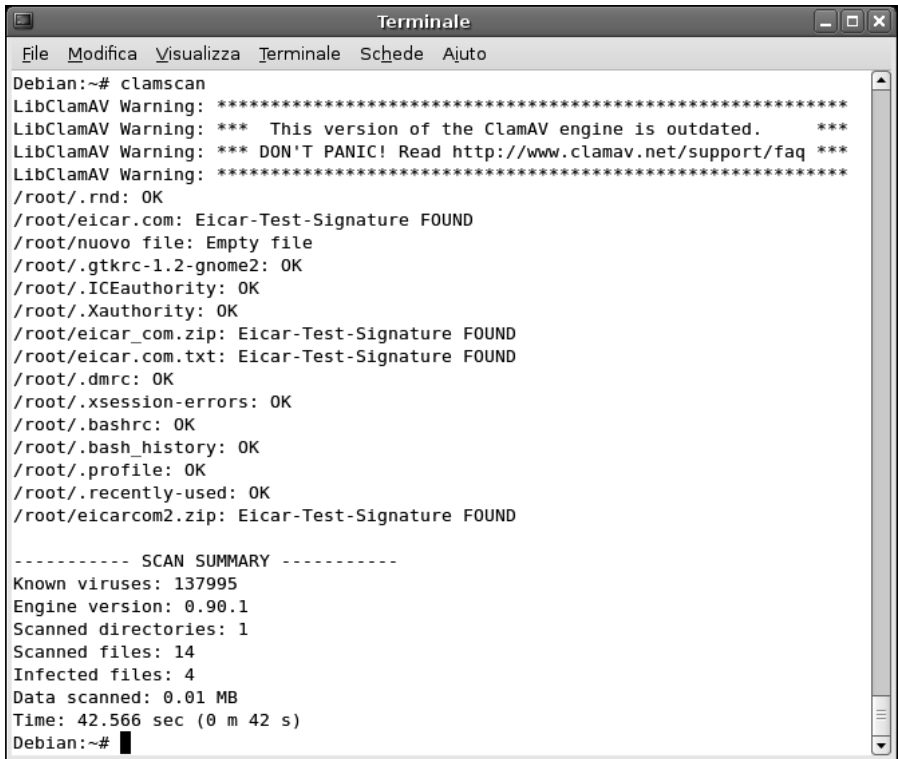
`anti_virus_test_file.htm`. Si tratta di un innocuo file di puro testo costituito da caratteri ASCII. Ogni anti-virus che supporti il test EICAR dovrebbe rilevare questo “virus” in ogni file che inizi con i seguenti caratteri:

```
X50!P%0AP[4!PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

Naturalmente non è il caso di scrivere manualmente tutta questa sequenza: possiamo scaricare i relativi file di test dalla pagina Web dell’EICAR e memorizzarli in una cartella sulla quale intendiamo eseguire la scansione.

In questo modo ci assicureremo che lo scanner anti-virus sia installato e che funzioni correttamente.

La Figura 6.7 mostra cosa accade dopo aver copiato i file di test nella directory dell’utente root e dopo aver lanciato manualmente l’antivirus.



```

Terminale
File Modifica Visualizza Terminale Schede Ajuto
Debian:~# clamscan
LibClamAV Warning: *****
LibClamAV Warning: *** This version of the ClamAV engine is outdated. ***
LibClamAV Warning: *** DON'T PANIC! Read http://www.clamav.net/support/faq ***
LibClamAV Warning: *****
/root/.rnd: OK
/root/eicar.com: Eicar-Test-Signature FOUND
/root/nuovo file: Empty file
/root/.gtkrc-1.2-gnome2: OK
/root/.ICEauthority: OK
/root/.Xauthority: OK
/root/eicar_com.zip: Eicar-Test-Signature FOUND
/root/eicar.com.txt: Eicar-Test-Signature FOUND
/root/.dmrc: OK
/root/.xsession-errors: OK
/root/.bashrc: OK
/root/.bash_history: OK
/root/.profile: OK
/root/.recently-used: OK
/root/eicarcom2.zip: Eicar-Test-Signature FOUND

----- SCAN SUMMARY -----
Known viruses: 137995
Engine version: 0.90.1
Scanned directories: 1
Scanned files: 14
Infected files: 4
Data scanned: 0.01 MB
Time: 42.566 sec (0 m 42 s)
Debian:~# █

```

Figura 6.7

Scansione di virus: ClamAV ha individuato con successo i file di test.

Con il daemon attivo e l'antivirus aggiornato, potremo tenere lontane tutte queste minacce.

SpamAssassin una difesa contro lo spamming

SpamAssassin è il più noto software anti-spamming open-source per Linux: essendo in grado di eliminare dall'80 al 95% dello spamming, è decisamente il miglior strumento gratuito, ma è anche considerato migliore di tanti prodotti commerciali. Di seguito vedremo come scaricare e installare SpamAssassin per filtrare la posta in arrivo.

Perché filtrare con un anti-spammer la posta elettronica

Se non abbiamo mai ricevuto messaggi di spamming, perché mai dovremmo filtrarli? Questo atteggiamento non è valido per un semplice motivo: gli spammer inviano un primo messaggio “a tappeto”, quindi scoprono se il loro messaggio è stato visualizzato, ora sanno che un determinato indirizzo di posta elettronica è valido e quindi partono con le successive raffiche di messaggi.

A questo punto saremo ormai in trappola. Ma se filtriamo lo spamming fin da subito, anche il primo messaggio di spamming non otterrà mai una risposta e, di conseguenza, lo spammer cancellerà tale indirizzo (il nostro indirizzo) dall’elenco dei sistemi che può utilizzare.

Il problema è che gli spammer si presentano sempre sotto nuove forme. Cambiano strategie ogni volta che scoprono nuovi metodi, per i quali, naturalmente, verranno poi sviluppate nuove contromisure. Per questo è importante impiegare strumenti anti-spamming aggiornati.

Ecco alcune contromisure impiegate da SpamAssassin per proteggerci dal fastidioso spamming.

- **Relay aperti** Server di posta elettronica che consentono a chiunque di inviare messaggi. Come contromisura sono state sviluppate blacklist in grado di filtrare i messaggi spamming.
- **Filtraggio a parole chiave** Gli spammer ripetono spesso le stesse parole e frasi. SpamAssassin utilizza delle regole in grado di rilevare queste frasi.
- **Liste nere e liste bianche** Elencano, rispettivamente, le fonti note di spamming e di messaggi corretti.
- **Filtri bayesiani** Sistemi automatici calcolano le probabilità che un messaggio sia di spamming sulla base di ciò che il filtro stesso ha esaminato in precedenza.
- **Database** Molti server di posta elettronica inoltrano i messaggi ad alcuni server centrali. Se il sistema si accorge che lo stesso messaggio viene inviato a migliaia di destinatari, molto probabilmente si tratta di spamming.

Installare SpamAssassin

Anche in questo caso utilizzeremo il comodo Gestore di pacchetti Synaptic. Facciamo clic su Desktop > Amministrazione > Gestore pacchetti Synaptic e, nella finestra del programma, facciamo clic sull'icona Cerca nella barra degli

strumenti. Nella finestra di dialogo Trova scriviamo spamassassin e poi facciamo clic sul pulsante Cerca. Verranno visualizzati tutti i programmi che hanno a che fare con l'anti-spammer SpamAssassin.

Ora facciamo clic sulla casella quadrata che si trova a lato della voce spamassassin e selezioniamo l'opzione Marca per l'installazione. Anche in questo caso, come nei casi precedenti, verranno segnalati altri pacchetti che devono essere installati insieme al pacchetto principale. Ci dovremmo trovare nella situazione rappresentata nella Figura 6.8.

Per avviare l'installazione, possiamo fare clic sul pulsante Applica nella barra degli strumenti e confermare facendo clic sul pulsante Applica nella finestra Riepilogo. In breve, i file verranno scaricati da Internet e poi inizierà l'installazione del software.

Subito dopo l'installazione, SpamAssassin sarà già stato configurato per filtrare tutti i messaggi in arrivo sul sistema.

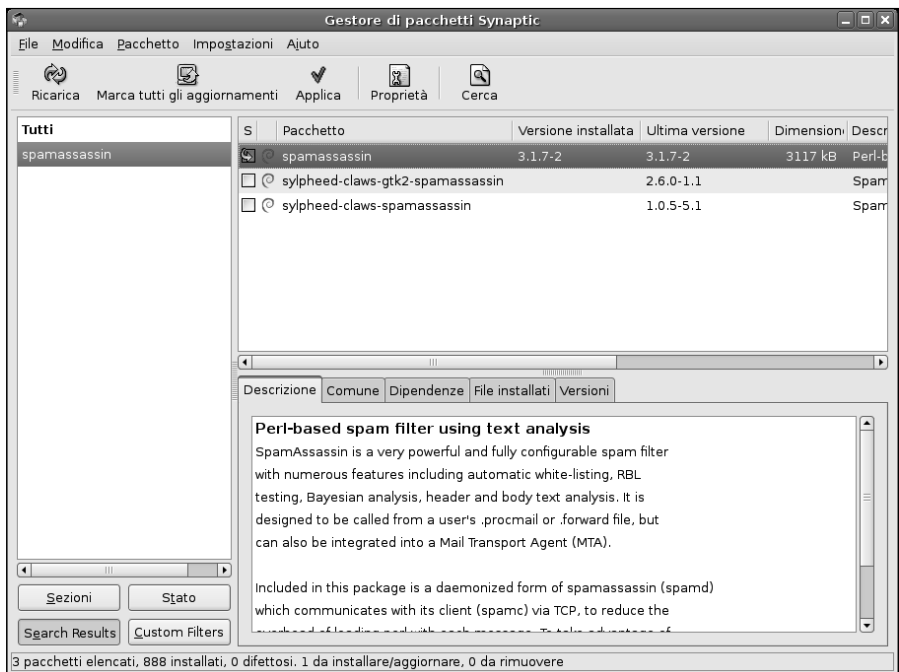


Figura 6.8

Installiamo l'anti-spammer SpamAssassin sul nostro sistema.

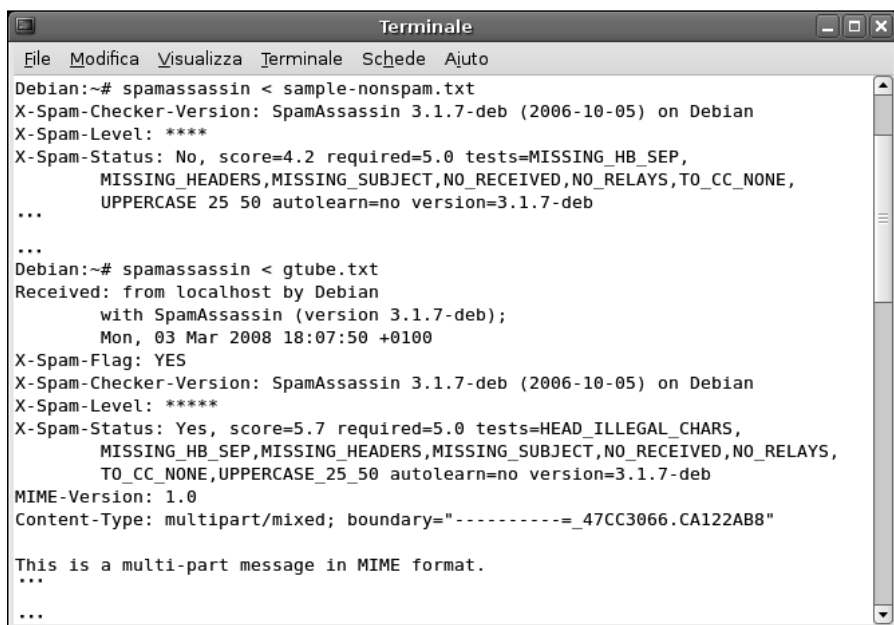
Collaudo dell'installazione

Possiamo anche svolgere dei test per garantire che SpamAssassin sia installato e che funzioni correttamente. Da Internet si possono scaricare e salvare due tipici file di esempio che dovrebbero essere riconosciuti da SpamAssassin, il cui nome è `sample-nospam.txt` e `sample-spam.txt`. Ovviamente il secondo è un esempio di spamming e il primo no. Basterà eseguire una ricerca di questi file con Google e scaricarli nella directory corrente, poi aprire una finestra Terminale (Applicazioni > Accessori > Terminale root) e infine “dare in pasto” i due file a SpamAssassin con i seguenti comandi:

```
spamassassin < sample-nospam.txt
```

```
spamassassin < sample-spam.txt
```

L'output del comando, depurato dal contenuto del messaggio, è rappresentato nella Figura 6.9. Come si può notare, il file `sample-nospam.txt` mostrerà l'indicazione X-Spam-Status: No e quello ottenuto utilizzando `sample-spam.txt` (in



```

Terminale
File Modifica Visualizza Terminale Schede Ajuto
Debian:~# spamassassin < sample-nospam.txt
X-Spam-Checker-Version: SpamAssassin 3.1.7-deb (2006-10-05) on Debian
X-Spam-Level: ****
X-Spam-Status: No, score=4.2 required=5.0 tests=MISSING_HB_SEP,
MISSING_HEADERS,MISSING_SUBJECT,NO_RECEIVED,NO_RELAYS,TO_CC_NONE,
...
UPPERCASE_25_50 autolearn=no version=3.1.7-deb
...
Debian:~# spamassassin < gtube.txt
Received: from localhost by Debian
with SpamAssassin (version 3.1.7-deb);
Mon, 03 Mar 2008 18:07:50 +0100
X-Spam-Flag: YES
X-Spam-Checker-Version: SpamAssassin 3.1.7-deb (2006-10-05) on Debian
X-Spam-Level: *****
X-Spam-Status: Yes, score=5.7 required=5.0 tests=HEAD_ILLEGAL_CHARS,
MISSING_HB_SEP,MISSING_HEADERS,MISSING_SUBJECT,NO_RECEIVED,NO_RELAYS,
TO_CC_NONE,UPPERCASE_25_50 autolearn=no version=3.1.7-deb
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="-----=_47CC3066.CA122AB8"

This is a multi-part message in MIME format.
...

```

Figura 6.9

L'output (parziale) dei due test di verifica del funzionamento di SpamAssassin.

questo caso il file si chiama GTUBE – Generic Test for Unsolicited Bulk Email) mostrerà le indicazioni X-Spam-Flag: YES e X-Spam-Status: Yes.

Configurazione dei client di posta elettronica

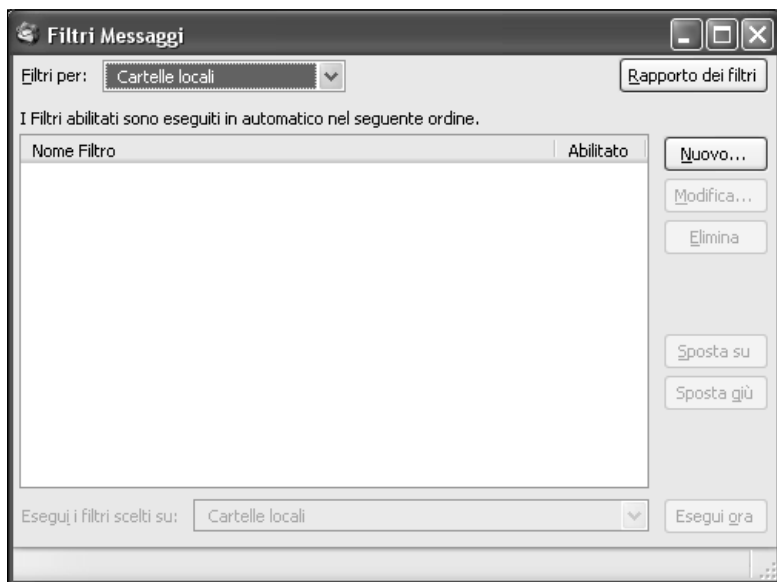
I messaggi individuati non verranno eliminati: semplicemente alla riga Oggetto (Subject) verrà aggiunta l'indicazione *****SPAM*****. Inoltre SpamAssassin inserisce nuovi campi X-Spam facilmente riconoscibili dai client predisposti. Basterà quindi istruire i client di posta elettronica in modo da eliminare o inserire in un'apposita cartella tutti i messaggi che hanno questa caratteristica.

Le operazioni da svolgere sono abbastanza semplici. In questo caso le figure rappresentano la procedura svolta con il client Mozilla Thunderbird su una macchina Windows, ma altri client impiegano procedure del tutto analoghe. Qualora non fosse possibile utilizzare il campo X-Spam, potremo comunque verificare che il campo Oggetto (Subject) contenga la stringa inserita da SpamAssassin.

1. Dobbiamo creare una cartella per i messaggi di spamming selezionando il comando **File | Nuovo | Cartella**. Nella finestra **Nuova cartella** indichiamo in **Nome** la cartella in cui dovrà trovarsi, per esempio **Posta in arrivo**. La cartella potrebbe chiamarsi semplicemente **Spam** (Figura 6.10). Al termine facciamo clic su **OK**.
2. Ora creiamo un filtro con il comando **Strumenti | Filtri**; nella finestra di dialogo **Filtri messaggi** (Figura 6.11) facciamo clic sul pulsante **Nuovo**.
3. Nella finestra di dialogo **Regole del filtro**, scegliamo innanzitutto il nome del filtro, per esempio **Spam** e quindi facciamo clic su **Soddisfano tutte le condizioni**. Nella prima casella digitiamo **X-Spam-Status**, nella casella centrale selezioniamo **“è”** e nella casella a destra digitiamo **Yes**. Nel riquadro



Figura 6.10
Creiamo la nuova cartella per i messaggi spamming.

**Figura 6.11**

Ora prepariamo un filtro apposito per i messaggi individuati come spamming da SpamAssassin.

sottostante, componiamo la frase *Sposta i messaggi in Spam su Cartelle locali*. (Figura 6.12).

4. Tornati nella finestra *Filtri messaggi*, facciamo clic sul pulsante Esegui ora in basso a destra (Figura 6.13) per provare il funzionamento della nostra nuova regola.

Conclusioni

In questo capitolo abbiamo imparato a proteggere il sistema e la casella di posta elettronica dai virus, utilizzando un potente software anti-virus come ClamAV. La seconda grande piaga dei messaggi di posta elettronica è rappresentata dallo spamming. Abbiamo imparato a debellarla grazie all'intervento di SpamAssassin.

Nel prossimo capitolo vedremo come usare un'altra modalità di accesso alla posta elettronica, sempre più comune: Webmail, ovvero la gestione della posta elettronica via Web, con SquirrelMail.

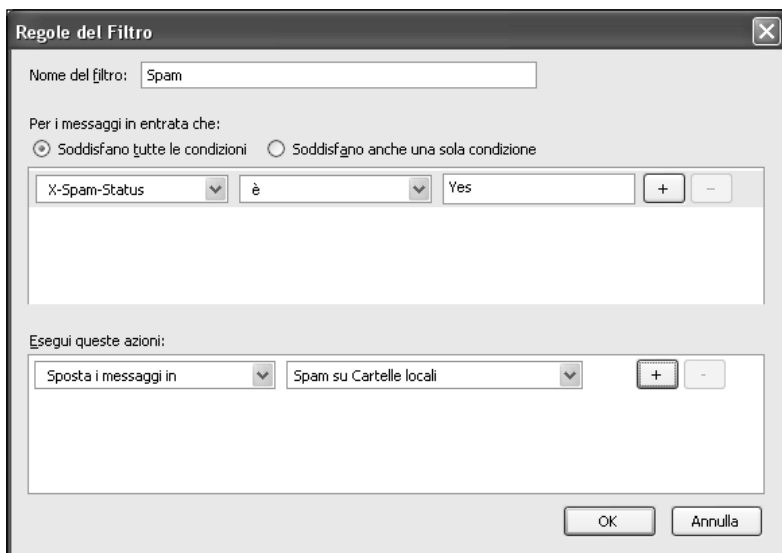


Figura 6.12

Il filtro è semplice e sfrutta la capacità di Mozilla Thunderbird di individuare i singoli campi del messaggio.

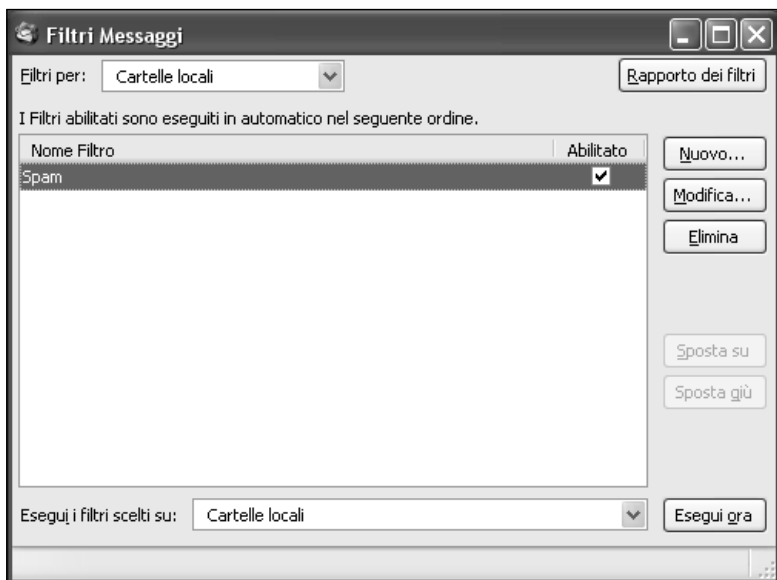


Figura 6.13

Possiamo finalmente eseguire la nuova regola sui messaggi già scaricati.

Capitolo 7

EMail + Web = SquirrelMail

L'accesso via Web alla posta elettronica è decisamente il metodo più utilizzato e semplice da usare.

Vediamo come aggiungerlo alla nostra macchina Debian.

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Webmail: che cos'è?
- ☑ SquirrelMail
- ☑ Installiamo SquirrelMail
- ☑ Plug-in per SquirrelMail

Abbiamo visto quali strumenti ci mette a disposizione Linux per gestire la posta elettronica su un server. In questo capitolo parleremo di una modalità di accesso sempre più comune alla casella di posta elettronica, ovvero tramite un'interfaccia Web, una vera e propria pagina Web. Il software che impiegheremo è il notissimo SquirrelMail, esaminandone la procedura di installazione e configurazione.

Webmail: che cos'è?

In una soluzione Webmail vi è un programma, in esecuzione sul server, che garantisce un accesso via Web alla nostra casella di posta elettronica, un po' come avviene nelle tipiche interfacce Webmail di Gmail (Google), di Hotmail (Microsoft) o del nostro provider Internet.

Una soluzione Webmail con SquirrelMail offre indubbi vantaggi ma anche qualche svantaggio, che è bene tenere in considerazione.

Vantaggi di SquirrelMail

Ecco i principali vantaggi di una soluzione Webmail con SquirrelMail.

- **Facilità di accesso** Le normali soluzioni di accesso alla posta elettronica presentano difficoltà di configurazione e manutenzione. È necessario installare e configurare adeguatamente il software client su tutti i computer locali. La cosa si complica se la rete è mista Linux/Windows/Mac/altro e se ognuno è libero di scegliere il client di posta che desidera (Outlook, Outlook Express, Thunderbird, Evolution e così via). La soluzione Webmail elimina questi problemi. Gli utenti potranno usare un comune browser (qualsiasi browser) e le operazioni di impostazione dovranno essere configurate in un solo luogo: nel server.
- **Accesso da posizioni remote** Abbiamo la nostra macchina e la usiamo per connetterci alla nostra rete. OK. Ma come possiamo accedere alla nostra posta mentre ci troviamo altrove, magari in un'altra sede dell'azienda, in vacanza, da un amico o a casa? Con una soluzione Webmail, potremo accedere alla posta elettronica da qualsiasi macchina connessa a Internet, da un Internet Cafè e perfino da macchine non "canoniche", come le console per videogame, che portano l'accesso a Internet sui teleschermi domestici. Senza naturalmente dimenticare le connessioni da Hotspot WiFi, sempre più comuni nei locali e negli ambienti pubblici.
- **Niente più client** Si elimina d'un tratto il problema dell'aggiornamento dei client di posta elettronica. La soluzione Webmail viene gestita e amministrata a livello centrale, sul server di posta elettronica. Tutto ciò che è necessario è un browser, di qualsiasi tipo.
- **Semplificazione dell'interfaccia utente** I client di posta elettronica potrebbero non supportare alcune attività fondamentali. Per esempio possono esservi problemi nell'impostazione della password di accesso o nel filtraggio dei messaggi. SquirrelMail offre vari plug-in che consentono, per esempio di modificare la password dall'interfaccia Webmail.
- **Vantaggi in termini di sicurezza** Tradizionalmente, scarichiamo la posta elettronica sul nostro computer locale. Ma se non facciamo attenzione, il nostro computer potrebbe cadere preda di un hacker "cattivo" che potrebbe utilizzarlo come testa di ponte per accedere al server centrale. Con una soluzione Webmail, il fattore sicurezza viene gestito centralmente.

Svantaggi di SquirrelMail

Una soluzione Webmail per la posta elettronica introduce però anche alcuni potenziali svantaggi, elencati di seguito.

- **Fattori prestazionali** Un client tradizionale di posta elettronica offre in genere funzionalità utili per migliorare la produttività personale, come l'ordinamento e la ricerca dei messaggi, la gestione dell'elenco dei contatti e degli allegati, il filtraggio dello spamming. Queste attività dovranno essere svolte dal server centrale e le relative informazioni devono passare da Internet. Ciò rappresenta un carico per il server di posta elettronica e introduce una latenza più o meno accentuata nei colloqui fra la macchina locale e il server.
- **Grandi quantitativi di messaggi** In genere la visibilità dei messaggi in una soluzione Webmail è piuttosto limitata. Caricare una casella di grandi dimensioni può richiedere del tempo e poi dobbiamo sempre passare attraverso una gestione a pagine Web, normalmente più macchinosa.
- **Problematiche nella gestione degli allegati** Se i messaggi risiedono sul server, il loro download sulla macchina locale non può essere immediato. In genere vi sono problemi a scaricare dal server di posta elettronica allegati di grandi dimensioni.
- **Sicurezza** Un accesso Webmail introduce potenziali problemi di sicurezza, tuttavia non dissimili da quelli presentati dai comuni client Web utilizzati da una postazione remota.

SquirrelMail

SquirrelMail è una piattaforma Webmail ben nota, stabile e matura, molto diffusa e basata su standard; inoltre genera pagine in puro codice HTML 4.0, leggibile da ogni tipo di browser. Fra le funzionalità offerte da SquirrelMail vi è un ottimo supporto dei tipi MIME per gli allegati, una gestione della rubrica dei contatti, un sistema di controllo ortografico dei messaggi, la possibilità di inviare e ricevere messaggi in formato HTML, il supporto dei temi e delle skin e il supporto degli host virtuali. La Figura 7.1 mostra l'aspetto della pagina di login di SquirrelMail.

Installiamo SquirrelMail

Anche in questo caso, per installare SquirrelMail sul nostro sistema Debian utilizzeremo il comodo Gestore di pacchetti Synaptic, che semplifica la risoluzione delle dipendenze da altri pacchetti.

Richiamiamo Desktop > Amministrazione > Gestore pacchetti Synaptic dal desktop Gnome di Debian e, nella finestra del programma, facciamo clic sull'icona Cerca nella barra degli strumenti. Nella finestra di dialogo Trova scriviamo

**Figura 7.1**

Possiamo connetterci a SquirrelMail utilizzando un comune browser Web.

squirrelmail e poi facciamo clic sul pulsante Cerca. Verranno visualizzati tutti i programmi che hanno a che fare con il pacchetto Webmail SquirrelMail.

Ora facciamo clic sulla casella quadrata che si trova a lato della voce squirrelmail e selezioniamo l'opzione Marca per l'installazione. Come di consueto, verranno segnalati tutti i pacchetti che devono essere installati insieme a quello principale. In particolare deve essere installato PHP per Apache. A questo punto la situazione dovrebbe essere quella rappresentata nella Figura 7.2.

Ora possiamo avviare l'installazione, facendo clic sul pulsante Applica nella barra degli strumenti; confermiamo facendo clic sul pulsante Applica nella finestra Riepilogo. In breve, i file verranno scaricati da Internet e inizierà l'installazione del software (Figura 7.3).

Ora SquirrelMail è stato correttamente installato.

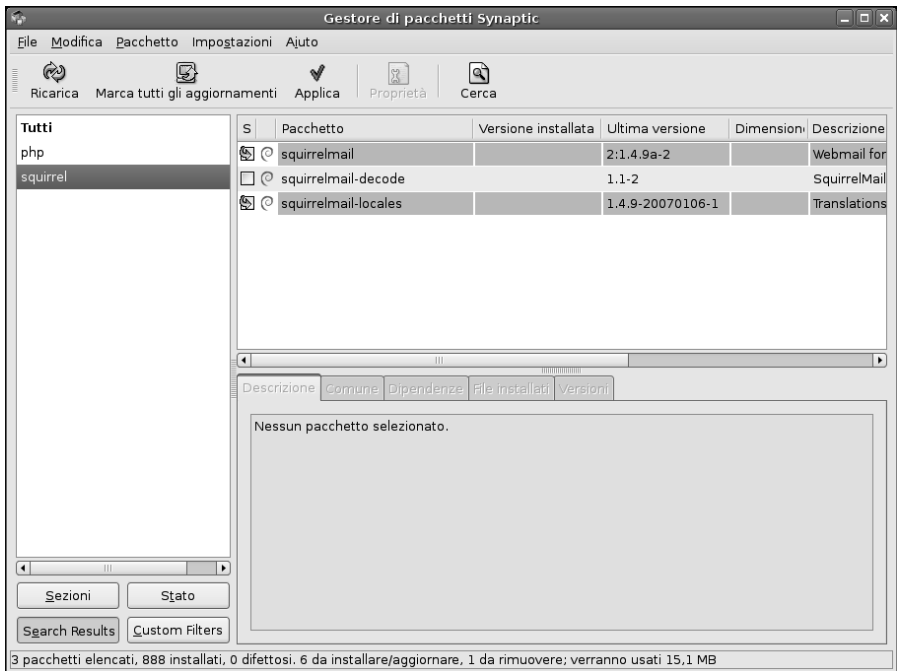


Figura 7.2

Installiamo SquirrelMail e dei pacchetti da cui dipende, fra i quali PHP per il server Web Apache.

Configurazione di SquirrelMail

La configurazione di SquirrelMail è contenuta nel file `/etc/squirrelmail/config.php` (Figura 7.4). Invece di intervenire direttamente su tale file possiamo però utilizzare il comodo script `Perl conf.pl` contenuto sempre nella stessa directory.

Facendo doppio clic su questo file nel desktop di Debian, viene presentata una finestra che recita "Eseguire <<conf.pl>> o mostrare il suo contenuto?". Facendo clic sul pulsante Esegui nel terminale si aprirà una finestra del Terminale che mostrerà il menu rappresentato nella Figura 7.5.

Il funzionamento del menu è intuitivo: basta specificare il numero o la lettera appropriati e premere INVIO. Per impostare le preferenze relative all'azienda, digitiamo 1 e premiamo INVIO.

In questo menu possiamo modificare i campi `Organization Name`, `Organization Logo` e `Organization Title`. Dopo averli modificati (Figura 7.6), possiamo premere R per tornare al menu principale.

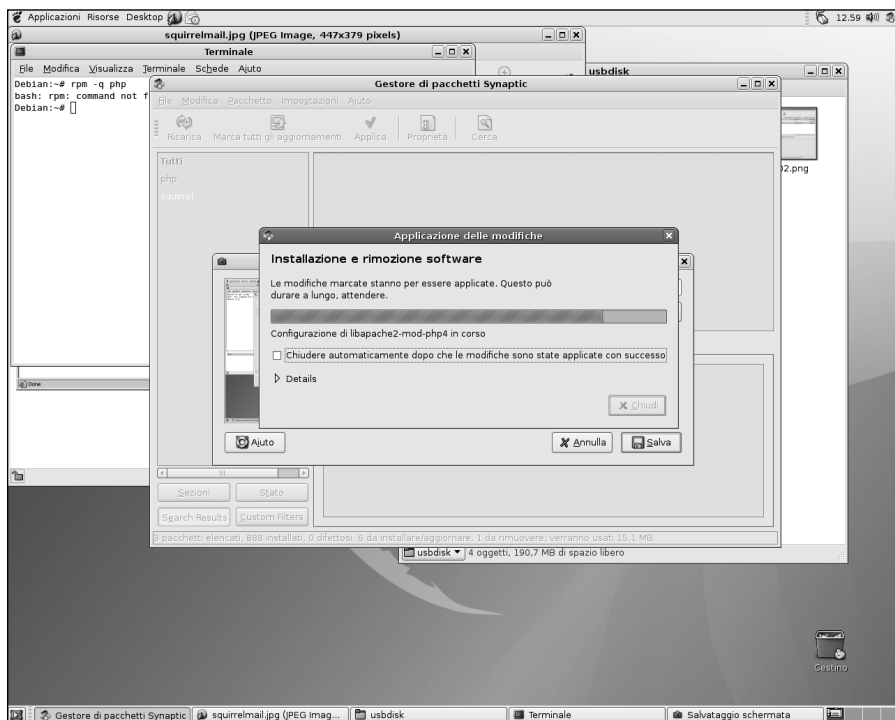


Figura 7.3
Installazione di SquirrelMail dal Gestore di pacchetti Synaptic.

Ora possiamo passare alle impostazioni del server digitando **2** e premendo INVIO. Compariranno le opzioni visibili nella Figura 7.7. Nel campo **Domain** dobbiamo specificare il valore corretto. Inoltre dobbiamo scegliere l'opzione **3** per selezionare **Sendmail**, il server di posta che abbiamo installato nel Capitolo 5. Premiamo **R** e poi INVIO per tornare al menu principale.

Con l'opzione **4** richiamiamo le opzioni generali di SquirrelMail. Qui troviamo le directory **Data** **Directory** **Attachment** **Directory** che conterranno rispettivamente i dati e gli allegati (Figura 7.8).

Per tornare al menu principale possiamo digitare **R** seguito da INVIO.

Per salvare le modifiche scegliamo **S** e premiamo INVIO; premiamo ancora INVIO per proseguire e infine scegliamo **Q** per uscire dalla configurazione.

Termina così la configurazione delle impostazioni di SquirrelMail. Potremo riutilizzare questo script per modificare la configurazione, per scegliere un nuovo tema o per attivare un plug-in.


```

conf.pl (/etc/squirrelmail) - gedit
File Modifica Visualizza Cerca Strumenti Documenti Ajuto
Nuovo Apri Salva Stampa Annulla Ripeti Taglia Copia Incolla Trova Sostituisce

conf.pl x
#!/usr/bin/env perl
# conf.pl
#
# Copyright (c) 1999-2006 The SquirrelMail Project Team
# Licensed under the GNU GPL. For full terms see COPYING.
#
# A simple configure script to configure SquirrelMail
#
# $Id: conf.pl,v 1.154.2.35 2006/10/07 11:58:42 tokul Exp $
#####
$conf_pl_version = "1.4.0";

#####
# Check what directory we're supposed to be running in, and
# change there if necessary. File::Basename has been in
# Perl since at least 5.003_7, and nobody sane runs anything
# before that, but just in case.
#####
my $dir;
if ( eval q{require "File/Basename.pm"} ) {
    $dir = File::Basename::dirname($0);
    chdir($dir);
}

#####
Rg 1, Col 1 INS

```

Figura 7.4

L'aspetto del file config.php, che contiene la configurazione di SquirrelMail.

```

Terminale
File Modifica Visualizza Terminale Schede Ajuto

SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Main Menu --
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color on
S Save data
Q Quit

Command >>

```

Figura 7.5

Il menu principale di configurazione di SquirrelMail.

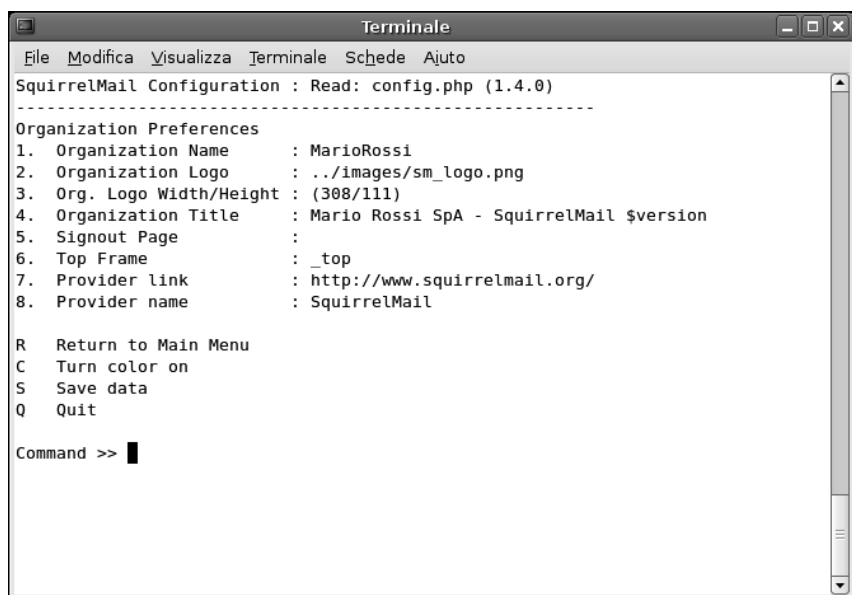


Figura 7.6

Configurazione delle preferenze relative alla nostra organizzazione/azienda/ente.

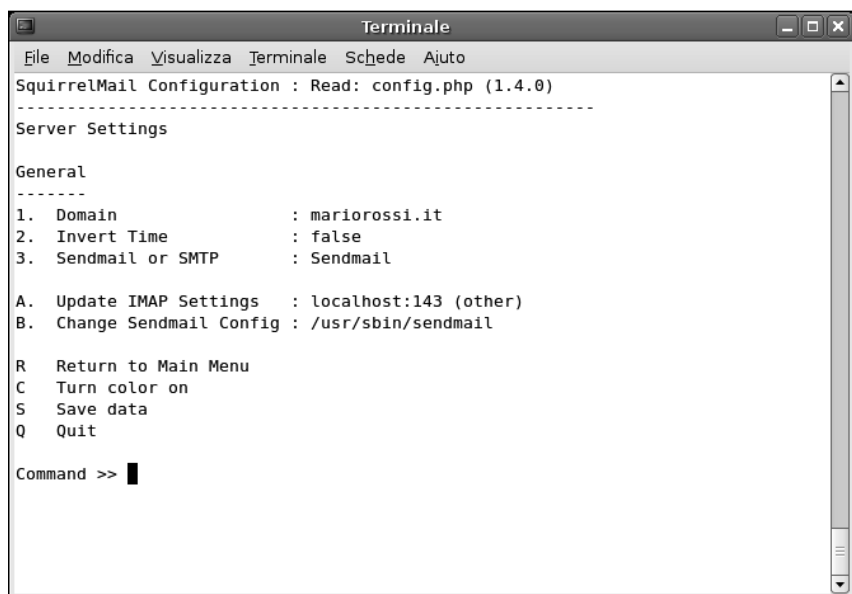


Figura 7.7

Le impostazioni del server per SquirrelMail.


```

Terminale
File Modifica Visualizza Terminale Schede Ajuto
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
General Options
1. Data Directory           : /var/lib/squirrelmail/data/
2. Attachment Directory    : /var/spool/squirrelmail/attach/
3. Directory Hash Level    : 0
4. Default Left Size       : 150
5. Usernames in Lowercase  : false
6. Allow use of priority    : true
7. Hide SM attributions    : false
8. Allow use of receipts   : true
9. Allow editing of identity : true
   Allow editing of name   : true
   Remove username from header : false
10. Allow server thread sort : false
11. Allow server-side sorting : false
12. Allow server charset search : true
13. Enable UID support      : true
14. PHP session name       : SQMSESSID
15. Location base          :

R  Return to Main Menu
C  Turn color on
S  Save data
Q  Quit

Command >> █

```

Figura 7.8

Le directory di lavoro utilizzate da SquirrelMail.

Plug-in per SquirrelMail

Tramite appositi plug-in possiamo estendere le funzionalità di SquirrelMail e infatti, dal sito Web di SquirrelMail possiamo scaricare oltre 200 plug-in, all'indirizzo <http://www.squirrelmail.org/plugins.php>.

I più importanti sono installabili direttamente dal menu di `conf.pl`. In particolare, l'opzione 8 del menu principale consente proprio di configurare e attivare i plug-in. Dobbiamo quindi lanciare il file `/etc/squirrelmail/conf.pl` come descritto in precedenza e scegliere l'opzione 8. Dovremmo ottenere il risultato rappresentato nella Figura 7.9. L'elenco `Available Plugins` presenta i plugin disponibili e non installati, mentre l'elenco `Installed Plugins` presenta tutti i plug-in che sono stati installati e attivati.

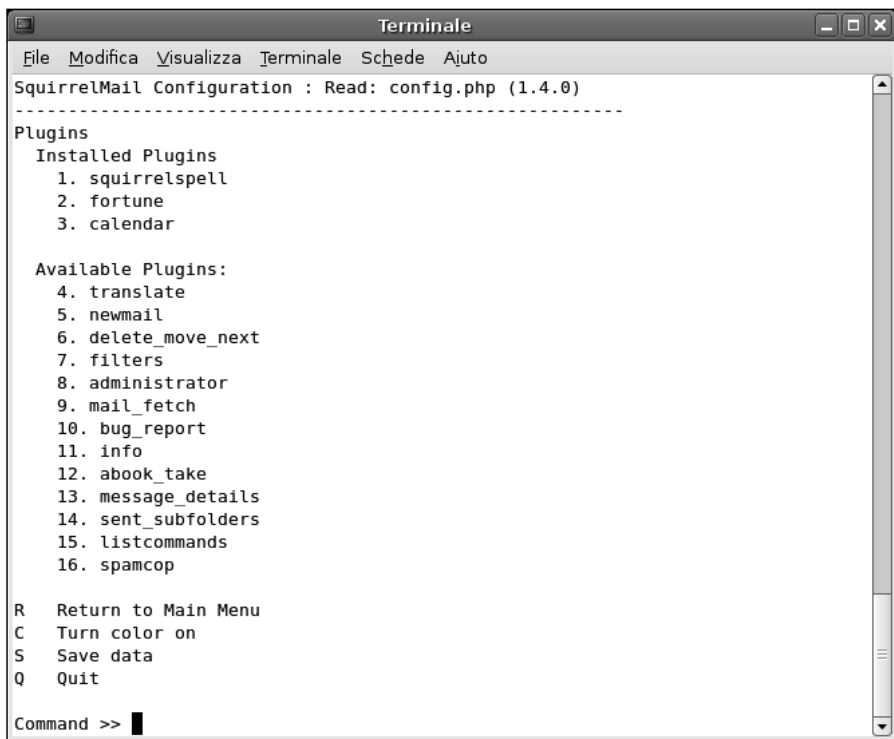


Figura 7.9

Il menu di installazione dei plug-in di SquirrelMail dopo aver installato tre plugin.

Conclusioni

In questo capitolo abbiamo presentato sommariamente il software Webmail SquirrelMail, presentando vantaggi e svantaggi di una soluzione di questo tipo. Dopo averne esaminato le procedure di installazione e configurazione, abbiamo introdotto anche la possibilità di installare i numerosi plug-in disponibili per questo software.

Nel prossimo capitolo proveremo ad affrontare un altro tipico utilizzo di una macchina Linux: come server FTP (File Transfer Protocol).

Capitolo 8

Un server FTP in Debian

Impariamo a creare un server FTP su una macchina Linux per trasferire i file più ingombranti

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Cos'è e a cosa serve il protocollo FTP
- ☑ FTP e i rischi
- ☑ Server FTP su Debian
- ☑ Configurazione di ProFTPD
- ☑ La directory FTP
- ☑ Limitare gli accessi anonimi
- ☑ Accesso al server FTP
- ☑ Un accesso in scrittura

Cos'è e a cosa serve il protocollo FTP

FTP (File Transfer Protocol) è un protocollo Internet normalmente utilizzato per trasferire dati da un luogo, una rete o un computer a un altro. Possiamo averne bisogno per consentire ai nostri collaboratori di scaricare e inviarci determinati dati, oppure per mettere in condivisione del materiale pubblicitario con i nostri agenti commerciali o con il mondo intero.

In questo modo il sito FTP aziendale diviene una sorta di grande contenitore di file al quale può accedere chiunque oppure solo determinati utenti selezionati, sia all'interno sia all'esterno dell'azienda.

Potremmo pensare che, in fin dei conti, ci scambiamo tutti i giorni file sfruttando gli allegati di posta elettronica. Ma come possiamo fare per scambiare

un file da 25 MB? O magari una raccolta di due o trecento MB di file? La posta elettronica non è in grado di far fronte a queste esigenze.

Il protocollo impiegato per i trasferimenti FTP fa sì che il trasferimento proceda tranquillamente per file di (quasi) qualsiasi dimensione. Sarà solo una questione di tempo e i file verranno trasferiti da una macchina all'altra. Nel frattempo possiamo naturalmente procedere con il nostro lavoro o i nostri passatempi: il trasferimento FTP in background non occupa particolarmente le risorse della macchina. Naturalmente cercherà di sfruttare al massimo la banda disponibile nella connessione a Internet e dunque, nel corso del trasferimento FTP, ogni attività nel Web risulterà rallentata.

FTP e i rischi

Bisogna anche considerare che un utilizzo troppo “aperto” di FTP espone ad alcuni rischi da non sottovalutare.

- In passato alcune implementazioni di FTP hanno sofferto di gravi punti deboli, che offrivano un accesso **root** alla macchina.
- Se lasciamo libero accesso alla nostra macchina, qualcuno potrebbe utilizzare lo spazio disponibile per riempirlo di “porcherie” di ogni tipo. Poco male, i file si cancellano, ma se tali file fossero di natura illegale, pesantemente illegale o addirittura penale? Cosa penserà la Polizia Postale nello scoprire che è in corso un'intensa attività di up/download di materiale pedopornografico da e verso la nostra macchina? “Diamine, davvero? Non ne sapevo nulla! Ma sul mio FTP?” Crederà alla nostra sincerità? Meglio non correre rischi.

Server FTP su Debian

I sistemi Unix hanno sempre in funzione un daemon FTP, ma sulla nostra macchina Debian, nata come un sistema desktop, non c'è ancora nulla. Nel corso di questo capitolo esamineremo l'installazione e l'uso del server FTP ProFTPD. Si tratta di un daemon molto noto e di largo utilizzo, che ha sostituito l'ormai desueto wu-ftp (Washington University FTP Daemon).

Installare ProFTPD

Volendo procedere il più possibile in modo grafico, sfruttando il desktop Gnome di Debian, installeremo ProFTPD utilizzando il Gestore di pacchetti Synaptic, che cura anche eventuali conflitti e dipendenze da altri pacchetti. Richiamiamo Desktop > Amministrazione > Gestore pacchetti Synaptic dal desktop Gnome di Debian e, nella finestra del programma, facciamo clic sull'icona

Cerca nella barra degli strumenti. Nella finestra di dialogo Trova scriviamo ftp o, meglio, proftpd e poi facciamo clic sul pulsante Cerca. Verranno visualizzati tutti i programmi che riguardano il pacchetto ProFTPD (Figura 8.1).

Ora facciamo clic sulla casella quadrata che si trova a lato della voce proftpd e selezioniamo l'opzione Marca per l'installazione. Come di consueto, verranno segnalati alcuni pacchetti che devono essere installati obbligatoriamente insieme al pacchetto principale.

Ora possiamo avviare l'installazione, facendo clic sul pulsante Applica nella barra degli strumenti; confermiamo facendo clic nuovamente su Applica nella finestra Riepilogo. In breve, i (pochi) file verranno scaricati da Internet e verrà eseguita l'installazione del software.

Configurazione di ProFTPD

Subito dopo l'installazione verrà lanciata automaticamente la procedura di configurazione del server FTP. La prima domanda che viene posta riguarda il

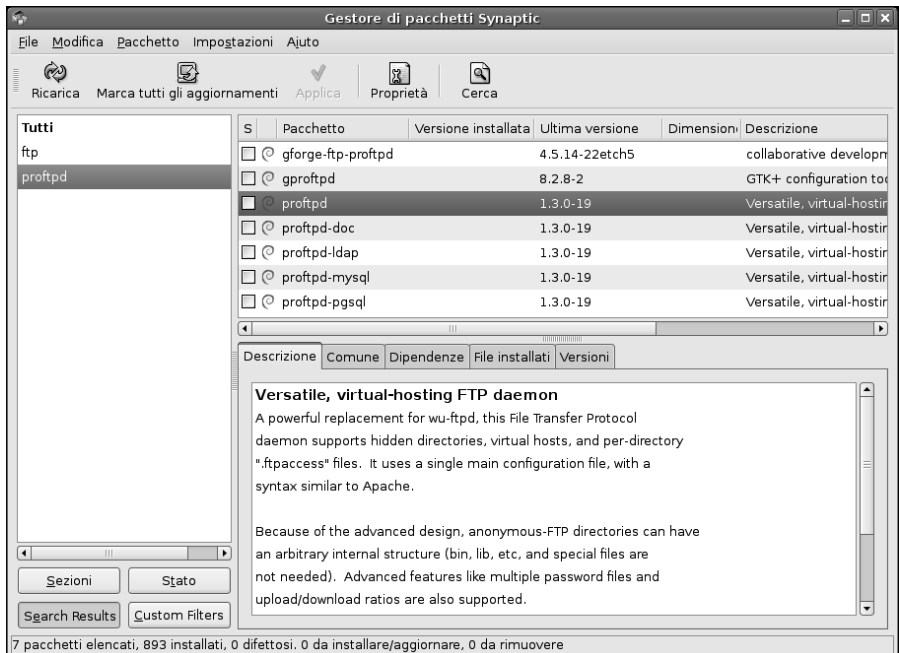


Figura 8.1

I pacchetti Linux che riguardano il server FTP ProFTPD.

fatto che il server debba essere eseguito da `inetd` (InterNET services Daemon) o in modo autonomo (Figura 8.2).

- Nel primo caso, il daemon FTP verrà lanciato automaticamente dal super-daemon `inetd` ogni volta che al sistema arriva una richiesta di connessione FTP (21).
- Nel secondo caso si potrà lasciare il daemon FTP in esecuzione costante sul sistema, in attesa di esaudire una richiesta.

Il tipo di scelta dipende quindi da quali sono le nostre esigenze.

- Se il sito si troverà ad accettare solo poche connessioni FTP al giorno, è meglio scegliere l'opzione `inetd`, che richiamerà automaticamente il server solo in caso di necessità; in questo modo non sprecheremo risorse preziose lasciando il server ProFTPD costantemente in funzione sul sistema in attesa di eventi tutto sommato molto rari.
- Se invece le sessioni FTP sono più intense e frequenti, è meglio scegliere l'opzione `autonomo`, che fa sì che ogni richiesta FTP ottenga una risposta

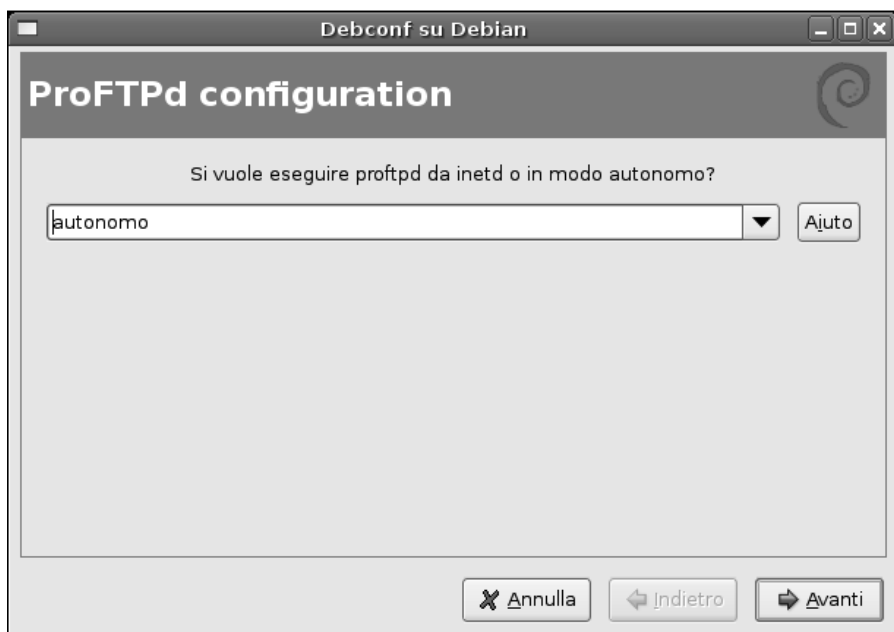


Figura 8.2

Viene lanciata automaticamente la procedura di configurazione di ProFTPD.

immediata; scegliendo l'opzione `inetd` si causerebbe uno spreco di risorse, poiché per ogni richiesta verrebbe creato un nuovo processo.

Sulla base di queste indicazioni, scegliamo l'opzione più appropriata e facciamo clic sul pulsante Avanti. In questo caso abbiamo scelto l'opzione `autonomo` perché la macchina è, tutto sommato, ben dotata di risorse e non dovrebbe soffrire troppo della presenza del server FTP.

Verranno installati alcuni pacchetti aggiuntivi specifici per il tipo di installazione che abbiamo scelto e, al termine, verrà avviato il server.

Possiamo verificare il funzionamento dei suoi servizi selezionando dal desktop di Gnome il comando Desktop > Amministrazione > Servizi; nella finestra di dialogo Impostazioni servizi (Figura 8.3), troveremo elencato il nuovo Ser-



Figura 8.3
Il nostro nuovo server FTP è attivo e funzionante.

vizio FTP (proftpd) che, come possiamo notare dalla casella a sinistra (che contiene un segno di spunta) e già attivo e operativo sul sistema. Terminata la fase di configurazione, torneremo al desktop di Debian dove possiamo chiudere la finestra del Gestore di pacchetti Synaptic.

I file di configurazione del server FTP

A questo punto, il nostro server FTP è predisposto per la connessione degli utenti noti al sistema. Dunque potremmo collegarci da qualsiasi altra macchina della rete locale o di Internet e accedere al server FTP utilizzando semplicemente il nostro nome utente e la nostra password. Con una connessione di questo tipo, ci ritroveremo direttamente nella nostra directory home, dove avremo accesso a tutti i file del nostro nome-utente.

Per esempio, la Figura 8.4 mostra una connessione eseguita direttamente da un browser digitando nella casella Indirizzo il seguente indirizzo specifico per la connessione FTP:

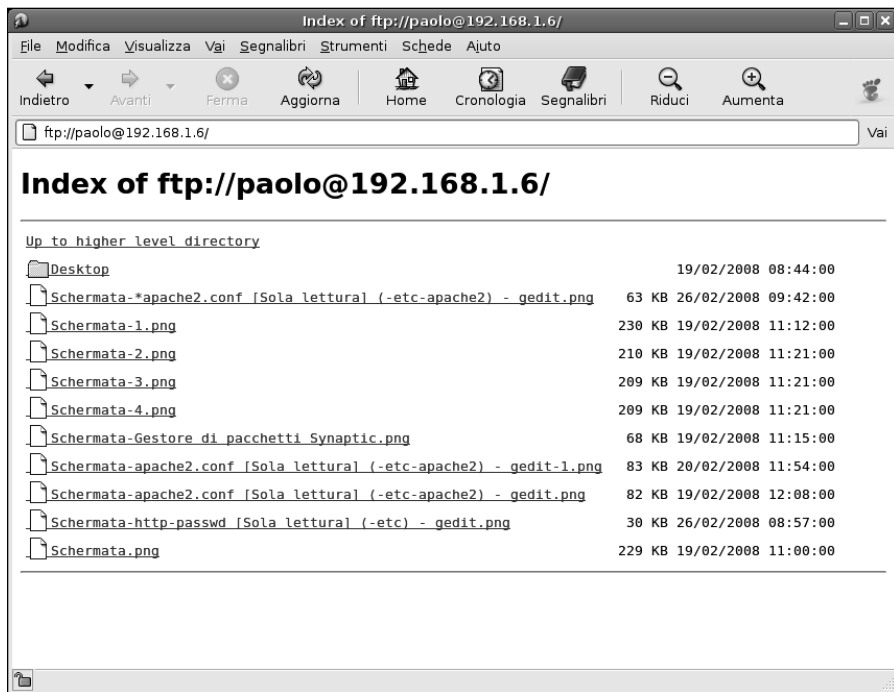


Figura 8.4

Ci siamo collegati da una macchina Windows della rete locale alla directory del nostro nome-utente sul server Linux. Niente di più facile.

`ftp://nome-utente:password@indirizzo-IP`

Il browser ci mostrerà il contenuto della nostra cartella home presentato con la consueta formattazione dei siti FTP.

La directory del sistema in cui si trovano i file di configurazione di ProFTPD è `/etc/proftpd` che contiene due soli file di configurazione, ovvero `modules.conf` e `proftpd.conf` (Figura 8.5).

Il file `modules.conf` si occupa del caricamento dei moduli esterni del programma, mentre il file `proftpd.conf` è il vero e proprio file di configurazione del programma.

Si tratta di un comune file di testo che normalmente non deve essere troppo modificato, in quanto è già impostato per funzionare in modo ottimale sul sistema. L'unica parte del file che richiede una vera e propria modifica è costituita dalla sezione `<Anonymous>` che definisce le possibilità di connessione per gli utenti anonimi. Si tratta di tutti gli utenti che non hanno un account sul sistema e che potranno utilizzare queste impostazioni anonime per accedere a una particolare directory FTP della macchina.

Inizialmente tutta questa sezione è protetta, in quanto trasformata in un commento: davanti a tutte le righe della sezione `<Anonymous>` si trova infatti il sim-

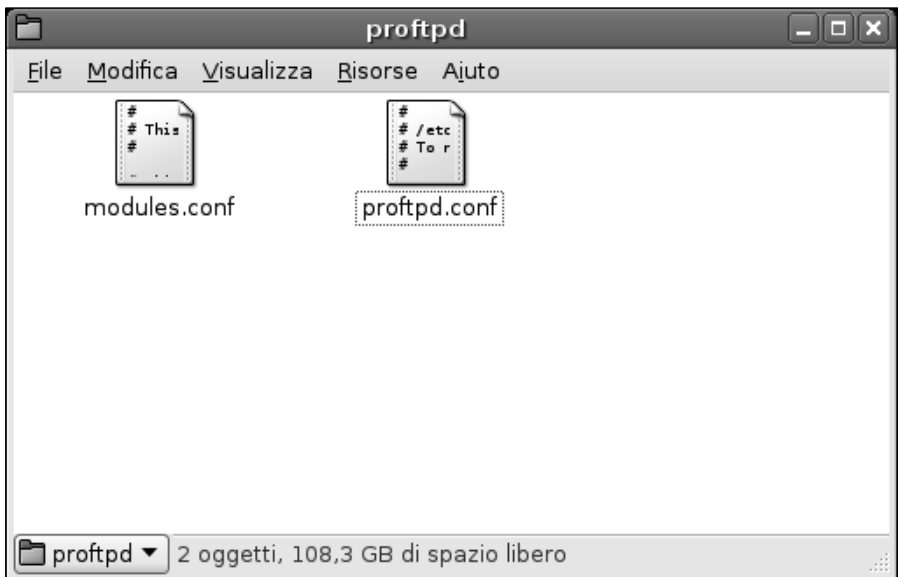


Figura 8.5

La directory contenenti file di configurazione di ProFTPD.

bolo "#". In altre parole, subito dopo l'installazione, ProFTPD consentirà solo l'accesso da parte degli utenti dotati di un account sulla macchina.

Togliendo tutti questi simboli "#" iniziali, si abilitano gli accessi anonimi al sistema. Per esempio, la Figura 8.6 mostra l'aspetto di questa sezione del file `proftpd.conf` dopo aver rimosso i caratteri di commento "#".

Come possiamo vedere osservando la sezione `<Anonymous>` rappresentata nella Figura 8.6, tutti gli accessi anonimi faranno riferimento all'utente `ftp`. Si tratta di un vero e proprio utente del computer locale, dotato di una propria directory. È proprio nella sua directory che si ritroveranno tutti coloro che chiederanno di ottenere un accesso anonimo alla macchina.

La directory FTP

Vediamo allora che cosa troviamo all'interno della directory predisposta per i trasferimenti FTP. La troviamo, insieme alle directory home di tutti gli altri utenti del sistema, scendendo lungo il percorso `/home` (Figura 8.7).

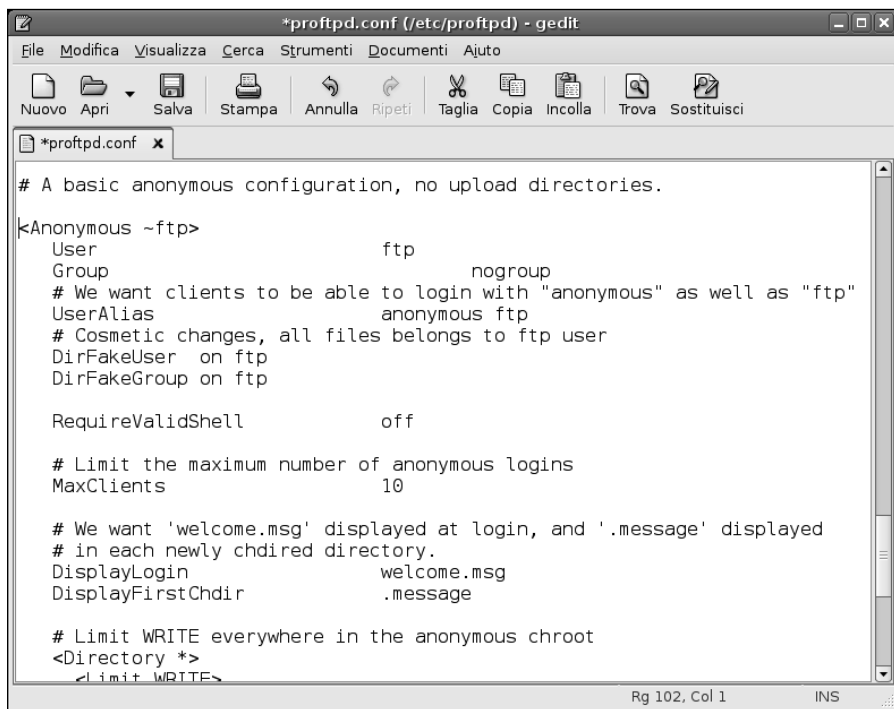


Figura 8.6

Togliendo il simbolo di commento abbiamo abilitato gli accessi anonimi al sito FTP della nostra macchina.



Figura 8.7

L'utente ftp è come un normale utente e ha la propria directory home.

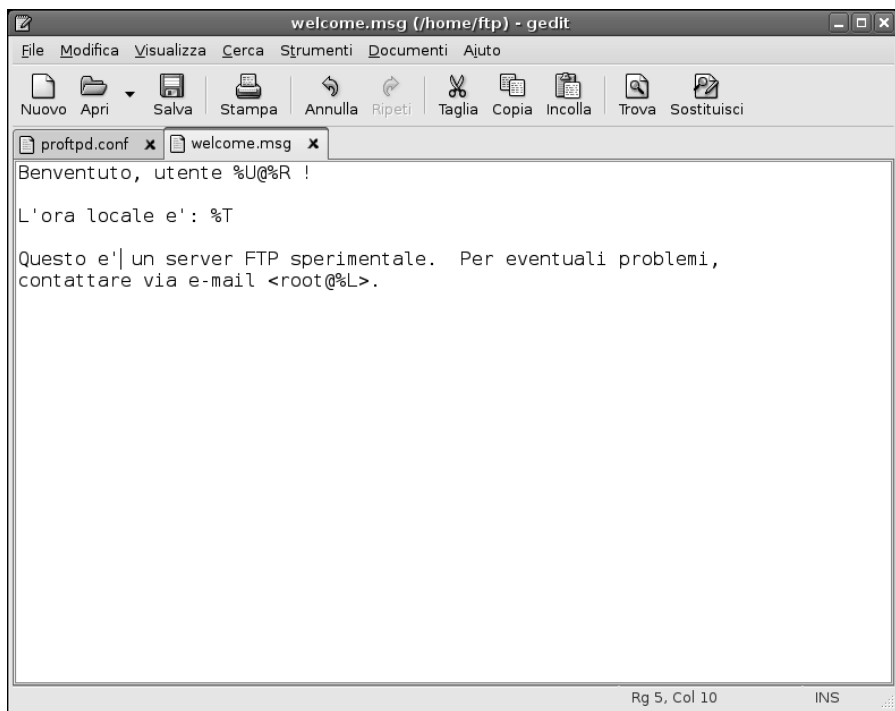
All'interno della directory home dell'utente ftp troviamo un unico file, ovvero `welcome.msg`, al quale fa riferimento anche file di configurazione `proftpd.conf`. Questo è il file che verrà presentato a ogni utente anonimo che si connette al sito FTP che stiamo predisponendo. La Figura 8.8 mostra l'aspetto del file, che abbiamo provveduto a localizzare in italiano.

All'interno di questa directory possiamo inserire tutti i file che intendiamo lasciare scaricare ai nostri utenti FTP anonimi.

Limitare gli accessi anonimi

Attualmente avranno accesso a nostro server FTP tutti coloro che lo desiderano e che conoscono il nostro indirizzo. Come possiamo proteggere un po' di più gli accessi, in modo da concederli unitamente a persone fidate?

Come abbiamo visto, tutti coloro che si collegano in forma anonima (ovvero come utenti `anonymous`, utilizzeranno in realtà l'utente Linux `ftp`. Questo è un comune utente del sistema, che ritroviamo nella finestra **Utenti e gruppi** (Figura 8.9) che possiamo richiamare col comando **Desktop > Amministrazione > Utenti e gruppi**.

**Figura 8.8**

Il messaggio che verrà visualizzato agli utenti anonimi che chiedono la connessione a questo sito FTP.

Nella finestra di dialogo **Utenti e gruppi** possiamo fare clic sul pulsante **Proprietà** per visualizzare la pagina **Account** della finestra di dialogo **Impostazioni** per l'utente **ftp** (Figura 8.10). Qui possiamo specificare nella sezione **Password** la password che vogliamo attribuire a questo utente. Dopo averla impostata, avranno accesso anonimo unicamente gli utenti che la conoscono, presumibilmente solo persone fidate.

Accesso al server FTP

Ora l'accesso è più protetto. Vediamo come impostare un accesso al server FTP da un'altra macchina che, in questo caso, appartiene alla stessa rete locale, ma potrebbe benissimo trovarsi all'altro capo del mondo ed essere connessa via Internet.



Figura 8.9
Individuiamo l'utente ftp del sistema.

Come abbiamo già visto in precedenza, per connettersi utilizzando un browser, possiamo utilizzare la seguente forma:

ftp://nome-utente:password@indirizzo-IP

Come si può vedere nella Figura 8.11, l'aspetto del server FTP da remoto e con una connessione anonima, non differisce troppo da quello che si ha nel caso di una connessione locale con l'account di un utente del sistema.

Se vogliamo utilizzare una visualizzazione più comoda e aprire la directory remota del server FTP in una finestra Windows del tutto analoga a quella che utilizziamo per esplorare i file, possiamo utilizzare il comando **Pagina > Apri sito FTP in Esplora risorse** di Internet Explorer: ritroveremo il contenuto della directory FTP della macchina Debian in una comune finestra di Windows! Possiamo apprezzare il risultato nella Figura 8.12.

Per provare una connessione con un client FTP, utilizzeremo invece il comodo software per Windows **FTP Commander**, che può essere comodamente scaricato all'indirizzo <http://www.internet-soft.com>.

In ogni caso si tratta di un esempio: altri client offrono procedure certamente differenti ma sostanzialmente analoghe.

Impostazioni per l'utente ftp

Account **Avanzato** Privilegi utente

Impostazioni di base

Nome ute:

Nome reale:

Informazioni sul contatto

Indirizzo ufficio:

Telefono di lavoro:

Telefono di casa:

Password

☒ Imposta password a mano

Password utente:

Conferma:

☐ Genera password casuale

Password impostata a:

Figura 8.10

Specifichiamo una password per l'utente anonimo ftp in modo da proteggere gli accessi.

Dobbiamo innanzitutto creare le opzioni per la connessione con il server, facendo clic sul pulsante **Nuovo server**.

Nella finestra di dialogo delle impostazioni (Figura 8.13), dovremo indicare un nome a piacere con il quale identificheremo il nostro server Debian remoto e l'indirizzo con il quale è raggiungibile.

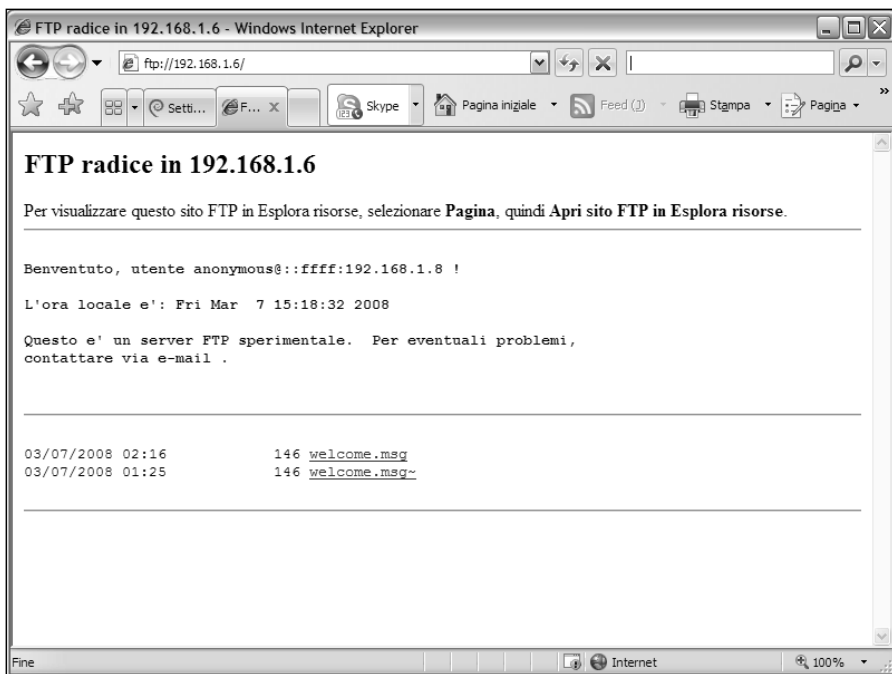


Figura 8.11

Internet Explorer, su una macchina Windows, si connette al server FTP della macchina Debian.

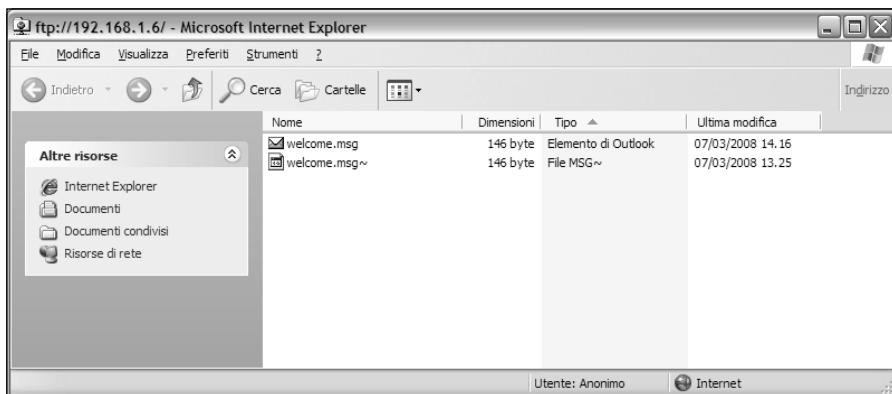


Figura 8.12

La directory FTP Debian non differisce affatto da una comune cartella Windows.

Sul lato destro della finestra di dialogo dobbiamo specificare le credenziali per la connessione, ovvero l'utente `anonymous` e la password che abbiamo precedentemente predisposto sul sistema Debian. Questo è tutto: tutte le altre opzioni sono già impostate correttamente.

Tornati alla finestra principale di **FTP Commander**, ci basterà fare doppio clic sulla voce che abbiamo appena creato e il programma attiverà la connessione con il nostro server FTP Debian. Nella Figura 8.14 possiamo notare sulla destra il contenuto della directory FTP e, nella parte inferiore, gli scambi di informazioni che sono intercorsi fra la macchina locale e il server FTP per stabilire la connessione.

Un accesso in scrittura

Se proviamo a utilizzare il server FTP, ci accorgeremo che possiamo scaricare tutti i file contenuti nella directory predisposta, ma non possiamo inviare, da remoto, alcun file. Questa è una precisa scelta: infatti chiunque riuscirà a ottenere un accesso al server FTP potrebbe, come abbiamo accennato all'inizio del capitolo, utilizzarlo per depositarvi ogni sorta di file di cui ci troveremo ad avere la responsabilità. Ma dato che abbiamo protetto con una password gli accessi anonimi, il rischio non è poi così elevato, specialmente se non ci

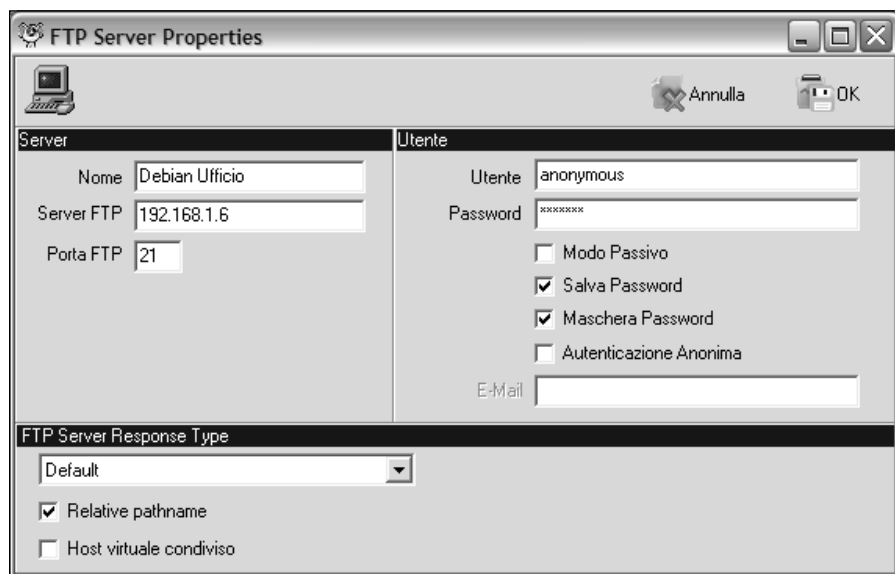


Figura 8.13

Definizione delle impostazioni di connessione a un nuovo server con **FTP Commander**.

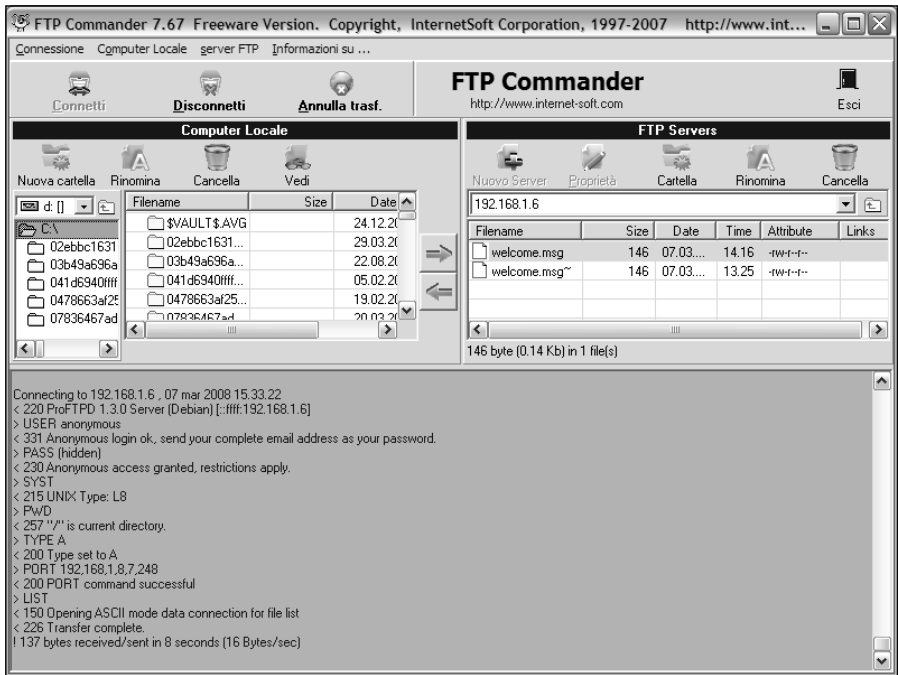


Figura 8.14

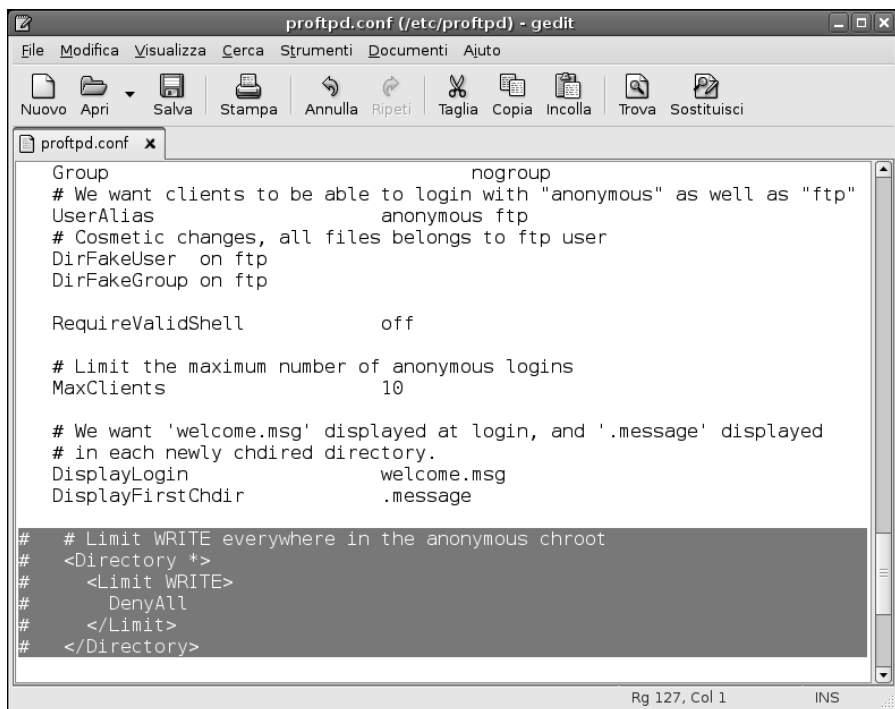
È avvenuta la connessione fra il client FTP Commander e il server FTP Debian.

mettiamo a diffondere troppo le credenziali di accesso, le quali rimarranno, sostanzialmente, in mani fidate.

Per liberalizzare le operazioni di upload e download di file da questa directory del server FTP, ci basta trasformare nuovamente in un commento le ultime righe del file `proftpd.conf`, evidenziate nella Figura 8.15. Queste righe infatti impongono un blocco in scrittura sulla directory in questione; trasformandole in commenti, elimineremo questo blocco e dunque chiunque potrà scaricare o anche caricare file sulla directory FTP.

DA SAPERE Ogni volta che si esegue una modifica anche a un piccolo elemento della configurazione del server FTP ProFTPD, è necessario riavviarlo. Il modo più comodo e grafico consiste nell'utilizzare il comando Desktop > Amministrazione > Servizi; nella finestra di dialogo Impostazioni servizi possiamo togliere e poi rimettere il segno di spunta dalla casella quadrata che si trova a sinistra della voce Servizio FTP, in modo da fermare e poi riavviare il server FTP con le nuove impostazioni.



**Figura 8.15**

Le righe del file `proftpd.conf` che bloccano l'upload di file sul server FTP.

Non bisogna dimenticare che questa liberalizzazione espone comunque a rischi e dunque occorre procedere in questo senso solo a ragion veduta.

Conclusioni

In questo capitolo abbiamo visto come definire, configurare e utilizzare un server FTP che consenta accessi in download oppure in upload/download da una directory protetta del sistema.

Questa può essere una soluzione molto comoda per scambiare file di grandi dimensioni o in grandi quantità con collaboratori remoti oppure clienti o fornitori. Nel prossimo capitolo impareremo a definire un server DHCP che si occupa di governare la distribuzione degli indirizzi di rete alle macchine un'intera rete locale.

Capitolo 9

A ciascuno il suo indirizzo: il protocollo DHCP

Per poter comunicare, le macchine di una rete (qualsiasi rete) devono avere un indirizzo. È qui che entra in gioco il server DHCP

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Il protocollo DHCP
- ☑ Configurazione DHCP su una macchina client Debian
- ☑ Un server DHCP Debian
- ☑ Installazione del server DHCP
- ☑ Configurazione di DHCP Server
- ☑ Avvio del server DHCP
- ☑ Proviamo se funziona?

Debian, come la maggior parte delle distribuzioni di Linux, si offre di configurare la rete al momento dell'installazione, cosicché il sistema sia pronto a partire. Questo prevede normalmente l'impiego di una "macchina" che svolga un servizio fondamentale per una rete, che assegni automaticamente un indirizzo a ogni sistema. Questo è il compito del protocollo DHCP (Dynamic Host Configuration Protocol). Se la rete è dotata di un router/gateway (magari anche wireless) che si occupa di questa incombenza e che assegna automaticamente un indirizzo a ogni macchina, non sarà necessario configurare nulla poiché tutte le informazioni necessarie saranno state inserite automaticamente nei file di configurazione della rete.

Altrimenti una macchina della rete dovrà occuparsi di assegnare a tutte un indirizzo. In caso contrario le macchine non potranno né vedersi né sentirsi.

Il protocollo DHCP

Il server DHCP (Dynamic Host Configuration Protocol) genera tutte le informazioni di configurazione che i sistemi hanno bisogno per connettersi a una rete TCP/IP, ovvero a qualsiasi rete, locale, intranet o Internet. Le macchine della rete diverranno dunque *client DHCP* e si configureranno sulla base di informazioni fornite dal *server DHCP* operativo sulla rete. Su ogni macchina della rete, il daemon del client DHCP contatterà automaticamente il server DHCP per prelevare automaticamente le proprie informazioni di configurazione (Figura 9.1).

Queste informazioni comprendono i seguenti elementi:

- indirizzo IP (l'informazione più importante);
- indirizzo del server DNS della rete;
- indirizzo del gateway della rete;
- indirizzo del proxy della rete;
- maschera di rete.

Dunque non dovremo introdurre mai nulla per configurare i sistemi locali e l'intera configurazione dei sistemi della rete sarà completamente centralizzata. In qualità di amministratori della rete potremo così gestire la configurazione di tutti i sistemi della rete da un unico server DHCP. Un server DHCP può offrire tre diversi metodi per l'allocazione degli indirizzi IP:

- automatica
- dinamica
- manuale

Con l'allocazione automatica, il server DHCP assegna a ogni macchina un indirizzo IP permanente.

Con l'allocazione dinamica, il server DHCP assegna a ogni macchina un indirizzo IP solo su richiesta e traendolo da un gruppo di indirizzi disponibili.

Con l'allocazione manuale, il server DHCP assegna a ogni macchina un indirizzo IP fisso e scelto esplicitamente dall'amministratore della rete.

L'allocazione dinamica degli indirizzi IP ha un grave limite. L'indirizzo IP non può essere sincronizzato con un server DNS. Il server DNS (argomento del

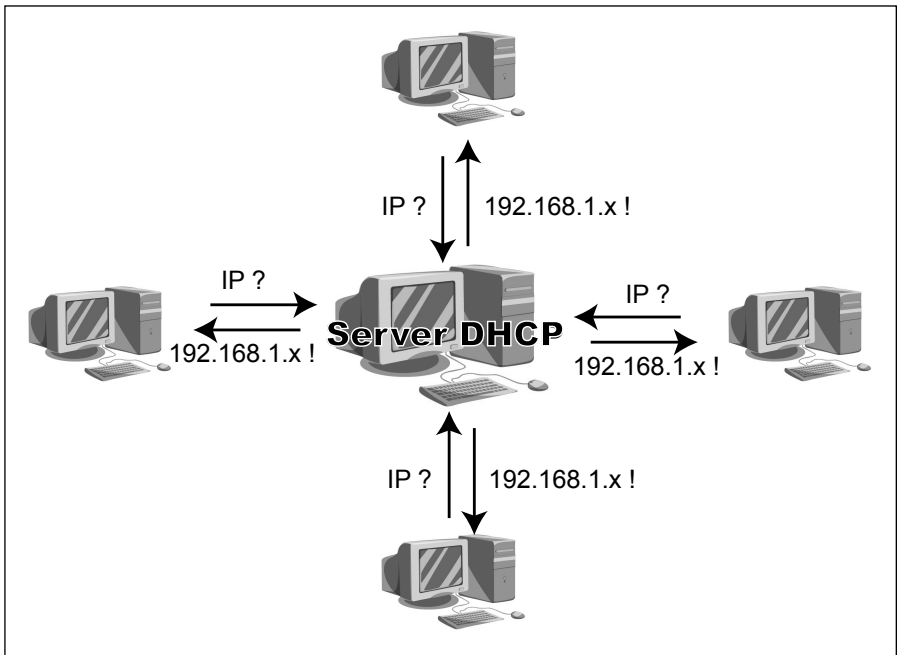


Figura 9.1

Le macchine interrogano il server DHCP che fornisce l'indirizzo IP e altre informazioni di connessione.

prossimo capitolo) crea associazioni fra nomi di host (macchine) e indirizzi IP. Con l'allocazione dinamica, il server DHCP assegna alle varie macchine un indirizzo IP casuale e quindi l'indirizzo IP cambierà di volta in volta e non sarà lo stesso indirizzo al quale il server DNS si aspetta di trovare una determinata macchina. Una soluzione è rappresentata dal Dynamic DNS che consente al server DHCP di informare automaticamente il server DNS degli indirizzi IP assegnati alle varie macchine.

Esistono vari server e client DHCP ma il più diffuso è il software DHCP ISC (Internet Software Consortium - www.isc.org). Il pacchetto software comprende un server e un client DHCP.

Le informazioni di configurazione della rete acquisite dal client DHCP della macchina Debian possono essere agevolmente consultate facendo clic con il pulsante destro del mouse sull'icona della connessione di rete (in alto a destra nel desktop Gnome) e selezionando dal menu rapido l'opzione Informazioni connessione. La Figura 9.2 mostra l'aspetto della finestra di dialogo Informazioni connessione.

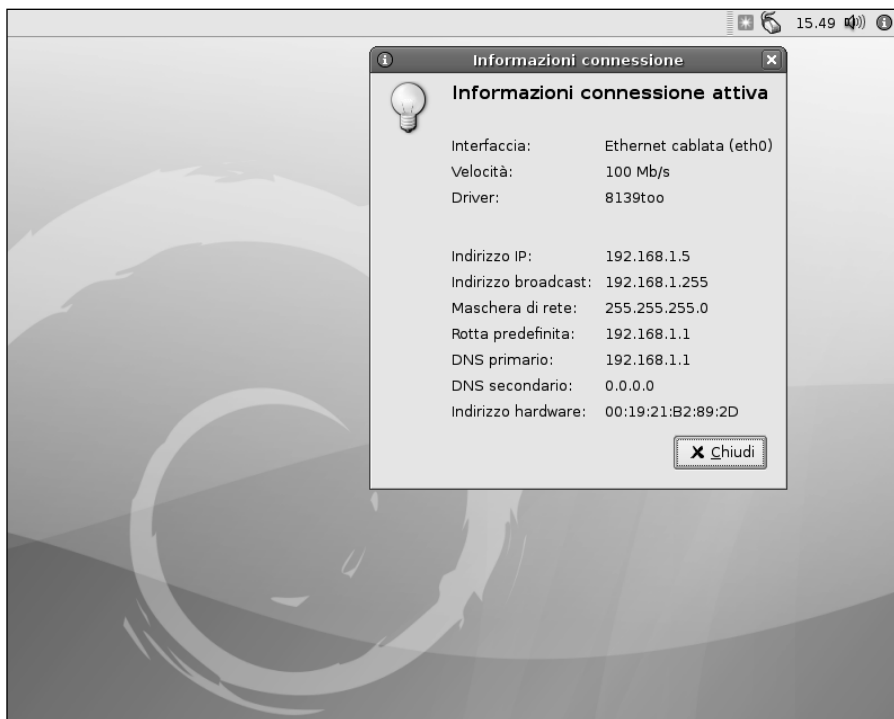


Figura 9.2

Nella finestra Informazioni connessione troviamo l'indirizzo IP assegnato alla nostra macchina e ogni altra informazione utile per la connessione di rete.

Configurazione DHCP su una macchina client Debian

Per sfruttare su una macchina Debian le funzionalità di un server DHCP disponibile nella rete (tipicamente il router/gateway residenziale) basta accedere alla configurazione delle opzioni per l'interfaccia di rete, ovvero la scheda Ethernet.

In particolare si può utilizzare un comodo comando dei menu del desktop Gnome, Desktop > Amministrazione > Rete, che richiama la finestra di dialogo Impostazioni di rete rappresentata nella Figura 9.3.

Le impostazioni predefinite fanno in modo che la nostra macchina acceda automaticamente a un server DHCP al quale richiede informazioni sulla rete. Nella finestra Impostazioni di rete, basta fare clic sul pulsante Proprietà per aprire la finestra Proprietà interfaccia (Figura 9.4). Qui troveremo nella casel-

**Figura 9.3**

La disponibilità di strumenti grafici come la finestra Impostazioni di rete semplifica notevolmente la configurazione delle funzionalità del sistema.

la Configurazione l'opzione DHCP: questo significa che nella nostra rete locale si trova un server DHCP il quale si occupa di assegnare un indirizzo IP alla nostra macchina, la quale non deve fare altro che contattare e accettare le informazioni da esso fornite.

L'altra opzione disponibile, Indirizzo IP statico, ci permette di forzare un indirizzo fisso per la nostra macchina. In questo caso si attiveranno anche i campi sottostanti: in Indirizzo IP dovremo specificare il nostro indirizzo IP statico, per esempio 192.168.1.123; in Maschera di rete dovremo specificare la maschera che, per una normale rete locale sarà con ogni probabilità 255.255.255.0; in Indirizzo del gateway dovremo specificare l'indirizzo IP della macchina che ci offre la connessione a Internet, per esempio 192.168.1.1 (Figura 9.5).

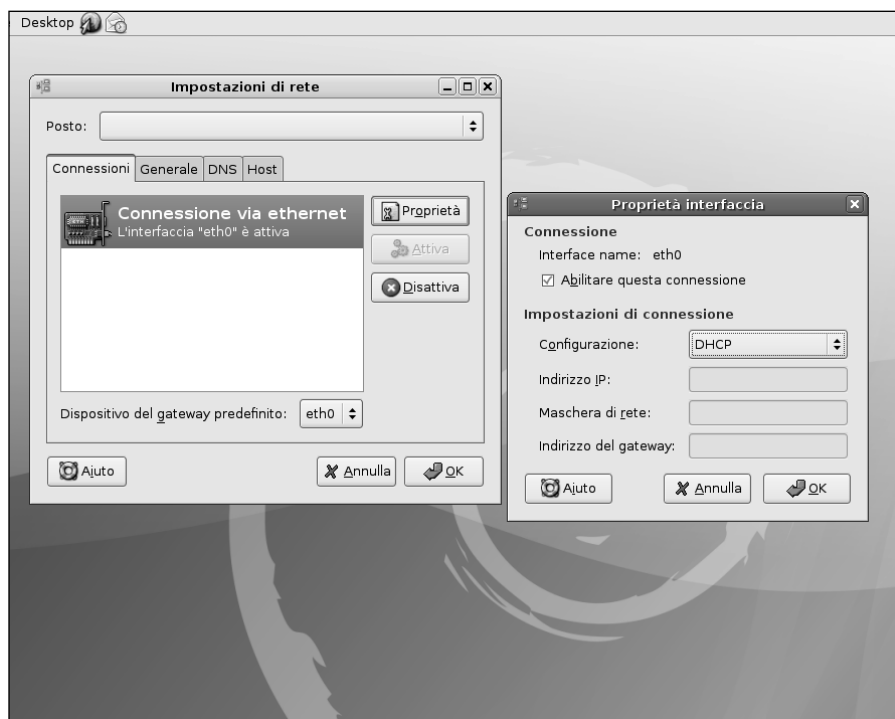


Figura 9.4

La finestra *Proprietà interfaccia* ci conferma che l'indirizzo viene assegnato alla macchina da un server DHCP.

L'impostazione di un indirizzo IP fisso può essere utile se, per esempio, la nostra macchina deve fornire alle altre macchine della rete o a Internet determinati servizi e deve pertanto essere sempre individuabile allo stesso indirizzo IP all'interno della rete o se abbiamo impostato il router/gateway della rete in modo da consentire il passaggio (ma solo verso una determinata macchina) di determinati tipi di connessioni "a rischio", normalmente filtrate, per esempio una delle famigerate applicazioni P2P.

Un server DHCP Debian

Se la rete non ha già un componente che funga da server DHCP, ovvero da "governatore" degli indirizzi di rete (magari nel router/gateway che ci garantisce la connessione a Internet), possiamo costruirlo sulla nostra macchina Debian, sempre meno simile a un sistema desktop. Nel corso di questo capitolo



Figura 9.5
Impostazione di un indirizzo IP fisso.

esamineremo l'installazione e l'uso del server DHCP dell'ISC (Internet Software Consortium), il software più utilizzato in ambiente Linux per questo compito e direttamente installabile dal desktop Gnome di Debian.

Installazione del server DHCP

Quando possibile sfruttiamo gli strumenti grafici del desktop Gnome di Debian e quindi anche questa volta utilizzeremo il Gestore di pacchetti Synaptic, avendo cura di *connetterci al sistema come utenti root*. Diversamente l'installazione e la configurazione di strumenti di sistema risulterà impossibile.

Richiamiamo Desktop > Amministrazione > Gestore pacchetti Synaptic dal desktop Gnome di Debian e, nella finestra del programma, facciamo clic sull'ico-

na Cerca nella barra degli strumenti. Nella finestra di dialogo Trova scriviamo dhcp e facciamo clic sul pulsante Cerca. Verranno visualizzati tutti i programmi client e server DHCP disponibili per la distribuzione Debian (Figura 9.6).

Ora facciamo clic sulla casella quadrata che si trova a lato della voce dhcp3-server e selezioniamo l'opzione Marca per l'installazione.

Ora possiamo avviare l'installazione, facendo clic sul pulsante Applica nella barra degli strumenti; confermiamo facendo nuovamente clic su Applica nella finestra Riepilogo. In breve, l'unico file necessario verrà scaricato da Internet e verrà eseguita l'installazione del software.

Configurazione di DHCP Server

Terminata l'installazione verrà richiamata la procedura di configurazione del server DHCP. La prima finestra della procedura è puramente informativa (Figura 9.7).

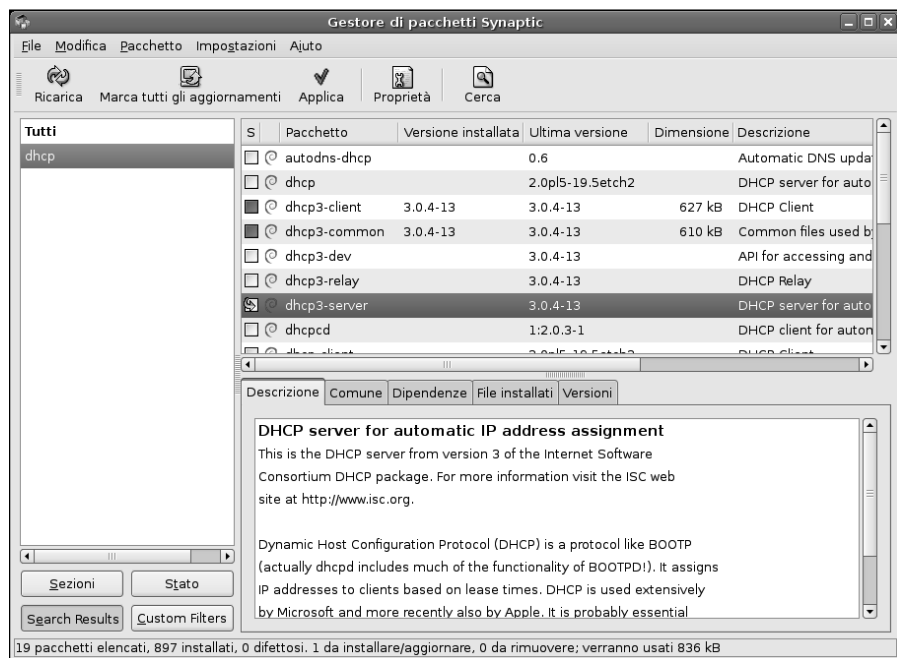


Figura 9.6
I pacchetti relativi a client e server DHCP per Debian Linux.



DA SAPERE Il server DHCP è predisposto per governare un segmento della rete. Questo significa che un client potrebbe anche richiedere un indirizzo che non rientra nella "giurisdizione" di questo server. Un server non autoritario (in inglese *authoritative*) risponderà: "Non è affar mio, ci pensi qualcun altro". Un server autoritario direbbe invece: "Le tue pretese sono irrilevanti! Lascia perdere: lo so io qual è l'indirizzo IP che fa per te".

Al termine dell'installazione la finestra dei dettagli mostrerà messaggi d'errore piuttosto preoccupanti:

```
Generating /etc/default/dhcp3-server...
```

```
Starting DHCP server: dhcpd3 failed to start - check syslog for diagnostics.
```

```
invoke-rc.d: initscript dhcp3-server, action "start" failed.
```



Figura 9.7
Parte la configurazione automatica del server DHCP.

Ma questo è perfettamente normale, dato che il server DHCP non è ancora configurato per l'uso. Questo è esattamente il prossimo passo della procedura. Dobbiamo dire al server quale intervallo di indirizzi IP può assegnare ai client, dove si trova il gateway per l'accesso a Internet, su quali server DNS può contare e così via.

Il file di configurazione del server DHCP è `/etc/dhcp3/dhcpd.conf`, che in questa fase contiene solo una configurazione base; non ce ne facciamo nulla, ma è sempre meglio copiare il file originale, operazione che possiamo eseguire anche dal desktop trascinando l'icona del file e mantenendo nel contempo premuto il tasto CTRL; verrà creato il file `dhcpd (copia).conf` (Figura 9.8).

Come abbiamo detto, l'attuale contenuto del file `dhcpd.conf` è sostanzialmente inutile. Possiamo aprirlo con l'editor Gedit, cancellarlo interamente e scrivere le seguenti direttive oppure utilizzare le direttive in esso contenute fino a ottenere, comunque, il seguente risultato:

```
ddns-update-style none;
```

```
option domain-name-servers xxx.xxx.xxx.xxx, xxx.xxx.xxx.xxx;
```

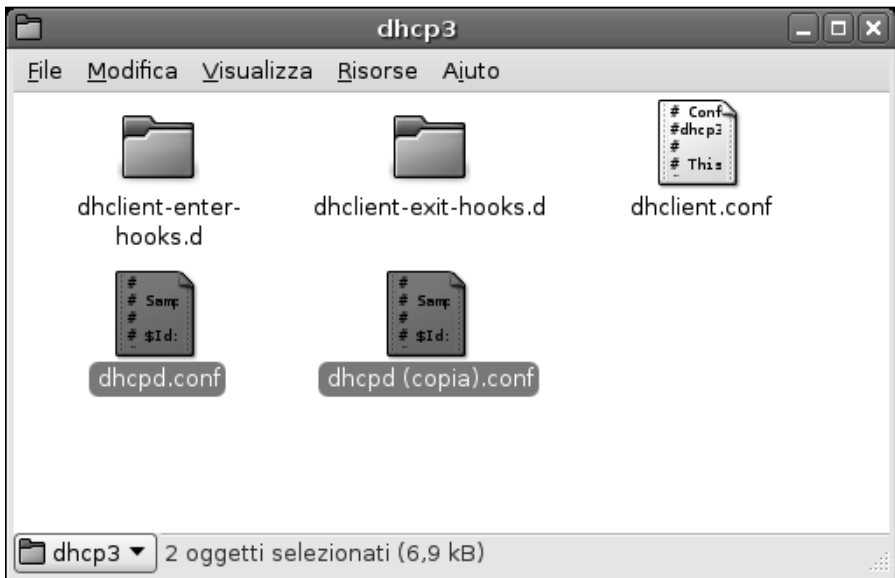


Figura 9.8

Il contenuto della directory di DHCP con il file di configurazione `dhcpd.conf` e la sua copia di backup.


```
default-lease-time 86400;
max-lease-time 604800;

authoritative;

subnet 192.168.1.0 netmask 255.255.255.0
{
    range 192.168.1.100 192.168.1.200;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
}
```

La Figura 9.9 mostra, per esempio, la configurazione di un server DHCP su una rete con indirizzi IP privati 192.168.1.x/255.255.255.0 e con provider TIN (lo si evince dai due server DNS specificati, 62.211.69.150 e 212.48.4.15)

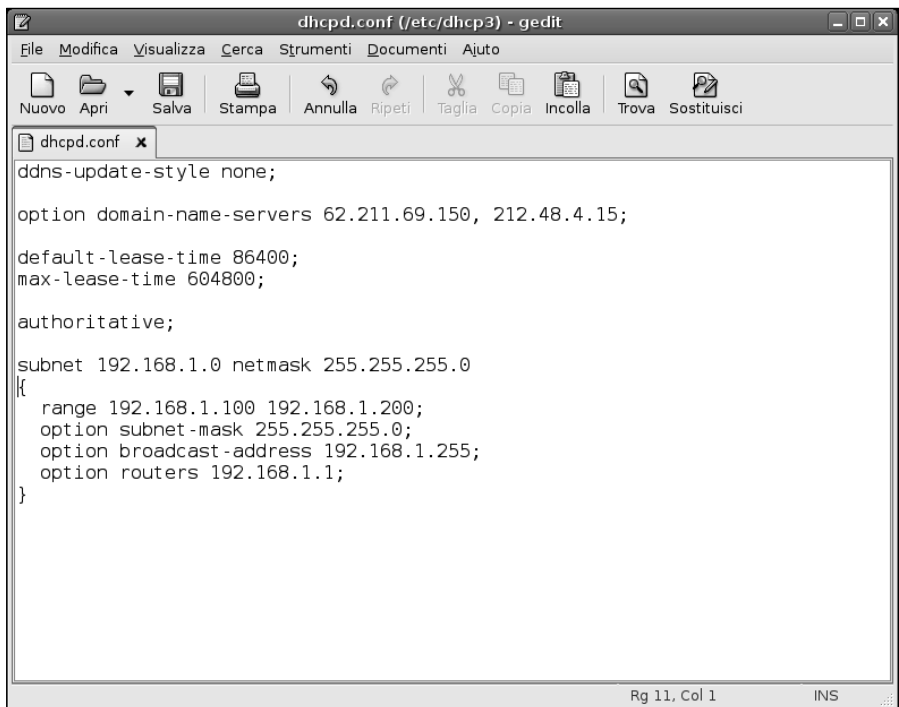


Figura 9.9

Un esempio di configurazione di rete.



DA SAPERE Possiamo scoprire i server DNS che dobbiamo specificare al posto di `xxx.xxx.xxx.xxx` nella seconda riga del file di configurazione consultando il sito del nostro provider Internet.

Le opzioni di configurazione

Ecco il significato delle opzioni di configurazione impiegate.

`ddns-update-style none`

Con `none`, il server DHCP *non* informerà il server DNS di eventuali cambiamenti negli indirizzi IP che assegna alle macchine.

`option domain-name-servers xxx.xxx.xxx.xxx, xxx.xxx.xxx.xxx`

I client DHCP vorranno sapere l'indirizzo IP dei server DNS al quale rivolgersi per accedere ai siti Internet; normalmente i provider Internet offrono un server DNS primario e uno secondario. Si tratta di informazioni da scaricare dal sito di assistenza tecnica del provider.

`default-lease-time 86400`

Un client può dire al server DHCP per quanti secondi vuole mantenere il proprio indirizzo IP. In caso contrario, sarà il server a concedere un indirizzo IP per `default-lease-time` secondi, in questo caso, $86400 \text{ sec} / 60 = 1440 \text{ min} / 60 = 24 \text{ ore}$

`max-lease-time 604800`

Se invece il client propone al server una determinata durata del proprio indirizzo IP, il server la concederà solo se è inferiore o uguale a `max-lease-time` secondi, in questo caso $604800 \text{ sec} / 60 = 10080 \text{ min} / 60 = 168 \text{ ore} / 24 = 7 \text{ giorni}$.

`authoritative`

Il nostro server sarà "autorevole" sul suo segmento di rete: se un client richiede un indirizzo di cui il server non sa nulla ed errato per il relativo segmento di rete, il server invierà al client un DHCPNAK, intimandogli di non usare tale indirizzo.


```
subnet 192.168.1.0
```

La sottorete privata utilizzata per la nostra LAN; ovviamente l'indirizzo può essere differente, ma questo indirizzo è una scelta molto comune.

```
netmask 255.255.255.0
```

La maschera della sottorete.

```
range 192.168.1.100 192.168.1.200
```

Dice l'intervallo degli indirizzi che il server DHCP può concedere ai client. In questo caso si tratta di un centinaio di indirizzi: 192.168.1.100, 192.168.1.101, 192.168.1.102, ... e 192.168.1.200.

```
option broadcast-address 192.168.1.255
```

L'indirizzo broadcast della rete. Normalmente, come in questo caso, si tratta dell'ultimo indirizzo disponibile nella sottorete: 192.168.1.255.

```
option routers 192.168.1.1
```

Un'altra informazione che i client vorranno sapere dal server DHCP sarà l'indirizzo del gateway, ovvero della macchina o del componente di rete che garantisce l'accesso a Internet. In questo caso il router/gateway si trova all'indirizzo 192.168.1.1.

Come si può vedere le direttive da utilizzare sono semplici e intuitive.

Avvio del server DHCP

Non rimane che avviare il server, che a questo punto può contare su parametri di funzionamento adatti alla rete in cui si trova a operare.

```
/etc/init.d/dhcp3-server restart
```

Possiamo verificare che il nostro server DHCP sia attivo con il comando:

```
netstat -uap
```

che produce le connessioni Internet attive (comprese quelle dei server). Il risultato di questi comandi dovrebbe essere simile a quello rappresentato nella Figura 9.10; la riga evidenziata è proprio quella relativa al nostro server DHCP.


```

Terminale
File Modifica Visualizza Terminale Schede Ajuto
Debian:~# /etc/init.d/dhcp3-server restart
Stopping DHCP server: dhcpd3.
Starting DHCP server: dhcpd3.
Debian:~# netstat -uap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
udp        0      0 *:32770                *:                        3032/avahi-daemon:
udp        0      0 *:32771                *:                        3125/rpc.statd
udp        0      0 *:bootps                *:                        24480/dhcpd3
udp        0      0 *:bootpc                *:                        3128/dhclient
udp        0      0 *:bootpc                *:                        4612/dhclient3
udp        0      0 *:mdns                  *:                        3032/avahi-daemon:
udp        0      0 *:sunrpc                *:                        2413/portmap
udp        0      0 *:757                   *:                        3125/rpc.statd
udp        0      0 *:ipp                   *:                        2955/cupsd
Debian:~#

```

Figura 9.10

Il server DHCP è attivo e funzionante.

Proviamo se funziona?

Prima considerazione: non possiamo avere due server DHCP nella stessa rete. Se è attivo quello del router/gateway dovremo disattivarlo, così da trasferire tutte le sue attività al nostro nuovo server DHCP.

A questo punto, per scoprire se il nostro server DHCP funziona nel modo previsto, ci basta avviare un altro PC della rete locale (Linux, ma anche Windows o Macintosh o, perché no, una console di videogiochi). Tipicamente non avrà un indirizzo IP statico e dunque cercherà di chiederlo a qualcuno. Sappiamo già chi è questo “qualcuno”: è proprio il nostro server DHCP Linux. Dopo qualche istante, nel registro degli eventi di Linux, ovvero nel file `/var/log/syslog` del server DHCP troveremo traccia dell’assegnamento dell’indirizzo IP al PC.

Ecco un esempio di un estratto del file `/var/log/syslog`:

```

Mar 13 10:39:26 debian dhcpd: DHCPDISCOVER from 00:13:ce:ea:5d:27 via eth0
Mar 13 10:39:26 debian dhcpd: DHCPOFFER on 192.168.1.123
to 00:13:ce:ea:5d:27 (NB-Paolo) via eth0

```



```
Mar 13 10:39:27 debian dhcpd: DHCPDISCOVER from 00:13:ce:ea:5d:27
(NB-Paolo) via eth0
Mar 13 10:39:27 debian dhcpd: DHCPOFFER on 192.168.1.123
to 00:13:ce:ea:5d:27 (NB-Paolo) via eth0
Mar 13 10:39:31 debian dhcpd: DHCPDISCOVER from 00:13:ce:ea:5d:27
(NB-Paolo) via eth0
Mar 13 10:39:31 debian dhcpd: DHCPOFFER on 192.168.1.123
to 00:13:ce:ea:5d:27 (NB-Paolo) via eth0
Mar 13 10:39:31 debian dhcpd: Wrote 1 leases to leases file.
Mar 13 10:39:31 debian dhcpd: DHCPREQUEST for 192.168.1.123
(192.168.0.100) from 00:13:ce:ea:5d:27 (NB-Paolo) via eth0
Mar 13 10:39:31 debian dhcpd: DHCPACK on 192.168.1.123
to 00:13:ce:ea:5d:27 (NB-Paolo) via eth0
```

Il server DHCP scrive tutte le concessioni di indirizzi IP sul file `/var/lib/dhcp3/dhcpd.leases`, nel quale possiamo naturalmente andare a curiosare:

```
# All times in this file are in UTC (GMT), not your local timezone. This is
# not a bug, so please don't ask about it. There is no portable way to
# store leases in the local timezone, so please don't request this as a
# feature. If this is inconvenient or confusing to you, we sincerely
# apologize. Seriously, though - don't ask.
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-V3.0.4
```

```
lease 192.168.1.123 {
    starts 2 2007/03/13 10:39:31;
    ends 3 2007/03/13 10:39:31;
    binding state active;
    next binding state free;
    hardware ethernet 00:13:ce:ea:5d:27;
    uid "\001\000\014v\213\304\026";
    client-hostname "NB-Paolo";
}
```

Conclusioni

In questo capitolo abbiamo descritto il funzionamento, l'installazione, la configurazione e l'uso di un server DHCP che distribuisce indirizzi IP corretti alle macchine della sottorete di competenza. Può essere una soluzione interessante per controllare la distribuzione degli indirizzi e per conoscere questo aspetto del funzionamento delle reti.

Nel prossimo capitolo vedremo come è possibile creare un server DNS: il vero e proprio anello di connessione fra la rete locale e i milioni di server disponibili in Internet.

Capitolo 10

Nomi e indirizzi IP: DNS, il grande traduttore

A ogni sito corrisponde un nome ma anche un indirizzo IP. Naturalmente è più facile ricordarsi il nome che non l'indirizzo IP di un sito. Se il Web è così amichevole, lo dobbiamo alla costante opera di traduzione fra nomi e indirizzi svolta dai server DNS.

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ✓ Ricerche DNS
- ✓ Creare un server DNS con BIND
- ✓ Installiamo BIND
- ✓ Configurazione di BIND
- ✓ Un server autorevole sulla zona
- ✓ Proviamo a usare il nuovo server DNS

DNS (Domain Name Service) è un servizio di rete che si occupa di convertire i nomi di dominio nei corrispondenti indirizzi IP. Tutti i computer connessi a Internet sono infatti identificati da un indirizzo IP (Internet Protocol) costituito da una sequenza di quattro numeri separati da un punto.



DA SAPERE *In base al tipo di rete, alcuni di questi numeri (i primi) vengono utilizzati per l'indirizzo della rete; gli ultimi indicano invece l'indirizzo della macchina vera e propria. In una piccola rete locale, i primi tre numeri corrispondono all'indirizzo della rete mentre l'ultimo identifica il computer.*

Per esempio, nell'indirizzo IP 192.168.1.2, la parte che indica la rete locale è 192.168.1 mentre la parte che indica il computer, all'interno di tale rete locale è 2. Insieme, questi numeri compongono l'indirizzo IP tramite il quale il computer può essere contattato nella rete locale o in Internet.

Gli indirizzi IP (numerici) sono sicuri ma difficili da ricordare; molto meglio impiegare dei nomi facili da memorizzare. Agli indirizzi IP sono stati pertanto associati dei nomi chiamati *nomi di dominio*. Ogni computer connesso a Internet utilizza un file all'interno del quale si trovano le associazioni fra indirizzi IP e relativi nomi di dominio. In Linux questo file si chiama `/etc/hosts` (Figura 10.1). In questo file possiamo introdurre l'indirizzo IP e il relativo nome di dominio dei computer che utilizziamo più frequentemente.

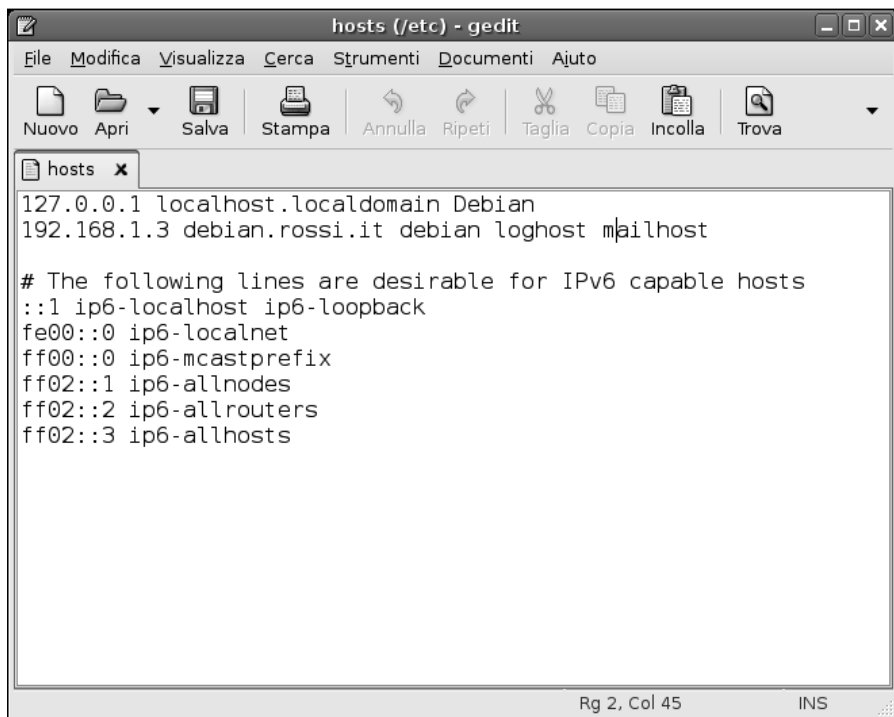


Figura 10.1

Il file `hosts` del nostro sistema Debian.

Questo semplice metodo può andare bene per una manciata di macchine, magari anche qualche decina, in pratica solo per le macchine presenti nella rete locale (e in questo caso funziona benissimo). Ma è chiaro che per i milioni di computer raggiungibili in Internet tale soluzione diventa ridicola. Il sistema sarebbe costretto a creare lunghi elenchi, da aggiornare costantemente.

È proprio qui che interviene il servizio offerto dai *server DNS*: la traduzione dei nomi di domini nei corrispondenti indirizzi IP. Internet è “coperta” da una rete di server DNS interconnessi che conservano grandi elenchi di nomi di domini e dei corrispondenti indirizzi IP. Immaginiamo Internet come una “rete di reti”, cioè una grande rete costituita da moltissime sottoreti; ogni sottorete ha i propri server DNS, autorevoli sul proprio segmento di rete, che memorizzano il nome e l’indirizzo IP di tutti i computer che compongono la sottorete.

Possiamo quindi individuare una struttura gerarchica di server DNS interconnessi che, nella sua globalità, conserva informazioni su tutte le macchine connesse a Internet.

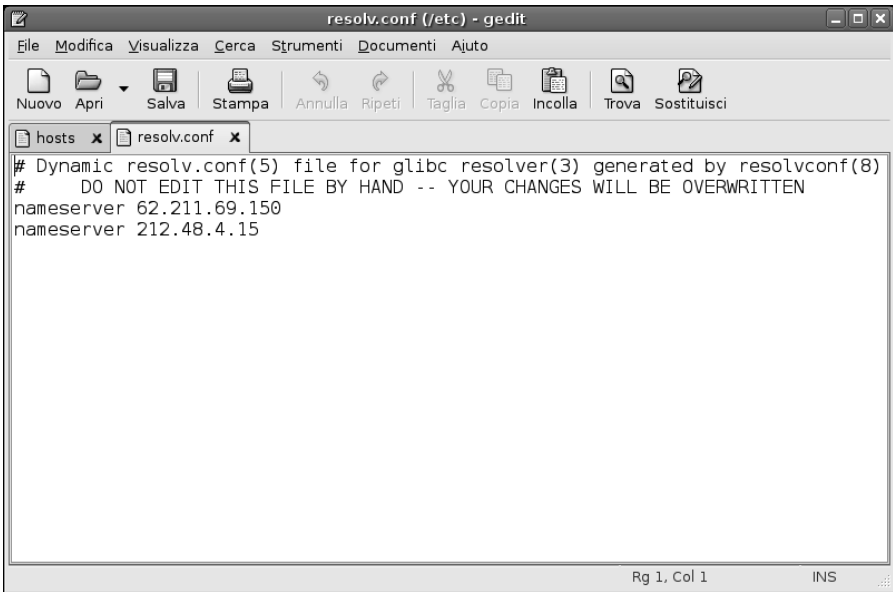
Quando introduciamo nel browser Web (per esempio) il nome di dominio di un sistema remoto, il nostro computer interroga il server DNS della rete locale per richiedere l’indirizzo IP corrispondente a tale nome. Se il server DNS locale non ha questa informazione, trasferisce la richiesta a un server DNS di livello superiore e poi ancora più su, fino a trovare l’indirizzo IP del sito desiderato.

I server DNS (impostati al momento della configurazione del sistema) che sono in grado di esaudire le nostre richieste li troviamo elencati nel file `/etc/resolv.conf` (Figura 10.2). Tipicamente, in questo file troveremo i server DNS primario e secondario (di riserva) del nostro provider. Tali server DNS sono proprio quelli che traducono il nome `www.google.it` nel corrispondente indirizzo IP: `209.85.135.103`.

Ricerche DNS

Per scoprire questo mondo numerico che si cela dietro i ben noti indirizzi testuali, possiamo utilizzare il comodo comando `nslookup` (Name Space Lookup, ovvero ricerca nello spazio dei nomi). Come possiamo vedere nella Figura 10.3, l’output del comando è piuttosto interessante.

Le prime due righe prodotte dal comando, indicano l’indirizzo (`62.211.69.150`) del server DNS che ha risolto l’indirizzo; si tratta ovviamente del server primario del nostro provider (TIN in questo caso) e dunque qui ognuno potrebbe trovare un indirizzo diverso (tranne ovviamente coloro che usano il provider TIN). La parte “#53” finale indica la porta utilizzata su tale server ed è la tipica porta dei servizi DNS.

**Figura 10.2**


Il file `resolv.conf` contiene l'indirizzo IP dei due server DNS impostati al momento della configurazione della connessione a Internet. In questo caso si tratta dei server del provider TIN.

Non-authoritative answer. Cosa significa che la risposta non è autorevole? Che non ci possiamo fidare? Significa semplicemente che il server DNS che abbiamo contattato (quello del nostro provider) non ha alcun controllo sul sito `www.google.it`. Conosce questa informazione perché è nelle sue tabelle o l'ha richiesta a un server più "in alto" di lui. Dunque l'informazione è sicura ma non è "di prima mano".

Le righe successive indicano il nome e il corrispondente indirizzo IP delle macchine che rispondono al nome `www.google.it`. Sono quattro (in questo caso) perché Google ha deciso di bilanciare il carico delle richieste in italiano su quattro macchine, scelte casualmente dal server DNS.

Creare un server DNS con BIND

Visto che i server DNS del nostro provider svolgono già un ottimo lavoro, perché mai dovremmo preoccuparci di gestire un nostro server DNS? Perché in questo modo avremo una risoluzione immediata degli indirizzi testuali nei corrispondenti indirizzi numerici e non dovremo pertanto contattare a ogni richiesta i server del provider.



```
Terminale
File Modifica Visualizza Terminale Schede Ajuto
Debian:~# nslookup www.google.it
Server:        62.211.69.150
Address:       62.211.69.150#53

Non-authoritative answer:
www.google.it canonical name = www.google.com.
www.google.com canonical name = www.l.google.com.
Name:   www.l.google.com
Address: 209.85.135.103
Name:   www.l.google.com
Address: 209.85.135.104
Name:   www.l.google.com
Address: 209.85.135.147
Name:   www.l.google.com
Address: 209.85.135.99

Debian:~# █
```

Figura 10.3

Ecco quali macchine rispondono quando digitiamo `www.google.it` nel nostro browser Web.

Ciò si tradurrà in un incremento sensibile (o anche notevole) della velocità di navigazione, specialmente nel caso di provider particolarmente congestionati, specialmente in determinati orari di picco.

Il software DNS più utilizzato su sistemi Linux si chiama BIND, che in inglese vuol dire, sì, opportunamente, “legare” (perché “lega” gli indirizzi IP ai corrispondenti nomi), ma in realtà è un acronimo: Berkeley Internet Name Domain. BIND è stato originariamente sviluppato presso la University of California, Berkeley e attualmente è distribuito e supportato dall’ISC (Internet Software Consortium - www.isc.org).

BIND è costituito da un daemon DNS, vari file di configurazione e da alcune librerie di risoluzione degli indirizzi. Debian, come la maggior parte delle distribuzioni Linux, installa la versione 9 di BIND. Il daemon del server DNS, `named`, ascolta le richieste in arrivo e restituisce l’indirizzo IP corrispondente al nome richiesto. Una volta lanciato il server DNS potremo verificarne il funzionamento con il comando `nslookup` (che abbiamo appena visto all’opera).

Installiamo BIND

Richiamiamo ancora una volta il Gestore di pacchetti Synaptic, ricordandoci di *connetterci al sistema come utenti root*. Diversamente non potremo installare e configurare alcuno strumento di sistema.

Richiamiamo Desktop > Amministrazione > Gestore pacchetti Synaptic dal desktop Gnome di Debian e, nella finestra del programma, facciamo clic sull'icona Cerca nella barra degli strumenti. Nella finestra di dialogo Trova scriviamo bind e facciamo clic sul pulsante Cerca. Verranno visualizzati tutti i software DNS BIND disponibili per Debian (Figura 10.4).

Ora facciamo clic sulla casella quadrata che si trova a lato della voce bind9 e selezioniamo l'opzione Marca per l'installazione. Ogni altro pacchetto necessario (bind9-host e dnstools) è già presente e installato sul nostro sistema.

Ora possiamo avviare l'installazione, facendo clic sul pulsante Applica nella barra degli strumenti; confermiamo facendo clic sul pulsante Applica nella finestra Riepilogo. In breve, il file di BIND verrà scaricato da Internet. L'installazione sarà rapida e "invisibile".

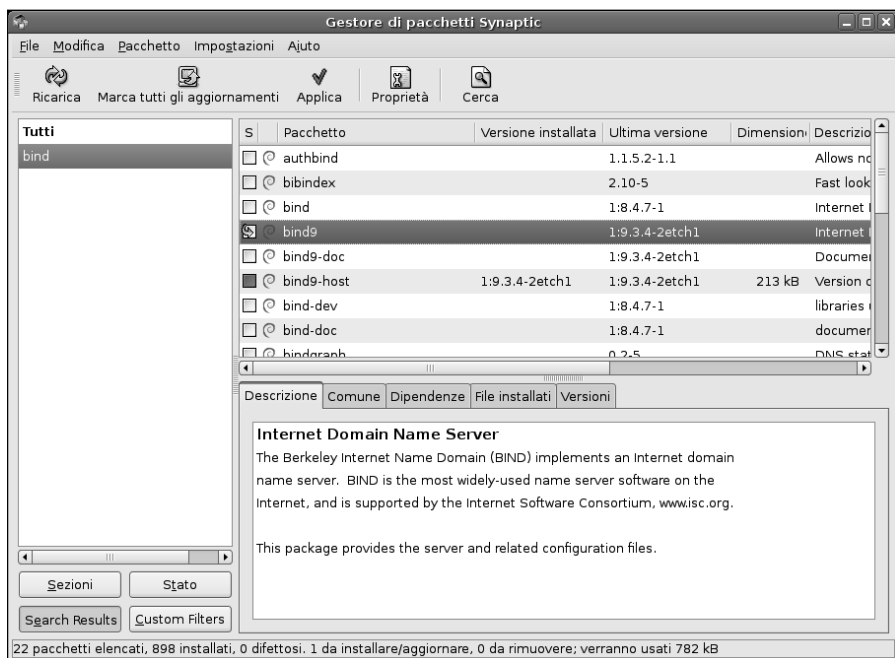


Figura 10.4

I pacchetti relativi al server DNS BIND per Debian Linux.

Configurazione di BIND

Il server che stiamo approntando può operare in tre diverse modalità.

`root only`

Ogni nostra richiesta verrà sottoposta ai grandi server DNS root autorevoli nei rispettivi domini e ai quali intendiamo appoggiarci per ottenere l'indirizzo IP corretto. In pratica in questo caso salteremo i server DNS del nostro provider.

`forward only`

Ogni nostra richiesta verrà sottoposta al server DNS del nostro provider; se ci fornisce la risposta, bene, altrimenti il risultato sarà un errore per mancata risoluzione del nome in un indirizzo IP.

`forward first`

Questa è l'opzione che sceglieremo: prima la nostra richiesta viene sottoposta al server DNS del nostro provider; se poi questo non sa nulla del nome che gli abbiamo sottoposto, dovranno essere interrogati i server DNS root.

BIND viene normalmente installato in modalità `root only`, ma lo imposteremo in modalità `forward first` in modo che contatti innanzitutto il server DNS del nostro provider che, in media, sarà sicuramente il più veloce a risponderci.

I file di configurazione di BIND si trovano nella directory `/etc/bind` (Figura 10.5); in particolare dovremo intervenire sul file `named.conf.options`.

Per prima cosa creiamo una copia di riserva del file originale trascinandolo con il puntatore del mouse mantenendo nel contempo premuto il tasto CTRL. A questo punto possiamo aprire con sicurezza il file con l'editor di testi di Gnome. Qui dobbiamo specificare l'opzione `forward first` e indicare il o i server DNS del nostro provider Internet. Tutte le altre righe del file possono essere eliminate. Il risultato dovrebbe essere simile a quello rappresentato nella Figura 10.6, tranne per i server DNS, dove ognuno dovrà specificare quelli del proprio provider Internet.

Abbiamo modificato la configurazione del server. Per far sì che la utilizzi dovremo riavviarlo con il comando:

```
/etc/init.d/bind9 restart
```

Il nostro nuovo server DNS verrà riavviato e a questo punto gestirà già localmente, tramite la cache che andrà progressivamente a costruire, tutte le richieste di nomi che gli sottoporremo dalla nostra macchina Debian. Possia-

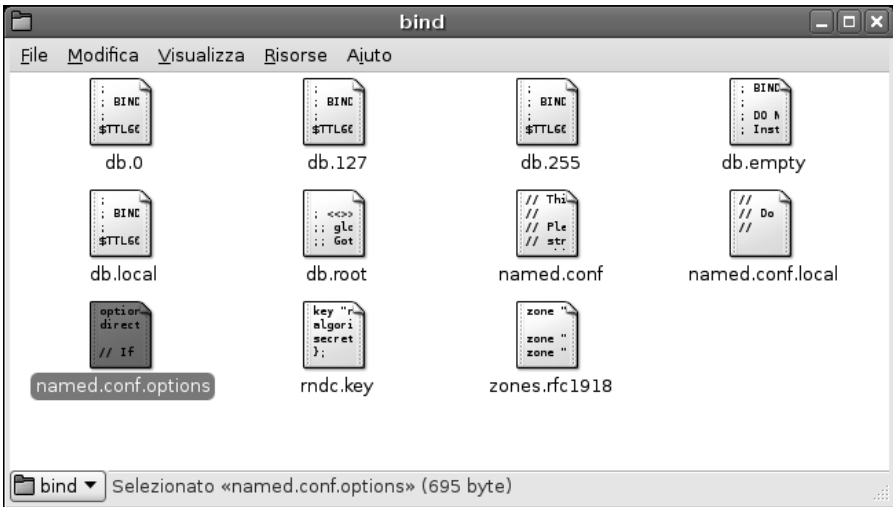


Figura 10.5
I file di configurazione di BIND nella directory /etc/bind.

mo verificarlo di nuovo con il comando `nslookup`, che questa volta utilizzerà il nostro server DNS tramite lo speciale indirizzo `localhost`, ovvero `127.0.0.1` (Figura 10.7).

Un server autorevole sulla zona

Ma naturalmente questo non ci basta.

Il nostro server DNS (che diverrà sempre più “intelligente” e conoscerà sempre più indirizzi con il passare del tempo), deve anche conoscere qualcosa di più sulle altre macchine della rete locale.

Possiamo creare una nuova “zona” intervenendo ancora sul file `/etc/bind/named.conf.options` e definiamo al suo interno la zona `zona-locale` sulla quale il nostro server DNS sarà autorevole; in altre parole potrà offrire una propria traduzione sicura ed esatta dei nomi in indirizzi IP:

```
zone "zona-locale" in
{
    type master;
    file "/etc/bind/db.zona-locale";
};
```

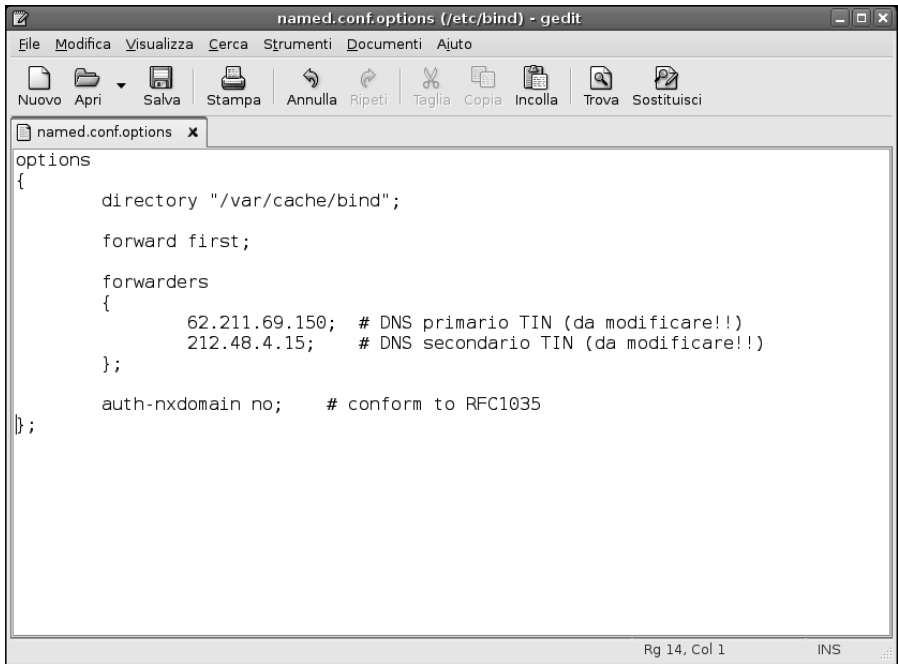



Figura 10.6

L'aspetto del file di configurazione named.conf.options dopo gli interventi necessari.

Il file `/etc/bind/named.conf.options` dovrebbe quindi avere l'aspetto rappresentato nella Figura 10.8.

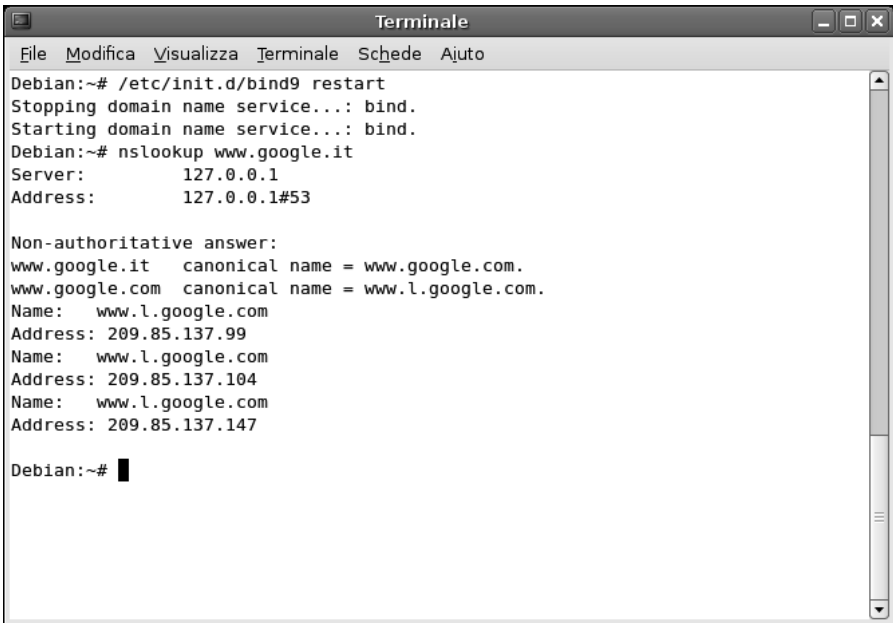
La nostra zona sarà quindi denominata `zona-locale`; il nostro server DNS sarà autorevole su tale zona, in quanto abbiamo utilizzato la clausola `type master`; definiremo la nuova zona nel file `/etc/bind/db.zona-locale`.

Non ci resta che creare il file di zona `/etc/bind/db.zona-locale`, il quale si occuperà di risolvere i nomi delle macchine della sottorete in indirizzi IP; il file dovrà avere l'aspetto rappresentato nella Figura 10.9.

In questo caso abbiamo impostato un tempo di "vita" (**TTL** = Time To Live) delle richieste su un arco di tre ore.

Poi abbiamo nominato `nome-sito.zona-locale` quale server DNS autorevole della zona `zona-locale`. Il simbolo `@` fa in modo che BIND utilizzi il nome che trova nel file `named.conf`.

Quindi indichiamo il nome e l'indirizzo del server DNS (**NS** = Name Server). Per esempio, in queste righe abbiamo specificato l'indirizzo IP di `pippo.zona-locale` (questa stessa macchina).



```

Terminale
File Modifica Visualizza Terminale Schede Ajuto
Debian:~# /etc/init.d/bind9 restart
Stopping domain name service...: bind.
Starting domain name service...: bind.
Debian:~# nslookup www.google.it
Server:          127.0.0.1
Address:         127.0.0.1#53

Non-authoritative answer:
www.google.it    canonical name = www.google.com.
www.google.com  canonical name = www.l.google.com.
Name:   www.l.google.com
Address: 209.85.137.99
Name:   www.l.google.com
Address: 209.85.137.104
Name:   www.l.google.com
Address: 209.85.137.147

Debian:~# █

```

Figura 10.7

Il comando `nslookup` ci conferma che ora la nostra macchina sta usando il server DNS interno che abbiamo appena configurato.

Poi abbiamo nominato `pluto.zona-locale` quale alias di `pippo.zona-locale`. Controlliamo il file `named.conf` con il comando `named-checkconf` e il file di zona `db.zona-locale` con `named-checkzone zona-locale db.zona-locale`. Se non ci sono errori riavviamo BIND digitando ancora il comando `/etc/init.d/bind9 restart` (Figura 10.10) e mettiamo alla prova il nostro nuovo server DNS.

Proviamo a usare il nuovo server DNS

Vediamo come il nuovo server DNS riesce a migliorare i tempi di accesso ai siti Internet.

Bisogna cominciare con una prima richiesta relativa a un sito, per controllare i relativi tempi di accesso:

```

Debian:/etc/bind# time nslookup www.flickr.com
Server:          127.0.0.1
Address:         127.0.0.1#53

```

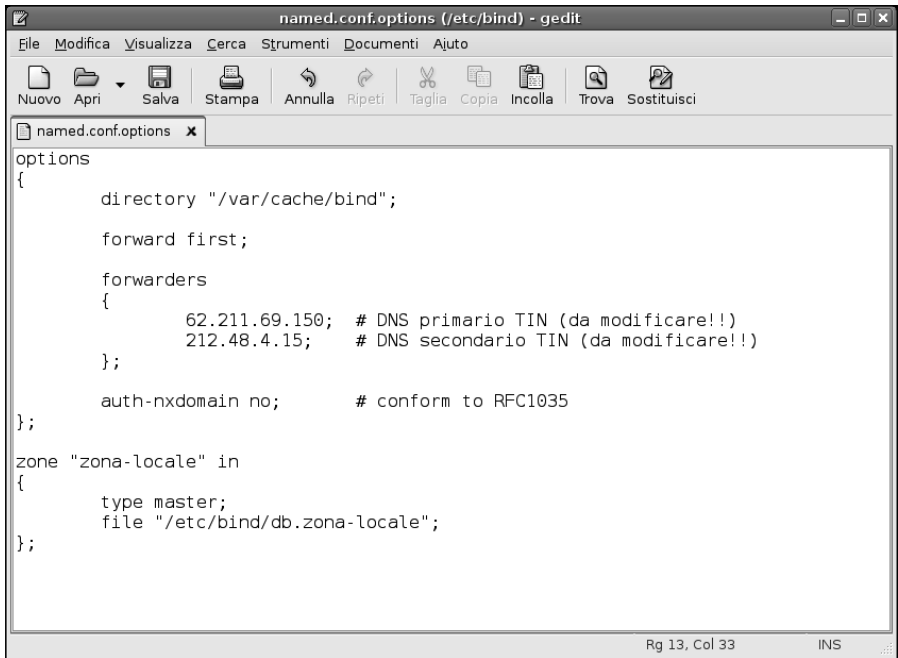



Figura 10.8

Aggiungiamo la zona e il relativo file alla configurazione del server DNS.

Non-authoritative answer:

www.flickr.com canonical name = www.flickr.vip.mud.yahoo.com.

Name: www.flickr.vip.mud.yahoo.com

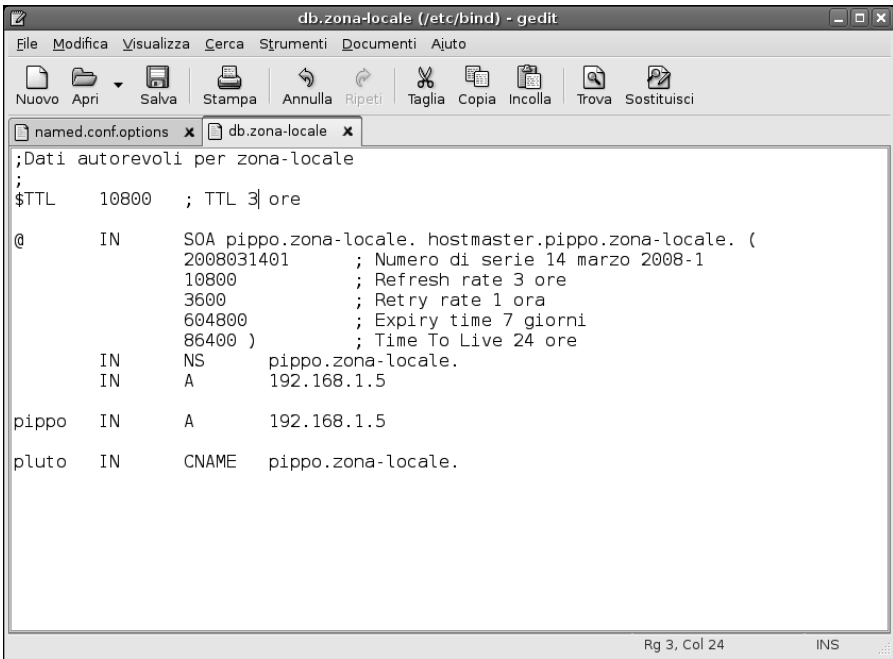
Address: 68.142.214.24

real 0m0.134s

user 0m0.000s

sys 0m0.004s

Un tempo di risposta di 134 ms per una richiesta che ha dovuto necessariamente risalire dal nostro server DNS a quello del provider non è affatto trascurabile.

**Figura 10.9**

Creiamo il file di zona `/etc/bind/db.zona-locale` per la nostra sottorete.

Ripetiamo ora questa stessa ricerca. Ora il nostro server DNS BIND conosce immediatamente la risposta e ci aspettiamo tempi di risposta inferiori. Di quanto? Vediamo subito:

```
Debian:/etc/bind# time nslookup www.flickr.com
```

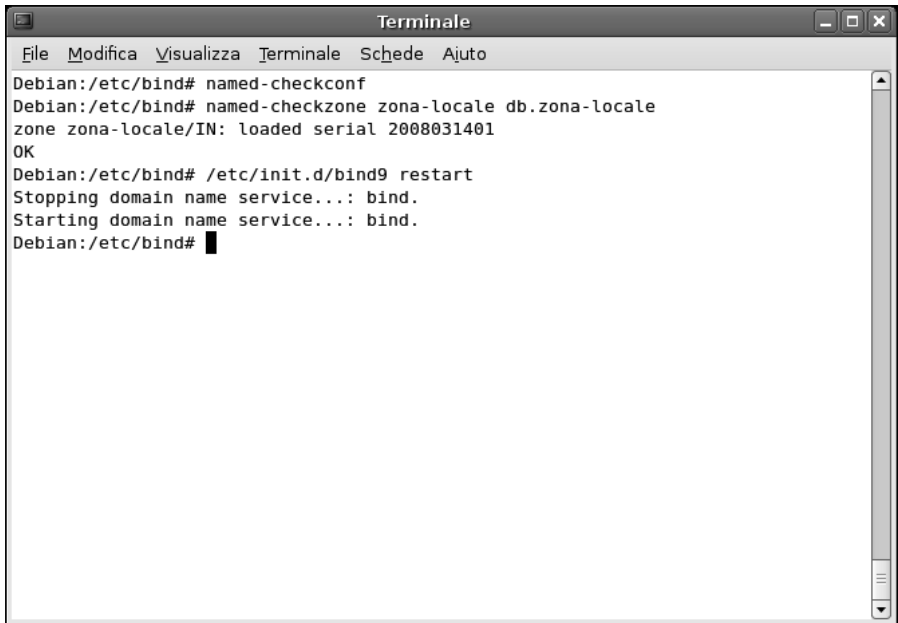
```
Server:      127.0.0.1
Address:     127.0.0.1#53
```

Non-authoritative answer:

```
www.flickr.com canonical name = www.flickr.vip.mud.yahoo.com.
Name:   www.flickr.vip.mud.yahoo.com
Address: 68.142.214.24
```

```
real    0m0.005s
user    0m0.000s
sys     0m0.004s
```

Un totale di 5 ms: oltre 20 volte più veloce (Figura 10.11)!



```
Terminale
File Modifica Visualizza Terminale Schede Ajuto
Debian:/etc/bind# named-checkconf
Debian:/etc/bind# named-checkzone zona-locale db.zona-locale
zone zona-locale/IN: loaded serial 2008031401
OK
Debian:/etc/bind# /etc/init.d/bind9 restart
Stopping domain name service...: bind.
Starting domain name service...: bind.
Debian:/etc/bind#
```

Figura 10.10
Test della configurazione e riavvio del server DNS BIND.

Conclusioni

In questo capitolo abbiamo imparato ad approntare un server DNS privato della rete locale, che garantirà tempi di accesso eccezionali (almeno a partire dal secondo accesso a un sito) nella navigazione Internet.

Dopo aver introdotto gli aspetti teorici di questo servizio abbiamo visto quanto sia facile installare e configurare il server DNS BIND (versione 9).

Non contenti, abbiamo messo alla prova il nuovo server per sperimentare anche quantitativamente le sue prestazioni.

Nel prossimo capitolo affronteremo la creazione e l'uso di un comodo server per messaggi istantanei/chat di utilizzo interno della rete locale.



Figura 10.11

Abbiamo messo alla prova il nostro server DNS con un sito: la prima volta ha dovuto chiedere informazioni al DNS del provider, ma la seconda volta è "schizzato" immediatamente all'indirizzo corretto.

Capitolo 11

Messaggi istantanei in rete locale: Jabber

Un server e un sistema IM “libero” per attivare un sistema di comunicazione interno alla rete locale.

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Il server IM Jabber
- ☑ Installare Jabber
- ☑ Configurazione di Jabber
- ☑ Installazione di un client per Jabber
- ☑ Primo avvio e configurazione di Gabber

Spesso farebbe comodo poter scambiare piccole comunicazioni di servizio interne all'azienda, una specie di chat agile e interna. La soluzione più semplice? Scaricare un client qualsiasi e connettersi a un account di quelli “ufficiali” come MSN, ICQ Yahoo e così via. Ma siamo proprio sicuri di voler far viaggiare i nostri messaggi per mezzo mondo solo per raggiungere il collega che sta a dieci metri da noi o anche meno.

La soluzione “vera”? Installare un server IM sulla nostra macchina Debian (che ormai sta diventando sempre più ricca di servizi). L'installazione e la gestione sono abbastanza semplici e, dopo aver scaricato un client adatto, potremo conversare amabilmente e utilmente in tutta rapidità e sicurezza con i nostri colleghi.

Il server IM Jabber

Il modo più comodo, sicuro e semplice per gestire un servizio di messaggi istantanei in Linux consiste nell'installare il server Jabber, uno dei tantissimi strumenti software disponibili gratuitamente per Linux e in generale per tutte le varianti di Unix.

A differenza degli altri sistemi di chat e messaggi istantanei come MSN, Jabber, in puro stile Linux, è completamente aperto, sia come protocollo sia come software: non ha nulla da nascondere e nulla da mantenere segreto.

Per prima cosa dovremo quindi scaricare e installare il server Jabber.

Installare Jabber

Anche in questo caso svolgeremo la procedura di installazione di Jabber in modo grafico, sfruttando il Gestore di pacchetti Synaptic, che si assicura di caricare tutti gli altri pacchetti necessari per il funzionamento di Jabber.

Richiamiamo Desktop > Amministrazione > Gestore pacchetti Synaptic dal desktop Gnome di Debian e, nella finestra del programma, facciamo clic sull'icona Cerca nella barra degli strumenti. Nella finestra di dialogo Trova scriviamo jabber e poi facciamo clic sul pulsante Cerca. Verranno visualizzati tutti i pacchetti legati in qualche modo a Jabber (Figura 11.1).

Ve ne sono molti, poiché Jabber può funzionare anche come agente di trasporto verso altri sistemi di Instant Messaging.

Ora facciamo clic sulla casella quadrata che si trova a lato della voce jabber e selezioniamo l'opzione Marca per l'installazione. Verranno segnalati due pacchetti che devono essere installati obbligatoriamente insieme a quello principale: jabber-common e libpth20. Facciamo clic sul pulsante Marca.

Ora possiamo avviare l'installazione, facendo clic sul pulsante Applica nella barra degli strumenti; confermiamo facendo clic sul pulsante Applica nella finestra Riepilogo. In breve, i tre file verranno scaricati da Internet e verrà eseguita l'installazione del software.

Concludiamo facendo clic sul pulsante Chiudi e poi usciamo dal Gestore di pacchetti Synaptic.

Configurazione di Jabber

Subito dopo l'installazione, il server viene automaticamente avviato, come possiamo agevolmente verificare utilizzando il seguente comando:

```
Debian:/etc/jabber# ps ax -l | grep jabber
```

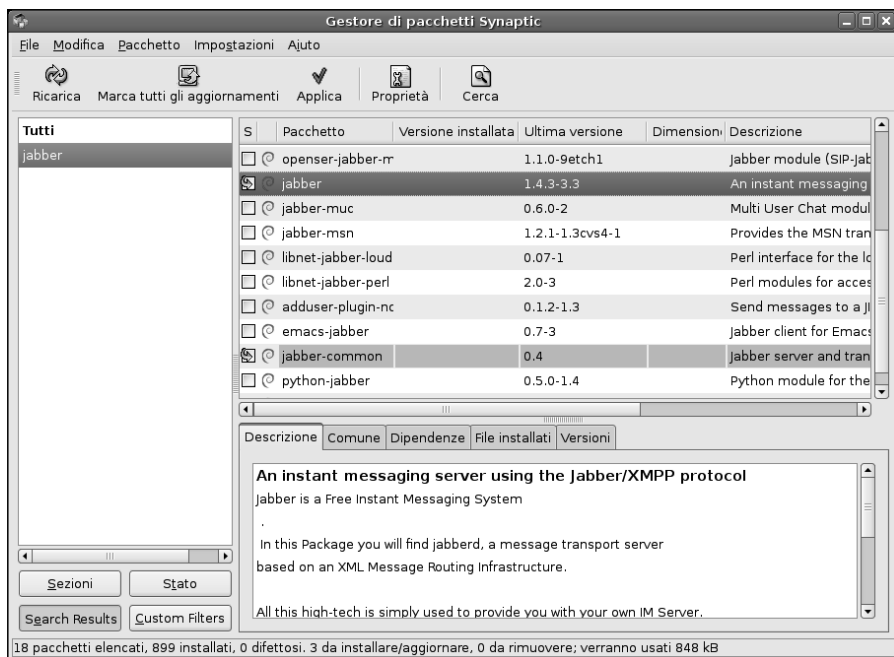



Figura 11.1

I pacchetti Linux che riguardano il server IM Jabber.

L'output del comando, rappresentato nella Figura 11.2, mostra che il daemon di Jabber è già in funzione.

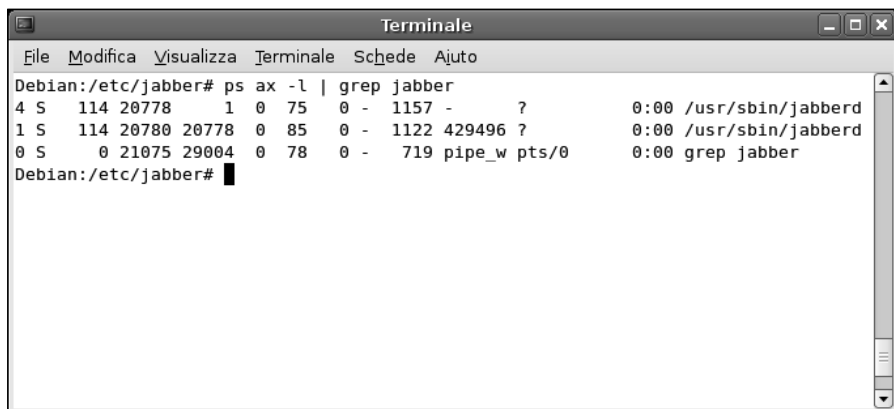


Figura 11.2

Fra i processi in esecuzione nel sistema troviamo già il server Jabber.

Per configurarlo, dobbiamo innanzitutto fermarlo utilizzando il seguente comando:

```
Debian:/etc/jabber# /etc/init.d/jabber stop
Stopping jabberd: jabberd.
Debian:/etc/jabber#
```

A questo punto possiamo operare sui file di configurazione che si trovano nella directory `/etc/jabber`. L'unica operazione che dobbiamo svolgere per configurare adeguatamente Jabber consiste nel definire la direttiva `JABBER_HOSTNAME`.

La configurazione del server avviene utilizzando in particolare il file `/etc/jabber/jabber.xml`, ma sempre nella stessa directory troviamo anche un altro file di configurazione: `jabber.cfg`, all'interno del quale dobbiamo definire il nome che corrisponderà al server.

Dunque dobbiamo aprire il file `/etc/jabber/jabber.cfg` e specificare il nome assegnato alla macchina in rete.

In particolare dobbiamo togliere il segno di commento “#” da una riga già predisposta all'interno di tale file e specificare dopo il segno “=” il nome della macchina. Al termine dell'operazione, il file dovrebbe avere l'aspetto rappresentato nella Figura 11.3.

Questo è tutto quanto è necessario per configurare il server di Jabber. A questo punto possiamo riavviarlo utilizzando il seguente comando:

```
Debian:/etc/jabber# /etc/init.d/jabber start
Starting jabberd: jabberd.
Debian:/etc/jabber#
```

Installazione di un client per Jabber

Ora che abbiamo il server, proviamo a usarlo! Dovremo scaricare e installare un client in grado di connettersi al nuovo server Jabber. Il compagno ideale del server Jabber è il client Gnome Gabber. Richiamiamo ancora Desktop > Amministrazione > Gestore pacchetti Synaptic dal desktop Gnome di Debian, facciamo clic su Cerca e nella finestra di dialogo Trova scriviamo `gabber`; poi facciamo clic sul pulsante Cerca. Verrà visualizzato il solo pacchetto Gabber (Figura 11.4).

Se facciamo clic sulla casella quadrata a lato della voce `gabber` e selezioniamo, come di consueto, l'opzione `Marca per l'installazione`, verrà segnalata una ventina (almeno) di pacchetti necessari per il funzionamento di Gabber (Figura 11.5); facciamo clic sul pulsante `Marca`.

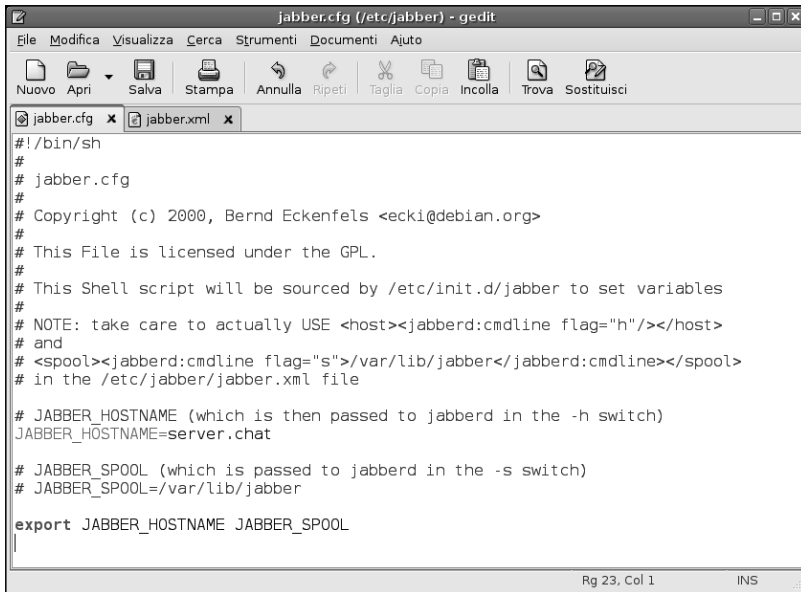


Figura 11.3

L'aspetto del file jabber.cfg dopo la configurazione del nome-host.

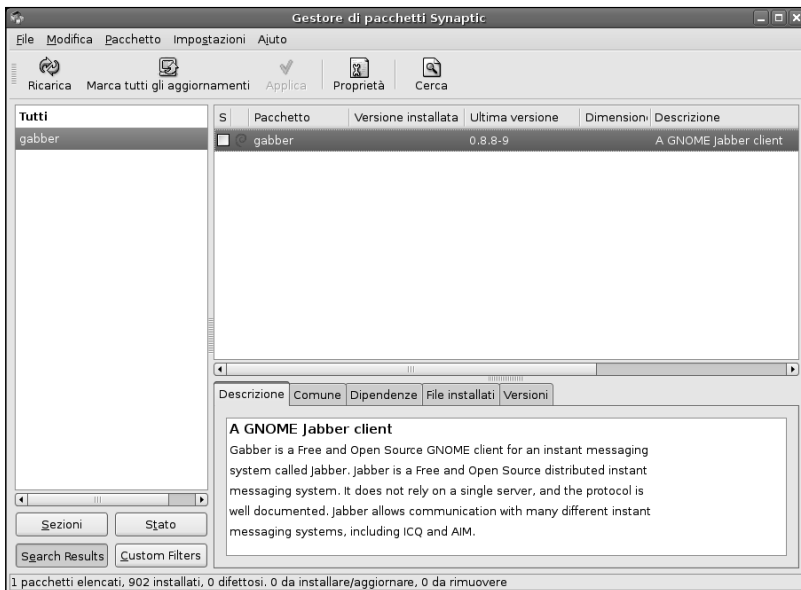
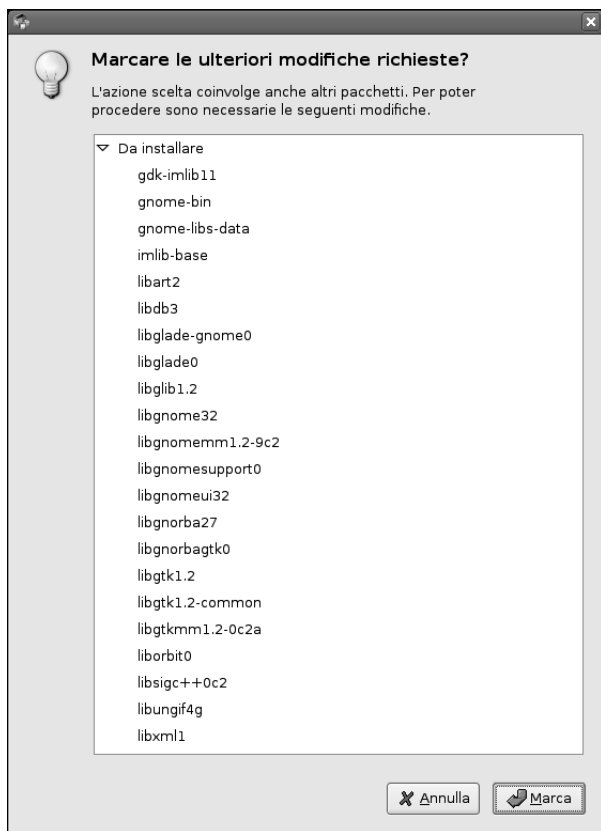


Figura 11.4

Installiamo il client Gabber per Jabber.

**Figura 11.5**

Per il funzionamento di Gabber è necessario installare parecchi altri pacchetti, una ventina circa.

Ora possiamo avviare l'installazione, facendo clic sul pulsante *Applica* nella barra degli strumenti; confermiamo facendo clic sul pulsante *Applica* nella finestra *Riepilogo*. Dopo il download dei pacchetti da Internet verrà eseguita l'installazione del software.

Concludiamo facendo clic sul pulsante *Chiudi* e poi usciamo dal Gestore di pacchetti Synaptic.

Primo avvio e configurazione di Gabber

Subito dopo l'installazione verrà lanciata automaticamente la procedura di configurazione del client Gabber. Se così non dovesse essere, digitiamo *gabber* al prompt; verrà lanciata la procedura di configurazione (Figura 11.6)

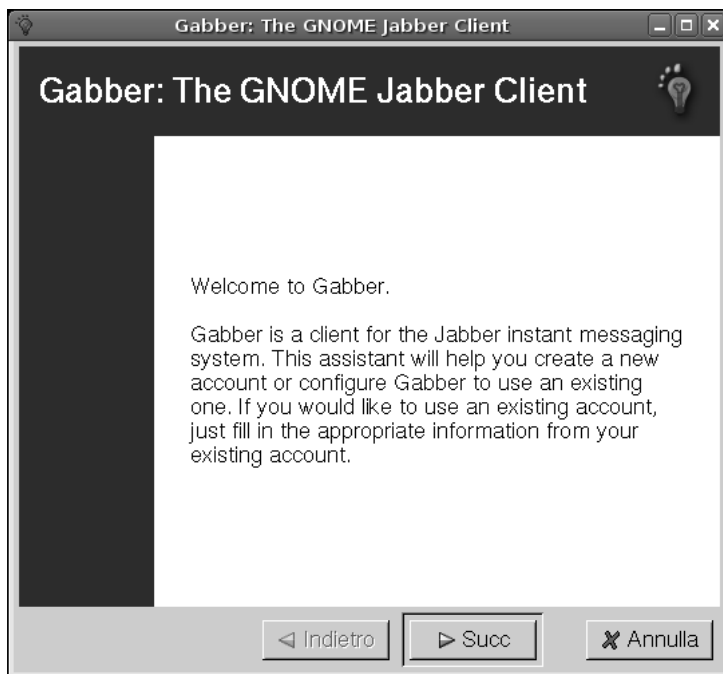


Figura 11.6
Configurazione del client per chat Gabber.

Facciamo clic sul pulsante **Succ** e verrà presentata la pagina di configurazione **Personal Information** (Figura 11.7) costituita da quattro caselle di testo.

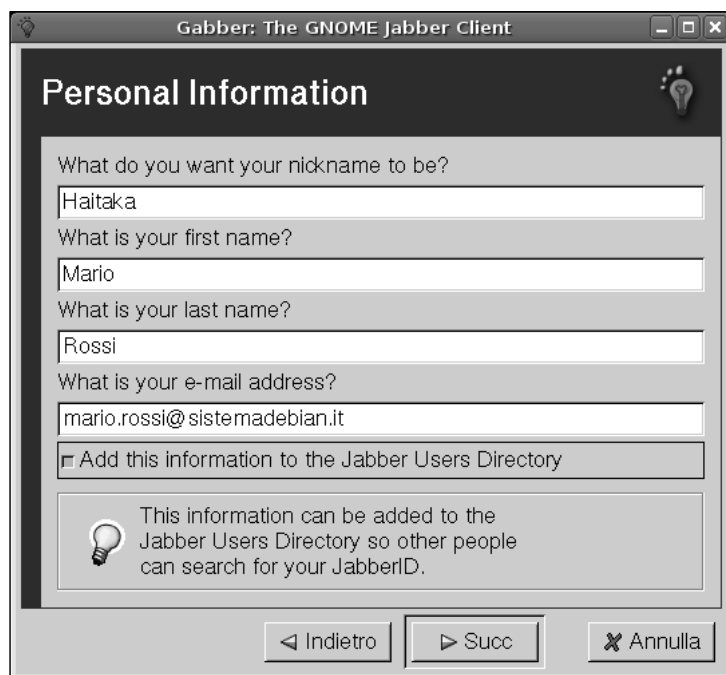
Nella prima dobbiamo specificare il nome con cui vogliamo essere identificati (nickname) nelle chat.

Nella seconda e nella terza casella dobbiamo specificare rispettivamente il nome e il cognome.

Nella quarta casella di testo dobbiamo digitare l'indirizzo di posta elettronica. In un sistema di chat esterno, probabilmente non dovremmo specificare tutte queste informazioni personali o al massimo dovremmo specificare un nome fasullo e un indirizzo email secondario, "sacrificabile". Ma trattandosi di un sistema interno, forse ci conviene scrivere dati effettivi.

La casella finale (Add this information to the Jabber Users Directory) consente di essere contattati tramite una rubrica comune.

Facciamo ancora clic sul pulsante **Succ** e verrà presentata la pagina di configurazione **Jabber account** (Figura 11.8) costituita da tre caselle.



The screenshot shows a window titled "Gabber: The GNOME Jabber Client". Inside, there is a section titled "Personal Information" with a lightbulb icon. The form contains the following fields and options:

- Question: "What do you want your nickname to be?"
Input: "Haitaka"
- Question: "What is your first name?"
Input: "Mario"
- Question: "What is your last name?"
Input: "Rossi"
- Question: "What is your e-mail address?"
Input: "mario.rossi@sistemadebian.it"
- Checkbox: ☒ "Add this information to the Jabber Users Directory"
- Information box with a lightbulb icon: "This information can be added to the Jabber Users Directory so other people can search for your JabberID."
- Navigation buttons at the bottom: "< Indietro", "> Succ", and "X Annulla".

Figura 11.7

La pagina delle informazioni personali per la configurazione del client Gabber.

Nella prima è indicato il nome-utente corrispondente all'account. In questo caso è root perché stiamo utilizzando il sistema come utenti root per svolgere le attività di installazione e configurazione del software.

Nella seconda casella dobbiamo indicare il nome del sistema sul quale abbiamo appena installato Jabber. Trattandosi dello stesso sistema possiamo tranquillamente specificare localhost, ovvero "questa stessa macchina". La terza casella indica la porta standard utilizzata dal server Jabber: la 5222. I pulsanti a freccia a lato di queste ultime due caselle ci consentono anche di utilizzare un altro server Jabber (esterno) e un'altra porta.

Facciamo nuovamente clic sul pulsante Succ e verrà presentata la pagina di configurazione della Password (Figura 11.9) nella quale dobbiamo indicare e ripetere la password e selezionare la casella Save Password per memorizzarla.



Figura 11.8
Configurazione dell'account Jabber.

Ancora un clic sul pulsante Succ e verrà presentata l'ultima pagina di configurazione, Resource, nella quale dobbiamo specificare il nome identificativo di questa risorsa. Possiamo accettare la scelta Gabber o specificare un nome personale. Premendo ancora Succ verrà presentata la pagina di riepilogo Your New JabberID, nella quale troviamo i dati identificativi del server e del nostro ID Jabber. Infine verrà visualizzata la pagina Logging In (Figura 11.10) dove il client effettuerà la connessione con il server Jabber che abbiamo appena impostato. Si aprirà la finestra principale di Gabber (Figura 11.11) dove potremo attivare una sessione di chat. Installiamo Gabber anche sulle altre macchine Linux della rete locale. Sulle macchine Windows della rete possiamo installare un client come Gajim o Coccinella (Figura 11.12).



Figura 11.9
La pagina per l'introduzione della password.

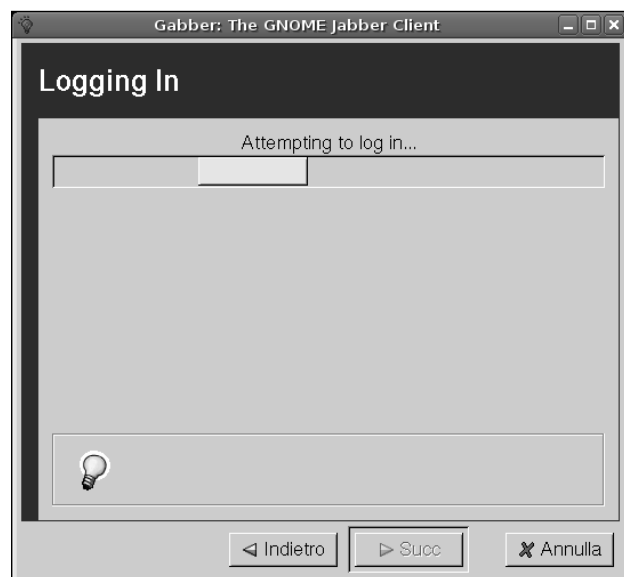


Figura 11.10
La finestra di connessione del client di chat/IM Gabber.

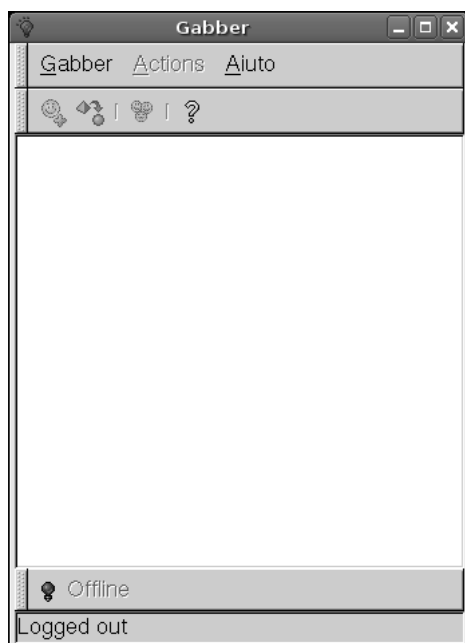


Figura 11.11
L'aspetto del client Gabber

Conclusioni

In questo capitolo abbiamo introdotto brevemente la creazione di un server IM/chat con Jabber per la rete locale aziendale. Una soluzione di questo tipo consente di attivare un sistema di scambio rapido di informazioni interno, per brevi comunicazioni di natura testuale.

Nel prossimo capitolo impareremo a trasformare la nostra macchina Debian in un server di database, con MySQL.

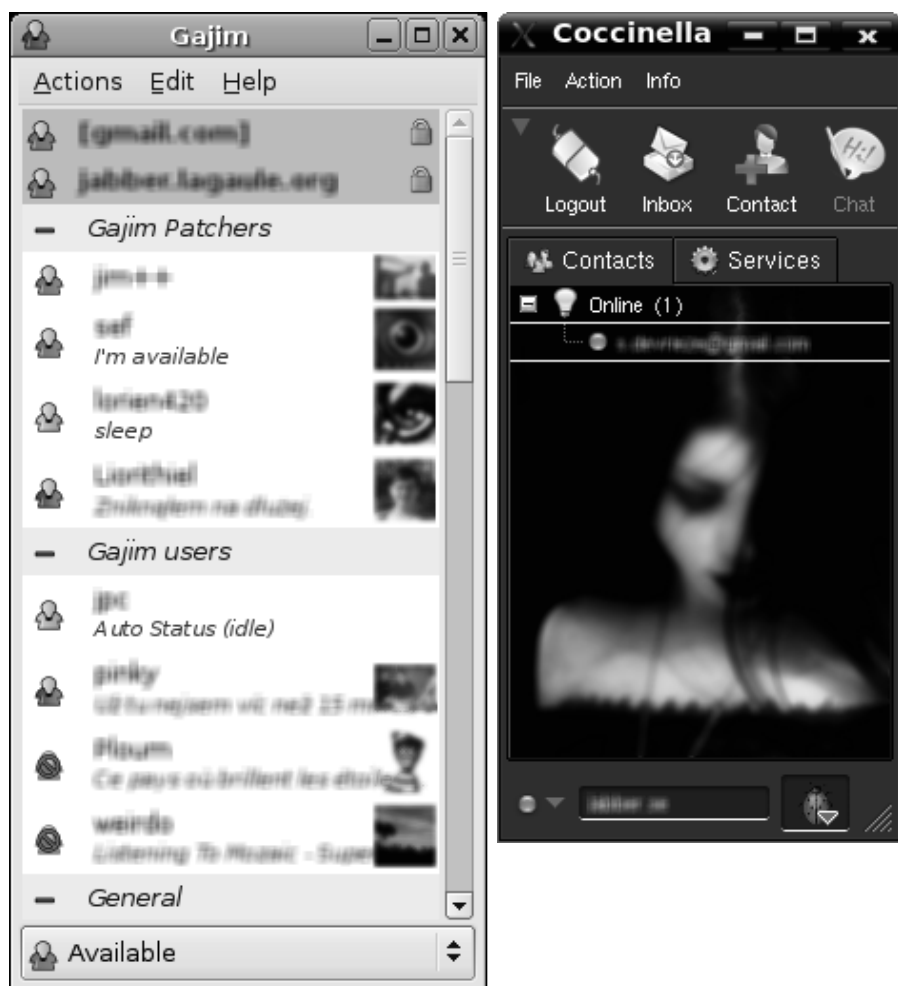


Figura 11.12
I client IM per Windows Gajim e Coccinella.

Capitolo 12

Creare un server per database MySQL

La gestione di database su un sistema Linux non si limita ad applicazioni database “pure”; un database può essere utile anche come supporto per un sito Web e le relative applicazioni.

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ✓ MySQL
- ✓ Installazione di MySQL
- ✓ Configurazione di MySQL
- ✓ Il programma di amministrazione MySQL Monitor
- ✓ Creazione di un database
- ✓ Utilizzo in rete del database
- ✓ Gli utenti del database
- ✓ Installiamo un client per Windows

I database per Linux sono applicazioni ormai molto diffuse: a parte il software commerciale *Oracle*, non bisogna dimenticare che su ogni sistema Linux viene ormai installato il pacchetto open-source *OpenOffice.org* che annovera al suo interno l'applicazione Base, studiata strategicamente per entrare in diretta competizione con Microsoft Access, quanto meno come componente di un grande pacchetto per l'ufficio.

MySQL

Ma il database più noto e gratuito per l'ambiente Linux è certamente *MySQL*, utilizzato sia per normali applicazioni a database sia come strumento di supporto per la realizzazione di siti Web.

A differenza di quanto si può pensare, la casa produttrice di MySQL non è, come sarebbe normale del mondo Linux, una società senza scopi di lucro o, meglio ancora, una comunità di programmatori e utilizzatori sparsa in tutto il mondo.

In altre parole, MySQL non è un software “alla Apache”, ma un prodotto software che la casa produttrice ha deciso di distribuire gratuitamente. Anche MySQL è un prodotto open-source, ma la casa produttrice svedese *MySQL AB* (Figura 12.1), recentemente passata sotto le grandi ali di Sun Microsystems, ha deciso di trarre profitto unicamente da servizi di supporto e dalle versioni più avanzate del database, rivolte alla grande azienda.

Il fatto di fornire gratuitamente il software sembra un controsenso per un'impresa commerciale, ma la realtà è che in questo modo MySQL è diventato il database open-source più diffuso al mondo, impiegato dai siti Web più importanti, come quelli di Yahoo, Alcatel, Google, Nokia e YouTube.



Figura 12.1

La home page (in italiano) di MySQL, dove da qualche tempo campeggia anche il logo Sun Microsystems.

Al di là di questo, MySQL vanta nella sua storia oltre 100 milioni di copie distribuite in tutto il mondo, 11 milioni totali di installazioni nel mondo e un tasso di crescita superiore ai 30.000 download al giorno.

Installazione di MySQL

La procedura di installazione di un software ampio e “pesante” come MySQL può comunque essere svolta in modo grafico, tramite il Gestore di pacchetti Synaptic, che, come di consueto, si preoccupa di segnalare e installare tutte le dipendenze del software.

Richiamiamo Desktop > Amministrazione > Gestore pacchetti Synaptic dal desktop Gnome di Debian e, nella finestra del programma, facciamo clic sull'icona Cerca nella barra degli strumenti. Nella finestra di dialogo Trova scriviamo `mysql-server`. Digitando solamente *mysql* verrebbero trovati troppi risultati. Lasciamo a Synaptic il compito di cercare il pacchetto per noi. Facendo clic sul pulsante Cerca verranno visualizzati i soli pacchetti server di MySQL (Figura 12.2), in pratica solo `mysql-server`, un metapacchetto per l'installazione

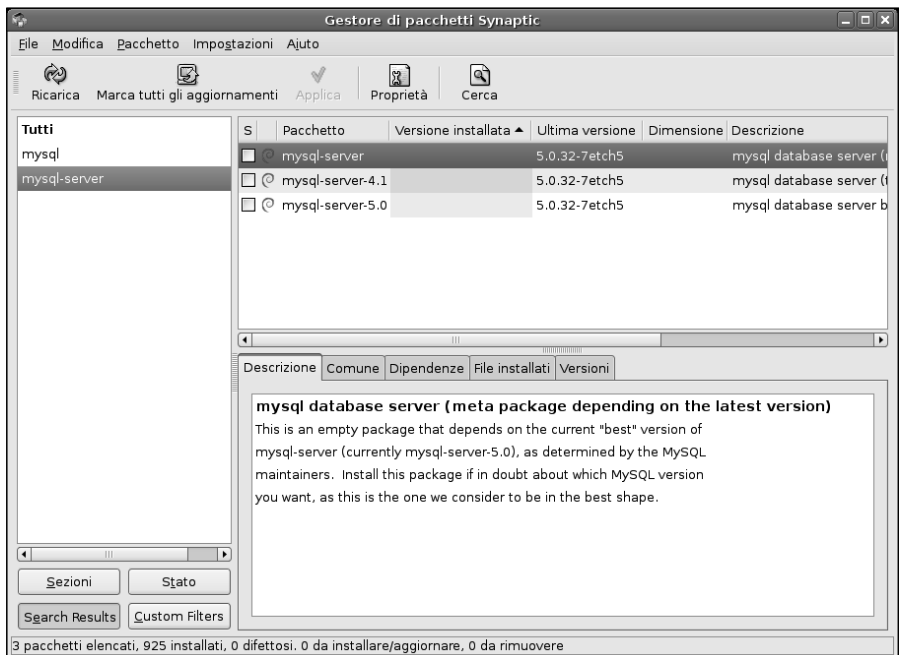


Figura 12.2

I pacchetti Linux relativi al server DB MySQL.

del software (quello che dobbiamo selezionare), `mysql-server-4.1` (una vecchia versione del pacchetto) e `mysql-server-5.0` (il pacchetto dei file binari).

Ora facciamo clic sulla casella quadrata che si trova a lato della voce `mysql-server` e selezioniamo l'opzione **Marca** per l'installazione. Verranno segnalati ben sei altri pacchetti che devono essere installati obbligatoriamente insieme al pacchetto principale (Figura 12.3). Facciamo clic sul pulsante **Marca**.

Ora possiamo avviare l'installazione, facendo clic sul pulsante **Applica** nella barra degli strumenti; confermiamo facendo clic sul pulsante **Applica** nella finestra **Riepilogo**. MySQL non è un'applicazione "leggera": i file da scaricare occupano oltre 30 MB e dunque il download potrebbe richiedere qualche minuto (Figura 12.4) a seconda della velocità della connessione Internet; terminato il download verrà eseguita l'installazione del software (Figura 12.5).

Concludiamo facendo clic sul pulsante **Chiudi** e poi usciamo dal **Gestore di pacchetti Synaptic**.

Terminata l'installazione, il server MySQL viene automaticamente avviato, come possiamo verificare utilizzando il seguente comando:

```
Debian:/etc/jabber# ps ax -l | grep mysql
```

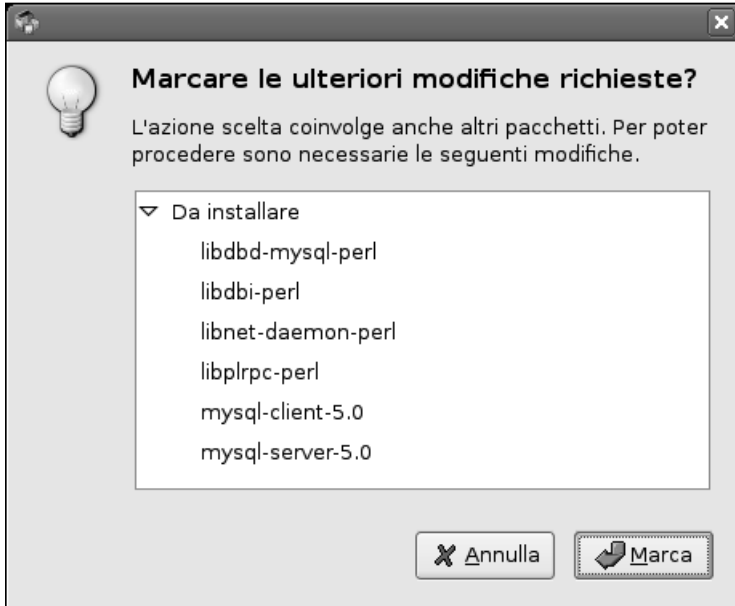
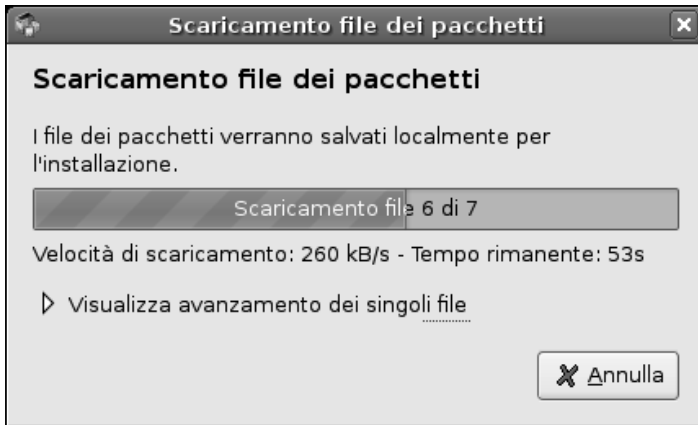
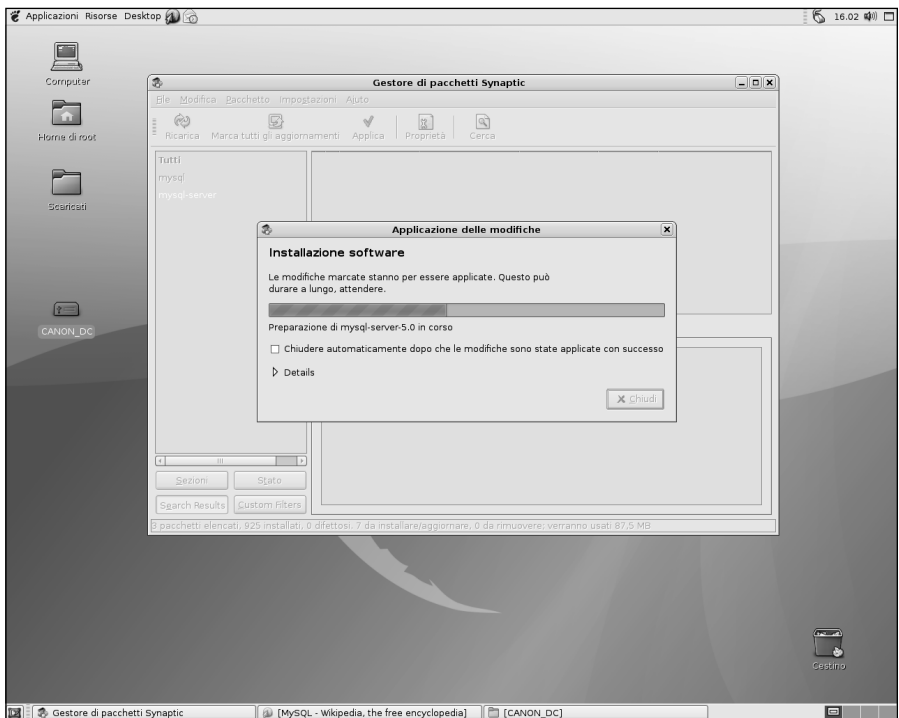


Figura 12.3

I pacchetti Linux che compongono il software MySQL.

**Figura 12.4**

Il download di MySQL può richiedere qualche minuto, a seconda della velocità della connessione Internet e del livello di carico dei server per il download.

**Figura 12.5**

Installazione del software MySQL.

Come possiamo vedere dall'output del comando, rappresentato nella Figura 12.6, Il daemon di MySQL, `mysqld`, è già attivo.

```

Terminale
File Modifica Visualizza Terminale Schede Aiuto
Debian:~# ps ax -l | grep mysql
4 S    0 14328    1 0 82 0 -   918 wait ?           0:00 /bin/sh /usr/bin/mysqld_safe
4 S   115 14440 14328 0 78 0 - 31548 429496 ?       0:00 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=mysql --pid-file=/var/run/mysqld/mysqld.pid --skip-external-locking --port=3306 --socket=/var/run/mysqld/mysqld.sock
0 S    0 14441 14328 0 85 0 -   687 pipe_w ?           0:00 logger -p daemon.err -t mysqld_safe -i -t mysqld
0 S    0 14732 14552 0 78 0 -   719 pipe_w pts/0       0:00 grep mysql
Debian:~#

```

Figura 12.6
Il daemon `mysqld` del server MySQL è già attivo e in attesa di “ordini”.

Configurazione di MySQL

Normalmente non vi è alcuna necessità di configurare in modo particolare MySQL: quello che abbiamo installato ora è solo il lato server di una applicazione, ovvero quella parte che si occuperà di gestire i database per conto dei client per tutte le operazioni di inserimento, manipolazione e ricerca delle informazioni. Tuttavia vi sono alcuni elementi della configurazione sui quali sarebbe utile intervenire. Come di consueto, il file di configurazione di MySQL si trova nella directory `/etc/mysql` (Figura 12.7). Dobbiamo intervenire solo sul file `my.cnf`.

Nella sezione Basic Settings del file `my.cnf`, troviamo l'opzione `datadir` che indica la directory in cui MySQL inserirà i database.

L'impostazione predefinita prevede che i database vengano collocati nella directory `/var/lib/mysql`.

Al momento dell'installazione, MySQL è impostato sulla lingua inglese.

Possiamo verificarlo controllando la voce `language` all'interno del file `my.cnf`.

Nella directory `/usr/share/mysql` troviamo varie subdirectory corrispondenti a tutte le principali lingue. Cambiamo questa opzione e al posto di `english` specifichiamo `italian`.



Figura 12.7

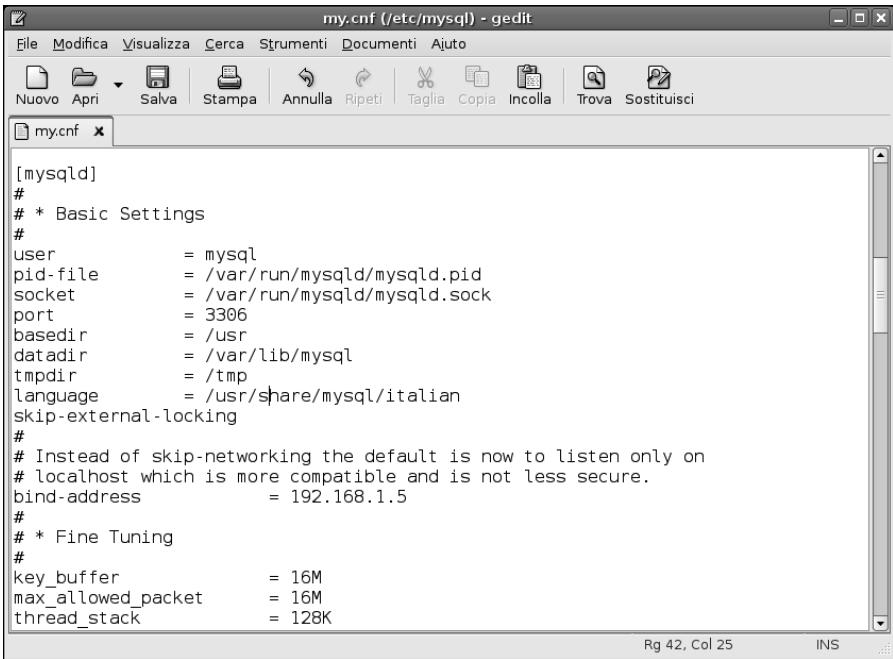
Tra tutti i file presenti nella directory `mysql` dobbiamo intervenire solo su `my.cnf`. L'altro file, `debian.cnf` non deve essere toccato per nessun motivo.

Per facilitare le connessioni con il database da parte dei client operanti su altre macchine della rete, possiamo trasformare l'opzione `bind-address`, che si trova appena sotto la sezione `Basic Settings`, sostituendo l'indirizzo `localhost` (`127.0.0.0`) con l'effettivo indirizzo utilizzato dalla macchina in rete, concesso dal server DNS (in questo caso `192.168.1.5`, ma su altre reti tale indirizzo sarà ovviamente differente).

La Figura 12.8 mostra l'aspetto del file `my.cnf` dopo queste tre modifiche. Ora dobbiamo riavviare MySQL utilizzando il seguente comando:

```
Debian:# /etc/init.d/mysql restart
Stopping MySQL database server: mysqld.
Starting MySQL database server: mysqld.
Checking for corrupt, not cleanly closed and upgrade needing tables...
Debian:#
```

Il messaggio finale non rappresenta un problema, in quanto non abbiamo ancora creato alcun database.

**Figura 12.8**

Adeguiamo la configurazione di MySQL a un ambiente di rete e alla lingua italiana.

Il programma di amministrazione MySQL Monitor

Anche MySQL, come Linux, deve controllare gli accessi al database per garantire il livello di sicurezza necessario. Così come non tutti possono avere accesso a un sistema, tanto meno con privilegi root, perché ciò metterebbe in grado chiunque di alterare e anche distruggere l'intero sistema, così MySQL deve proteggere il database tramite account e password, per evitare che qualcuno possa alterare o cancellare i dati del database senza averne l'autorità.

All'inizio MySQL prevede un unico account “plenipotenziario” il cui nome è, intuitivamente ma incidentalmente, root. Non si tratta dello stesso root che ha accesso al sistema operativo, ma di un account root specifico per MySQL. Inizialmente tale account non ha alcuna password.

Come abbiamo visto, al momento MySQL è attivo. Questo ci consente di richiamare il suo programma di amministrazione **MySQL Monitor** che consente di

gestire i database e le relative tabelle. Per richiamare il programma di amministrazione si utilizza seguente comando:

```
mysql -u root
```

Verrà presentato il messaggio di conferma “Welcome to MySQL Monitor” e il tipico prompt di Debian cambierà in:

```
mysql>
```

con un cursore lampeggiante in attesa di comandi.

Il primo comando, il più ovvio, è `help`, che chiede la visualizzazione dell’elenco dei comandi disponibili.

La Figura 12.9 mostra l’avvio della sessione di MySQL Monitor e l’elenco dei comandi disponibili

Questi non sono gli unici comandi accettati in questo ambiente operativo di MySQL. Qui infatti si possono utilizzare tutti i comuni comandi SQL impiegabili per creare ed elaborare un database.

DA SAPERE *I comandi SQL sono normalmente piuttosto lunghi. Possiamo inserirli anche su più righe. Per indicare la fine di un comando dobbiamo utilizzare il segno “;”.*

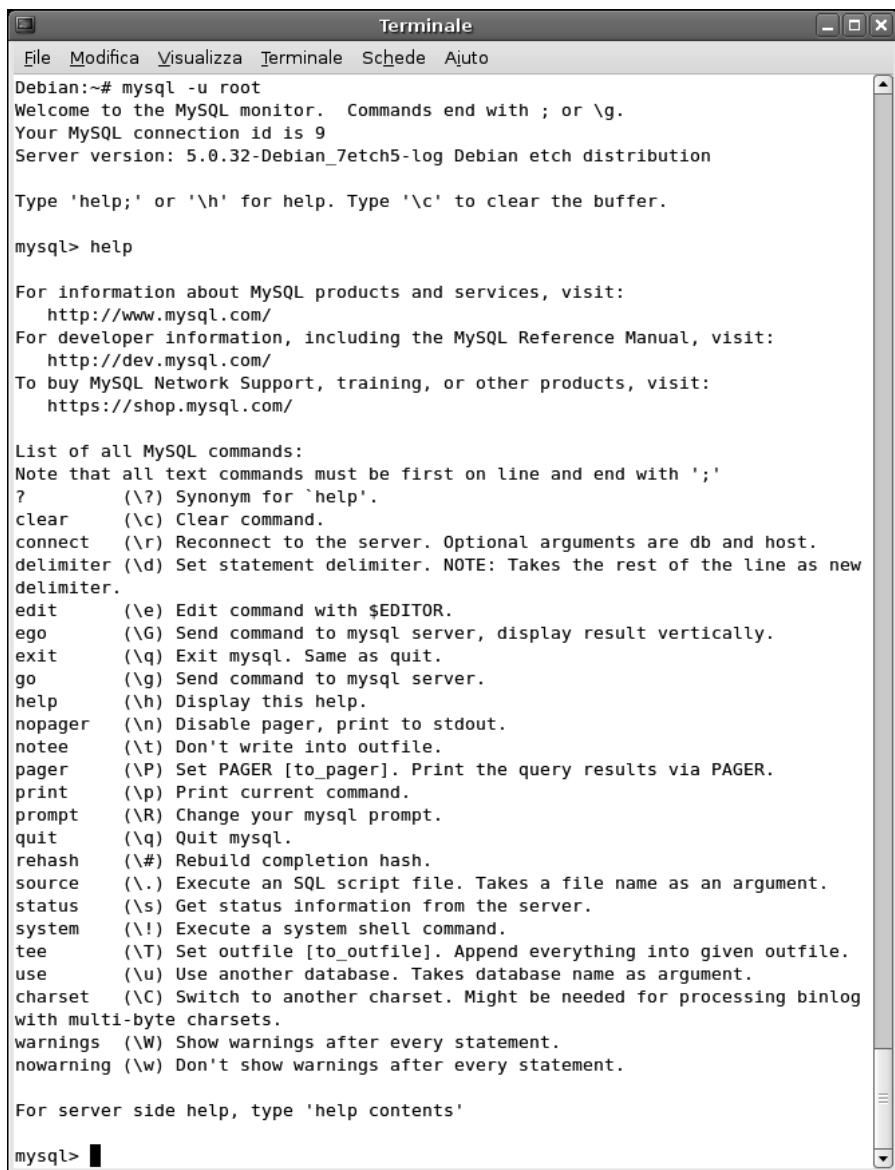


Creazione di un database

Per fare un esempio, proveremo a creare un semplice database costituito da due tabelle per memorizzare una semplice rubrica telefonica. Tale database potrà essere poi collegato a una pagina Web che consenta agli utenti interni della rete dei consultare la rubrica telefonica comodamente, utilizzando i servizi del server Web Apache attivo nella intranet, che abbiamo predisposto nei primi capitoli di questo libro. Innanzitutto possiamo dunque creare il database utilizzando il seguente comando:

```
mysql> create database Rubrica;
```

In questo modo abbiamo creato un database vuoto; l’operazione sarà confermata da un apposito messaggio “Query OK”. Il database si chiama Rubrica (attenzione, con l’iniziale maiuscola).



```
Terminale
File Modifica Visualizza Terminale Schede Ajuto
Debian:~# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.0.32-Debian_7etch5-log Debian etch distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> help

For information about MySQL products and services, visit:
  http://www.mysql.com/
For developer information, including the MySQL Reference Manual, visit:
  http://dev.mysql.com/
To buy MySQL Network Support, training, or other products, visit:
  https://shop.mysql.com/

List of all MySQL commands:
Note that all text commands must be first on line and end with ';'
?          (\?) Synonym for 'help'.
clear      (\c) Clear command.
connect    (\r) Reconnect to the server. Optional arguments are db and host.
delimiter  (\d) Set statement delimiter. NOTE: Takes the rest of the line as new
delimiter.
edit       (\e) Edit command with $EDITOR.
ego        (\G) Send command to mysql server, display result vertically.
exit       (\q) Exit mysql. Same as quit.
go         (\g) Send command to mysql server.
help       (\h) Display this help.
nopager    (\n) Disable pager, print to stdout.
notee      (\t) Don't write into outfile.
pager      (\P) Set PAGER [to_pager]. Print the query results via PAGER.
print      (\p) Print current command.
prompt     (\R) Change your mysql prompt.
quit       (\q) Quit mysql.
rehash     (\#) Rebuild completion hash.
source     (\.) Execute an SQL script file. Takes a file name as an argument.
status     (\s) Get status information from the server.
system     (\!) Execute a system shell command.
tee        (\T) Set outfile [to_outfile]. Append everything into given outfile.
use        (\u) Use another database. Takes database name as argument.
charset    (\C) Switch to another charset. Might be needed for processing binlog
with multi-byte charsets.
warnings   (\W) Show warnings after every statement.
nowarning   (\w) Don't show warnings after every statement.

For server side help, type 'help contents'

mysql> █
```

Figura 12.9

Abbiamo fatto accesso al programma di amministrazione di MySQL e abbiamo visualizzato l'elenco dei comandi con `help`.

Il database esiste ma non lo stiamo ancora utilizzando. Poco male, basta introdurre il comando:

```
mysql> use Rubrica;
```

Come abbiamo detto, il database ora è vuoto e dovremmo crearvi delle tabelle. La prima tabella potrebbe chiamarsi `NumeriInterni`. Per fare ciò si utilizza il comando `create table` che richiede il nome e la definizione del contenuto della tabella. Si tratta dunque di un comando che si estende naturalmente su più righe. Dunque dovremo scrivere ogni riga senza mai terminarla con un “;”. Potremo inserire il “;” di chiusura del comando al termine, su una riga a parte. Ecco dunque il comando necessario per creare la nuova tabella:

```
mysql> create table NumeriInterni (
```

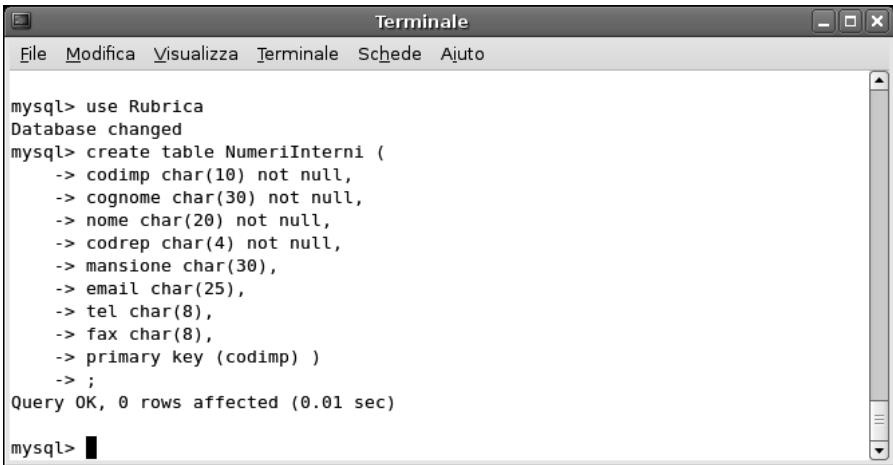
Il prompt si trasformerà in una freccia, rientrata alla stessa altezza del prompt precedente (Figura 12.10). A questo prompt, introduciamo una per una le seguenti righe; al termine del comando, possiamo inserire il “;” di chiusura.

```
-> codimp char(10) not null,  
-> cognome char(30) not null,  
-> nome char(20) not null,  
-> codrep char(4) not null,  
-> mansione char(30),  
-> email char(25),  
-> tel char(8),  
-> fax char(8),  
-> primary key(codimp) )  
-> ;
```

DA SAPERE *Nell’inserimento di comandi questo tipo è facile commettere errori. A volte basta dimenticare un “;” o una parentesi. In tal caso il messaggio verrà rifiutato con un errore, ma avremo la possibilità di correggerlo e reintrodurlo: premendo FRECCIA SU, il comando verrà però presentato su un’unica lunga riga che, se necessario (e quasi sempre lo è) verrà mandata a capo più volte.*



La penultima riga del comando nomina la chiave primaria della tabella. In tale riga si trovano due parentesi chiuse, in quanto la seconda chiude quella che abbiamo aperto sulla prima riga del comando (`create table NumeriInterni`).



```

Terminale
File Modifica Visualizza Terminale Schede Ajuto

mysql> use Rubrica
Database changed
mysql> create table NumeriInterni (
  -> codimp char(10) not null,
  -> cognome char(30) not null,
  -> nome char(20) not null,
  -> codrep char(4) not null,
  -> mansione char(30),
  -> email char(25),
  -> tel char(8),
  -> fax char(8),
  -> primary key (codimp) )
  -> ;
Query OK, 0 rows affected (0.01 sec)

mysql> █

```

Figura 12.10

Abbiamo creato una nuova tabella e MySQL ha confermato l'operazione con la risposta "Query OK".

Per rendere più interessante il database creiamo anche una seconda tabella legata alla prima che elenca i reparti dell'azienda. Ecco il comando da utilizzare:

```

mysql> create table Reparti (
  -> codrep char(4) not null,
  -> nomerep char(30) not null,
  -> indirizzo char(40),
  -> citta char(25),
  -> provincia char(2),
  -> cap char(5),
  -> prefisso char(4),
  -> responsabile char(8),
  -> primary key(codrep) )
  -> ;

```

Abbiamo così creato una seconda tabella `Reparti` che utilizza come chiave primaria il campo `codrep`. Le due tabelle hanno un campo comune che le mette in relazione l'una con l'altra; si tratta proprio del campo `codrep`, che lega un record di `NumeriInterni` alle informazioni contenute nei campi di un record `Reparti`. Per vedere l'aspetto del nostro database a questo punto, ovvero dopo avere inserito due tabelle, utilizziamo seguente comando:

```
mysql> show tables;
```


Verrà visualizzato lo schemino rappresentato nella Figura 12.11, semplice ma efficace: il database Rubrica contiene due tabelle: NumeriInterni e Reparti. Finora tutto bene.

Possiamo anche osservare l'aspetto delle due tabelle, ovvero i campi che vi abbiamo creato. A tale scopo possiamo utilizzare i seguenti comandi:

```
mysql> show columns from NumeriInterni;  
mysql> show columns from Reparti;
```

Il risultato è rappresentato nella Figura 12.12.



```
Terminale
File Modifica Visualizza Terminale Schede Ajuto

mysql> create table NumeriInterni (
-> codimp char(10) not null,
-> cognome char(30) not null,
-> nome char(20) not null,
-> codrep char(4) not null,
-> mansione char(30),
-> email char(25),
-> tel char(8),
-> fax char(8),
-> primary key (codimp) )
-> ;
Query OK, 0 rows affected (0.01 sec)

mysql> create table Reparti (
-> codrep char(4) not null,
-> nomerep char(30) not null,
-> indirizzo char(40),
-> citta char(25),
-> provincia char(2),
-> cap char(5),
-> prefisso char(4),
-> responsabile char(8),
-> primary key(codrep) )
-> ;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-----+
| Tables_in_Rubrica |
+-----+
| NumeriInterni      |
| Reparti            |
+-----+
2 rows in set (0.00 sec)

mysql>
```

Figura 12.11

Abbiamo creato la struttura delle due tabelle all'interno del database Rubrica.

The screenshot shows a MySQL terminal window with the following content:

```

mysql> show columns from NumeriInterni;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| codimp | char(10) | NO | PRI | | |
| cognome | char(30) | NO | | | |
| nome | char(20) | NO | | | |
| codrep | char(4) | NO | | | |
| mansione | char(30) | YES | | NULL | |
| email | char(25) | YES | | NULL | |
| tel | char(8) | YES | | NULL | |
| fax | char(8) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> show columns from Reparti;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| codrep | char(4) | NO | PRI | | |
| nomerep | char(30) | NO | | | |
| indirizzo | char(40) | YES | | NULL | |
| citta | char(25) | YES | | NULL | |
| provincia | char(2) | YES | | NULL | |
| cap | char(5) | YES | | NULL | |
| prefisso | char(4) | YES | | NULL | |
| responsabile | char(8) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>

```

Figura 12.12

L'aspetto delle due tabelle che abbiamo creato nel database Rubrica, con tutti i loro campi.

Ora possiamo inserire dei dati all'interno del nostro nuovo database; prima nella tabella NumeriInterni:

```

mysql> insert into NumeriInterni (
-> codimp, cognome, nome, codrep, mansione, email, tel, fax)
-> values ("123456", "Rossi", "Mario", "1234", "Tecnico",
-> "mario.rossi@esempio.it", "1234.567", "1234.568")
-> ;

```

e poi nella tabella Reparti:

```

mysql> insert into Reparti (
-> codrep, nomerep, indirizzo, citta, provincia, cap, prefisso)
-> values ("1234", "Tecnici", "via Garibaldi, 123", "Milano", "MI",

```



```
-> "20100", "02")
-> ;
```

MySQL risponderà con la classica conferma “Query OK”.

DA SAPERE *Non è necessario specificare i valori per i campi non obbligatori, quelli per cui non abbiamo specificato in fase di definizione la parte “not null”. Nel primo caso, per esempio, abbiamo specificato tutti i campi, mentre nel secondo caso abbiamo tralasciato il campo “responsabile” che, come vedremo, sarà nullo.*



A questo punto vogliamo vedere il contenuto del nostro piccolo database. A tale scopo utilizziamo i seguenti comandi:

```
mysql> select * from NumeriInterni;
mysql> select * from Reparti;
```

Il risultato mostra i record che abbiamo inserito nelle due tabelle. Possiamo notare che la colonna “responsabile” della seconda tabella contiene il valore NULL, ma questo non pregiudica il funzionamento del database.

Utilizzo in rete del database

Finora abbiamo messo all’opera i due componenti del database MySQL (il server e il client).

Il server sta operando in modo invisibile. Il programma di amministrazione MySQL Monitor che abbiamo utilizzato per introdurre comandi è invece un client che crea, manipola e interroga il database, gestito dal server.

Dunque sulla stessa macchina stanno operando due diverse entità: il server si occupa, giustamente, di “servire” i dati, mentre il client si frappone fra noi e il server e consente invece di inviare le nostre richieste. Non è affatto necessario che il client e il server si trovino sulla stessa macchina.

MySQL è in grado di gestire interrogazioni multiple sugli stessi database, bloccando al livello minimo possibile i dati in esso contenuti.

Per rendere la cosa un po’ più varia, proveremo a utilizzare un client situato su una macchina Windows; ma utilizzando un altro sistema Linux o Macintosh o un qualsiasi altro tipo di macchina supportata da MySQL (e sono molte), ci troveremmo a eseguire operazioni certamente diverse ma sostanzialmente analoghe per la connessione al nostro database centralizzato. Per prima cosa dobbiamo però considerare un piccolo dettaglio riguardante gli accessi.

Gli utenti del database

MySQL riconosce gli utenti per nome e per password, ma anche sulla base del sistema da cui si connettono. Così come è impostato adesso, il server può essere utilizzato solo localmente, ovvero sulla macchina in cui è installato. Per liberalizzare gli accessi dobbiamo comunicare a MySQL che vogliamo consentire l'accesso anche se gli utenti (per il momento l'utente è uno solo, `root`) si connettono da altre macchine.

Dunque al momento abbiamo operato come utenti `root@localhost`.

Il seguente comando chiede a MySQL di accettare le connessioni dell'utente `root` da qualsiasi macchina egli si connetta:

```
grant all on *.* to root@"%";
```

In una frase, il comando ha il seguente significato: “Concedi (`grant`) ogni permesso (`all`) su (`on`) ogni tabella di ogni database (`*.*`) all'utente (`to`) `root` quando si collega (`@`) da qualsiasi macchina (`"%"`)”. Un comando più semplice di quello che sembrava a prima vista.

Ora dobbiamo vedere se è veramente possibile connettersi al database utilizzando una macchina qualsiasi.

Installiamo un client per Windows

Basta una semplice ricerca in Internet per trovare un client MySQL che renda più comodo l'utilizzo del database, senza dover ricorrere alla digitazione di comandi in MySQL e, comunque, impiegando un'interfaccia grafica.

Per esempio in questo caso utilizzeremo il programma *SQL-Front*, scaricabile liberamente all'indirizzo <http://www.sql-front.com> (Figura 12.13).

Nella pagina *Download* del sito abbiamo la possibilità di scaricare l'ultima versione del client anche da un sito italiano. Seguiamo la procedura di download e poi lanciamo il programma utilizzando i comandi offerti dal browser. In questo caso, utilizzando Mozilla Firefox, ci basta fare clic sul comando *Apri* della finestra *Download*.



DA SAPERE *Notoriamente, Windows è un ambiente più pericoloso di Linux e pertanto quasi sempre il browser o il sistema operativo cercheranno di proteggerci contro l'esecuzione di file eseguibili potenzialmente pericolosi. Possiamo proseguire facendo clic su *Avanti* o *OK*, a seconda del browser impiegato.*

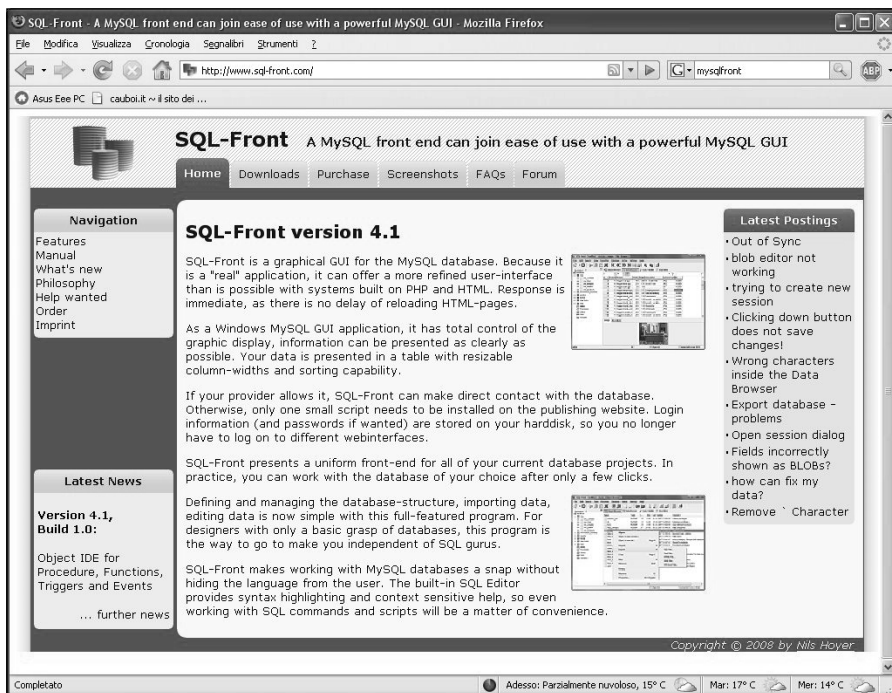


Figura 12.13
Scarichiamo SQL-Front, un client grafico per Windows.

Nella Figura 12.14, possiamo vedere sia i server da cui possiamo scaricare il front-end (compreso quello italiano) sia la finestra di download dopo che è stato completato lo scaricamento del file.

Si aprirà la pagina iniziale della procedura di installazione di SQL-Front, rappresentata nella Figura 12.15. Qui dobbiamo semplicemente fare clic su **Avanti** per proseguire nell'installazione.

Nella seconda pagina, **Selezione della cartella di installazione**, dobbiamo specificare nella casella di testo la cartella nella quale vogliamo installare il front-end. Quella proposta (C:\Programmi\SQL-Front) è adattissima allo scopo e dunque possiamo fare clic sul pulsante **Avanti**.

Analogamente, nella pagina **Selezione della cartella del Menu Avvio/Start**, ci viene proposta la creazione di una nuova voce nel menu, il cui nome sarà, giustamente, SQL-Front. Anche qui possiamo fare clic sul pulsante **Avanti**.

La pagina **Selezione processi aggiuntivi** consente di stabilire dei legami fra il front-end MySQL e l'ambiente operativo Windows. L'opzione **Desktop Icon**

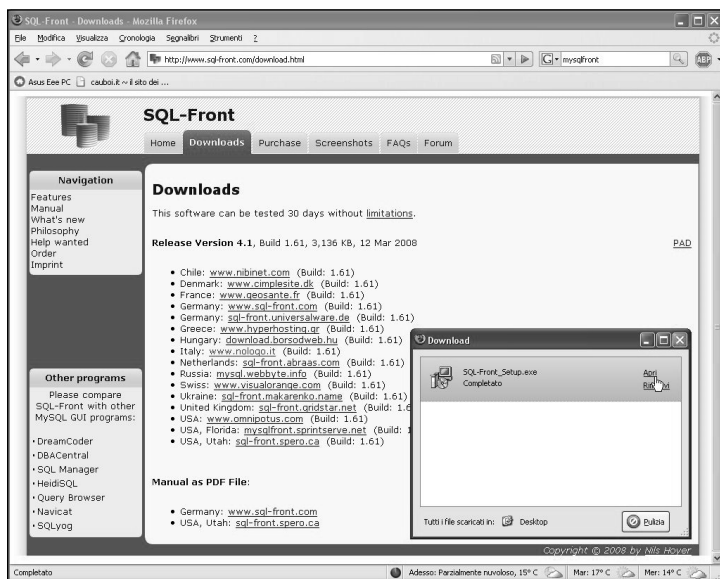


Figura 12.14

Lanciamo la procedura di installazione del front-end grafico per MySQL.



Figura 12.15

Parte la procedura di installazione del client per MySQL.

crea un'icona sul desktop. Non necessariamente questa può essere una soluzione appropriata, specialmente se il desktop è già congestionato dalle troppe icone delle applicazioni che vi abbiamo installato.

L'opzione Quick Launch Icon crea invece un'icona nella barra Avvio veloce, che si trova in prossimità del pulsante Start. Questa forse è una collocazione migliore per il front-end MySQL.

Le ultime due opzioni, Associate .sql (SQL File) with MySQL-Front e Associate mysql://protocol with SQL-Front sono normalmente selezionate e consentono di associare al programma che stiamo installando e il protocollo utilizzato per la connessione al database MySQL.

Queste opzioni sono rappresentate nella Figura 12.16.

Segue la pagina finale di riepilogo, il cui nome è Pronto per l'installazione. Al posto del pulsante Avanti troveremo il pulsante Installa che darà avvio all'installazione del programma (Figura 12.17).

L'intera procedura di installazione durerà pochi istanti; in fin dei conti si tratta di un'interfaccia grafica con il database e non del vero database. Al termine verrà presentata la pagina Completamento dell'installazione di SQL-Front che, con la casella Launch SQL-Front (normalmente selezionata), ci offre la possibilità di



Figura 12.16

Le opzioni aggiuntive che è opportuno configurare per personalizzare il funzionamento di SQL-Front.

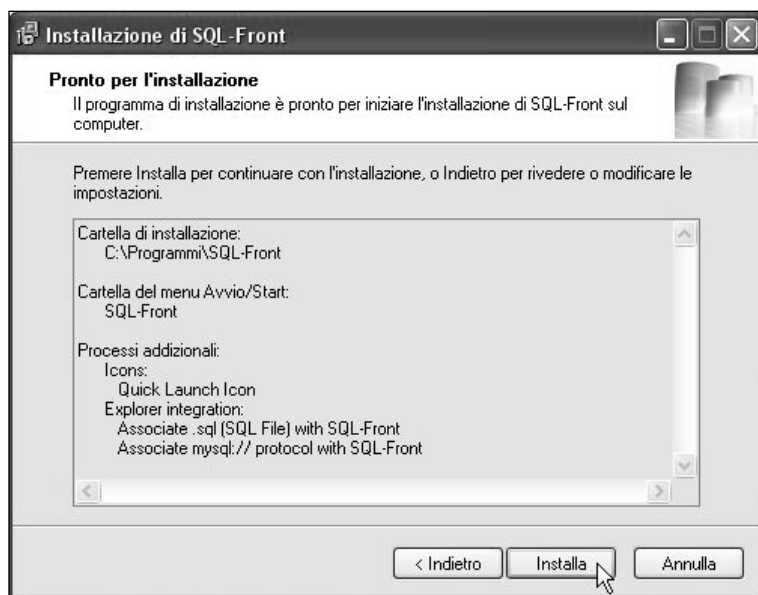


Figura 12.17
La pagina di riepilogo delle opzioni di installazione.

lanciare immediatamente il programma. Facciamo clic sul pulsante Fine per chiudere l'installazione e avviare SQL-Front (Figura 12.18).

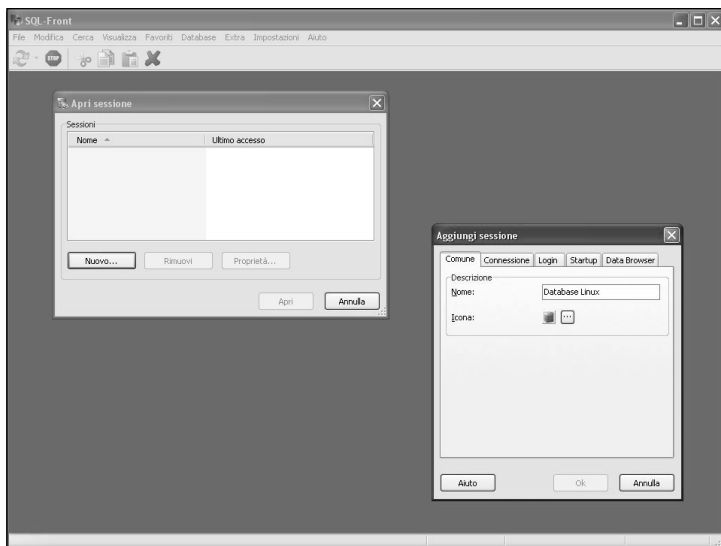
Si aprirà l'interfaccia di SQL-Front, rappresentata nella Figura 12.19; per prima cosa verrà richiamata la finestra di dialogo **Aggiungi sessione**, aperta sulla scheda **Comune**. Qui, nella casella **Nome** possiamo specificare il nome che intendiamo attribuire alla connessione con il database che abbiamo installato precedentemente sulla macchina Debian Linux. In questo caso abbiamo specificato semplicemente il nome **Database Linux**.

Nella pagina **Connessione** della finestra di dialogo **Aggiungi sessione** dobbiamo specificare nella casella di testo **Server** l'indirizzo del server MySQL. Nel nostro caso si tratta della macchina che risponde all'indirizzo 192.168.1.5. Tutti gli altri parametri sono corretti: occorre utilizzare la porta 3306 con una connessione diretta; il Timeout di 30 secondi sembra appropriato, la compressione è attiva e il set di caratteri avrà l'impostazione automatica. L'aspetto di questa scheda della finestra di dialogo **Aggiungi sessione** è rappresentato nella Figura 12.20.

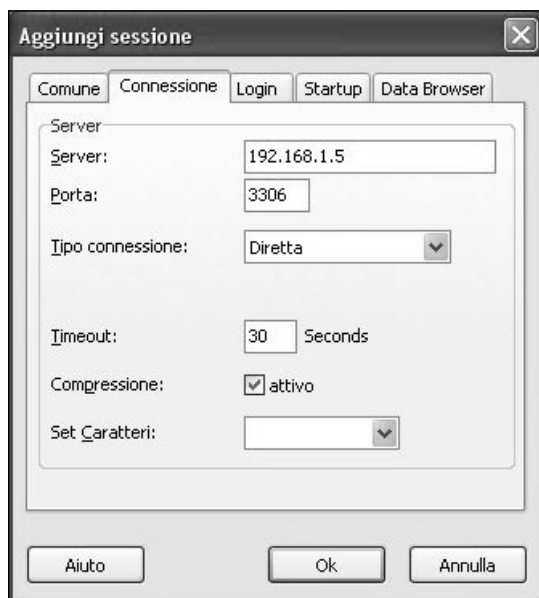
Passati alla scheda **Login** dobbiamo specificare il nome utente e la password più, opzionalmente, il nome del database cui vogliamo connetterci. Come sappiamo, l'unico utente attualmente autorizzato a connettersi è root; la sua pas-

**Figura 12.18**

È finita la procedura di installazione e lanciamo il front-end per MySQL sul sistema Windows.

**Figura 12.19**

La finestra Aggiungi sessione consente di definire i parametri di connessione con il database MySQL sulla macchina Linux.

**Figura 12.20**

Ora dobbiamo impostare i parametri di connessione con il database.

sword non è ancora stata impostata e dunque deve rimanere (per il momento) vuota.

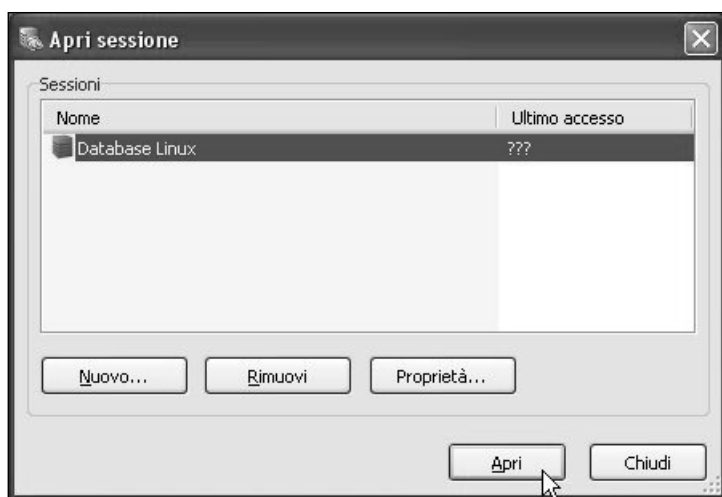
La Figura 12.21 mostra l'aspetto della scheda Login della finestra di dialogo Aggiungi sessione.

Le ultime due schede della finestra di dialogo Aggiungi sessione (Startup e Data Browser) non contengono opzioni utili per la connessione e dunque a questo punto possiamo fare clic sul pulsante OK. Abbiamo così creato una nuova sessione, che comparirà nella finestra di dialogo Apri sessione, rappresentata nella Figura 12.22. Per connetterci al database non ci rimane che selezionare la sessione che abbiamo appena creato e fare clic sul pulsante Apri.

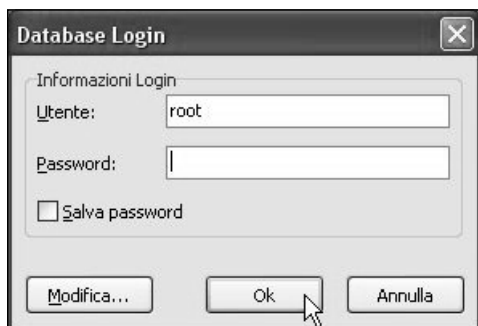
Facendo clic sul pulsante Apri viene richiamata la piccola finestra di connessione Database Login (Figura 12. 23) dove dobbiamo specificare il nome utente e la password. Il nome utente è preimpostato, correttamente, su root, in base alle specifiche con cui abbiamo creato questa sessione . Ricordiamoci che, a questo punto, la password non è ancora impostata e dunque non dobbiamo specificare nulla nella casella di testo Password. Facendo clic sul pulsante OK dovrebbe essere aperta la connessione con il database Linux.

**Figura 12.21**

In questa pagina dobbiamo specificare il nome utente e l'eventuale password per la connessione.

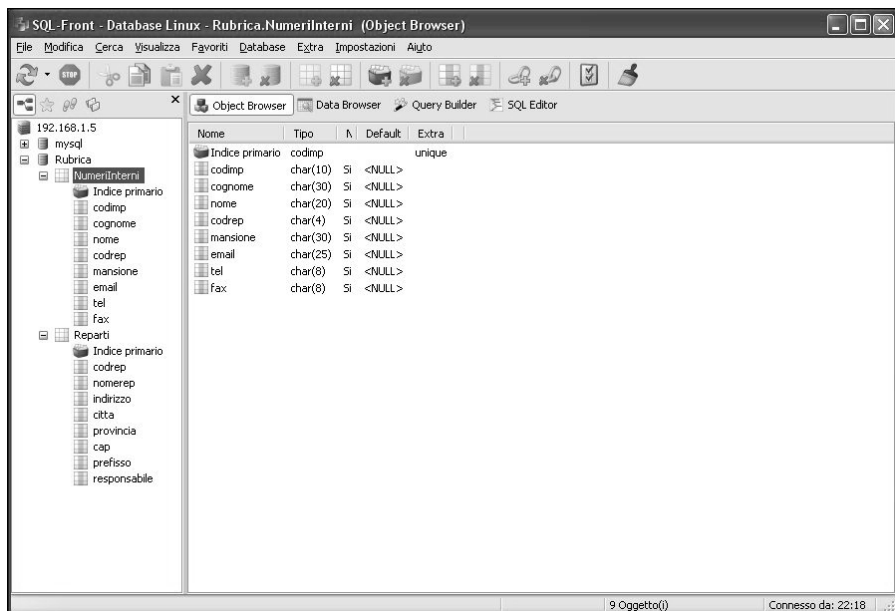
**Figura 12.22**

Siamo a un passo dalla connessione con il database MySQL sulla macchina Linux.

**Figura 12.23**

La finestra di connessione Database Login ci dovrebbe dare accesso al database.

In effetti la connessione funziona: la Figura 12.24 mostra ciò che compare subito dopo la connessione: sul lato sinistro della finestra di SQL-Front troviamo, sotto l'indirizzo della macchina Debian Linux, due database: il database di base mysql e il database Rubrica che abbiamo creato all'inizio di questo capitolo.

**Figura 12.24**

La modalità di visualizzazione degli oggetti (database e tabelle) di SQL-Front.

Abbiamo espanso questo database per mostrare le sue tabelle (NumeriInterni e Reparti) e i campi che abbiamo predisposto all'interno di queste tabelle.

La modalità in cui stiamo osservando i dati ci mostra gli oggetti. Nella parte superiore del lato destro della finestra troviamo infatti selezionata l'azione Object Browser. Proprio a lato si trova l'opzione Data Browser, all'interno della quale possiamo visualizzare i semplici dati che abbiamo introdotto in precedenza (Figura 12.25).

Al momento attuale, troviamo solo un record per tabella, quello che abbiamo inserito in precedenza, ma naturalmente i dati visualizzati possono essere molti di più.

Possiamo notare che i campi modificabili sono rappresentati in giallo, mentre il campo chiave è rappresentato con uno sfondo grigio.

Possiamo utilizzare la terza modalità Query Builder per creare una query, ovvero un'interrogazione al database. Per esempio possiamo scrivere nella riga gialla in basso la seguente query SQL:

```
select cognome from NumeriInterni, Reparti where Reparti.prefixo='02'
```

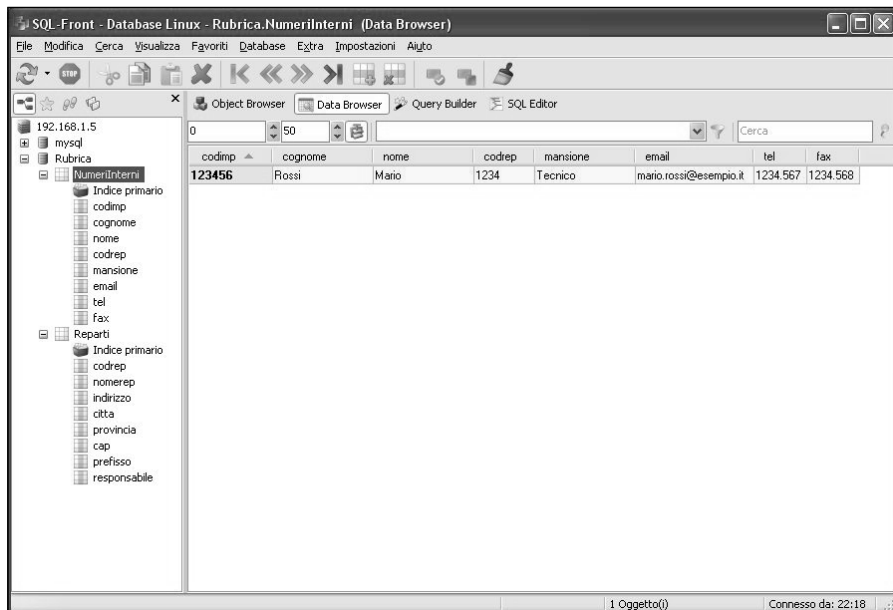


Figura 12.25

La pagina Data Browser di SQL-Front.

Nella parte superiore della finestra si comporrà un’espressione formata da una serie di caselle contenenti dei campi compilati sulla base delle indicazioni presenti nella query stessa (Figura 12.26).

Ci basta fare clic sul pulsante **Esegui** (o premere il tasto funzione F9) per applicare l’interrogazione al database e vedere i risultati che fornisce. Correttamente viene restituito il cognome richiesto (Figura 12.27). Dobbiamo ricordarci che mentre l’interrogazione viene svolta su una macchina, la risposta viene generata dalla macchina Debian Linux sulla quale è in esecuzione il server database MySQL.

La cosa interessante di questa query è che sfrutta il fatto che il database è relazionale: richiede infatti di fornire un’informazione (cognome) che è contenuta in una tabella (**Numeri Interni**), ponendo però una condizione su un’altra tabella (**Reparti**). La query funziona proprio perché le due tabelle sono correlate tramite il campo comune **codrep**.

L’ultima modalità, **SQL Editor** ci permette infine di comporre delle istruzioni SQL per la manipolazione del database. La Figura 12.28 mostra per esempio l’output di un comando che abbiamo precedentemente eseguito nella modalità testuale di MySQL Monitor, sulla macchina Debian Linux.

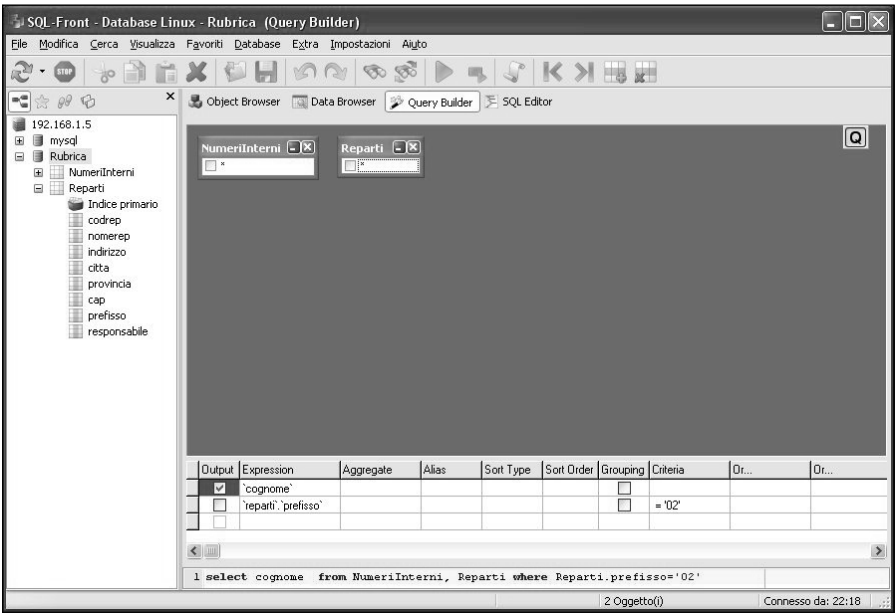


Figura 12.26
Abbiamo creato una semplice query con la quale vogliamo interrogare il database.

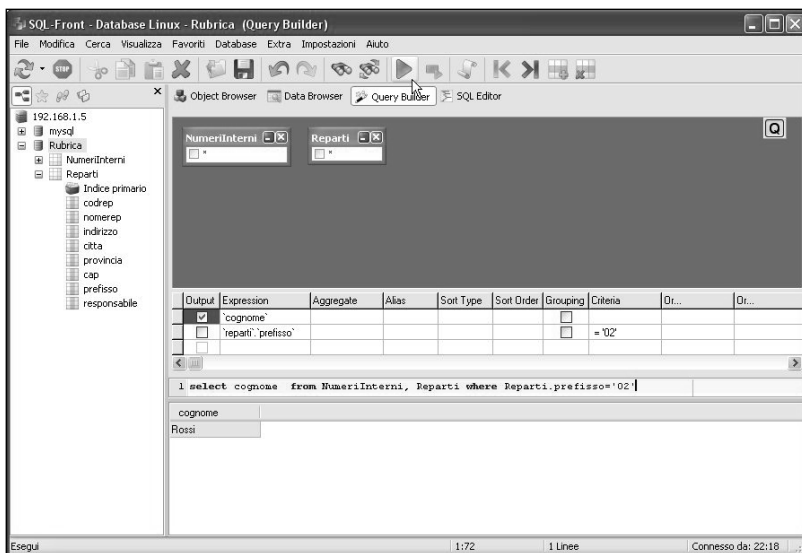


Figura 12.27
Esecuzione e risultato della query.

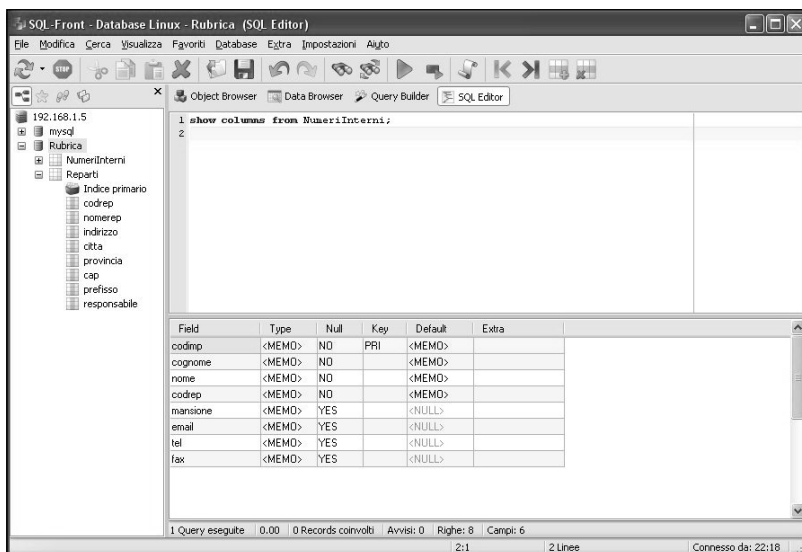


Figura 12.28
Esecuzione di un'istruzione SQL in SQL-Front.

Particolarmente interessante è la modalità di inserimento ed eliminazione dei record. Basta aprire una tabella in modalità di visualizzazione Data Browser e fare clic sull'icona **Inserisci nuovo record** (Figura 12.29): nella finestra si aprirà una nuova riga pronta per l'inserimento dei dati del nuovo record. Premendo **Invio** i dati verranno inseriti nel database che si trova sulla nostra macchina Linux. Proprio a lato di questo pulsante si trova **Elimina record** che consente di svolgere l'operazione inversa. Per modificare i dati già contenuti nel database ci basta invece fare clic su una casella e alterare a piacere i dati desiderati.

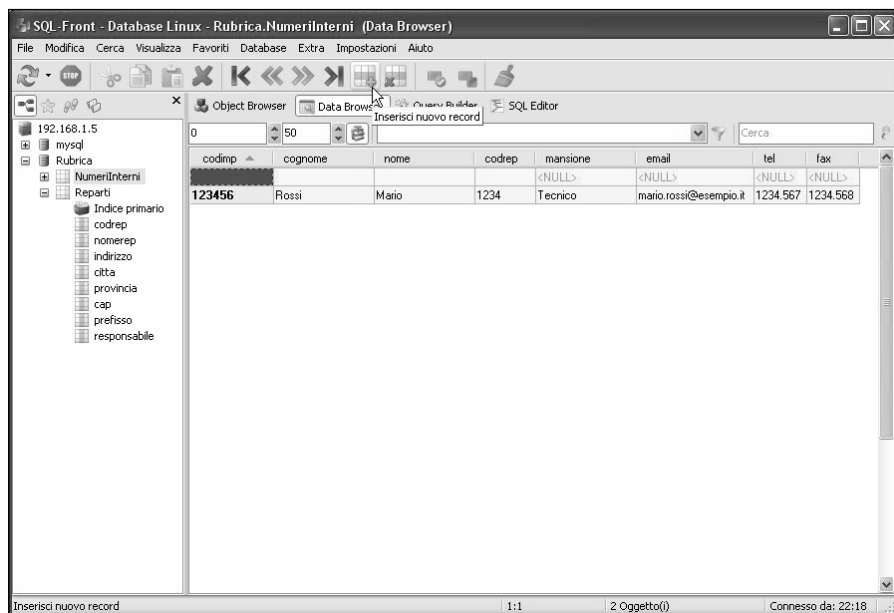


Figura 12.29
Inserimento di un nuovo record nel database MySQL dal front-end situato sulla macchina Windows.

Conclusioni

In questo lungo capitolo abbiamo trattato con un certo dettaglio le procedure di installazione, configurazione e utilizzo di uno degli strumenti più importanti a disposizione degli amministratori e utenti di sistemi e reti Linux: il database open-source MySQL.

Oltre ad avere esaminato le operazioni necessarie per utilizzare il database sulla macchina stessa, abbiamo visto come quello stesso database può essere manipolato a piacere anche da una o più macchine connesse in rete, non necessariamente Linux, tramite le quali possiamo aggiungere o cancellare record, modificare la struttura del database e interrogarlo a piacere.

Questo capitolo conclude la prima parte del libro. A partire dal prossimo capitolo affronteremo più in dettaglio le possibilità di integrazione di una macchina Linux in una rete mista costituita da macchine Linux e Windows.

Capitolo 13

E ora Samba!

Windows, nel bene e nel male, rappresenta una realtà che è impossibile ignorare: in molte aziende vi è la necessità di creare reti miste Linux/Windows.

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Linux e Windows
- ☑ Samba!
- ☑ Installazione di Samba
- ☑ Configurazione iniziale di Samba
- ☑ Utilizzo del client Samba
- ☑ Configurazione di un utente per il server Samba
- ☑ Cartelle condivise
- ☑ Accesso da Windows alla macchina Debian Linux

In molti ambienti professionali, l'idea di creare reti mono-sistema solo Linux è un'utopia.

Vi sono software che “girano” solo su Windows e per i quali non esistono alternative (quanto meno facilmente praticabili) per Linux.

Se in azienda è previsto, per qualsiasi tipo di vincolo interno o esterno, l'impiego dei programmi del pacchetto Microsoft Office o di Adobe Photoshop o di un software di CAD o DTP disponibile solo per Windows, il fatto di sostenere che per Linux esistano applicazioni analoghe, di potenza uguale o superiore rispetto a quelle del mondo Windows diviene, purtroppo una sorta di posizione troppo “ideologica” e troppo poco “logica”; soprattutto poco “pratica”.

Meglio dunque accettare una situazione mista Linux/Windows, con l'obiettivo di massimizzare la comodità d'uso, la potenza del sistema-rete e la produttività personale.

Linux e Windows

Nei capitoli precedenti, abbiamo già provato a far comunicare fra loro macchine Windows e macchine Linux, ma si trattava di una relazione finalizzata a scopi ben precisi: la nostra macchina Windows interrogava un server Web o un server di database e riceveva risposte in rete.

Le necessità possono però essere diverse. Per esempio potremmo avere un archivio aziendale (di documenti, fogli elettronici, fotografie o file audio o video) disponibile su macchine Windows, cui sarebbe utile poter accedere anche dalle macchine Linux o viceversa.

- È dunque importante che i sistemi Linux possano accedere alle risorse delle macchine Windows, per esempio ai file server e alle stampanti.
- Ma Linux rappresenta una risorsa insostituibile e pertanto è importante che anche i sistemi Windows possano accedere ai file server e alle stampanti connesse ai sistemi Linux.

L'*interoperabilità* con gli altri sistemi operativi è un aspetto importante della potenza e della flessibilità di Linux; inoltre è uno degli aspetti che più hanno promosso l'impiego di Linux in ambito aziendale (e anche domestico).

Samba!

L'interoperabilità fra Linux e Windows è garantita da un insieme di applicazioni client/server che formano il pacchetto *Samba*.

- Quando le macchine Linux devono accedere a risorse Windows, devono “fingersi” client Windows e a tale scopo devono impiegare un'applicazione *client Samba*.
- Quando invece le macchine Linux devono concedere le proprie risorse alle richieste provenienti da macchine Windows, devono utilizzare un *server Samba*.

In pratica client e server Samba svolgono tutte le operazioni di traduzione fra i diversi requisiti di una rete Linux e di una rete Windows, consentendo alle macchine di comunicare fra loro.

In questo capitolo vedremo come configurare i client e i server Samba sulla nostra macchina Debian, mostrando come attivare le funzioni di connettività fra queste macchine.

Le macchine Windows connesse in rete comunicano utilizzando il protocollo SMB (Server Message Block). Il protocollo SMB rientra in realtà nel più generale sottosistema di rete Microsoft CIFS (Common Internet File Services), tuttavia proprio dall'acronimo SMB deriva il nome del pacchetto *SaMBa*.

Samba è stato sviluppato a partire dal 1991 (ma allora si chiamava semplicemente NetBIOS per Unix o smbserver) da Andrew Tridgell. Oggi Samba è disponibile per Linux ma anche per ogni sistema derivato da Unix.

Attualmente (marzo 2008) è disponibile per Debian la versione 3.0.24 che offre importanti miglioramenti rispetto alle versioni precedenti, i più importanti dei quali sono i seguenti:

- supporto per Microsoft Active Directory; i server Samba possono essere membri di domini Active Directory;
- supporto avanzato delle codifiche Unicode e delle lingue locali;
- supporto avanzato dei servizi di stampa Windows.

Informazioni più dettagliate su Samba 3 si possono trovare nella documentazione (disponibile sotto forma di file PDF nel pacchetto *samba-doc-pdf*). Dopo aver installato il pacchetto *samba-doc-pdf*, troveremo la documentazione di Samba nella directory `/usr/share/doc/samba-doc-pdf` all'interno di cinque file compresi. Il sito ufficiale di Samba (Figura 13.1) si trova al seguente indirizzo:

<http://www.samba.org>

il cui contenuto è disponibile anche sul mirror italiano:

<http://it.samba.org>

Installazione di Samba

Come sempre, sfruttiamo l'interfaccia grafica di Gnome per scaricare e installare sulla nostra macchina Debian il pacchetto di Samba.

DA SAPERE *In molte altre distribuzioni (per esempio la Ubuntu, derivata dalla Debian, ma anche la SuSE) Samba si trova già installato e pronto all'uso. Purtroppo questo non è il caso della distribuzione Debian.*



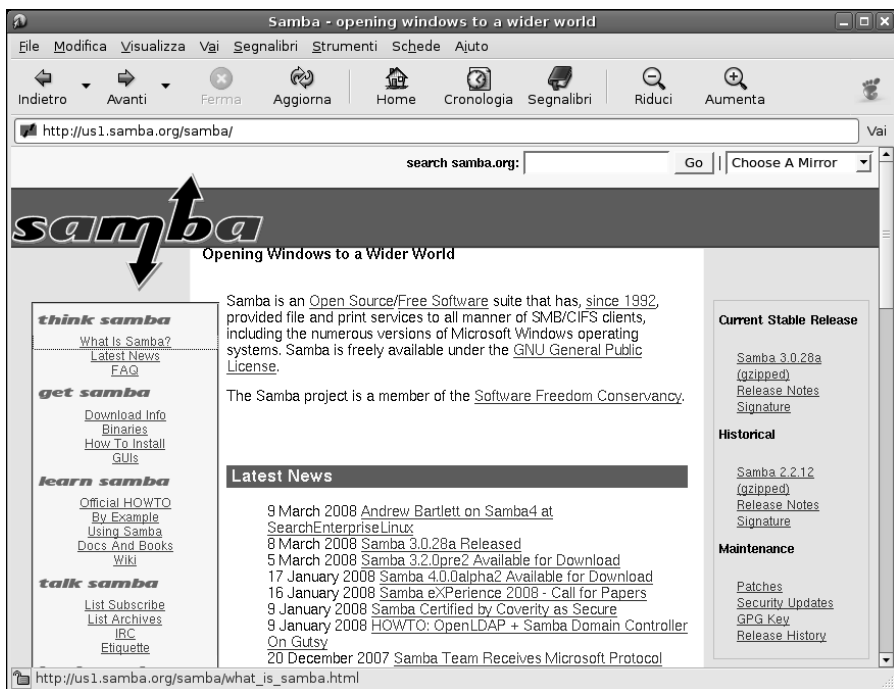


Figura 13.1

Il sito ufficiale di Samba, dove possiamo trovare utili informazioni sul pacchetto.

Utilizzeremo dunque il Gestore di pacchetti Synaptic, che, come sempre, si preoccupa di segnalare e installare tutte le dipendenze del software.

Richiamiamo Desktop > Amministrazione > Gestore pacchetti Synaptic dal desktop Gnome di Debian e, nella finestra del programma, facciamo clic sull'icona Cerca nella barra degli strumenti. Nella finestra di dialogo Trova scriviamo semplicemente samba (i pacchetti Samba, in totale, sono solo otto). Facendo clic sul pulsante Cerca verranno visualizzati tutti i pacchetti legati a Samba (Figura 13.2).

Ora facciamo clic sulla casella quadrata che si trova a lato della voce samba e selezioniamo l'opzione Marca per l'installazione. Come possiamo vedere nella finestra di dialogo Marcare le ulteriori modifiche richieste?, il gioco delle dipendenze fra pacchetti fa sì che dovremo scaricare ben tredici altri pacchetti, principalmente relativi a script Perl di supporto (Figura 13.3). Facciamo clic sul pulsante Marca.

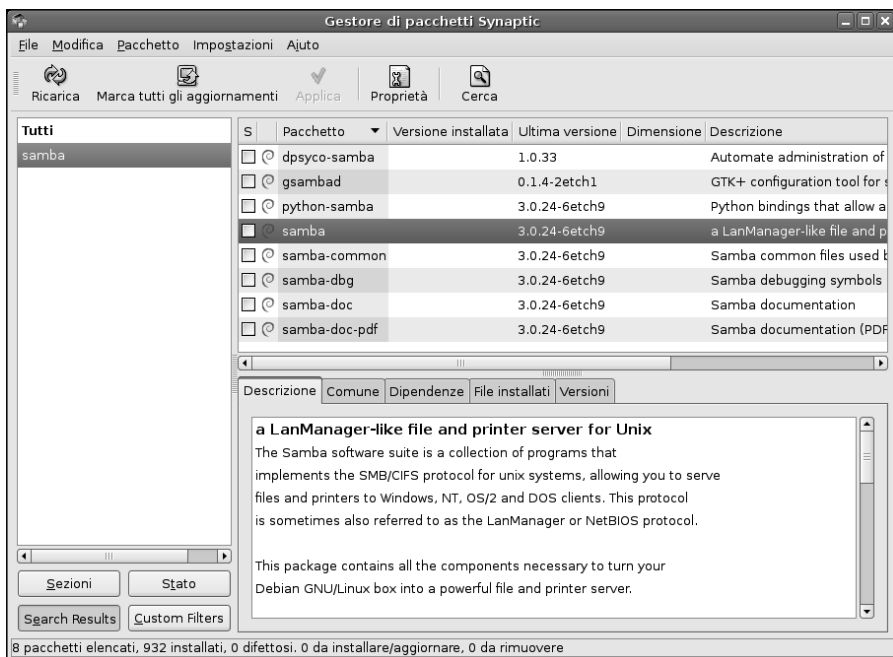


Figura 13.2

I pacchetti Linux relativi a Samba.

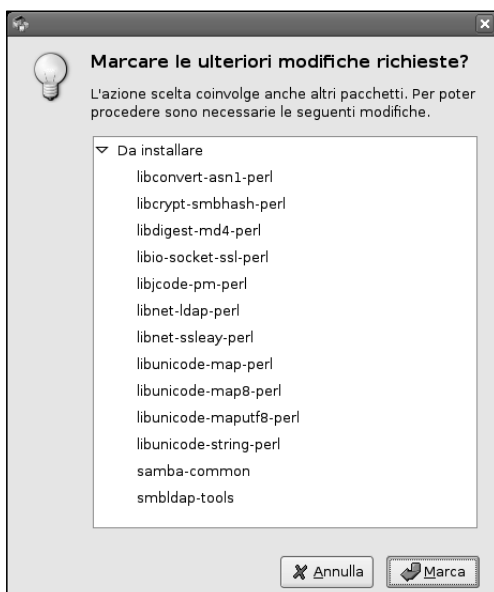


Figura 13.3

Tutti i pacchetti da installare a corollario di Samba.

Già che ci siamo, scarichiamo anche la documentazione PDF, selezionando l'opzione **Marca** per l'installazione anche per il pacchetto `samba-doc-pdf`. Ora possiamo avviare l'installazione, facendo clic sul pulsante **Applica** nella barra degli strumenti; confermiamo facendo clic sul pulsante **Applica** nella finestra **Riepilogo**. Il download di tutti i pacchetti potrebbe richiedere un po' più del normale (Figura 13.4), specialmente se non possiamo contare su una connessione Internet molto veloce; terminato il download verrà lanciata la procedura di configurazione del software.

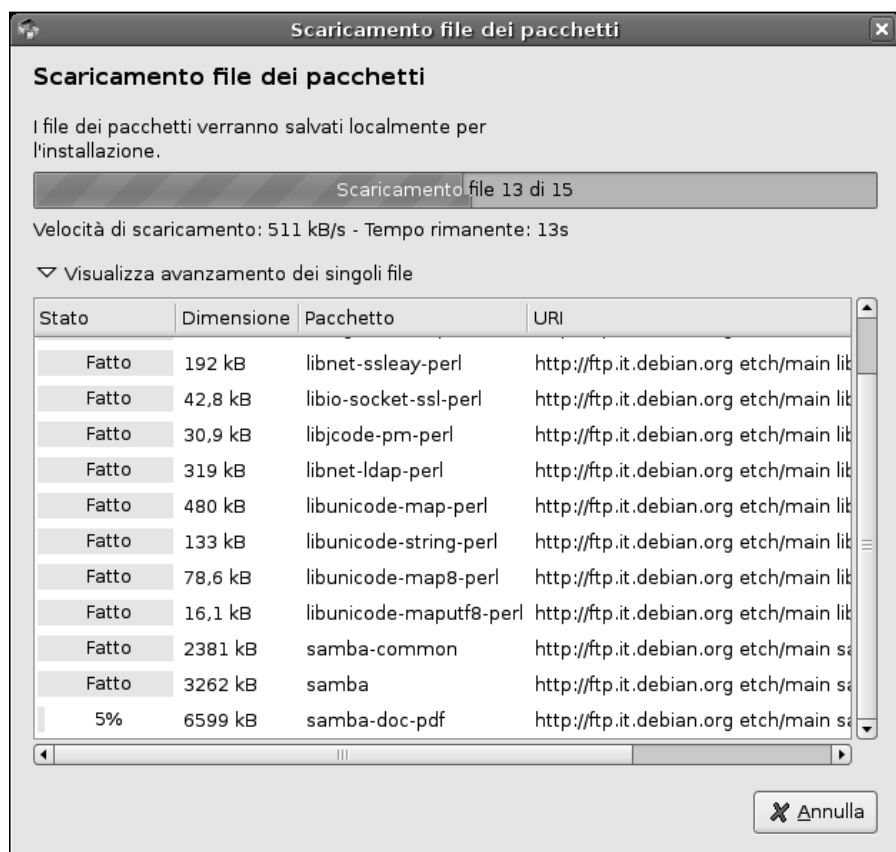


Figura 13.4

Il download di Samba può richiedere qualche istante in più del solito, dato il numero e il peso dei pacchetti interessati.

Configurazione iniziale di Samba

La prima pagina della procedura di configurazione (Figura 13.5) chiede di specificare il nome del dominio o del workgroup.

Supponendo di aver inserito la macchina Debian Linux in una rete a prevalenza Windows, dovremo specificare il nome del gruppo di lavoro Windows, informazione che possiamo ricavare da una macchina Windows qualsiasi, selezionando il Pannello di controllo, poi l'icona Sistema e infine aprendo la pagina Nome computer della finestra di dialogo Proprietà del sistema. L'informazione che cerchiamo si trova alla riga Gruppo di lavoro (Figura 13.6).

Tornati alla macchina Debian, scriviamo il nome del gruppo di lavoro nella casella di testo Nome del Workgroup/Dominio della finestra di dialogo Samba Server. Quindi facciamo clic sul pulsante Avanti per passare alla seconda pagina della procedura di configurazione (Figura 13.7).

In questa pagina, l'opzione Modificare smb.conf per usare le impostazioni WINS da DHCP? è forse un po' criptica per i non addetti ai lavori, ma in sostanza dice che se nella rete è attivo un server DHCP (che si occupa della distribuzione degli indirizzi IP interni e privati della rete), allora la procedura di configura-



Figura 13.5

Avvio della configurazione di Samba.

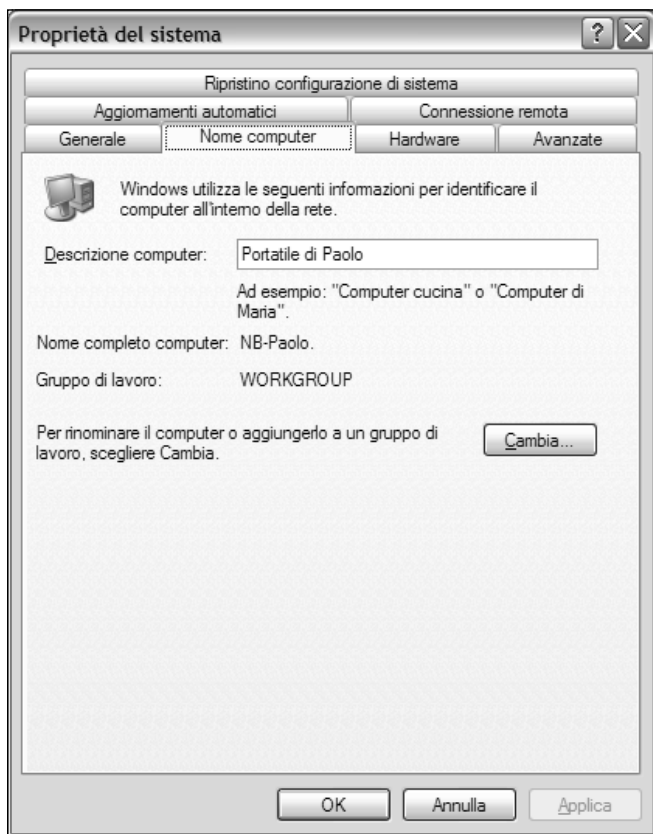


Figura 13.6

Scopriamo (se ne abbiamo bisogno) il nome del gruppo di lavoro da una qualsiasi macchina Windows.

zione può utilizzare tale risorsa per informarsi sulla composizione della rete, sfruttando il protocollo WINS (il “DNS” di Windows).

Se gli indirizzi di rete sono distribuiti dal nostro server DHCP Debian o dal router/gateway, selezioniamo senz’altro questa casella. Per proseguire facciamo clic sul pulsante **Avanti**.

Verrà così perfezionata la procedura di installazione.

Concludiamo facendo clic sul pulsante **Chiudi** e poi usciamo dal **Gestore di pacchetti Synaptic**.

Terminata l’installazione, il server Samba viene automaticamente avviato, come possiamo agevolmente verificare con il comando `Desktop > Amministra-`



Figura 13.7

Sfruttiamo i servizi DHCP già disponibili in rete.

zione > Servizi che apre la finestra di dialogo Impostazioni servizi dove troveremo il nuovo Servizio condivisione cartelle (samba) (vedere la Figura 13.8). A questo punto, dunque, abbiamo attivato sul sistema il *client Samba* che ci consentirà di interrogare dalla macchina Linux Debian le macchine Windows della rete per accedere alle risorse che esse offrono in condivisione.

Inoltre abbiamo attivato e configurato (automaticamente) il *server Samba* che ci consentirà di offrire in condivisione delle risorse alle macchine e agli utenti Windows che le richiederanno.

Poiché l'utilizzo del client è decisamente il più semplice, partiremo da qui. Successivamente affronteremo le tecniche legate all'uso del server.

Utilizzo del client Samba

La configurazione del client Samba è già stata svolta automaticamente durante la procedura di configurazione e dunque, in realtà, occorre semplicemente verificare che tutto funzioni: il sistema è già configurato come un client del



Figura 13.8

Il server Samba è attivo e pronto all'uso.

dominio o del gruppo di lavoro. Le credenziali e i diritti di accesso alle risorse disponibili nel dominio o nel gruppo di lavoro verranno controllate ogni volta che cercheremo di accedere a tali risorse.

Curiosare fra le risorse disponibili in Windows

Per sfogliare le risorse della rete locale possiamo utilizzare il file manager Nautilus di Gnome, ovvero lo strumento grafico che utilizziamo normalmente per aprire le directory interne del sistema Linux. Abbiamo due possibilità, entrambe ugualmente rapide.

- Possiamo fare clic sull'icona Computer che si trova nel desktop Gnome. Si aprirà la finestra Computer di Nautilus che mostra tutte le risorse richiamabili dal computer in uso. Una di queste voci è Rete che aprirà la finestra Rete, dove troveremo tutte le macchine connesse alla rete locale. Nulla distinguerà le macchine Windows da quelle Linux. Nella Figura 13.9 possiamo vedere aperta proprio la cartella del materiale di questo libro, situata su una macchina Windows.

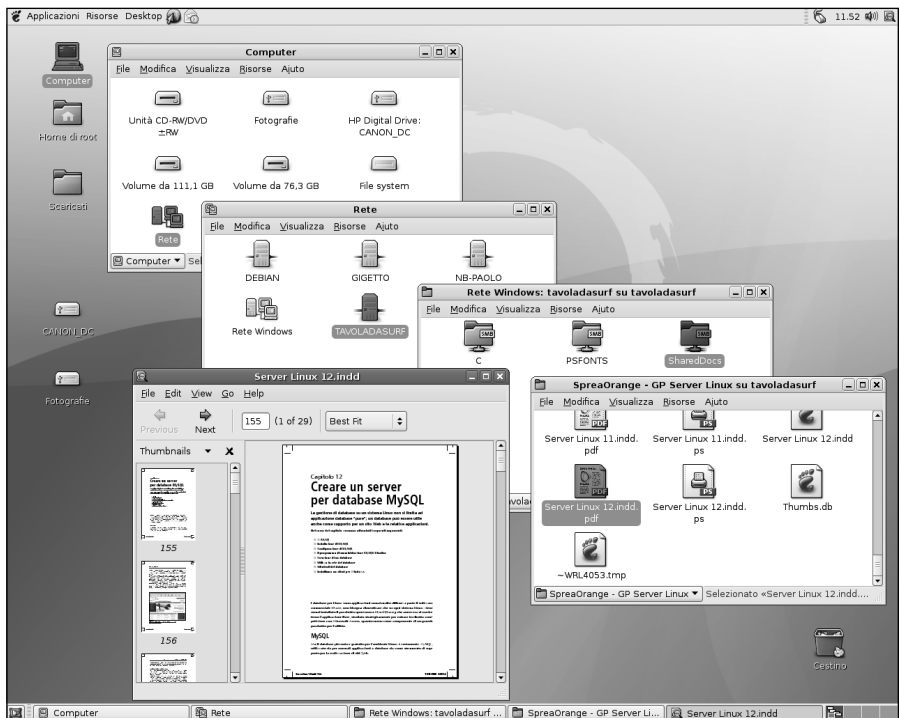


Figura 13.9

Dal desktop possiamo richiamare le risorse delle macchine Windows come se si trovassero sulla macchina locale, con la stessa facilità e naturalezza.

- In alternativa possiamo selezionare dal menu di Gnome il comando **Risorse > Server di rete**, che ci porterà direttamente alla finestra Rete rappresentata nella Figura 13.10.



Figura 13.10

La finestra Rete dalla quale possiamo richiamare tutte le risorse condivise disponibili in rete.

- Volendo, possiamo utilizzare il protocollo **smb** ora disponibile nella macchina Linux per accedere alle risorse condivise da qualsiasi browser. Basta aprire il browser del sistema (in questo caso Epiphany, il browser di dotazione) e digitare nella casella di testo degli indirizzi un particolare indirizzo: **smb://nome-workgroup** per accedere all'intera rete e **smb://nome-computer** per accedere a una specifica macchina e alle sue risorse. Avremo così accesso alla rete dal browser, in uno stile simile a quello di una connessione FTP via browser (Figura 13.11).

A questo punto le cartelle condivise sulle macchine Windows avranno un comportamento analogo a quello delle directory locali: potremo copiarvi file, visualizzare file, trascinare file per operazioni di copia e spostamento (Figura 13.12).

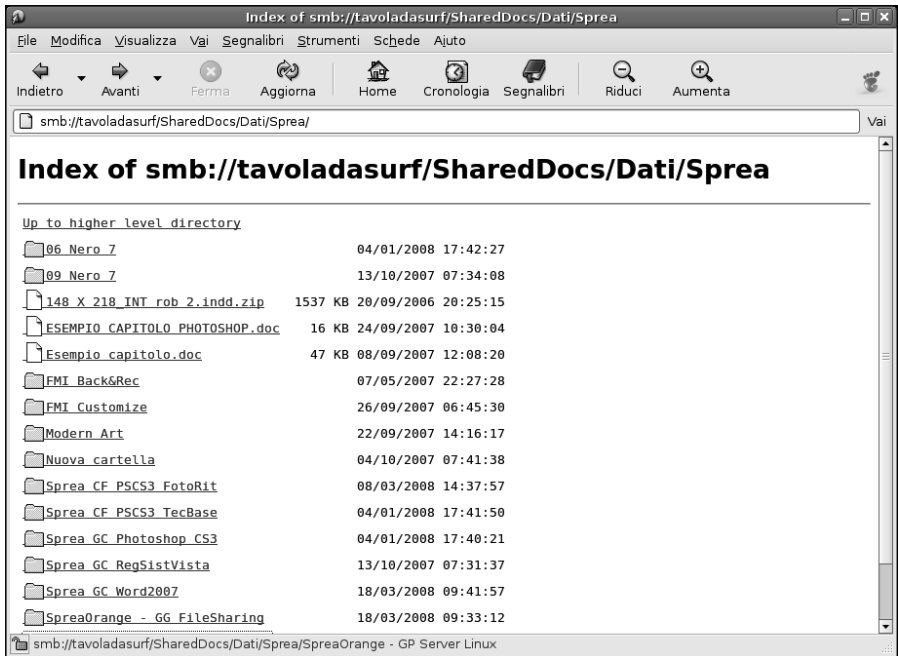


Figura 13.11

Possiamo esplorare la rete anche tramite il browser, utilizzando il protocollo smb.

Uso di una stampante Windows

Per accedere a una stampante Windows da un sistema Linux utilizzando Samba, dobbiamo creare la definizione di una stampante da Linux e collegarla tramite Samba a una stampante fisica connessa alla una macchina Windows.

A questo punto potremo utilizzarla come se fosse una stampante locale (o di rete, con un proprio indirizzo IP).

Ecco come possiamo creare la definizione di una stampante Windows dopo aver installato e configurato il client Samba.

1. Dal menu del desktop Gnome di Debian Linux, selezioniamo il comando Desktop > Amministrazione > Printing. Verrà visualizzata la finestra Stampanti, rappresentata nella Figura 13.13.
2. Ora facciamo doppio clic sull'icona Nuova stampante nella finestra di dialogo Stampanti per avviare la procedura Aggiunta di una stampante.

**Figura 13.12**

Abbiamo aperto una fotografia che si trova su un disco condiviso di una macchina Windows.

**Figura 13.13**

La finestra delle stampanti nel desktop Gnome. Attualmente non è installata alcuna stampante.



Figura 13.14

Dobbiamo indicare a Linux dove può trovare la stampante.

Verrà visualizzata la finestra di dialogo rappresentata nella Figura 13.14: Fase 1 di 2: Connessione della stampante.

3. Per l'opzione **Tipo stampante**, indichiamo **Stampante di rete** e nella casella a lato selezioniamo l'opzione **Stampante Windows (SMB)**; si aprirà la finestra di dialogo **Richiesta autenticazione** (Figura 13.15), nella quale dobbiamo specificare il **Nome utente** e la **Password** di accesso alla stampante. Concludiamo facendo clic sul pulsante **Connetti**.
4. Ora facciamo clic sulla freccia rivolta verso il basso che si trova all'estrema destra della casella di testo **Host** e selezioniamo la macchina alla quale è connessa la stampante. Quindi facciamo clic sulla freccia rivolta verso il basso a lato della casella **Stampante** e selezioniamo la stampante desiderata. Alla fine di questi passaggi, la finestra di dialogo **Aggiunta di una stampante** dovrebbe avere un aspetto simile a quello rappresentato nella Figura 13.16. Facciamo clic sul pulsante **Avanti** per proseguire.
5. Verrà visualizzata la pagina **Fase 2 di 2: Driver della stampante** della finestra di dialogo **Aggiunta di una stampante** (Figura 13.17). Qui nelle caselle

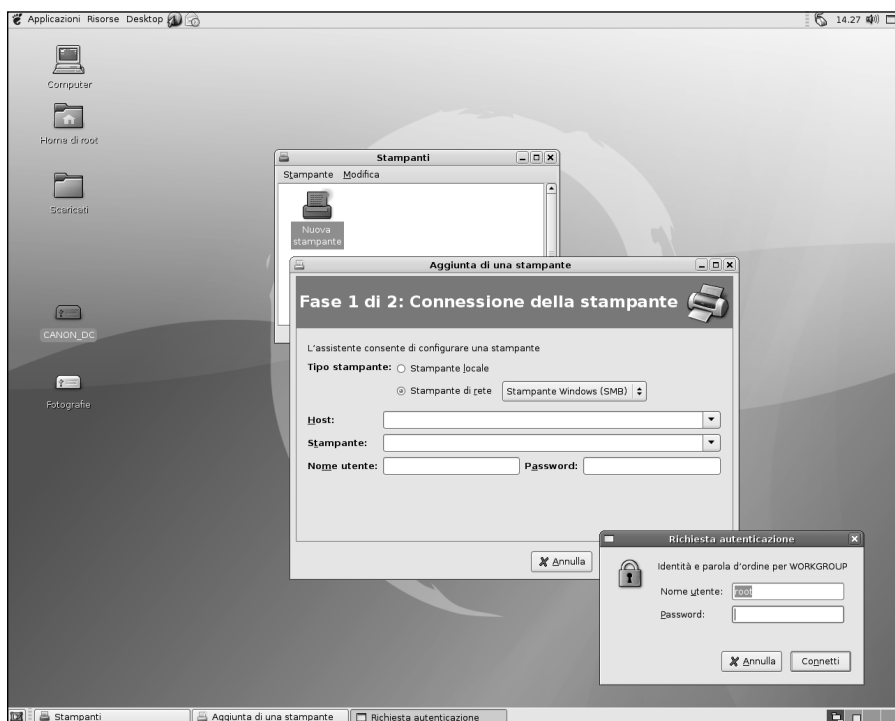


Figura 13.15
Autenticazione e connessione alla rete Windows.

Produttore e Modello dobbiamo specificare il tipo di stampante in uso; il relativo driver verrà indicato nella terza casella di testo: **Driver**. Al termine facciamo clic su **Applica** per concludere la procedura.

6. La nuova stampante SMB comparirà nella finestra **Stampanti**. Per verificare la connessione, facciamo clic sulla sua icona con il pulsante destro del mouse e selezioniamo dal menu rapido il comando **Proprietà**. Verrà visualizzata la finestra delle proprietà della stampante, dove possiamo fare clic sul pulsante **Stampa pagina di prova** (Figura 13.18). Dalla stampante Windows uscirà la pagina di prova Linux.



Figura 13.16
Definizione di una stampante Samba Windows da Linux.



Figura 13.17
Scelta del driver per la stampante SMB.

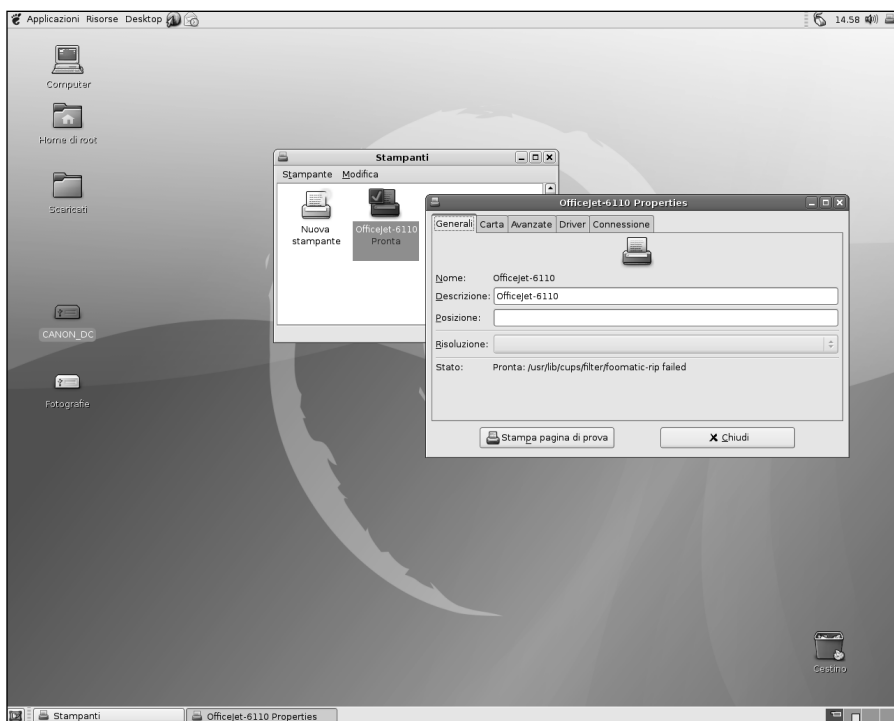


Figura 13.18
La stampante è connessa e funzionante.

7. Per far sì che diventi la stampante predefinita del sistema, facciamo clic sulla sua icona con il pulsante destro del mouse e selezioniamo il comando **Rendi predefinita**.

Ora la stampante verrà impiegata da tutte le applicazioni disponibili sul sistema. Per esempio, la Figura 13.19 mostra la finestra di dialogo **Stampa di Impress**, il programma per presentazioni del pacchetto OpenOffice.org.

Configurazione di un utente per il server Samba

Accedere alle risorse di una macchina Windows via rete da Linux è piuttosto semplice, mentre l'operazione inversa è leggermente più complessa per il fatto che Linux adotta una gestione più rigorosa degli accessi.

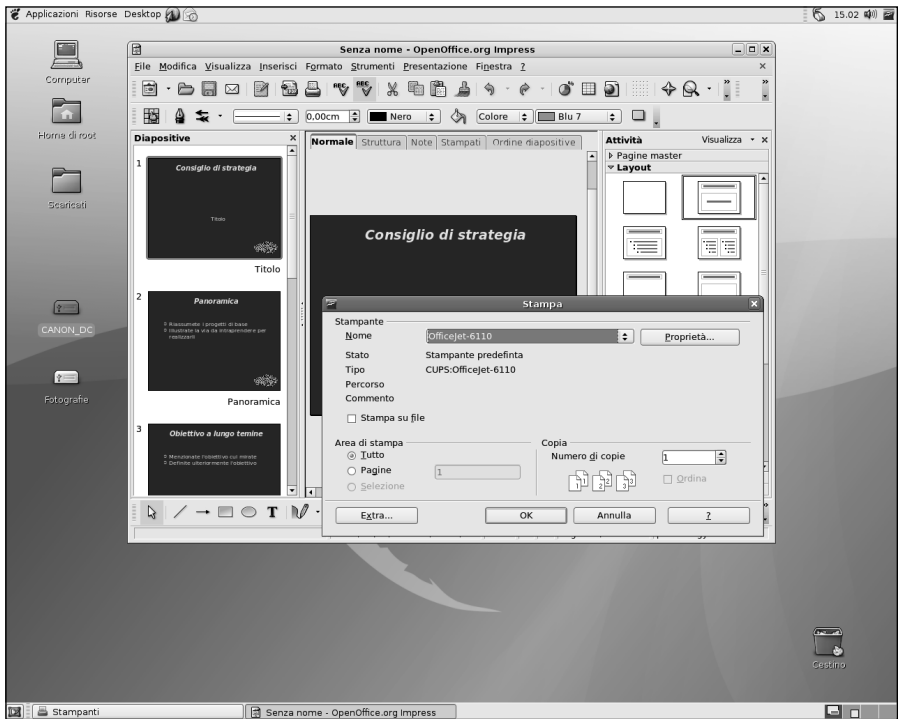


Figura 13.19

La stampante è ora disponibile per tutte le applicazioni.

In questo caso, Samba funge da *server*, ovvero deve concedere l'uso delle risorse a un utente che le richiede dall'esterno. Ovviamente deve fare attenzione a concedere le risorse giuste all'utente giusto. Nulla di più e nulla di meno.

Anche Samba, come altri software, ha una propria gestione specifica delle password. Naturalmente può concedere l'accesso al sistema solo da parte di utenti dotati di un account sul sistema stesso, concedendo loro di accedere alle risorse di loro proprietà e alle eventuali cartelle condivise (ci torneremo fra un attimo). Per prima cosa dobbiamo quindi far conoscere a Samba gli utenti che hanno diritto di accesso al sistema.

A tale scopo dobbiamo utilizzare il comando `smbpasswd`, che attribuisce una password Samba per gli accessi da macchine Windows. La Figura 13.20 mostra per esempio l'aspetto del comando necessario per configurare l'accesso da Windows via rete all'utente `paolo`. L'opzione `-a` aggiunge una nuova coppia nome-utente/password; l'opzione `-h` presenta invece l'elenco delle opzioni disponibili.


```

Terminale
File Modifica Visualizza Terminale Schede Ajuto
Debian:~# smbpasswd -h
When run by root:
    smbpasswd [options] [username]
otherwise:
    smbpasswd [options]

options:
  -L                local mode (must be first option)
  -h                print this usage message
  -s                use stdin for password prompt
  -c smb.conf file  Use the given path to the smb.conf file
  -D LEVEL          debug level
  -r MACHINE        remote machine
  -U USER           remote username
extra options when run by root or in local mode:
  -a                add user
  -d                disable user
  -e                enable user
  -i                interdomain trust account
  -m                machine trust account
  -n                set no password
  -W                use stdin ldap admin password
  -w PASSWORD       ldap admin password
  -x                delete user
  -R ORDER          name resolve order
Debian:~# smbpasswd -a paolo
New SMB password:
Retype new SMB password:
Debian:~#

```

Figura 13.20

Il comando `smbpasswd` consente di definire gli account Samba.

A questo punto abbiamo un utente, accreditato già sul sistema, che può connettersi da una macchina Windows per aver accesso alle proprie risorse e alle altre risorse condivise del computer.

Cartelle condivise

In una connessione Samba da una macchina Windows a una macchina Linux, ogni utente avrà sostanzialmente accesso alle proprie risorse: la macchina Windows si comporterà in pratica come un terminale grafico connesso alla macchina Linux, la quale consentirà a ciascun utente di lavorare in modo remoto con i propri dati.

Ma possiamo anche definire delle risorse condivise alle quali potranno accedere tutti gli utenti. Per esempio potremmo avere un disco o una directory

contenente un archivio coerente dei dati (tabelle, fogli elettronici, documenti standard aziendali ma anche a fotografie, filmati o brani musicali).

Il modo migliore e più rapido per condividere una risorsa di questo tipo consiste nell'utilizzare la finestra **Impostazioni cartelle condivise**.

Dal desktop Gnome di Debian Linux, selezioniamo il comando **Desktop > Amministrazione > Cartelle condivise**. Si aprirà la finestra di dialogo **Impostazioni cartelle condivise**, inizialmente vuota (Figura 13.21). Il nostro sistema non mette in condivisione proprio nulla.

Facciamo clic sul pulsante **Aggiungi** e si aprirà la finestra di dialogo **Condividi cartella**, rappresentata nella Figura 13.22. Qui, nella casella **Percorso**, dobbiamo specificare la posizione in cui si trova la cartella (in questo caso un intero disco) da condividere. Facendo clic sulla casella vengono proposti i principali elementi condivisibili: la cartella home dell'utente, la sua **Scrivania**, l'intero **Filesystem** e gli eventuali dischi connessi al sistema (in questo caso il disco **Fotografie**). L'ultima opzione, **Altro**, consente di individuare un'altra cartella da condividere.



Figura 13.21

L'aspetto della finestra di dialogo **Impostazioni cartelle condivise**. La prima volta che viene richiamata è ovviamente vuota: il sistema non condivide nulla.

**Figura 13.22**

L'aspetto della finestra di dialogo *Condividi cartella*, dove possiamo specificare il percorso e le proprietà della cartella condivisa.

Nella seconda casella, *Condividi con*, è già impostato il protocollo SMB.

La seconda parte della finestra di dialogo consente di specificare il *Nome* e un *Commento* per questa condivisione. Trattandosi di fotografie possiamo scrivere qualcosa come *Foto* e *Archivio fotografico aggiornato*.

Le due caselle *Sola lettura* e *Consenti di sfogliare cartella* consentono rispettivamente di bloccare eventuali operazioni di scrittura di questi file e di consentire la libera consultazione delle cartelle e delle relative sottocartelle.

Questo è tutto ciò che bisogna impostare per attivare una condivisione.

Il grosso pulsante *Impostazioni generali delle condivisioni Windows* apre la finestra di dialogo *Impostazioni condivisioni Windows* che normalmente contiene opzioni già impostate correttamente e che dunque non dovrebbe essere necessario modificare.

Per attivare la condivisione facciamo dunque clic sul pulsante OK della finestra di dialogo *Condividi cartella*. L'elemento verrà aggiunto immediatamente alla finestra *Impostazioni cartelle condivise*, dove avrà l'aspetto rappresentato nella Figura 13.23: l'icona di una cartella, con una connessione di rete e il segnalino SMB. Sempre in questa figura si vede l'aspetto della finestra *Condividi cartella*, che si ottiene facendo clic sul pulsante *Proprietà* dopo aver selezionato una condivisione. Come si può vedere, a parte la casella *Percorso*, non più attiva, questa finestra ricalca l'aspetto della finestra di dialogo utilizzata per definire la condivisione.

A questo punto possiamo chiudere la finestra di dialogo *Impostazioni cartelle condivise* facendo clic sul pulsante OK. Ora vedremo come si comporta la nostra macchina Debian in risposta alle richieste SMB provenienti da una macchina Windows.

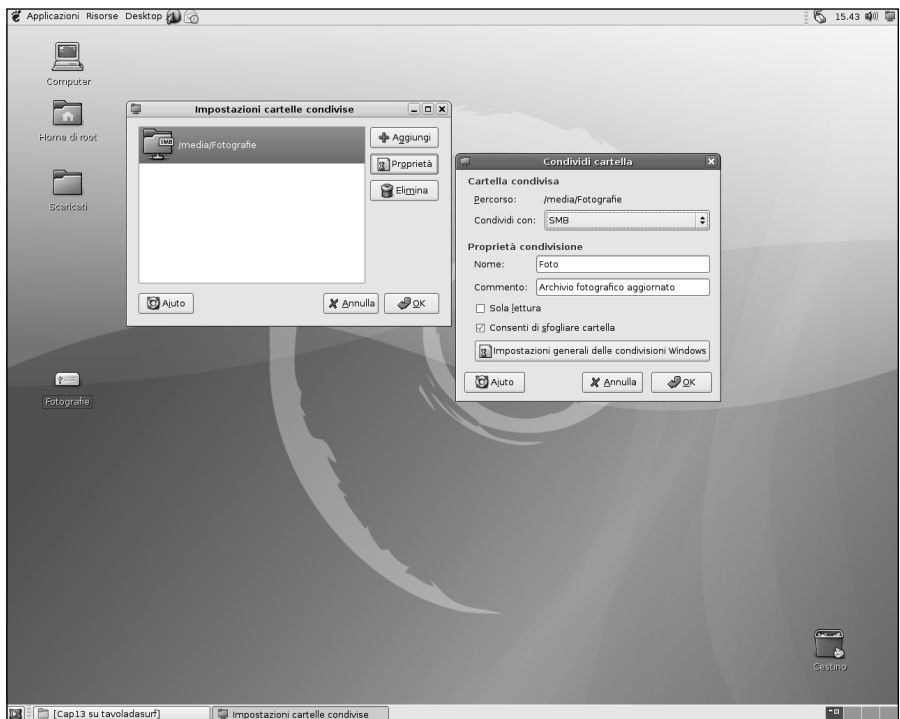


Figura 13.23

L'aspetto della finestra di dialogo *Impostazioni cartelle condivise* dopo aver definito una condivisione; a lato è visualizzata anche la finestra *Condividi cartella*, richiamata tramite il pulsante *Proprietà*.

Accesso da Windows alla macchina Debian Linux

Collegiamoci ora a una macchina Windows connessa in rete e situata sullo stesso gruppo di lavoro della macchina Linux. Selezioniamo il comando Start > Risorse di rete per vedere se la nostra rete manifesta qualche cambiamento dopo l'installazione del server Samba sulla macchina Debian. In effetti è così; come possiamo vedere nella Figura 13.24, la macchina il cui nome è Debian server offre in condivisione una cartella.

Facciamo clic su Visualizza computer del gruppo di lavoro nel riquadro Operazioni di rete in alto a sinistra nella finestra Risorse di rete. In effetti verranno visualizzate tutte le macchine della rete, compresa la nuova macchina Debian

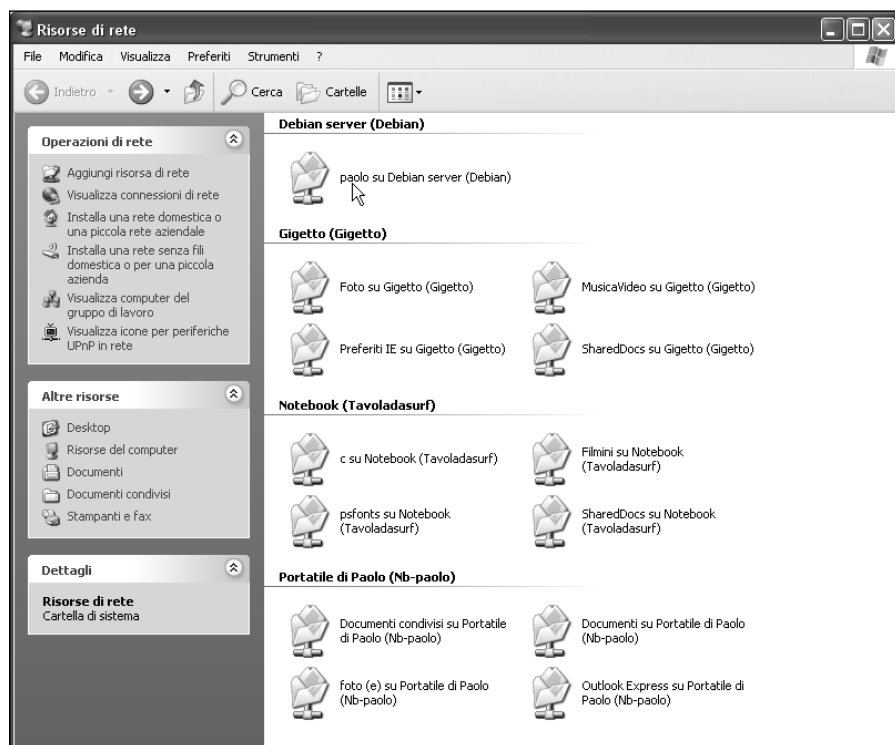


Figura 13.24

La macchina Debian è comparsa fra le Risorse di rete del sistema Windows connesso in rete.

server. Nella Figura 13.25 si può vedere anche la finestra di dialogo Proprietà che si ottiene facendo clic col pulsante destro del mouse sull'icona di una delle macchine e scegliendo dal menu rapido il comando Proprietà.

Un doppio clic sulla macchina e verranno visualizzate le risorse condivise. Come possiamo vedere nella Figura 13.26, si tratta della cartella home dell'utente che abbiamo definito sul sistema Samba e della cartella condivisa Foto che abbiamo appena definito sulla macchina Debian Linux.

Provando ad accedere a una delle risorse condivise, comparirà la finestra di connessione al sistema Debian, rappresentata nella Figura 13.27, che riporta le caselle di testo Nome utente e Password.

Qui dobbiamo specificare non il nostro nome-utente per il sistema Windows ma quello che abbiamo impostato per accedere al nostro account Linux tramite Samba, ovvero il nostro nome-utente Linux e la password che abbiamo impostato con il comando `smbpasswd`.

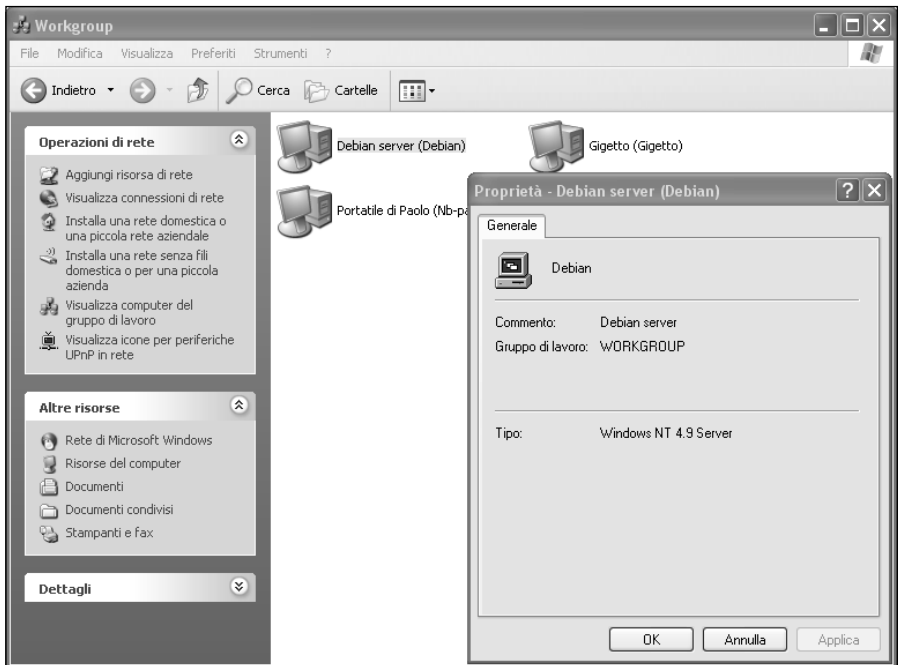


Figura 13.25
La macchina Debian e la relativa finestra delle proprietà.

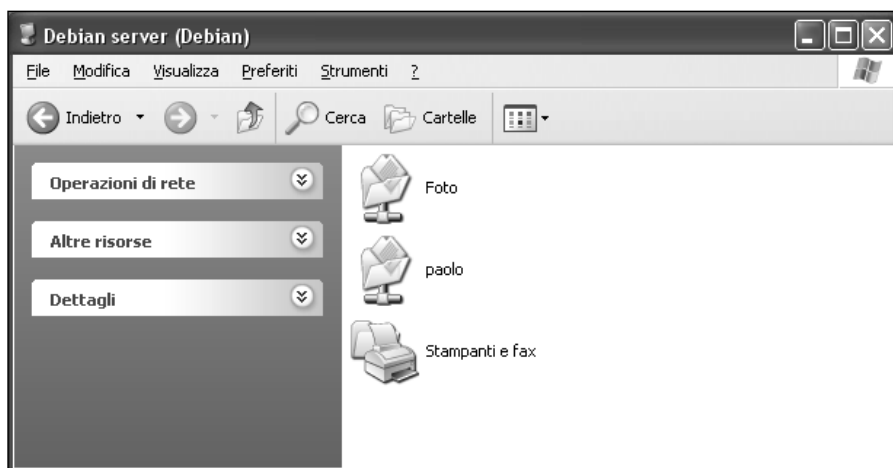


Figura 13.26
Le risorse condivise della nostra macchina Debian Linux.



Figura 13.27
Tramite la piccola finestra Connect to Debian, possiamo introdurre il nostro nome-utente e la nostra password per la connessione al sistema Linux da Windows via Samba.

Eseguita la connessione, potremo accedere alle risorse condivise e alle nostre risorse sull'account Linux. La Figura 13.28 mostra una finestra di Explorer di Windows mentre accediamo a una cartella di fotografie presente sulla macchina Debian, come si può vedere dall'indirizzo riportato nella barra del titolo. A questo punto l'integrazione fra la macchina Linux e le macchine Windows in una rete mista Linux/Windows è completa: abbiamo utilizzato il client Samba

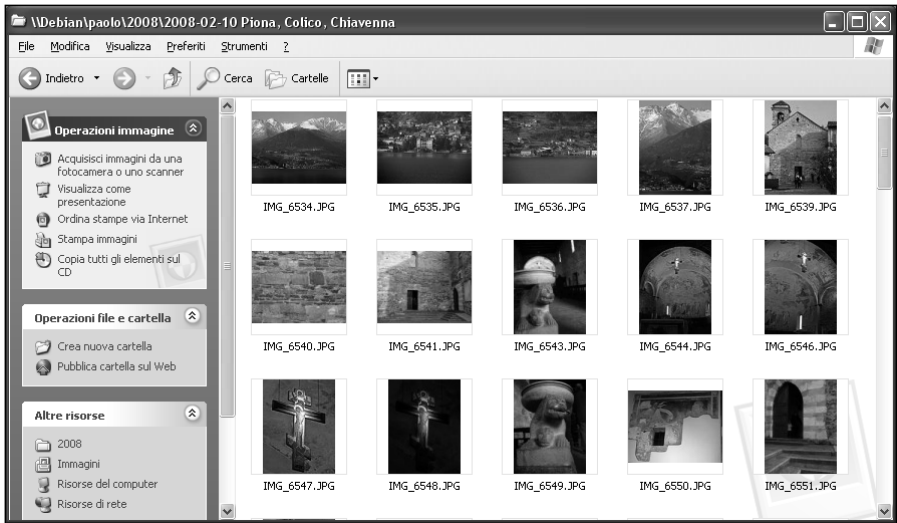


Figura 13.28

Le fotografie di questa finestra Windows sono contenute nella macchina Debian Linux, alla quale abbiamo fatto accesso tramite il server Samba.

per accedere da Debian alle risorse condivise (cartelle, dischi ma anche stampante) presenti su una macchina Windows e, viceversa, abbiamo utilizzato una macchina Windows per accedere al nostro account e alle risorse condivise della macchina Debian Linux, tramite il server Samba.

Manca qualcosa? Se da un lato abbiamo provato a stampare da Linux su una stampante connessa a una macchina Windows, dall'altro non abbiamo provato a utilizzare la macchina Linux come un server di stampa. Questo sarà l'argomento del prossimo capitolo.

Conclusioni

In questo capitolo abbiamo esaminato le molte interessanti funzionalità del pacchetto Samba per l'integrazione fra macchine Windows e Linux sulla stessa rete. Abbiamo descritto l'utilizzo del client Samba che consente alla macchina Linux di accedere al contenuto delle macchine Windows e quindi abbiamo esaminato la configurazione e l'utilizzo del server Samba che consente alle macchine Windows di accedere alle risorse contenute in una macchina Linux.

Nel prossimo capitolo esamineremo l'utilizzo di una macchina Linux come un server di stampa tramite un servizio CUPS.

Capitolo 14

Condividere le stampanti con CUPS

Un tempo, configurare e amministrare un server di stampa Linux era un'operazione complessa. Poi è arrivato CUPS

Nel corso del capitolo verranno affrontati i seguenti argomenti.

- ☑ Il sistema di stampa CUPS
- ☑ Installazione dei driver per una stampante locale
- ☑ Stampa dalla macchina locale
- ☑ Stampa da una macchina Windows

Il vecchio sistema di stampa di Linux (LPD - Line Printer Daemon) rendeva problematica ogni operazione di stampa, specialmente in rete, tanto che prima di arrivare a una configurazione accettabile bisognava preventivare il disboscamento di una piccola fetta di Amazzonia, in termini di carta sprecata.

Lo standard CUPS (Common Unix Print System) ha semplificato molto le cose ed è utilizzato, oltre che da Linux e Unix, anche da Mac OS X.

Definendo sul sistema Debian un server CUPS, potremo offrire servizi di stampa sia per la macchina principale sia per le altre connesse in rete, indipendentemente dal sistema operativo che impiegano.



DA SAPERE *Inizialmente Linux gestiva solo la stampa a caratteri. Le prime stampanti dotate di funzionalità grafiche utilizzavano il linguaggio di definizione della pagina PostScript introdotto nel 1982 da Adobe. PostScript è tuttora un elemento fondamentale del sistema di stampa in Linux.*

Il sistema di stampa CUPS

CUPS (Common Unix Printing System) è un sistema di stampa in rete. Indipendentemente dal fatto che richiederemo la stampa su una stampante locale o remota, CUPS impiegherà sempre un protocollo di rete (normalmente IPP, Internet Printing Protocol, un'estensione di HTTP) che richiama il daemon CUPS, *cupsd*. Il servizio CUPS richiede l'uso della porta 631.

Un server CUPS che si trova a gestire una o più stampanti può pubblicizzare nella rete locale la loro presenza tramite messaggi broadcast, grazie ai quali le macchine sapranno dell'esistenza del server; se il server lo consente, potranno così richiedere la stampa senza ulteriori configurazioni.

Il servizio di stampa CUPS è già installato sul sistema, come possiamo verificare richiamando il comando Desktop > Amministrazione > Servizi che apre la finestra di dialogo Impostazioni servizi dove troveremo il Servizio di stampa (*cupsys*) (vedere la Figura 14.1).

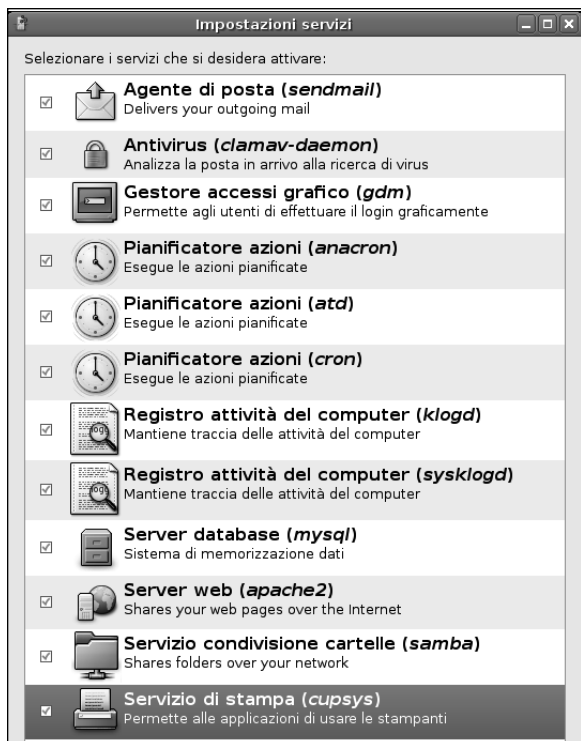


Figura 14.1

Il server CUPS è attivo e in attesa di richieste di stampa.

Installazione dei driver per una stampante locale

Per iniziare, dobbiamo connettere la stampante al sistema Linux e installare i relativi driver. Poi potremo gestirla tramite CUPS (già preinstallato sul sistema), in modo che offra i servizi di stampa a tutte le macchine situate sulla stessa rete locale. Ecco come possiamo installare una stampante utilizzando CUPS sul sistema Linux.

1. Colleghiamo e accendiamo la stampante alla porta appropriata (tipicamente la porta USB o parallela Centronics) della macchina Debian Linux
2. Apriamo il browser (Epiphany in questo caso, ma possiamo naturalmente usare qualsiasi altro browser) e digitiamo lo speciale indirizzo che richiama l'interfaccia di configurazione di CUPS: `http://localhost:631`. Verrà visualizzata la pagina Principale di CUPS, rappresentata nella Figura 14.2. Una bella sorpresa: è una comoda pagina Web interamente (o quasi) localizzata in italiano.

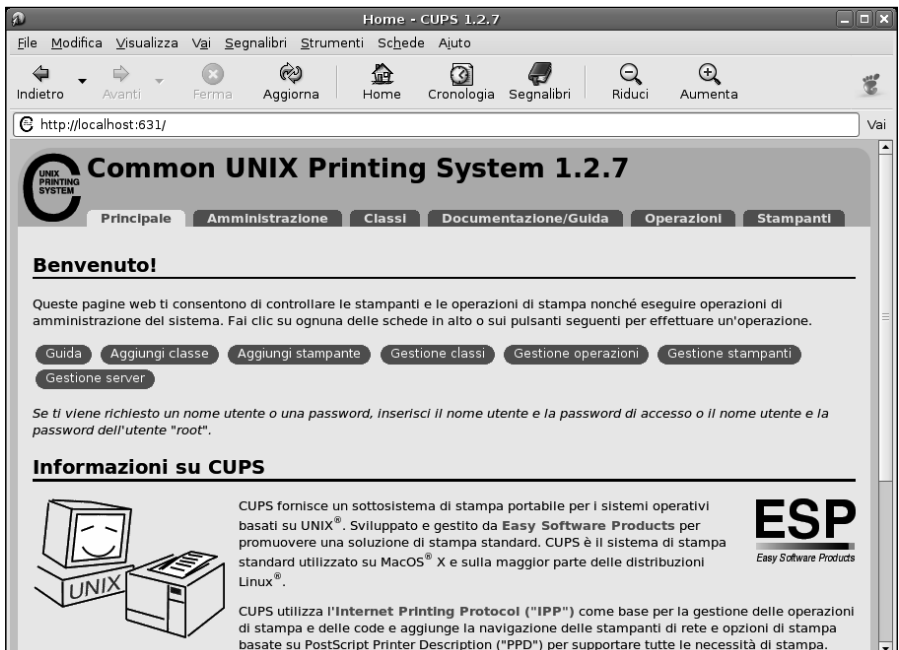


Figura 14.2

La pagina principale per l'installazione delle stampanti in CUPS.

- Facciamo clic sul pulsante ovale **Aggiungi stampante** per avviare la procedura omonima. Verrà visualizzata la pagina rappresentata nella Figura 14.3: **Amministrazione – Aggiungi stampante (Add New Printer)**. Nella casella **Name**, dobbiamo indicare la sigla (senza spazi) con cui verrà individuata la stampante; in **Location** possiamo specificare il nome della macchina sulla quale stiamo operando o la posizione/ufficio in cui si trova; in **Description** indichiamo invece un nome più descrittivo ed esteso per la stampante. Al termine facciamo clic sul pulsante **Continua**.



Figura 14.3

La pagina Web per l'aggiunta di una nuova stampante.

- Verrà visualizzata la pagina **Amministrazione – Dispositivo** per *nome stampante*, rappresentata nella Figura 14.4. Qui, nella casella **Dispositivo** dobbiamo specificare la porta alla quale è connessa la stampante. Facendo clic sulla casella verrà proposta una serie di opzioni, fra le quali dobbiamo solo scegliere la porta alla quale è connessa la nostra stampante. Al termine facciamo clic sul pulsante **Continua**.
- Nella pagina **Amministrazione – Modello/Driver** per *nome stampante*, rappresentata nella Figura 14.5, dobbiamo specificare nella grossa casella **Modello** il produttore e il modello della stampante. Quindi facciamo clic sul pulsante **Aggiungi stampante** per concludere l'installazione.
- Comparirà la finestra dei permessi (Figura 14.6), che concede l'accesso alla stampante all'utente corrente. Poiché stiamo eseguendo l'installazione



Figura 14.4
Scelta della porta cui è connessa la stampante locale.

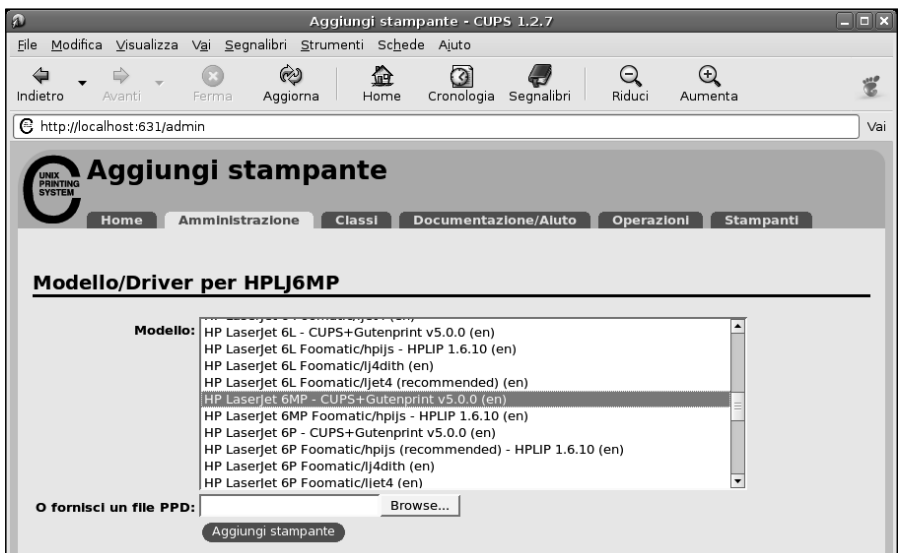


Figura 14.5
Scelta del driver per la stampante.

come utenti root, dovremo specificare la relativa password. La casella Use Password Manager to remember this password permette di memorizzare tale password per il futuro.

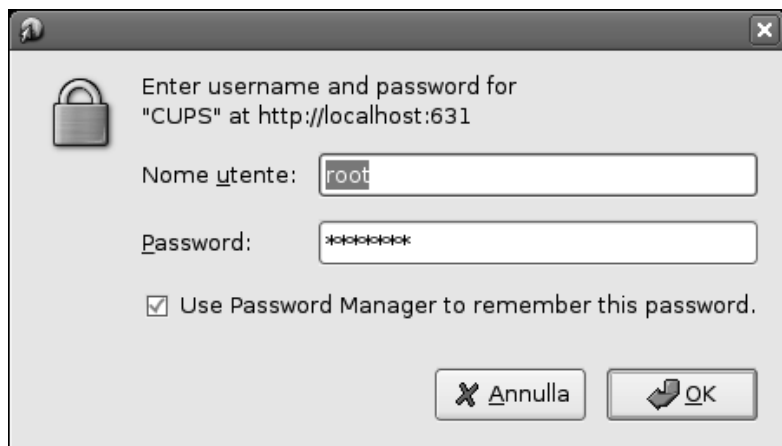


Figura 14.6
La finestra di accesso per la stampante.

7. Dopo una pagina intermedia di conferma dell'installazione, si aprirà la pagina Amministrazione – Imposta opzioni stampante (Figura 14.7) dove vengono proposte tutte le opzioni di stampa disponibili nel driver per la stampante appena installata. Le opzioni disponibili possono essere molto numerose e controllano aspetti specifici della stampante installata, quali la qualità di stampa, il formato della carta, il vassoio da cui verrà tratto il foglio, la stampa fronte/retro e così via. Dopo aver eventualmente personalizzato il comportamento della stampante, facciamo clic sul pulsante Imposta opzioni stampante per proseguire.
8. Dopo un breve messaggio di conferma, si aprirà la pagina principale della stampante, che possiamo vedere nella Figura 14.8. Qui troviamo un riepilogo della stampante che abbiamo appena installato e un insieme di comandi utili per gestire la stampante e per verificarne il funzionamento. Il primo pulsante è naturalmente Stampa pagina di prova, sul quale facciamo immediatamente clic per verificare che tutto abbia funzionato correttamente. La stampante produrrà un'apposita pagina di test contenente informazioni grafiche e geometriche sulle funzionalità disponibili per la stampante.
9. Un altro utile pulsante è Imposta come predefinita, che consente di nominare la stampante come predefinita per il sistema locale. Ciò significa che tutte le applicazioni in esecuzione sul sistema proporranno come prima cosa la stampa su questa stampante locale. Possiamo verificare che la nuova stampante sia stata riconosciuta dall'intero sistema richiamando una qualsiasi attività di stampa (per esempio dallo stesso browser Web Epiphany e veri-



Figura 14.7

La pagina contenente le opzioni di stampa specifiche della stampante in uso.

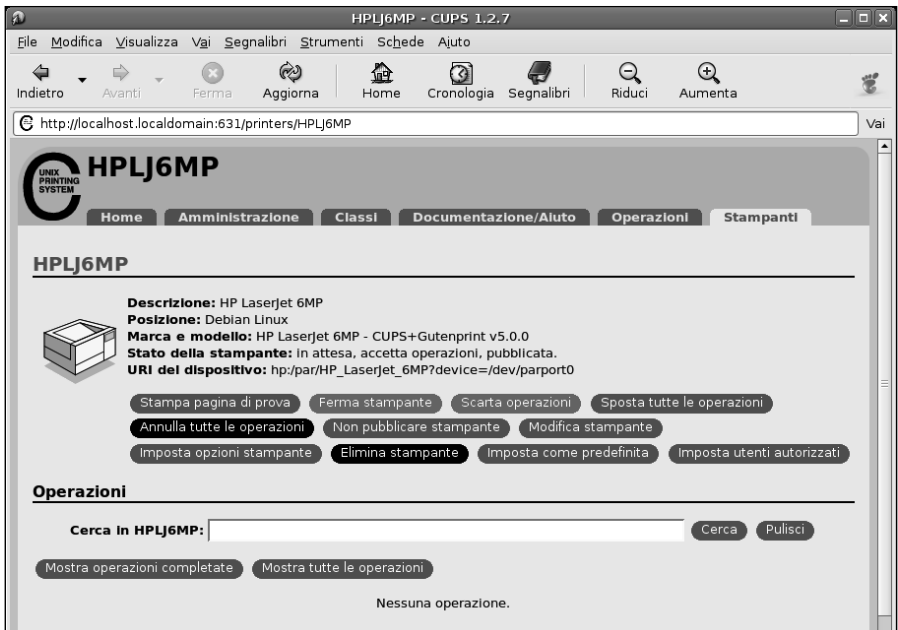


Figura 14.8

La pagina di amministrazione della stampante che abbiamo appena installato.

ficare che nella finestra di dialogo Stampa venga proposta proprio la stampante che abbiamo appena installato. Inoltre, se selezioniamo dal desktop Gnome di Debian il comando Desktop > Amministrazione > Printing, verrà visualizzata la finestra di dialogo Stampanti nella quale troveremo proprio la stampante che abbiamo appena installato, sormontata dal segno di spunta che la indica come stampante predefinita del sistema. Questa situazione è rappresentata nella Figura 14.9.

10. Poi facciamo clic sul pulsante Imposta utenti autorizzati per richiamare la pagina omonima del software di gestione CUPS. Nella casella Utenti dobbiamo specificare il nome degli utenti che possono impiegare la stampante, separati da una virgola. Selezioniamo l'opzione Consenti a questi utenti di stampare e poi facciamo clic sul pulsante Imposta utenti autorizzati. Tali utenti verranno aggiunti all'elenco degli utenti per i quali è disponibile

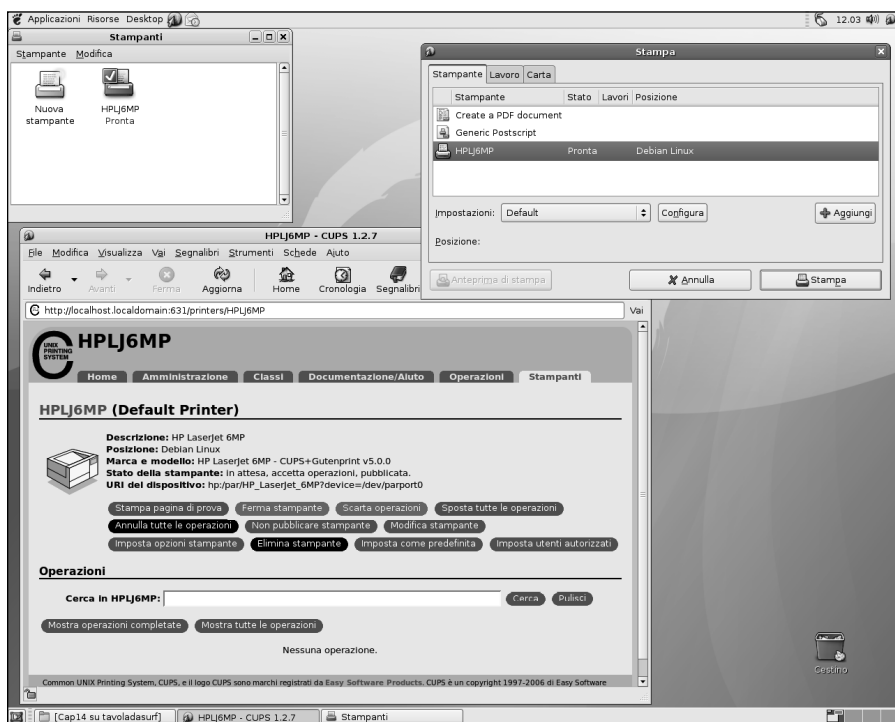


Figura 14.9

In questa figura si nota, in basso a sinistra il browser Web Epiphany aperto sulle pagine di configurazione della stampante e la sua finestra di dialogo Stampa dove possiamo trovare la stampante appena installata; in alto a sinistra mostriamo anche la finestra Stampanti, che conferma che questa stampante disponibile sul sistema.

la stampante. Analogamente possiamo ripetere questa stessa operazione selezionando la casella Impedisci a questi utenti di stampare e indicare tutti quegli utenti cui non è consentita la stampa su questa stampante. La situazione è rappresentata nella Figura 14.10.

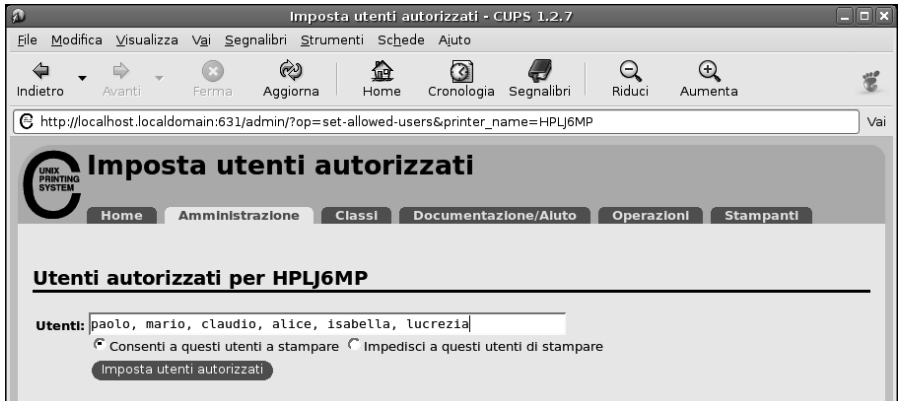


Figura 14.10

Indichiamo gli utenti in cui è consentito l'utilizzo della stampante.

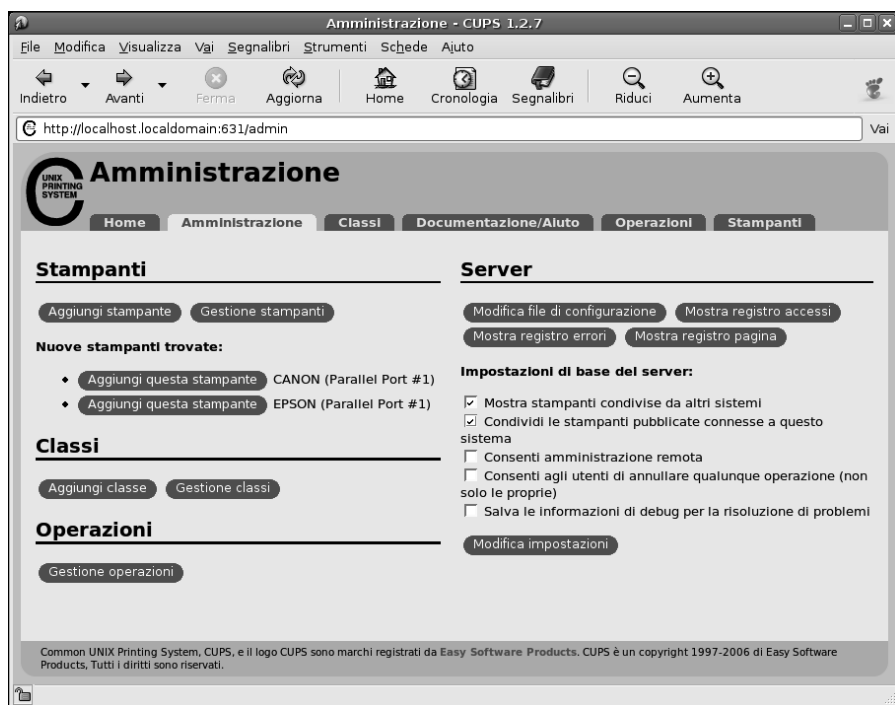
11. Tornati alla pagina principale, facciamo clic sulla scheda **Amministrazione** in alto, dove possiamo amministrare il comportamento del server CUPS. In particolare, in questo caso, dobbiamo intervenire sulle opzioni presentate nella colonna **Server** che si trova sul lato destro della pagina (Figura 14.11). In particolare dobbiamo selezionare le seguenti opzioni.

Condividi le stampanti pubblicate connesse a questo sistema - Se, come in effetti vogliamo, la stampante che abbiamo configurato deve risultare visibile anche da parte delle altre macchine della rete, dobbiamo selezionare questa casella.

Consenti amministrazione remota - Qui è solo una questione di scelte: se pensiamo di configurare sempre il sistema di stampa dalla macchina Debian locale, non vi è alcun motivo per selezionare questa casella.

Consenti agli utenti di annullare qualunque operazione (non solo le proprie) - In genere non si deve offrire questo permesso agli utenti: ognuno deve poter annullare solo le proprie operazioni di stampa.

Salva le operazioni di debug per la risoluzione dei problemi - Non è un'opzione necessaria per il funzionamento del server, possiamo attivarla solo nel caso dovessimo incontrare dei problemi.

**Figura 14.11**

Attiviamo la condivisione della stampante sulla rete locale selezionando l'opzione Condividi le stampanti pubblicate connesse a questo sistema.

Al termine facciamo clic sul pulsante **Modifica impostazioni** per attivare le opzioni modificate del server, principalmente la condivisione della stampante nella rete locale. Il server verrà automaticamente riavviato per comprendere le modifiche appena apportate.

Questo conclude la configurazione del server CUPS.

Stampa dalla macchina locale

Per prima cosa, proviamo a utilizzare un'applicazione qualsiasi per sperimentare il funzionamento locale della stampante che abbiamo appena installato tramite CUPS.

Per esempio proviamo ad aprire un documento qualsiasi con l'applicazione di OpenOffice.org. Scegliamo il comando **File | Stampa** e nella finestra di dialogo **Stampa** dovrebbe essere selezionata proprio la stampante che abbiamo appena installato. Configuriamo le altre opzioni di stampa e proviamo a lanciare la

stampa facendo clic sul pulsante OK (Figura 14.12). Il documento verrà così inviato alla stampante. Tutto molto semplice e immediato.

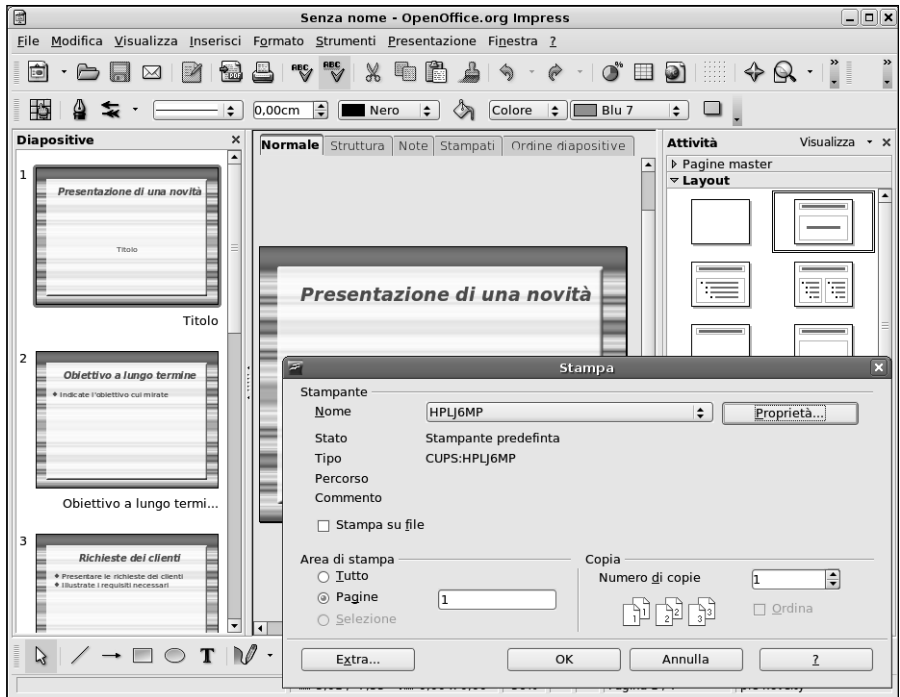


Figura 14.12

Le operazioni di stampa da un'applicazione del pacchetto OpenOffice.org, in questo caso Impress.

Stampa da una macchina Windows

Le operazioni di stampa da una macchina Windows sono leggermente complicate dal fatto che la stampante non risulterà visibile come una risorsa di rete disponibile sulla macchina Debian.

Questo in realtà non rappresenta un grosso problema, ma ci costringe a specificare manualmente la posizione in cui la macchina Windows dovrà andare a cercare il server CUPS e chiedergli di utilizzare proprio la stampante che abbiamo appena installato. Per prima cosa dobbiamo installare sulla macchina Windows una stampante di rete. L'operazione viene in questo caso svolta su una macchina Windows XP, ma con Windows Vista dovremo svolgere operazioni sostanzialmente identiche.

Prima di tutto vediamo cosa ci mostra la finestra **Risorse di rete**: scegliamo nel riquadro **Operazioni di rete** sul lato sinistro il comando **Visualizza computer del gruppo di lavoro**. Come possiamo vedere nella Figura 14.13, verranno visualizzate le macchine che compongono la rete, fra le quali si trova anche il nostro server Debian.

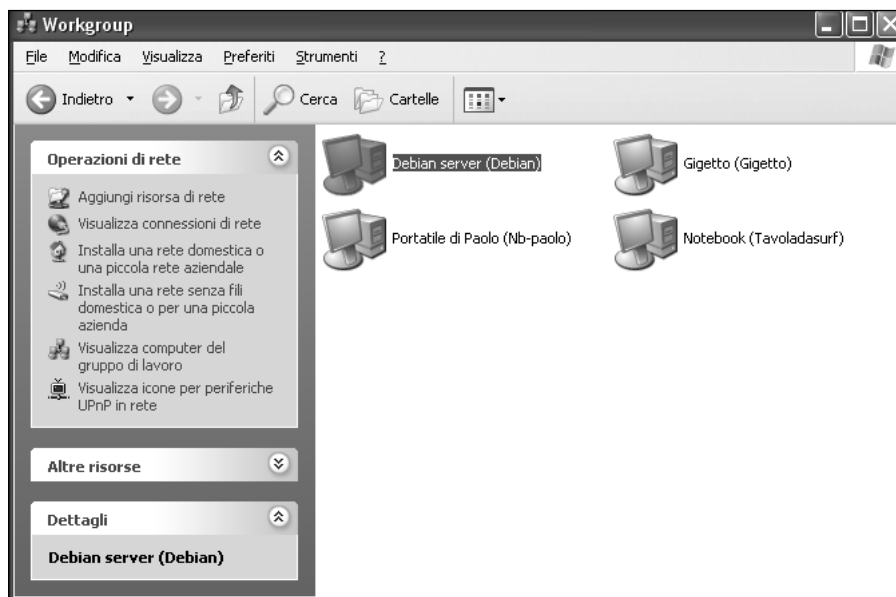


Figura 14.13
Il server è visibile tra le macchine che formano la rete.

Un doppio clic sulla macchina Debian, ci aprirà le risorse condivise, tra le quali troveremo anche la cartella **Stampanti e fax**. Fiduciosi, facciamo doppio clic su questa icona, ma la Figura 14.14 mostra il risultato: assolutamente nulla; sembra che sulla macchina Debian non sia disponibile alcuna stampante. Non è esattamente così, anzi, tutt'altro.

Apriamo ora il **Pannello di controllo** di Windows e qui selezioniamo la categoria **Stampanti e fax**. Si aprirà la pagina delle stampanti installate sul nostro sistema. Nel riquadro **Operazioni stampante**, in alto a sinistra, selezioniamo il comando **Aggiungi stampante**, visibile nella Figura 14.15. Si aprirà la procedura **Installazione guidata stampante**. La prima pagina è di presentazione e possiamo proseguire facendo clic semplicemente sul pulsante **Avanti**. La seconda pagina della procedura, **Stampante locale o di rete**, chiede di indicare la col-

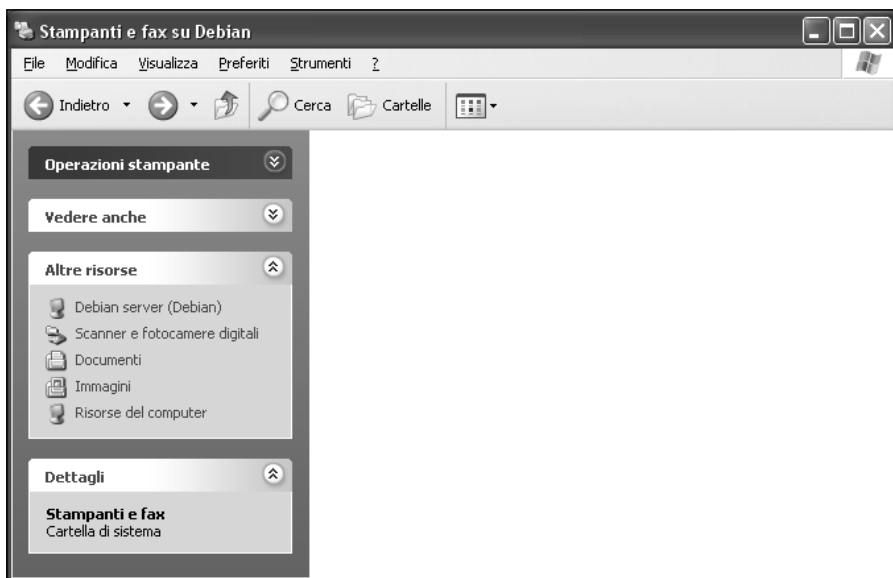


Figura 14.14

È proprio vero che sulla macchina Debian non è disponibile alcuna stampante?

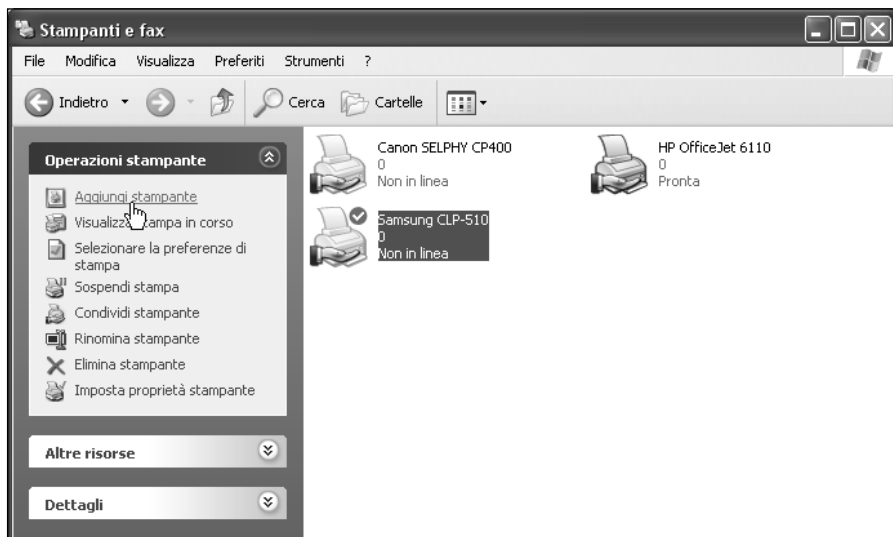


Figura 14.15

Aggiungiamo al nostro sistema la stampante CUPS installata sul server Debian.

locazione della stampante che intendiamo installare. Naturalmente dobbiamo scegliere l'opzione Stampante di rete o stampante collegata a un altro computer (Figura 14.16) e poi fare clic sul pulsante Avanti.

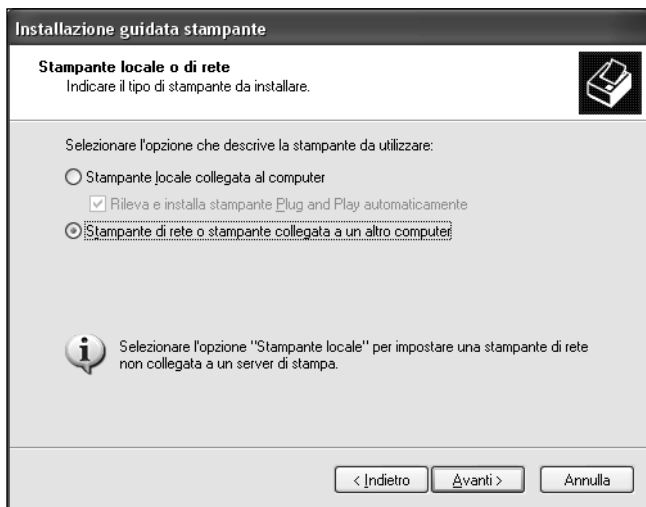


Figura 14.16
Scegliamo di installare una stampante di rete.

Nella pagina Specificare una stampante, selezioniamo la seconda opzione: Connetti alla stampante. Per cercarne una in rete selezionare l'opzione e scegliere Avanti. Qui, nella casella Nome, (Figura 14.17) dobbiamo specificare un particolare indirizzo che identifica la stampante di rete:

`http://192.168.1.5:631/printers/HPLJ6MP`

L'esempio si riferisce ovviamente a una determinata macchina su una determinata rete con una determinata stampante. Nella casella Nome dovremo pertanto scrivere qualcosa di simile a:

`http://indirizzo-ip:631/printers/nome-stampante`

sostituendo a *indirizzo-ip* e *nome-stampante* rispettivamente l'indirizzo IP della macchina Debian e il nome che abbiamo attribuito alla stampante da CUPS.

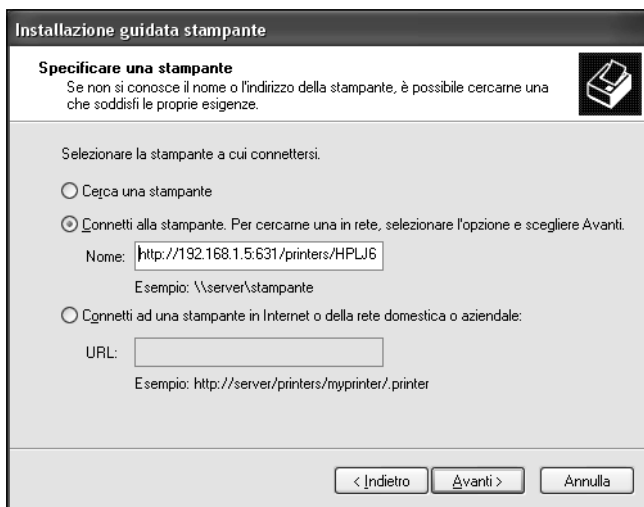


Figura 14.17

Indichiamo l'indirizzo di rete e il nome della stampante cui vogliamo accedere.

Il numero 631, obbligatorio, indica la porta sul sistema Debian da cui il server CUPS si aspetta di essere contattato. Per proseguire facciamo clic sul pulsante **Avanti**. Windows contatterà il server CUPS e si assicurerà che tale stampante sia effettivamente presente sul sistema. Nella pagina successiva (Figura 14.18), dobbiamo indicare il **Produttore** e il modello della stampante in questione.

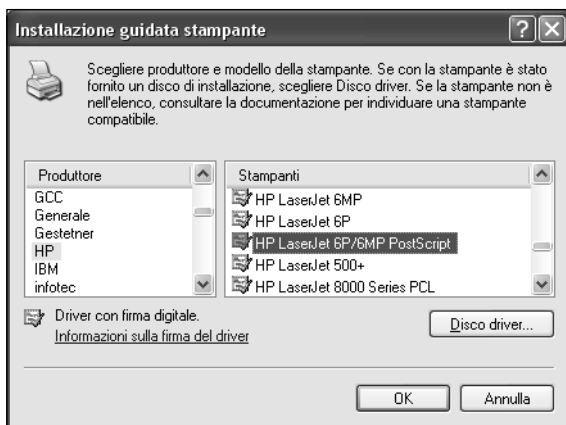


Figura 14.18

Scelta del driver corretto per la stampante.

Naturalmente qui dovremo semplicemente scegliere il driver corretto per la stampante. Per proseguire, facciamo clic sul pulsante **Avanti**.

La pagina successiva ci chiede se vogliamo impostare questa stampante come la Stampante predefinita del sistema. Rispondiamo in base alle nostre esigenze e poi facciamo clic sul pulsante **Avanti**. La pagina finale conferma l'installazione della stampante sulla macchina Debian. Per concludere possiamo fare clic sul pulsante **Fine**.

Ora possiamo osservare l'aspetto della finestra **Stampanti e fax** del Pannello di controllo per confermare il fatto che la stampante di rete è installata e operativa (Figura 14.19). Non ci rimane che provare a stampare un documento qualsiasi utilizzando le metodologie di stampa tipiche delle applicazioni Windows. L'applicazione contatterà il computer Debian sulla porta 631; lì troverà in ascolto il server CUPS che, obbediente, eseguirà l'operazione di stampa.

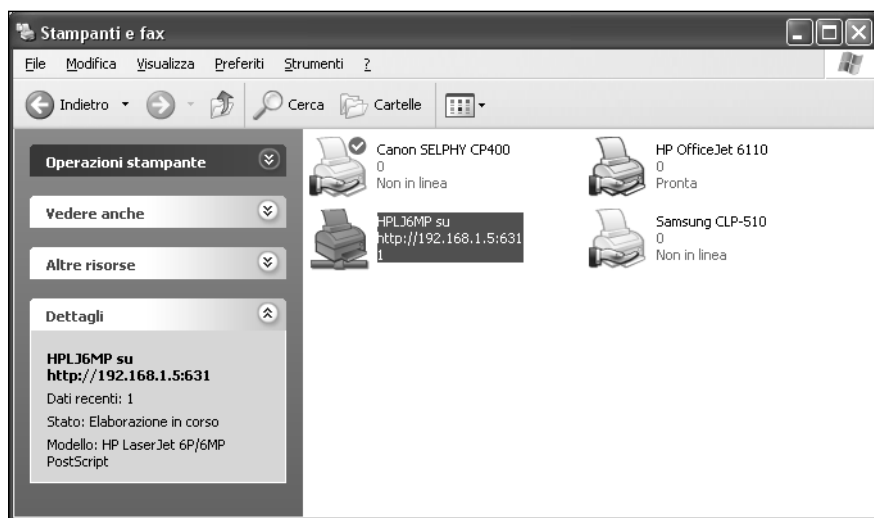


Figura 14.19

La nuova stampante di rete installata sulla macchina Windows.

Conclusioni

In questo capitolo abbiamo trattato le attività di stampa e le tecniche di condivisione della stampante Linux tramite un server CUPS

Glossario

I termini tecnici impiegati nel volume.

Access Point

Un sistema wireless che offre una connessione a una rete o a Internet. La connessione offerta da un nostro router/gateway wireless è un tipico esempio di Access Point privato. Quella invece offerta da un provider Internet in un aeroporto o in un altro ambiente è un tipico esempio di Access Point pubblico.

Accesso remoto

Connessione di rete effettuata tramite sessioni tradizionali punto-a-punto, telefoniche o a rete privata virtuale.

ACL (Access Control List)

In un firewall protegge gli accessi stabilendo chi può e chi non può accedere a una determinata risorsa.

Ampiezza di banda

La velocità teorica di trasmissione di un elemento della rete.

Architettura client/server

Architettura che segue il modello di flussi fra client e server. Una macchina (il client) esegue richieste o interrogazioni e l'altra (il server) risponde fornendo (o meno) le informazioni richieste. Un tipico esempio è rappresentato dal Web: il client Web (il browser) richiede una pagina e il server Web risponde restituendo la pagina Web richiesta o impedendo l'accesso a una pagina riservata.

Architettura intranet/extranet

Architettura che privilegia la sicurezza e la privacy, comprendendo la separazione di utenti, dispositivi e applicazioni sulla base di accessi sicuri.

Broadcast

Speciale indirizzo di rete tramite il quale vengono raggiunte tutte le macchine della rete. Viene utilizzato per comunicare contemporaneamente con tutte le macchine della rete. Per esempio, nella rete locale con indirizzi privati 192.168.1.x, l'indirizzo broadcast sarà l'ultimo disponibile, ovvero 192.168.1.255.

Client

In una relazione client/server si tratta della parte richiedente. Può trattarsi di un client Web (browser), di posta elettronica (il software che usiamo per scaricare la posta), FTP e così via.

Crittografia

Meccanismo di sicurezza che prevede l'applicazione di algoritmi di cifratura e di una chiave segreta per codificare i dati in modo che risultino illeggibili anche se dovessero essere intercettati.

Daemon

Applicazioni di sistema che restano costantemente in funzione in background, offrendo un servizio ad altri programmi o sistemi.

DHCP (Dynamic Host Configuration Protocol)

Protocollo di assegnazione degli indirizzi IP alle macchine di una rete locale o pubblica.

Dipendenze fra componenti

Requisiti imposti da un componente su uno o più altri componenti per poter funzionare. In Linux, i pacchetti dipendono normalmente da altri pacchetti per

poter funzionare. Per questo motivo, quando si chiede l'installazione di un pacchetto, normalmente viene installato un corollario di pacchetti dipendenti.

Distribuzione

Linux è liberamente distribuibile e personalizzabile, in base a una licenza che obbliga a fornirlo gratuitamente imponendo eventualmente solo il costo dei servizi offerti (CD/DVD, documentazione, confezione ma, soprattutto assistenza tecnica professionale).

DNS (Domain Name System)

Servizio di traduzione dei nomi comunemente impiegati in Internet (per esempio per i siti Web) nei corrispondenti indirizzi IP.

Downstream

Traffico che segue un percorso da un server alla macchina locale.

DSL (Digital Subscriber Loop)

Tecnica digitale per la trasmissione delle informazioni ad alta velocità tramite un normale doppino telefonico.

Ethernet

Tecnologia di rete largamente impiegata per la realizzazione di reti locali. Col tempo ha offerto velocità sempre maggiori, cambiando anche le caratteristiche dell'infrastruttura di rete impiegata (cablaggi e schede).

Extranet

Rete locale aziendale che si estende su più sedi o che comprende i fornitori e clienti più importanti dell'azienda. Per raggiungerli sfrutta passaggi crittografati all'interno della rete pubblica Internet. Utilizzata per lo scambio di informazioni strategiche dell'azienda e delle sue associate.

Filtraggio di routing

Il router rappresenta e applica un filtro per nascondere la rete interna alla rete esterna (tipicamente Internet).

Firewall

Si tratta di uno o più meccanismi di sicurezza, implementati in software nei dispositivi o negli elementi di rete (normalmente nei router) collocati in punti strategici di accesso alla rete.

FTP (File Transfer Protocol)

Protocollo per il trasferimento di file. Rende possibile il trasferimento di file di grandi dimensioni nell'ambito di una connessione client/server.

Gateway

Elemento della rete attraverso il quale una rete può connettersi a una rete più grande (come Internet).

Indirizzamento

Assegnazione alle macchine della rete di indirizzi locali o globali, privati o pubblici, temporanei o persistenti.

Indirizzo IP

Numero a 32 bit che identifica un dispositivo a livello della rete. Normalmente rappresentato con una sequenza di quattro numeri separati da un punto, come in 192.168.1.1.

Indirizzo IP privato

Indirizzo IP che non può essere pubblicizzato dagli elementi della rete locale al pubblico dominio (normalmente Internet).

Indirizzo IP pubblico

Indirizzo IP che può e deve essere pubblicizzato nel dominio pubblico (tipicamente Internet) dagli elementi e dai dispositivi di rete.

Intranet

Rete locale interna all'azienda che funziona come una Internet privata e in miniatura. Viene utilizzata per lo scambio di informazioni all'interno dell'azienda.

ISP (Internet Service Provider)

Società che ci offre la connessione alla rete Internet.

LAN (Local Area Network)

Rete locale di computer. Quella che creiamo all'interno dell'azienda o dell'abitazione quando colleghiamo insieme più macchine è un tipico esempio di LAN.

Latenza

Velocità di risposta di un sistema o di un componente: valuta il ritardo che considera l'elaborazione svolta dal dispositivo e dall'applicazione e comprende il tempo necessario per completare un compito.

Maschera di sottorete

Maschera per gli indirizzi IP che aggiunge un ulteriore livello di gerarchia all'insieme degli indirizzi IP.

Peer

Utenti, applicazioni, dispositivi o reti che agiscono allo stesso livello. Per esempio, in un sistema client/server, i server che agiscono allo stesso livello (ovvero sopra i loro client) vengono considerati peer.

Peer-to-peer

Modello di flussi in cui gli utenti e le applicazioni si connettono direttamente fra loro senza l'ausilio di un server centralizzato.

PING (Packet InterNet Groper)

Pacchetto inviato in rete per sondare la risposta del sistema posto all'altra estremità.

Reti private virtuali

Tramite sistemi a tunnel è possibile unire reti isolate che comunicano attraverso un'infrastruttura comune.

Server

In una relazione client/server si tratta della parte che fornisce le informazioni. Può trattarsi di un server Web (come Apache), di posta elettronica (il software che contattiamo per scaricare la nostra posta), FTP e così via.

Servizi di rete

Ogni server in funzione sulla macchina e sulla rete fornisce un servizio. Per esempio, il server DHCP offre un servizio di assegnazione degli indirizzi e il server DNS offre un servizio di traduzione fra nomi e indirizzi IP.

Sicurezza

Requisito che garantisce l'integrità (precisione e autenticità) delle informazioni, delle risorse e dell'accesso a tali risorse da parte degli utenti del sistema e della rete.

TCP (Transmission Control Protocol)

Il protocollo su cui si basa l'intera infrastruttura di Internet.

Tempo di risposta umana

Stima del tempo trascorso il quale gli utenti iniziano a percepire un ritardo nel sistema.

Throughput

Misura la produttività (velocità) del sistema o degli elementi della rete.

Traduzione dell'indirizzo di rete

Conversione degli indirizzi IP da un ambito a un altro. Normalmente è necessaria fra lo spazio di indirizzamento della rete locale privata e quella pubblica (Internet).

Upstream

Traffico che segue un percorso dalla macchina locale a un server.

VLAN (Virtual LAN)

Una rete locale virtuale creata interconnettendo sistemi distanti tramite tunnel aperti nella rete pubblica Internet. Indipendentemente dalla distanza (100 m o 100/1000/10000 km) le macchine connesse sembreranno risiedere sulla stessa rete locale. Necessariamente la velocità di connessione delle macchine remote subirà le limitazioni del collegamento via Internet.

VoIP (Voice over IP)

Protocollo di comunicazione audio digitale via Internet. Rende possibili conversazioni “telefoniche” a costo zero sfruttando la banda disponibile in Internet. È alla base di sistemi come il notissimo Skype.

WAN (Wide Area Network)

Una rete che si estende su un'area geografica di dimensioni considerevoli. Spesso è formata da tante sottoreti locali (LAN) più piccole.

Workgroup

Gruppo di lavoro: il nome di una rete alla quale possono connettersi le macchine di una LAN.

