Virtual Services Howto

Brian Ackerman, `brian@nycrc.net` v2.1, 15 Agosto 1998

Questo documento è stato scritto per soddisfare la crescente richiesta di informazioni sui servizi virtuali. Traduzione a cura di Riccardo Fabris `skizzo at mail.seta.it`, su contributo iniziale di gAsp `gasp at eponet.it`, nel Maggio 1999. Un grazie ai correttori della traduzione (l'onnipresente Giovanni Bortolozzo e gAsp).

Indice

1 Introduzione

1.1 Conoscenze richieste

Creare un sistema per servizi virtuali non è troppo difficile, tuttavia è richiesto qualcosa di più che una conoscenza di base. Questo documento non è un'introduzione alla configurazione completa di una macchina Linux.

Per una piena comprensione di questo documento è necessario avere assoluta familiarità con quanto segue:

- Compilazione del kernel Linux e aggiunta del supporto IP aliasing *IP alias mini-HOWTO*
- Installazione e configurazione di dispositivi di rete *NET-3 HOWTO*
- Configurazione di inetd *NET-3 HOWTO*
- tradotto in italiano: *NET-3 HOWTO*
- Pacchetti vari per il networking quali: *Sendmail*
  Apache
  Qmail
  SAMBA
- Impostazione del DNS *DNS HOWTO*
- Conoscenze base di amministrazione di sistemi *Linux Systems Administrators's Guide*

- tradotto in italiano: *Guida dell'Amministratore di Sistema*
- Conoscenze base sulla configurazione di un server web *WWW HOWTO*
- tradotto in italiano: *WWW-HOWTO*

Se non si è sicuri di conoscere le procedure concernenti uno qualsiasi dei componenti citati sopra, è FORTEMENTE raccomandato di prendere confidenza con tutti i pacchetti facendo riferimento ai link riportati. Io NON risponderò a messaggi di posta riguardanti gli argomenti sopra indicati. Rivolgete le vostre domande agli autori dei rispettivi HOWTO.

1.2 Scopo

La funzione dei servizi virtuali è quella di permettere ad una singola macchina di riconoscere indirizzi IP multipli senza il bisogno di schede di rete multiple. L'IP aliasing è un'opzione di compilazione del kernel che permette di assegnare a ciascuna interfaccia di rete più di un'indirizzo IP. Esso permette al kernel di gestire simultaneamente più indirizzi IP in modo trasparente, saltando da uno all'altro in rapida successione ('multiplexing'). All'utente sembrerà che ci sia più di un server.

Il 'multiplexing' permette che domini multipli (www.dominio1.com, www.dominio2.com eccetera) vengano ospitati sulla stessa macchina allo stesso costo di un unico dominio. Sfortunatamente la maggior parte dei servizi (FTP, web, mail) non sono stati progettati per gestire domini multipli. Allo scopo di farli lavorare correttamente è quindi necessario modificare sia i file di configurazione che il codice sorgente. Questo documento descrive come apportare queste modifiche nel corso dell'impostazione di una macchina virtuale.

Per poter far funzionare i servizi virtuali è anche richiesto un demone. I sorgenti di questo demone (virtuald) vengono forniti più avanti in questo documento.

1.3 Commenti e critiche

Questo documento si espanderà man mano che i pacchetti verranno aggiornati e cambieranno le modifiche da apportare ai sorgenti o alla configurazione. Se ci sono parti di questo documento che non sono chiare potete mandarmi una e-mail con suggerimenti o domande. Allo scopo di facilitarmi il lavoro siete pregati di fare commenti specifici e di includere la sezione in questione. È importante che il messaggio di posta contenga la dicitura 'VIRTSERVICES HOWTO' nel soggetto. Qualunque altro messaggio verrà considerato personale e, come sanno bene tutti i miei amici, io non sono solito leggere la mia posta personale ed è perciò probabile venga scartato assieme ai loro.

Notate anche che i miei esempi sono solo esempi, quindi non dovrebbero essere copiati pari pari. Potreste avere infatti bisogno di inserire i vostri valori. Se avete dei problemi, potete inviarmi un'e-mail. Allegate tutti i file di configurazione pertinenti e i messaggi di errore che ricevete durante l'installazione. Ci darò un occhiata e vi invierò i miei suggerimenti.

1.4 Archivio storico delle revisioni

V1.0

Versione originaria

V1.1

Corretto un errore nella sezione Web Virtuale.

V1.2

Corretta la data.

V2.0

Aggiornati i link html.

Aggiornamenti alla sezione Web.

Nuova opzione di Sendmail.

Nuova sezione su Qmail.

Aggiornata la sezione sul Syslog.

Aggiornata la sezione sull'FTP.

Opzione predefinita di Virtuald.

Nuova sezione su Samba.

Aggiornate le FAQ.

V2.1

Cambiati tutti i percorsi in /usr/local.

Aggiunta l'opzione di compilazione VERBOSELOG a virtuald.

Corretto un bug di setuid/setgid in virtmailfilter.

Corretto un bug di execl in virtmailfilter.

Corretto un bug nella trasformazione minuscole/maiuscole in virtmailfilter.

Corretta la variabile di ambiente sanity check in virtmailfilter.

Tolto il codice mbox da virtmailfilter/virtmaildelivery.

Aggiunta la sezione tcpserver.init pop per Qmail.

Aggiunta la sezione riguardante gli alias dei nomi di dominio alle FAQ.

1.5 Copyright/Distribuzione

[Questa parte viene lasciata in originale per motivi legali N.d.T.]

2 IP Aliasing

L'IP Aliasing è un'opzione di compilazione del kernel che deve essere abilitata per permettere il 'virtual hosting'. Esiste già un mini-HOWTO sull'argomento *IP aliasing* . Si prega di consultarlo per questioni che riguardano la sua impostazione.

3 Virtuald

3.1 Introduzione

Ogni connessione di rete è basata su due coppie di indirizzi IP e porte. L''API' (Applications Program Interface) per la programmazione di rete viene chiamata 'Socket API'. Un 'socket' si comporta come un file aperto, con operazioni di lettura/scrittura su di esso è possibile scambiare dati su una connessione di rete. C'è una funzione chiamata `getsockname` che restituisce l'indirizzo IP del socket locale. Virtuald in primo luogo utilizza `getsockname` per determinare a quale indirizzo IP della macchina locale si vuole accedere. Quindi legge da un file di configurazione quale directory è associata a tale indirizzo IP. Virtuald fa `chroot` a quella directory e passa la connessione al servizio. `Chroot` reimposta '/', la directory radice, in un nuovo punto dell'albero delle directory, in modo tale che il programma in esecuzione non possa accedere a nulla fuori da questo ramo. Quindi ogni indirizzo IP è associato ad un proprio filesystem virtuale. Tutto ciò è trasparente per il programma di rete, che si comporterà come se niente fosse successo. Virtuald può quindi essere utilizzato insieme con un programma come inetd per rendere virtuale un servizio.

3.2 Inetd

Inetd è un super server di rete che sta in ascolto su varie porte e, quando riceve una connessione (ad esempio, una richiesta pop in entrata), effettua la fase di negoziazione e passa la connessione ad un programma che gestisce lo specifico servizio. Questo per evitare che vengano eseguiti dei servizi che restano inattivi quando inutilizzati.

Un file /etc/inetd.conf standard appare così:

```
ftp stream tcp nowait root /usr/sbin/tcpd \
        wu.ftpd -l -a
pop-3 stream tcp nowait root /usr/sbin/tcpd \
        in.qpop -s
```

Il file /etc/inetd.conf di un sistema in cui si utilizza virtuald appare così:

```
ftp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.ftp wu.ftpd -l -a
pop-3 stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.pop in.qpop -s
```

3.3 File di configurazione

Ciascun servizio ha un file di configurazione che controlla quali indirizzi IP e directory sono autorizzati per quel servizio. Si può avere o un unico file principale oppure più file di configurazione, se si desidera che ad ogni servizio sia associato una lista diversa di domini. Un tipico file di configurazione appare così:

```
# Questo è un commento e allo stesso modo vengono trattate le linee vuote

# Formato: IndirizzoIP [spazio] directory [nessun spazio]
10.10.10.129 /virtual/domain1.com
10.10.10.130 /virtual/domain2.com
10.10.10.157 /virtual/domain3.com

# Opzione predefinita per tutti gli altri indirizzi IP
default /
```

3.4 Codice sorgente

Questo è il codice sorgente in C del programma virtuald. È necessario compilarlo e installarlo in /usr/local/bin con permessi 0755, utente root, e gruppo root. L'unica opzione di compilazione è VERBOSELOG che attiva/disattiva la registrazione delle connessioni nei file di log:

```
#include <netinet/in.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdarg.h>
#include <unistd.h>
#include <string.h>
#include <syslog.h>
#include <stdio.h>

#undef VERBOSELOG

#define BUFSIZE 8192

int getipaddr(char **ipaddr)
{
        struct sockaddr_in virtual_addr;
        static char ipaddrbuf[BUFSIZE];
        int virtual_len;
        char *ipptr;

        virtual_len=sizeof(virtual_addr);
        if (getsockname(0,(struct sockaddr *)&virtual_addr,&virtual_len)<0)
        {
                syslog(LOG_ERR,"getipaddr: getsockname failed: %m");
                return -1;
        }
        if (!(ipptr=inet_ntoa(virtual_addr.sin_addr)))
        {
                syslog(LOG_ERR,"getipaddr: inet_ntoa failed: %m");
                return -1;
        }
        strncpy(ipaddrbuf,ipptr,sizeof(ipaddrbuf)-1);
        *ipaddr=ipaddrbuf;
        return 0;
```

```
}

int iptodir(char **dir,char *ipaddr,char *filename)
{
        char buffer[BUFSIZE],*bufptr;
        static char dirbuf[BUFSIZE];
        FILE *fp;

        if (!(fp=fopen(filename,"r")))
        {
                syslog(LOG_ERR,"iptodir: fopen failed: %m");
                return -1;
        }
        *dir=NULL;
        while(fgets(buffer,BUFSIZE,fp))
        {
                buffer[strlen(buffer)-1]=0;
                if (*buffer=='#' || *buffer==0)
                        continue;
                if (!(bufptr=strchr(buffer,' ')))
                {
                        syslog(LOG_ERR,"iptodir: strchr failed");
                        return -1;
                }
                *bufptr++=0;
                if (!strcmp(buffer,ipaddr))
                {
                        strncpy(dirbuf,bufptr,sizeof(dirbuf)-1);
                        *dir=dirbuf;
                        break;
                }
                if (!strcmp(buffer,"default"))
                {
                        strncpy(dirbuf,bufptr,sizeof(dirbuf)-1);
                        *dir=dirbuf;
                        break;
                }
        }
        if (fclose(fp)==EOF)
        {
                syslog(LOG_ERR,"iptodir: fclose failed: %m");
                return -1;
        }
        if (!*dir)
        {
                syslog(LOG_ERR,"iptodir: ip not found in conf file");
                return -1;
        }
        return 0;
}
```

```
int main(int argc,char **argv)
{
        char *ipaddr,*dir;

        openlog("virtuald",LOG_PID,LOG_DAEMON);

#ifdef VERBOSELOG
        syslog(LOG_ERR,"Virtuald Starting: $Revision: 1.49 $");
#endif
        if (!argv[1])
        {
                syslog(LOG_ERR,"invalid arguments: no conf file");
                exit(0);
        }
        if (!argv[2])
        {
                syslog(LOG_ERR,"invalid arguments: no program to run");
                exit(0);
        }
        if (getipaddr(&ipaddr))
        {
                syslog(LOG_ERR,"getipaddr failed");
                exit(0);
        }
#ifdef VERBOSELOG
        syslog(LOG_ERR,"Incoming ip: %s",ipaddr);
#endif
        if (iptodir(&dir,ipaddr,argv[1]))
        {
                syslog(LOG_ERR,"iptodir failed");
                exit(0);
        }
        if (chroot(dir)<0)
        {
                syslog(LOG_ERR,"chroot failed: %m");
                exit(0);
        }
#ifdef VERBOSELOG
        syslog(LOG_ERR,"Chroot dir: %s",dir);
#endif
        if (chdir("/")<0)
        {
                syslog(LOG_ERR,"chdir failed: %m");
                exit(0);
        }
        if (execvp(argv[2],argv+2)<0)
        {
                syslog(LOG_ERR,"execvp failed: %m");
                exit(0);
```

```
        }

        closelog();

        exit(0);
}
```

4 Gli script di shell

4.1 Virtfs

Ciascun dominio dovrebbe avere una propria struttura di directory. Dal momento che si sta usando `chroot` bisognerà inserirvi un duplicato di tutti i file necessari, come librerie condivise, file binari, file di configurazione eccetera. Io utilizzo /virtual/domain1.com per ciascun dominio che creo.

Tutto ciò occupa dello spazio su disco, ma è comunque meno costoso di una nuova macchina con tanto di schede di rete. Se è veramente necessario risparmiare spazio su disco, si possono collegare insieme tutte le copie dei file con degli hard link, in modo che esista effettivamente solo una copia di ogni file binario. Il filesystem che utilizzo io occupa poco più di 2Mbyte. Comunque lo script che segue tenta di copiare tutti i file dal filesystem principale in modo da essere il più generico possibile.

Ecco un esempio di semplice script virtfs:

```
#!/bin/sh

echo '$Revision: 1.49 $'

echo -n "Inserisci il nome di dominio: "
read domain

if [ "$domain" = "" ]
then
        echo Non è stato inserito niente: esecuzione interrotta
        exit 0
fi

leadingdir=/virtual

echo -n "Inserire la directory principale: (Scelta predefinita: $leadingdir): "
read ans

if [ "$ans" != "" ]
then
        leadingdir=$ans
fi

newdir=$leadingdir/$domain

if [ -d "$newdir" ]
then
        echo La nuova directory: $newdir: è già esistente
```

```
        exit 0
else
        echo La nuova directory è: $newdir
fi

echo Crea $newdir
mkdir -p $newdir

echo Crea bin
cp -pdR /bin $newdir

echo Crea dev
cp -pdR /dev $newdir

echo Crea dev/log
ln -f /virtual/log $newdir/dev/log

echo Crea etc
mkdir -p $newdir/etc
for i in /etc/*
do
        if [ -d "$i" ]
        then
                continue
        fi
        cp -pd $i $newdir/etc
done

echo Crea etc/skel
mkdir -p $newdir/etc/skel

echo Crea home
for i in a b c d e f g h i j k l m n o p q r s t u v w x y z
do
        mkdir -p $newdir/home/$i
done

echo Crea home/c/crc
mkdir -p $newdir/home/c/crc
chown crc.users $newdir/home/c/crc

echo Crea lib
mkdir -p $newdir/lib
for i in /lib/*
do
        if [ -d "$i" ]
        then
                continue
        fi
        cp -pd $i $newdir/lib
```

```
done

echo Crea proc
mkdir -p $newdir/proc

echo Crea sbin
cp -pdR /sbin $newdir

echo Crea tmp
mkdir -p -m 0777 $newdir/tmp
chmod +t $newdir/tmp

echo Crea usr
mkdir -p $newdir/usr

echo Crea usr/bin
cp -pdR /usr/bin $newdir/usr

echo Crea usr/lib
mkdir -p $newdir/usr/lib

echo Crea usr/lib/locale
cp -pdR /usr/lib/locale $newdir/usr/lib

echo Crea usr/lib/terminfo
cp -pdR /usr/lib/terminfo $newdir/usr/lib

echo Crea usr/lib/zoneinfo
cp -pdR /usr/lib/zoneinfo $newdir/usr/lib

echo Crea usr/lib/\*.so\*
cp -pdR /usr/lib/*.so* $newdir/usr/lib

echo Crea usr/sbin
cp -pdR /usr/sbin $newdir/usr

echo Fa un link a usr/tmp
ln -s /tmp $newdir/usr/tmp

echo Crea var
mkdir -p $newdir/var

echo Crea var/lock
cp -pdR /var/lock $newdir/var

echo Crea var/log
mkdir -p $newdir/var/log

echo Crea var/log/wtmp
cp /dev/null $newdir/var/log/wtmp
```

```
echo Crea var/run
cp -pdR /var/run $newdir/var

echo Crea var/run/utmp
cp /dev/null $newdir/var/run/utmp

echo Crea var/spool
cp -pdR /var/spool $newdir/var

echo Fa un link a var/tmp
ln -s /tmp $newdir/var/tmp

echo Crea var/www/html
mkdir -p $newdir/var/www/html
chown webmast.www $newdir/var/www/html
chmod g+s $newdir/var/www/html

echo Crea var/www/master
mkdir -p $newdir/var/www/master
chown webmast.www $newdir/var/www/master

echo Crea var/www/server
mkdir -p $newdir/var/www/server
chown webmast.www $newdir/var/www/server

exit 0
```

4.2 Virtexec

Per poter eseguire dei comandi in un ambiente virtuale bisogna prima fare `chroot` nella directory prefissata e poi eseguire il comando. Ho scritto un apposito script di shell chiamato virtexec che fa questo per un qualsiasi comando:

```
#!/bin/sh

echo '$Revision: 1.49 $'

BNAME=`basename $0`
FIRST4CHAR=`echo $BNAME | cut -c1-4`
REALBNAME=`echo $BNAME | cut -c5-`

if [ "$BNAME" = "virtexec" ]
then
        echo Non si può eseguire direttamente virtexec: è NECESSARIO un link simbolico
        exit 0
fi

if [ "$FIRST4CHAR" != "virt" ]
then
```

```
        echo Il link simbolico non è a una funzione virt
        exit 0
fi

list=""
num=1
for i in /virtual/*
do
        if [ ! -d "$i" ]
        then
                continue
        fi
        if [ "$i" = "/virtual/lost+found" ]
        then
                continue
        fi
        list="$list $i $num"
        num=`expr $num + 1`
done

if [ "$list" = "" ]
then
        echo Non esistono ambienti virtuali
        exit 0
fi

dialog --clear --title 'Virtexec' --menu Pick 20 70 12 $list 2> /tmp/menu.$$
if [ "$?" = "0" ]
then
        newdir=`cat /tmp/menu.$$`
else
        newdir=""
fi
tput clear
rm -f /tmp/menu.$$

echo '$Revision: 1.49 $'

if [ ! -d "$newdir" ]
then
        echo La nuova directory: $newdir: NON ESISTE
        exit 0
else
        echo Nuova directory: $newdir
fi

echo bname: $BNAME

echo realbname: $REALBNAME
```

```
if [ "$*" = "" ]
then
        echo args: none
else
        echo args: $*
fi

echo Spostamento in $newdir
cd $newdir

echo Esecuzione del programma $REALBNAME

chroot $newdir $REALBNAME $*

exit 0
```

Si prega di notare che lo script funziona solo se si ha installato sul proprio sistema il programma `dialog`. Per usare virtexec basta collegare con un link simbolico un programma a virtexec. Ad esempio:

```
ln -s /usr/local/bin/virtexec /usr/local/bin/virtpasswd
ln -s /usr/local/bin/virtexec /usr/local/bin/virtvi
ln -s /usr/local/bin/virtexec /usr/local/bin/virtpico
ln -s /usr/local/bin/virtexec /usr/local/bin/virtemacs
ln -s /usr/local/bin/virtexec /usr/local/bin/virtmailq
```

In questo modo quando si digiterà virtvi o virtpasswd o virtmailq si potrà editare un file con vi, cambiare la password di un utente o controllare la coda di posta sul proprio sistema virtuale. Si possono creare tanti link simbolici a virtexec quanti occorrono. Da notare che, se il programma richiede una libreria condivisa, essa deve trovarsi nel filesystem virtuale, così come il file binario stesso.

4.3 Note

Di solito io installo tutti gli script in /usr/local/bin. Tutto ciò che non voglio compaia nel filesystem virtuale lo metto in /usr/local. Lo script non copia alcun file di /usr/local nel filesystem virtuale. È importante che ogni file che non deve trovarsi in tutti i filesystem virtuali venga rimosso. Ad esempio, sul mio sistema è installato ssh ed io non voglio che la chiave privata per il server sia disponibile su tutti i filesystem virtuali, così lo cancello da ciascun filesystem virtuale dopo aver lanciato virtfs. Oltre a questo, cambio anche resolv.conf e rimuovo per ragioni legali tutto ciò che contiene un riferimento ad un altro dominio. Ad esempio, /etc/hosts /etc/HOSTNAME.

Questi sono i programmi che ho collegato con un link simbolico a virtexec:

- virtpasswd – cambia la password di un utente
- virtadduser – crea un utente
- virtdeluser – cancella un utente
- virtsmbstatus – visualizza lo stato di SAMBA
- virtvi – edita un file
- virtmailq – controlla var/spool/mqueue
- virtnewaliases – ricostruisce la tabella degli alias

5 DNS

Il DNS può essere configurato normalmente. C'è un HOWTO sul *DNS* .

6 Syslogd

6.1 Problema

Syslogd è il programma utilità di registrazione dei messaggi dei servizi tipicamente utilizzato sui sistemi UNIX. Syslogd è un demone che apre un file speciale chiamato FIFO. Una FIFO è un file speciale che si comporta come una 'pipe'. Tutto ciò che viene mandato sul lato scrittura uscirà sul lato lettura. Ci sono delle funzioni C che scrivono sul lato scrittura. Se un programma utilizza tali funzioni C l'output verrà mandato al syslogd.

Ci si ricordi che si è impostato un ambiente `chroot` e che la FIFO da cui syslogd sta leggendo (/dev/log) non è presente. Questo significa che [in assenza di opportune modifiche N.d.T.] a syslogd non giungeranno i messaggi provenienti dagli ambienti virtuali.

6.2 Soluzione

6.2.1 Impostare dei link

Syslogd è in grado di utilizzare una FIFO differente se specificata sulla riga di comando:

```
syslogd -p /virtual/log
```

Poi si colleghi con un link simbolico /dev/log a /virtual/log con:

```
ln -sf /virtual/log /dev/log
```

Infine si colleghino con hard link tutte le copie di /dev/log a questo file con:

```
ln -f /virtual/log /virtual/domain1.com/dev/log
```

Lo script virtfs soprariportato fa già tutto questo. Dato che /virtual si trova su un unico disco e i file /dev/log sono collegati con hard link, essi hanno lo stesso numero di inode e puntano agli stessi dati. `Chroot` non può impedirlo, così ora tutti i /dev/log virtuali funzioneranno. Si noti che tutti i messaggi provenienti dai vari ambienti virtuali verranno registrati assieme. È possibile comunque ideare programmi separati per filtrare le informazioni che interessano.

6.2.2 Syslogd.init

Questa versione di syslogd.init effettua un hard link a /dev/log ad ogni suo avvio poiché syslogd cancella e crea la FIFO /dev/log ad ogni sua nuova esecuzione. Ecco una versione modificata del file syslogd.init:

```
#!/bin/sh

. /etc/rc.d/init.d/functions
```

```
case "$1" in
  start)
        echo -n "Ora viene fatto l'hard link a dev log: "
        ln -sf /virtual/log /dev/log
        echo done
        echo -n "Lancio dei demoni di log di sistema: "
        daemon syslogd -p /virtual/log
        daemon klogd
        echo
        echo -n "Ora viene fatto il link dei dev log virtuali: "
        for i in /virtual/*
        do
                if [ ! -d "$i" ]
                then
                        continue
                fi
                if [ "$i" = "/virtual/lost+found" ]
                then
                        continue
                fi
                ln -f /virtual/log $i/dev/log
                echo -n "."
        done
        echo " done"
        touch /var/lock/subsys/syslogd
        ;;
  stop)
        echo -n "Arresto dei demoni di log di sistema: "
        killproc syslogd
        killproc klogd
        echo
        rm -f /var/lock/subsys/syslogd
        ;;
  *)
        echo "Impiego: syslogd {start|stop}"
        exit 1
esac

exit 0
```

6.3 Syslogd multipli

6.3.1 Uno per disco

Se c'è carenza di spazio in un filesystem e bisogna suddividere i domini virtuali su più dischi, ci si ricordi che gli hard link non funzionano tra dischi diversi. Questo significa che bisognerà lanciare un syslogd distinto per ogni gruppo di domini di un disco. Ad esempio, se ci fossero tredici domini su /virtual1 e quindici su /virtual2, si dovrebbero collegare tramite hard link i tredici domini a /virtual1/log e lanciare un syslogd con `syslogd -p /virtual1/log`, poi collegare con hard link gli altri quindici domini a /virtual2/log e lanciare un altro syslogd con `syslogd -p /virtual2/log`.

6.3.2 Uno per dominio

Se si preferisce non accentrare i log in un unico posto è possibile lanciare un syslogd per dominio. Questo metodo comporta uno spreco di risorse di sistema (ci sono più processi attivi), quindi non lo raccomando, ma è più facile da implementare di quello precedente. È necessario modificare il file syslogd.init affinché il syslogd venga mandato in esecuzione con `chroot /virtual/domain1.com syslogd` e questo per ciascun dominio. Così facendo ogni syslogd verrà eseguito all'interno dell'ambiente di `chroot` e i log dei vari ambienti virtuali si troveranno singolarmente in /virtual/domain1.com/var/log piuttosto che tutti assieme in un solo /var/log. Non bisogna dimenticare di lanciare un `syslogd` normale per il sistema principale e un demone di log del kernel `klogd`.

7 FTP virtuale

7.1 Inetd

Wu-ftpd viene fornito con un supporto interno alla virtualizzazione. Ad ogni modo non si possono avere file di password separati per ogni dominio. Ad esempio, se `bob@domain1.com` and `bob@domain2.com` vogliono entrambi un account, è necessario assegnare nomiutente diversi, come bob e bob2, o chiedere ad uno dei due utenti di scegliere un nomeutente diverso. Ora invece abbiamo un filesystem virtuale per ogni dominio, quindi file delle password separati, e questo problema non sussiste. È sufficiente creare gli script virtnewuser e virtpasswd nel modo summenzionato e la configurazione è completa.

Le voci di inetd.conf per wu-ftpd:

```
ftp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.ftp wu.ftpd -l -a
```

7.2 FTP anonimo

Le cose non cambiano usando virtuald. È sufficiente creare l'utente FTP in /virtual/domain1.com/etc/passwd come si farebbe normalmente.

```
ftp:x:14:50:Anonymous FTP:/var/ftp:/bin/false
```

Poi bisogna configurare la directory per l'FTP anonimo. Ci sono file delle password distinti per ogni singolo dominio, per cui è possibile limitare l'FTP anonimo a un qualsivoglia numero di essi. Si noti che, dato che il server FTP si trova già in un ambiente di `chroot` nella directory /virtual/domain1.com, non è necessario premettere alcun percorso.

7.3 Utenti dell'FTP virtuale

Wu-ftpd supporta l'utilizzo del gruppo guest. Ciò permette di creare aree FTP differenti per ciascun utente. Il server FTP effettua un `chroot` sull'area specificata in modo che l'utente non possa uscire da quel ramo dell'albero delle directory. Gli utenti creati in questo modo all'interno di un dominio virtuale non potranno vedere i file di sistema.

Si aggiunga il gruppo guest al file /virtual/domain1.com/etc/ftpaccess file.

Si crei una voce /virtual/domain1.com/etc/passwd con la directory di `chroot` e la directory home di partenza separate da `/./`:

```
guest1:x:8500:51:Guest FTP:/home/g/guest1/./incoming:/bin/false
```

Infine si configuri la directory home di guest come si farebbe per l'FTP anonimo. Ci sono file delle password distinti per ciascun dominio, quindi è possibile specificare quali domini hanno account guest e quali utenti sono utenti guest all'interno di un dominio. Si noti che, dato che il server FTP si trova già in un ambiente di `chroot` nella directory /virtual/domain1.com, non è necessario premettere alcun percorso.

8 Web virtuale

8.1 Usando virtuald

8.1.1 Non raccomandabile

Apache ha un supporto interno per i domini virtuali. È il solo programma di cui raccomando di usare le funzionalità interne per la gestione dei domini virtuali. Ogniqualvolta si lancia qualcosa attraverso inetd c'è un prezzo da pagare: il programma deve ripartire da zero ogni volta che ne viene richiesta l'esecuzione. Questo causa un rallentamento nel tempi di risposta, che è accettabile per la gran parte dei servizi, ma inaccettabile per quello web. Apache ha anche un meccanismo per impedire connessioni quando ce ne siano troppe in entrata, che potrebbe essere un fattore critico anche per siti con un volume di traffico medio.

Detto in poche parole, rendere virtuale Apache con virtuald è una pessima idea. Virtuald trova la sua ragion d'essere nel colmare le lacune di servizi che non hanno la capacità di gestire in proprio i domini virtuali. Virtuald non è pensato per rimpiazzare del codice di buona qualità che sia in grado di svolgere da sé questo compito.

Per coloro che sono abbastanza sconsiderati da farlo comunque, malgrado quanto detto sopra, ecco come fare:

8.1.2 Inetd

Modificare /etc/inetd.conf

```
vi /etc/inetd.conf # Aggiungi questa linea
www stream tcp nowait www /usr/local/bin/virtuald \
        virtuald /virtual/conf.www httpd -f /var/www/conf/httpd.conf
```

8.1.3 Httpd.conf

Modificare /var/www/conf/httpd.conf

```
vi /var/www/conf/httpd.conf # O dovunque si trovino i file di configurazione

Dovrebbe esserci:
ServerType standalone

Rimpiazzare la riga con:
ServerType inetd
```

8.1.4 Configurazione

Si configuri poi ogni singola istanza del server Apache come si farebbe usandolo per un singolo dominio.

8.1.5 Httpd.init

Non è necessario un file httpd.init, dato che il programma server viene eseguito attraverso inetd.

8.2 Usando Apache VirtualHost

Apache ha tre file di configurazione `access.conf`, `httpd.conf`, e `srm.conf`. Le versioni recenti di Apache hanno reso non necessari i tre file di configurazione. Comunque ho trovato che suddividere la configurazione in tre sezioni ne semplifica la gestione, per cui continuerò a fare così in questo HOWTO.

8.2.1 Access.conf

Questo file di configurazione è usato per controllare l'accesso alle directory della struttura del sito. Ecco una configurazione di esempio che mostra come si possano gestire opzioni differenti per ciascun dominio:

```
# /var/www/conf/access.conf: Configurazione di accesso globale

# Le opzioni sono ereditate dalla directory genitore
# Configura la directory principale con le opzioni predefinite
<Directory />
AllowOverride None
Options Indexes
</Directory>

# Fornisce a un dominio una directory protetta da password
<Directory /virtual/domain1.com/var/www/html/priv>
AuthUserFile /var/www/passwd/domain1.com-priv
AuthGroupFile /var/www/passwd/domain1.com-priv-g
AuthName PRIVSECTION
AuthType Basic
<Limit GET PUT POST>
require valid-user
</Limit>
</Directory>

# Permette i Server Side Include in un altro dominio
<Directory /virtual/domain2.com/var/www/html>
Options IncludesNOEXEC
</Directory>
```

8.2.2 Httpd.conf

Questo file di configurazione è usato per gestire le opzioni principali del server Apache. Ecco una configurazione di esempio che mostra come si possano gestire opzioni differenti per ciascun dominio:

```
# /var/www/conf/httpd.conf: File principale di configurazione del server

# Inizio: sezione principale di configurazione

# La riga seguente è necessaria dato che non si sta usando inetd
```

```
ServerType standalone

# Porta sulla quale gira il server
Port 80

# Registra nei log gli host dei client con i loro nomi piuttosto che con
# gli indirizzi IP
HostnameLookups on

# Utente con i privilegi del quale gira il server
User www
Group www

# Collocazione dei file di configurazione, di errore e di log
ServerRoot /var/www

# File in cui si trova l'identificatore di processo (Process Id) del server
PidFile /var/run/httpd.pid

# File di informazioni sullo stato interno del server
ScoreBoardFile /var/www/logs/apache_status

# Opzioni di Timeout e KeepAlive
Timeout 400
KeepAlive 5
KeepAliveTimeout 15

# Limitazioni per i server in esecuzione
MinSpareServers 5
MaxSpareServers 10
StartServers 5
MaxClients 150
MaxRequestsPerChild 30

# Fine: sezione principale di configurazione

# Inizio: sezione host virtuale

# Specifica le coppie IP:porta su cui il demone accetta connessioni
# Io ho una direttiva per ogni IP necessario in modo da poter
# ignorare esplicitamente certi domini
Listen 10.10.10.129:80
Listen 10.10.10.130:80

# La direttiva VirtualHost permette di specificare un altro dominio
# virtuale sul server. La maggior parte delle opzioni di Apache possono
# essere specificate all'interno di questa sezione.
<VirtualHost www.domain1.com>

# Indirizzo di e-mail da contattare in caso di errori
```

```
ServerAdmin webmaster@domain1.com

# Collocazione dei documenti web nel dominio virtuale
DocumentRoot /virtual/domain1.com/var/www/html

# Nome di dominio del server
ServerName www.domain1.com

# File di Log relativi alla direttiva ServerRoot
ErrorLog logs/domain1.com-error_log
TransferLog logs/domain1.com-access_log
RefererLog logs/domain1.com-referer_log
AgentLog logs/domain1.com-agent_log

# Usa gli script CGI in questo dominio
ScriptAlias /cgi-bin/ /var/www/cgi-bin/domain1.com/
AddHandler cgi-script .cgi
AddHandler cgi-script .pl
</VirtualHost>

<VirtualHost www.domain2.com>

# Indirizzo di e-mail da contattare in caso di errori
ServerAdmin webmaster@domain2.com

# Collocazione delle pagine web nel dominio virtuale
DocumentRoot /virtual/domain2.com/var/www/html

# Nome di dominio del server
ServerName www.domain2.com

# File di Log relativi alla direttiva ServerRoot
ErrorLog logs/domain2.com-error_log
TransferLog logs/domain2.com-access_log
RefererLog logs/domain2.com-referer_log
AgentLog logs/domain2.com-agent_log

# Niente script CGI per questo host virtuale
</VirtualHost>
# Fine: sezione host virtuale
```

8.2.3 Srm.conf

Questo file di configurazione viene usato per controllare il modo in cui vengono processate le richieste e il formato dei risultati. Non ci sono modifiche particolari da apportare per i domini virtuali. Il file di configurazione di esempio dovrebbe andar bene.

8.2.4 Httpd.init

Non si devono apportate modifiche particolari al file httpd.init. Si può usare quello standard, compreso nella configurazione di Apache.

8.3 Overflow dei descrittori di file

8.3.1 Attenzione!

Quanto si dirà si applica solo al server Apache eseguito come 'standalone' (indipendente). Se il programma server viene eseguito attraverso inetd, esso non interagisce con gli altri domini, per cui ha un'intera tabella di descrittori di file per ogni dominio.

Ogni file di log che il server Apache apre significa un descrittore di file in più per il processo. C'è un limite di 256 descrittori di file per processo in Linux. Dato che si gestiscono più domini con un unico server web, si usano un mucchio di descrittori di file. Se un solo server web Apache, che è un processo singolo, supporta troppi domini, è possibile causare un overflow in questa tabella. Ciòsignificherebbe la mancata registrazione di alcuni log e l'impossibilità di eseguire script CGI.

8.3.2 Server Apache multipli

Se si ipotizza l'uso di cinque descrittori di file per dominio, si possono gestire 50 domini su un solo server Apache senza nessun problema. Comunque, nel caso si riscontrino problemi del genere, si può creare /var/www1 con un server Apache che s'incarichi dei domini da domain1 a domain25 e /var/www2 con un server Apache che gestisca i domini dal domain26 al domain50 e così via. Così facendo ogni server avrà la propria directory di file di configurazione, di errore e di log. Ogni server dovrà essere configurato separatamente, ognuno con le proprie direttive Listen e VirtualHost. Non ci si dimentichi di lanciare più server tramite il proprio file httpd.init.

8.4 Server che condividono un unico IP

8.4.1 Risparmiare indirizzi IP

HTTP (HyperText Transfer Protocol) versione 1.1 fornisce una funzionalità per comunicare il nome di dominio del server al client. Ciò significa che il client non ha necessità di risolvere il nome del server a partire dall'indirizzo IP. Perciò due server virtuali potranno avere gli stessi indirizzi IP ed essere siti web diversi. La configurazione di Apache è la stessa di sopra eccetto che non sarà necessario inserire direttive Listen differenti, dato che i due domini avranno lo stesso IP.

8.4.2 Inconveniente!

Il solo problema è che virtuald usa gli indirizzi IP per distinguere tra i vari domini. Nella sua stesura attuale, [nel caso si condividano indirizzi IP N.d.T.] virtuald non sarebbe in grado di eseguire il `chroot` a differenti directory di spool per ogni dominio. Perciò il servizio di posta risponderebbe solo a livello di singolo indirizzo IP e non ci sarebbe più una singola directory di spool per ogni dominio. Tutti i client del medesimo IP condiviso sul web dovrebbero condividere la medesima directory di spool. Ciò significa che duplicati di nomiutente costituerebbero nuovamente un problema [non si potrebbero usare gli stessi nomiutente in domini virtuali diversi N.d.T.]. Comunque questo è il prezzo da pagare per condividere lo stesso indirizzo IP.

8.5 Maggiori informazioni

Questo HOWTO mostra come implementare il supporto ai domini virtuali solo con il web server Apache. La maggior parte dei server web usano un'interfaccia simile. Per maggiori informazioni sul web hosting virtuale si consulti *WWW HOWTO* , la documentazione di Apache presso *Sito web di Apache* , o la documentazione presso *ApacheWeek* .

9 Mail/Pop virtuale

9.1 Problema

La domanda per il supporto alla posta elettronica virtuale è in continua crescita. Sendmail dice di supportare la posta virtuale. Ciò che supporta in realtà è la ricezione di messaggi per domini diversi. Quindi si può specificare di reinoltrare la posta altrove. Comunque, se i messaggi vengono reinoltrati alla macchina locale e ci sono dei messaggi per bob@domain1.com e bob@domain2.com, essi finiranno nello stesso folder. Questo è un problema, dato che i 'bob' sono persone diverse con posta diversa.

9.2 Soluzione

Ci si può accertare che ogni nomeutente sia unico, usando uno schema di numerazione: bob1, bob2 eccetera o preponendo pochi caratteri a ciascun nomeutente: dom1bob, dom2bob eccetera. Si potrebbe anche smanettare sui singoli programmi coinvolti, facendo in modo che eseguano queste conversioni per conto loro dietro le quinte, ma ciò potrebbe causare confusione. Inoltre i messaggi di posta in uscita hanno l'intestazione di dominio maindomain.com, mentre si vorrebbe che la posta in uscita avesse le intestazioni diversificate secondo i diversi sottodomini.

Propongo due soluzioni. Una funziona con sendmail e l'altra con Qmail. La soluzione che usa sendmail dovrebbe funzionare su un'installazione di base di sendmail. Comunque essa condivide tutte le limitazioni implicite di sendmail. Questa soluzione richiede inoltre che per ogni dominio venga eseguito un sendmail in modalità coda. Avere 50 o più processi di sendmail che si risvegliano ad ogni ora può sottoporre una macchina ad un carico non indifferente.

La soluzione che contempla l'uso di Qmail non richiede l'esecuzione di istanze multiple di Qmail e può fare a meno di una directory di coda. Richiede invece un programma extra, dato che Qmail non si appoggia a virtuald. Suppongo che una soluzione simile possa essere affrontata anche con sendmail. Ad ogni modo Qmail si presta a tale soluzione in modo più pulito.

Non appoggio comunque l'uso di un programma piuttosto che dell'altro. L'installazione di sendmail fila un po' più liscia ma Qmail è probabilmente il più potente dei due pacchetti.

9.3 La soluzione con Sendmail

9.3.1 Introduzione

Un filesystem virtuale per ogni dominio permette a quest'ultimo di avere il suo proprio /etc/passwd. Questo vuol dire che bob@domain1.com e bob@domain2.com sono utenti diversi presenti in file /etc/passwd diversi cosicché gestire la posta non sarà un problema. Inoltre i domini hanno ciascuno le proprie directory di spool, in modo che i folder di posta saranno file diversi in filesystem virtuali diversi.

9.3.2 Creare il file di configurazione di Sendmail

Si crei il file /etc/sendmail.cf come si farebbe normalmente usando m4. Io ho usato:

```
divert(0)
VERSIONID(‘tcpproto.mc’)
OSTYPE(linux)
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(local)
MAILER(smtp)
```

9.3.3 Modificare il file di configurazione di Sendmail

Si modifichi /virtual/domain1.com/etc/sendmail.cf in modo che risponda con le intestazioni appropriate al proprio dominio virtuale:

```
vi /virtual/domain1.com/etc/sendmail.cf # Circa alla riga 86
Dovrebbe esserci:
#Dj$w.Foo.COM

Rimpiazzarlo con:

Djdomain1.com
```

9.3.4 Consegna locale con Sendmail

Si introducano in /virtual/domain1.com/etc/sendmail.cw i nomi host locali.

```
vi /virtual/domain1.com/etc/sendmail.cw
mail.domain1.com
domain1.com
domain1
localhost
```

9.3.5 Posta tra domini virtuali con Sendmail: il trucco (Versioni precedenti la 8.8.6)

In ogni caso, sendmail richiede una piccola modifica al codice sorgente. Sendmail ha un file chiamato /etc/sendmail.cw che contiene tutti i nomi delle macchine cui sendmail consegnerà la posta posta localmente invece di reindirizzarla ad un'altra macchina. Sendmail fa un controllo interno di tutti i dispositivi della macchina per inizializzare questa lista con gli indirizzi IP locali. Ciò causa un problema nel caso di invii di messaggi di posta tra domini virtuali sulla stessa macchina. Sendmail sarà portato a credere che l'altro dominio virtuale sia un indirizzo locale e tratterà i messaggi localmente. Ad esempio, bob@domain1.com invia un'e-mail a fred@domain2.com. Dato che il sendmail di domain1.com crede che domain2.com sia un indirizzo locale, metterà il messaggio nella directory di spool di domain1.com e non lo invierà mai a domain2.com. È necessario modificare sendmail (io l'ho fatto su una versione 8.8.5 senza problemi):

```
vi v8.8.5/src/main.c # Circa alla riga 494
Dovrebbe esserci:

load_if_names();

Rimpiazzarlo con:

/* load_if_names(); Commentato perché da problemi con i domini virtuali */
```

Da notare che questo passo è necessario solo se si vuole essere in grado di spedire posta tra i domini virtuali, cosa che ritengo probabile.

Ciò risolverà il problema. Comunque il device ethernet principale eth0 non viene rimosso. Quindi se si invia un messaggio di posta da un IP virtuale a quello usato da eth0 sulla stessa macchina, esso verrà consegnato localmente. Per questo io non faccio altro che usarlo come un IP posticcio virtual1.maindomain.com (10.10.10.157). Non invierò mai posta a questo host, né lo faranno i domini virtuali. Questo è anche l'IP che userei per collegarmi alla macchina a mezzo ssh per controllare se tutto va bene.

9.3.6 Posta tra domini virtuali con Sendmail: Nuove funzionalità (Versioni successive alla 8.8.6)

Dalla versione 8.8.6 di Sendmail è disponibile una nuova opzione, che permette di disabilitare il caricamento delle interfacce extra di rete. Ciò significa che NON è più necessario modificare il sorgente in alcun modo. Tale opzione è chiamata `DontProbeInterfaces`.

Modificare /virtual/domain1.com/etc/sendmail.cf

```
vi /virtual/domain1.com/etc/sendmail.cf # Aggiungere la linea
O DontProbeInterfaces=True
```

9.3.7 Sendmail.init

Sendmail non può più essere lanciato come demone 'standalone' (indipendente), è necessario eseguirlo attraverso inetd. Ciò è inefficiente e causerà un peggioramento dei tempi di avvio, ma nel caso si avesse un sito con traffico piuttosto alto non gli si dovrebbe far comunque condividere un box virtuale con altri domini. È da notare che sendmail NON viene eseguito con l'opzione `-bd`. Si noti anche che è necessario venga eseguito un `sendmail -q` per ogni dominio, per processare la coda dei messaggi da consegnare. Ecco il nuovo file sendmail.init:

```
#!/bin/sh

. /etc/rc.d/init.d/functions

case "$1" in
  start)
        echo -n "Avvio di sendmail: "
        daemon sendmail -q1h
        echo
        echo -n "Avvio del sendmail virtuale: "
        for i in /virtual/*
        do
```

```
                if [ ! -d "$i" ]
                then
                        continue
                fi
                if [ "$i" = "/virtual/lost+found" ]
                then
                        continue
                fi
                chroot $i sendmail -q1h
                echo -n "."
        done
        echo " done"
        touch /var/lock/subsys/sendmail
        ;;
  stop)
        echo -n "Arresto di sendmail: "
        killproc sendmail
        echo
        rm -f /var/lock/subsys/sendmail
        ;;
  *)
        echo "Utilizzo: sendmail {start|stop}"
        exit 1
esac

exit 0
```

9.3.8 Configurazione di inetd

Il servizio pop si dovrebbe installare normalmente senza lavoro aggiuntivo. Basta solo che alla sua voce in inetd venga aggiunta la parte per virtuald. Ecco le voci di inetd.conf per sendmail e pop:

```
pop-3 stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.pop in.qpop -s
smtp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.mail sendmail -bs
```

9.4 La soluzione con Qmail

9.4.1 Introduzione

Questa soluzione scavalca qmail-local nelle mansioni di consegna, quindi i file .qmail nelle directory home virtuali non funzioneranno più. Comunque ogni dominio avrà ancora un utente responsabile del controllo sugli alias dell'intero dominio. A tale scopo verrano usati due programmi esterni per i file .qmail-default di tali utenti responsabili. La posta passerà attraverso questi due programmi per essere consegnata correttamente ad ogni dominio.

Sono richiesti due programmi poiché uno di essi viene eseguito con i privilegi di root. È un piccolo programma che cambia di volta in volta i propri privilegi ad un utente non root e manda in esecuzione il secondo. Si consulti un sito di documentazione sulla sicurezza per una disamina dei motivi per cui ciò è necessario.

Questa soluzione evita il bisogno di usare virtuald. Qmail è abbastanza flessibile da non richiere una configurazione tramite virtuald. Il modello progettuale su cui è basato Qmail utilizza il concatenamento di vari programmi per consegnare la posta. Questo modello rende molto facile inserire una sezione virtuale nel processo di consegna della posta di Qmail senza alterare l'installazione di base.

Occorre ricordare che, dato che si sta usando un unico server Qmail, qualunque nome di dominio non completamente qualificato verrà espanso usando il nome di dominio del server principale. Questo perché non si utilizza un server Qmail separato per ogni dominio. Perciò bisogna assicurarsi che i propri client (Eudora, elm, mutt, ecc.) siano configurati per espandere tutti i propri nomi di dominio non completamente qualificati.

9.4.2 Configurare i domini virtuali

Qmail dev'essere configurato per accettare messaggi di posta per ciascuno dei domini virtuali cui si vuole fornire il servizio. Si digitino i seguenti comandi:

```
echo "domain1.com:domain1" >> /var/qmail/control/virtualdomains
```

9.4.3 Configurare l'utente responsabile per il dominio

Si aggiunga al file /etc/passwd principale l'utente domain1. Meglio attribuirgli la shell /bin/false in modo che tale utente non possa accedere ad una console. Tale utente potrà aggiungere file .qmail e tutta la posta indirizzata al dominio virtuale domain1 passerà attraverso tale account. Si noti che i nomiutente possono essere lunghi solo otto caratteri mentre i nomi di dominio possono essere più lunghi. I caratteri che avanzano vengono troncati. Ciò significa che gli utenti dominio12 e dominio123 finiranno per essere lo stesso utente e Qmail potrebbe far confusione. Bisogna perciò fare attenzione a scegliere bene le proprie regole di denominazione dell'utente responsabile del dominio.

Si creino i file .qmail del responsabile di dominio con i seguenti comandi. Si aggiunga qualsiasi altro alias di sistema a questo punto, per es. webmaster o hostmaster.

```
echo "user@domain1.com" > /home/d/domain1/.qmail-mailer-daemon
echo "user@domain1.com" > /home/d/domain1/.qmail-postmaster
echo "user@domain1.com" > /home/d/domain1/.qmail-root
```

Si crei il file .qmail-default del responsabile di dominio. Questo file filtrerà tutta la posta indirizzata al dominio virtuale.

```
echo "| /usr/local/bin/virtmailfilter" > /home/d/domain1/.qmail-default
```

9.4.4 Tcpserver

Qmail richiede uno speciale programma pop, in grado di supportare il formato Maildir. Il programma pop dev'essere reso virtuale. L'autore di Qmail raccomanda di usare a questo scopo tcpserver (un rimpiazzo di inetd) con Qmail, quindi nei miei esempi userò tcpserver e NON inetd.

Tcpserver non richiede un file di configurazione. Tutte le informazioni necessarie gli possono essere passate da riga di comando. Segue il file tcpserver.init che si dovrebbe usare per i demoni di consegna e prelievo della posta ('mail demon' e 'popper'):

```
#!/bin/sh
```

```
. /etc/rc.d/init.d/functions

QMAILDUSER=‘grep qmaild /etc/passwd | cut -d: -f3‘
QMAILDGROUP=‘grep qmaild /etc/passwd | cut -d: -f4‘

# Dare uno sguardo a come vengono chiamati.
case "$1" in
  start)
        echo -n "Avvio di tcpserver: "
        tcpserver -u 0 -g 0 0 pop-3 /usr/local/bin/virtuald \
                /virtual/conf.pop qmail-popup virt.domain1.com \
                /bin/checkpassword /bin/qmail-pop3d Maildir &
        echo -n "pop "
        tcpserver -u $QMAILDUSER -g $QMAILDGROUP 0 smtp \
                /var/qmail/bin/qmail-smtpd &
        echo -n "qmail "
        echo
        touch /var/lock/subsys/tcpserver
        ;;
  stop)
        echo -n "Arresto di tcpserver: "
        killall -TERM tcpserver
        echo -n "killing "
        echo
        rm -f /var/lock/subsys/tcpserver
        ;;
  *)
        echo "Utilizzo: tcpserver {start|stop}"
        exit 1
esac

exit 0
```

9.4.5 Qmail.init

Si può utilizzare l'"init script' standard fornito con Qmail. La documentazione che accompagna Qmail è descrive ottimamente come farlo.

9.4.6 Sorgenti

Per far funzionare i servizi di posta virtuali con Qmail sono richiesti altri due programmi. Essi sono virtmailfilter e virtmaildelivery. Segue sotto il sorgente C di virtmailfilter. Il programma andrebbe installato in /usr/local/bin con modi 4750, utente root e gruppo nofiles.

```
#include <sys/wait.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
```

```
#include <ctype.h>
#include <pwd.h>

#define VIRTPRE                 "/virtual"

#define VIRTPWFILE              "etc/passwd"
#define VIRTDELIVERY            "/usr/local/bin/virtmaildelivery"
#define VIRTDELIVERY0           "virtmaildelivery"

#define PERM                    100
#define TEMP                    111
#define BUFSIZE                 8192

int main(int argc,char **argv)
{
        char *username,*usernameptr,*domain,*domainptr,*homedir;
        char virtpath[BUFSIZE];
        struct passwd *p;
        FILE *fppw;
        int status;
        gid_t gid;
        pid_t pid;

        if (!(username=getenv("EXT")))
        {
                fprintf(stdout,"environment variable EXT not set\n");
                exit(TEMP);
        }

        for(usernameptr=username;*usernameptr;usernameptr++)
        {
                *usernameptr=tolower(*usernameptr);
        }

        if (!(domain=getenv("HOST")))
        {
                fprintf(stdout,"environment variable HOST not set\n");
                exit(TEMP);
        }

        for(domainptr=domain;*domainptr;domainptr++)
        {
                if (*domainptr=='.' && *(domainptr+1)=='.')
                {
                        fprintf(stdout,"environment variable HOST has ..\n");
                        exit(TEMP);
                }
                if (*domainptr=='/')
                {
                        fprintf(stdout,"environment variable HOST has /\n");
```

```
                exit(TEMP);
        }

        *domainptr=tolower(*domainptr);
}

for(domainptr=domain;;)
{
        snprintf(virtpath,BUFSIZE,"%s/%s",VIRTPRE,domainptr);
        if (chdir(virtpath)>=0)
                break;

        if (!(domainptr=strchr(domainptr,'.')))
        {
                fprintf(stdout,"domain failed: %s\n",domain);
                exit(TEMP);
        }

        domainptr++;
}

if (!(fppw=fopen(VIRTPWFILE,"r+")))
{
        fprintf(stdout,"fopen failed: %s\n",VIRTPWFILE);
        exit(TEMP);
}

while((p=fgetpwent(fppw))!=NULL)
{
        if (!strcmp(p->pw_name,username))
                break;
}

if (!p)
{
        fprintf(stdout,"user %s: not exist\n",username);
        exit(PERM);
}

if (fclose(fppw)==EOF)
{
        fprintf(stdout,"fclose failed\n");
        exit(TEMP);
}

gid=p->pw_gid;
homedir=p->pw_dir;

if (setgid(gid)<0 || setuid(p->pw_uid)<0)
{
```

```
                fprintf(stdout,"setuid/setgid failed\n");
                exit(TEMP);
        }

        switch(pid=fork())
        {
                case -1:
                        fprintf(stdout,"fork failed\n");
                        exit(TEMP);
                case 0:
                        if (execl(VIRTDELIVERY,VIRTDELIVERY0,username,homedir,NULL)<0)
                        {
                                fprintf(stdout,"execl failed\n");
                                exit(TEMP);
                        }
                default:
                        if (wait(&status)<0)
                        {
                                fprintf(stdout,"wait failed\n");
                                exit(TEMP);
                        }
                        if (!WIFEXITED(status))
                        {
                                fprintf(stdout,"child did not exit normally\n");
                                exit(TEMP);
                        }
                        break;
        }

        exit(WEXITSTATUS(status));
}
```

9.4.7 Sorgenti

Per far funzionare i servizi di posta virtuali con Qmail sono richiesti altri due programmi. Essi sono virtmailfilter e virtmaildelivery. Segue sotto il sorgente C di virtmaildelivery. Andrebbe installato in /usr/local/bin con modi 0755, utente root e gruppo root.

```
#include <sys/stat.h>
#include <sys/file.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <stdio.h>
#include <errno.h>
#include <time.h>

#define TEMP                    111
#define BUFSIZE                 8192
#define ATTEMPTS                10
```

```
int main(int argc,char **argv)
{
	char *user,*homedir,*dtline,*rpline,buffer[BUFSIZE],*p,mail[BUFSIZE];
	char maildir[BUFSIZE],newmaildir[BUFSIZE],host[BUFSIZE];
	int fd,n,nl,i,retval;
	struct stat statp;
	time_t thetime;
	pid_t pid;
	FILE *fp;

	retval=0;

	if (!argv[1])
	{
		fprintf(stdout,"invalid arguments: need username\n");
		exit(TEMP);
	}

	user=argv[1];

	if (!argv[2])
	{
		fprintf(stdout,"invalid arguments: need home directory\n");
		exit(TEMP);
	}

	homedir=argv[2];

	if (!(dtline=getenv("DTLINE")))
	{
		fprintf(stdout,"environment variable DTLINE not set\n");
		exit(TEMP);
	}

	if (!(rpline=getenv("RPLINE")))
	{
		fprintf(stdout,"environment variable RPLINE not set\n");
		exit(TEMP);
	}

	while (*homedir=='/')
		homedir++;
	snprintf(maildir,BUFSIZE,"%s/Maildir",homedir);
	if (chdir(maildir)<0)
	{
		fprintf(stdout,"chdir failed: %s\n",maildir);
		exit(TEMP);
	}
```

```
time(&thetime);
pid=getpid();
if (gethostname(host,BUFSIZE)<0)
{
        fprintf(stdout,"gethostname failed\n");
        exit(TEMP);
}

for(i=0;i<ATTEMPTS;i++)
{
        snprintf(mail,BUFSIZE,"tmp/%u.%d.%s",thetime,pid,host);
        errno=0;
        stat(mail,&statp);
        if (errno==ENOENT)
                break;

        sleep(2);
        time(&thetime);
}
if (i>=ATTEMPTS)
{
        fprintf(stdout,"could not create %s\n",mail);
        exit(TEMP);
}

if (!(fp=fopen(mail,"w+")))
{
        fprintf(stdout,"fopen failed: %s\n",mail);
        retval=TEMP; goto unlinkit;
}

fd=fileno(fp);

if (fprintf(fp,"%s",rpline)<0)
{
        fprintf(stdout,"fprintf failed\n");
        retval=TEMP; goto unlinkit;
}

if (fprintf(fp,"%s",dtline)<0)
{
        fprintf(stdout,"fprintf failed\n");
        retval=TEMP; goto unlinkit;
}

while(fgets(buffer,BUFSIZE,stdin))
{
        for(p=buffer;*p=='>';p++)
                ;
```

```
		if (!strncmp(p,"From ",5))
		{
			if (fputc('>',fp)<0)
			{
				fprintf(stdout,"fputc failed\n");
				retval=TEMP; goto unlinkit;
			}
		}

		if (fprintf(fp,"%s",buffer)<0)
		{
			fprintf(stdout,"fprintf failed\n");
			retval=TEMP; goto unlinkit;
		}
	}

	p=buffer+strlen(buffer);
	nl=2;
	if (*p=='\n')
		nl=1;

	for(n=0;n<nl;n++)
	{
		if (fputc('\n',fp)<0)
		{
			fprintf(stdout,"fputc failed\n");
			retval=TEMP; goto unlinkit;
		}
	}

	if (fsync(fd)<0)
	{
		fprintf(stdout,"fsync failed\n");
		retval=TEMP; goto unlinkit;
	}

	if (fclose(fp)==EOF)
	{
		fprintf(stdout,"fclose failed\n");
		retval=TEMP; goto unlinkit;
	}

	snprintf(newmaildir,BUFSIZE,"new/%u.%d.%s",thetime,pid,host);
	if (link(mail,newmaildir)<0)
	{
		fprintf(stdout,"link failed: %s %s\n",mail,newmaildir);
		retval=TEMP; goto unlinkit;
	}

unlinkit:
```

```
        if (unlink(mail)<0)
        {
                fprintf(stdout,"unlink failed: %s\n",mail);
                retval=TEMP;
        }

        exit(retval);
}
```

9.5 Ringraziamenti

Ringrazio *Vicente Gonzalez (vince@nycrc.net)* per l'aiuto che ha reso possibile la soluzione presentata per Qmail. È certo possibile ringraziare Vince tramite e-mail, comunque le domande e i commenti su questioni che riguardano Qmail nel contesto di questo HOWTO dovrebbero essere indirizzati al sottoscritto.

10 Samba virtuale

10.1 Configurazione

Il SAMBA virtuale è molto semplice da installare. Ci si assicuri che i seguenti file siano configurati nel modo opportuno:

- /virtual/domain1.com/etc/smb.conf FILE
- /virtual/domain1.com/var/lock/samba DIRECTORY
- /virtual/domain1.com/var/log DIRECTORY
- /usr/local/bin/virtsmbstatus SYMLINK /usr/local/bin/virtexec

10.2 Inetd

Modificare così /etc/inetd.conf

```
vi /etc/inetd.conf # Aggiungere questa linea
netbios-ssn stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.smbd smbd
```

10.3 Smb.init

Non è necessario un file smb.init in quanto il programma server è lanciato tramite inetd.

11 Altri servizi virtuali

Per ogni altro servizio si dovrebbe seguire una procedura simile.

- Lanciare virtfs per aggiungere i file binari e le librerie al filesystem virtuale
- Aggiungere il servizio a /etc/inetd.conf

- Creare un file /virtual/conf.service
- Creare eventuali script virtuali ove siano necessari.

12 Conclusione

Questi sono tutti i passi necessari. Ricordo nuovamente di inviare qualunque commento a: *Computer Resource Center* . Se avete una qualche correzione o un aggiornamento da proporre, fatemelo sapere e lo aggiungerò al documento.

Questo documento ha ricevuto un'ottima accoglienza. Ringrazio tutti coloro che mi hanno inviato domande, dato che hanno permesso che il documento venisse incontro alle necessità comuni degli utenti. Prima di interpellarmi su una questione Vi prego però di leggere la FAQ per vedere se la domanda ha già ricevuto risposta. Grazie di nuovo. *Brian* .

13 FAQ

D1. Ho creato sendmail.init e syslogd.init. Li ho messi in /usr/local/bin e ho cercato di eseguirli, ma ottengo degli errori.

R1. Questi file sono chiamati 'init script'. Sono eseguiti dal programma init nella fase di inizializzazione del sistema. Non c'entrano con i file binari di /usr/local. Consulta la 'Linux System Administrators Guide' o la 'Linux Getting Started Guide' [anche in italiano su *Guide LDP tradotte* N.d.T.] per informazioni sull'uso degli 'init script'.

D2. Ho messo queste linee in /etc/sendmail.cf

```
divert(0)
VERSIONID(`tcpproto.mc')
OSTYPE(linux)
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(local)
MAILER(smtp)
```

E ho ricevuto degli strani messaggi di output. Perché?

R2. Non devi mettere queste linee direttamente in /etc/sendmail.cf. Il file sendmail.cf è stato ideato per essere di facile comprensione per sendmail e di difficile lettura per gli umani. Dunque per facilitare la configurazione noi umani usiamo un programma chiamato m4 e le sue capacità di gestione tramite macro per creare il file sendmail.cf. Le linee che iniziano con FEATURE sono in effetti delle macro che devono essere espanse in istruzioni di configurazione di sendmail. Esamina la documentazione su sendmail per capire come configurare sendmail con questo metodo. Nota inoltre che così creerai un file di configurazione principale /etc/sendmail.cf, file che lo script virtfs poi copierà in /virtual/domain1.com/etc/sendmail.cf. Quindi devi modificare il sendmail.cf della directory /virtual affinché sendmail risponda in modo appropriato.

D3. Dove trovo virtuald, che cos'è e come lo devo usare?

R3. Virtuald è un programma da me scritto per eseguire un servizio virtuale. L'ho incluso come codice sorgente in linguaggio C in questo HOWTO. Lo puoi compilare come un normale programma in C con `make` `virtuald`. Il file binario risultante viene installato in /usr/local/bin. Aggiungendo delle apposite linee in /etc/inetd.conf potrai usarlo come 'wrapper' per un normale programma server di rete.

D4. E se non ho dialog installato sul mio sistema?

R4. Dialog è un programma che permette di utilizzare finestre di dialogo a scomparsa ('dialog pop-up window') negli script di shell. È richiesto per il funzionamento degli script di shell di esempio che si trovano in questo HOWTO. Puoi ottenere una copia di dialog presso *sunsite* . È di facile compilazione e installazione.

D5. Come posso sapere se il syslogd virtuale funziona?

R5. Quando virtuald parte dovrebbe inviare i seguenti messaggi a syslogd (/var/log/messages):

```
Nov 19 17:21:07 virtual virtuald[10223]: Virtuald Starting: $Revision: 1.49 $
Nov 19 17:21:07 virtual virtuald[10223]: Incoming ip: 204.249.11.136
Nov 19 17:21:07 virtual virtuald[10223]: Chroot dir: /virtual/domain1.com
```

Il messaggio circa la directory su cui viene fatto `chroot` è inviato da virtuald dopo l'esecuzione della chiamata di funzione `chroot`. Se appare questo messaggio, il syslogd virtuale funziona. Se si possono vedere i messaggi che il servizio che si sta rendendo virtuale passa a syslogd, anche questo è un segno che il syslogd virtuale è configurato correttamente.

Nota che se non è stata attivata l'opzione di compilazione VERBOSELOG virtuald non passerà nessun messaggio a syslogd. In questo caso si può dire che il syslogd virtuale funziona correttamente se il programma demone che si sta rendendo virtuale riesce di suo a passare qualche messaggio a syslogd.

D6. Come posso configurare le quote disco tenendo conto dei vari filesystem virtuali?

R6. Puoi configurare le quote disco come faresti normalmente. Puoi consultare il *Quota mini-HOWTO* . Comunque devi essere sicuro che non ci siano conflitti di uid tra i vari domini. Se ci sono conflitti avrai più utenti che condividranno una stessa quota. Tieni da parte un intervallo di uid riservati agli utenti che avranno la quota disco abilitata e fa in modo che i tuoi domini non abbiano altri utenti in tale intervallo tranne quelli registrati per avere una quota disco.

D7. Che cosa sono i '\' nelle voci di inetd.conf?

R7. È solo un metodo per spezzare su più righe una singola linea di configurazione. L'ho usato in modo da suddividere a piacere la riga. Puoi ignorare il '\' e riunire nuovamente le due righe insieme.

D8. Quando lancio passwd o altri programmi di login ricevo un `permission denied`. Quando lancio FTP o 'su' ricevo un `no modules loaded for service XXX`. Perché?

R8. Sono messaggi di errore di PAM. Ho ideato questi script prima che uscisse PAM. Il mio script virtfs non copia /etc/pam.d, /usr/lib/cracklib_dict.*, /lib/security e nessun altro dei file richiesti per il corretto funzionamento di PAM. Modificando lo script virtfs in modo da copiare anche questi file il problema si risolverà.

D9. Virtuald può lavorare assieme ai file di tcpd: hosts.allow e hosts.deny?

R9. Sì, con opportune modifiche lo può fare.

Per prima cosa il sorgente dev'essere modificato in due punti.

Dev'essere inserito quanto segue nel punto in cui gli argomenti vengono controllati.

```
        if (!argv[3])
        {
                syslog(LOG_ERR,"invalid arguments: no program to run");
                exit(0);
        }
```

La linea ‘exec’ dev’essere cambiata da:

```
        if (execvp(argv[2],argv+2)<0)
```

in:

```
        if (execvp(argv[2],argv+3)<0)
```

Come secondo passo le linee di inetd.conf devono essere modificate da:

```
ftp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.ftp wu.ftpd -l -a
```

in:

```
ftp stream tcp nowait root /usr/local/bin/virtuald \
        virtuald /virtual/conf.ftp tcpd wu.ftpd -l -a
```

Come terzo passo modifica in modo appropriato i file /virtual/domain1.com/etc/hosts.allow e /virtual/domain1.com/etc/hosts.deny.

D10. I miei host virtuali possono eseguire script CGI?

R10. Sì, lo possono fare, ma ti raccomando di mettere i /cgi-bin in un posto non accessibile dopo il `chroot`, al quale abbia accesso solo tu. Ad esempio /var/www/cgi-bin/domain1.com. Permettere ai client l’accesso a /cgi-bin significa dare loro la possibilità di eseguire programmi sul tuo server. Ciò costituirebbe un grosso problema di sicurezza. Fa’ attenzione. Personalmente non lascio eseguire nessun cgi sui miei sistemi se non dopo aver controllato di persona l’assenza di bug.

D11. I miei file di configurazione sono diversi dagli esempi riportati. Che devo fare?

R11. Ci sono due stili fondamentali di configurazione: SystemV e BSD. Gli esempi riportati in questo HOWTO sono basati sui file di configurazione nello stile SystemV. I servizi virtuali funzionano ugualmente bene in entrambi i sistemi. Per informazioni sui file di configurazione stile BSD consulta le fonti della tua distribuzione o il più vicino sito LDP.

D12. Ti ho scritto una e-mail e non ho ricevuto alcuna risposta o c’è voluto molto tempo per averla. Perché?

R12. Probabilmente perché non hai messo VIRTSERVICES HOWTO nel soggetto del messaggio. Ti prego di tenere a mente che sono un amministratore di reti e che, tra le altre cose che faccio nelle mie giornate sempre troppo brevi [“in my 20 hour days” nell’originale N.d.T.], mi prendo cura dei box virtuali miei e dei miei clienti. I messaggi correttamente indirizzati trovano sempre risposta in due o tre giorni. I messaggi che invece non contengono il soggetto di cui sopra non vengono depositati nella mia casella VIRTSERVICES e possono non ricevere alcuna attenzione per giorni o anche settimane.

D13. Virtuald funziona su connessioni sotto i 100Mbit?

R13 La velocità della scheda di rete non è correlata al funzionamento di virtuald. Prova ad assicurarti che il tuo server lavori sotto i 10 Mbit e che la tua scheda di rete a 100 Mbit funzioni normalmente in assenza di un server virtuale.

D14. Dovrei usare la tabella virthost di sendmail?

R14. No. Quella è una funzionalità di sendmail che gli permette di ricevere informazioni per la gestione di domini multipli. Virtuald fornisce ad ogni sendmail il suo proprio ambiente, separato dagli altri tramite `chroot`. Installa virtuald e poi configura sendmail come faresti normalmente per ogni singolo dominio.

D15. Posso configurare un telnet virtuale sulla mia macchina? Che ne pensi della creazione di un account di root virtuale che permetta ai clienti di amministrare i propri domini?

R15. Questo genere di domande mi vengono fatte piuttosto spesso e, per essere onesti, mi stanno un po' stufando. La risposta, come espresso già parecchie volte nella documentazione, è che qualsiasi servizio eseguito attraverso inetd può essere reso virtuale usando lo script virtuald, quindi non c'è nulla che impedisca di farlo. Nulla eccetto il buon senso. Qualunque beneficio possa derivare dal permettere il telnet è ampiamente superato dai costi in termini di sicurezza per il box virtuale e di conseguenza per i siti, che si suppone debbano essere gestiti in modo responsabile. Di seguito cito solo alcuni dei punti in discussione:

- Allo scopo di ingannare compiutamente una sessione telnet in entrata si dovrebbe fare qualche modifica al kernel, aggiustare l'indirizzo IP sorgente per le connessioni in uscita, ingannare gethostname per fargli usare l'hostname virtuale e non quello del sistema reale, eccetera. Se si hanno conoscenze avanzate, ci si diverta pure a smanettare sul kernel, ma non lo consiglio certo a principianti o simili.
- Permettere agli utenti di accedere alla propria macchina via telnet significa consentire loro di lanciare programmi a piacere. Usando trucchi ben noti essi potrebbero acquistare i privilegi di root e procurare danni al sistema.
- Dare un account telnet con privilegi di root su un box virtuale è malsano. Come root nell'ambiente virtuale un utente può leggere comunque i file di device a basso livello, annullando in pratica il `chroot`, può spegnere il sistema, e può uccidere altri processi in esecuzione.
- I programmi che vengono eseguiti in queste sessioni telnet occupano un certo tempo di CPU, prezioso per i servizi di rete.
- Telnet è un servizio insicuro. Lungo la rete le password vengono trasmesse in chiaro. Se un utente malintenzionato riuscisse ad ottenerle, potrebbe utilizzare i trucchi sopramenzionati per mettere in pericolo il sistema.
- Gli ambienti virtuali avranno bisogno di uno spazio maggiore. Serve lo spazio per un numero maggiore di librerie condivise, file di configurazione e file binari. Anche su un disco da sei gigabyte lo spazio disponibile può esaurirsi molto rapidamente.

L'idea di fondo è che permettere il login su un box virtuale è una pessima cosa. Ove lo si permettesse, ogni sito ospitato sulla macchina sarebbe a rischio. Se si desidera permettere al titolare di un sito di amministrare da sè i propri utenti allora raccomando di scrivere il codice (non si tratta di script) necessario a lanciare i processi virtuali che permetteranno di aggiungere, eliminare o modificare gli account degli utenti su un collegamento in ssh. Dovrebbe essere completamente guidato da menu, non dovrebbe permettere l'accesso ad una console e non dovrebbe girare con i permessi di root. Per ottenere ciò si dovà cambiare il proprietario dei file opportuni da root a qualche altro utente. Se fatto in questo modo, sarà forse abbastanza sicuro da poter essere usato su una macchina virtuale. Non è mai il caso di permettere il collegamento sulla macchina come root, sia in telnet che in ssh. Permetterlo vuol dire semplicemente che si stanno cercando dei guai. Se c'è una ragione schiacciante per dover far girare telnet, allora il sito dovrebbe essere ospitato su una macchina a parte, così da limitare il rischio a quella macchina. Nessun amministratore responsabile dovrebbe fare altrimenti, quindi non spenderò altro tempo sull'argomento.

D16. C'è da qualche parte un file rpm o tar, un sito web, una lista di discussione ecc. che si occupi di virtuald e del Virtual-Services HOWTO?

R16. Attualmente non è disponibile nulla del genere. Questo HOWTO è la sola fonte di informazione per quanto riguarda questo progetto. Trovo che questo HOWTO sia abbastanza autonomo da rendere superflue altre fonti di informazioni più frammentarie.

D17. Quando lancio virtexec come utente normale ricevo il messaggio `chroot: operation not permitted`. Perché?

R17. `Chroot` è una chiamata di sistema ristretta ai privilegi di root. Solo il superutente può eseguirla. Lo script virtexec lancia il programma `chroot`, per cui è necessario essere root per eseguirlo con successo.

D18. Ho configurato pop e sendmail ma il prelievo della posta a mezzo pop non sembra funzionare. Che succede?

R18. Alcuni programmi che gestiscono il servizio pop usano /usr/spool/mail come directory per i file di posta. So ad esempio che qpop dev'essere modificato a livello sorgente per risolvere il problema. Quindi ricompila il sorgente opportunamente modificato o collega con un link simbolico /virtual/domain1.com/usr/spool a /virtual/domain1.com/var/spool.

D19. Non ho usato il programma citato nel tuo HOWTO. Ho usato il programma XXX. Non funziona. Perché?

R19. Nei miei esempi ho cercato di fare in modo di usare i più generici e diffusi tra i vari server a disposizione. Ad ogni modo mi rendo conto che ognuno ha il suo programma preferito. Cerca di inviarmi quante più informazioni utili è possibile. Proverò ad immaginare una soluzione al tuo problema e la riporterò in questa FAQ. L'informazione più importante da inviarmi è dove trovare la versione del software che stai usando (nella forma ftp://ftp.domain1.com/subdir/subdir/file.tgz).

D20. Quando lancio virtexec mi dice: `Il link simbolico non è a una funzione virt`. Cosa significa e come posso risolvere il problema?

R20. Virtexec è un programma che prende l'argomento zero [il nome con cui viene invocato da riga di comando N.d.T.], elimina i suoi primi quattro caratteri, ed esegue il programma dal nome rimanente nell'ambiente virtuale. Ad esempio, virtpasswd fa eseguire passwd. Se i primi quattro caratteri che deve eliminare non sono `virt` allora emette quel messaggio di errore. Virtexec è uno script di shell e dovrebbe risultare di facile comprensione. Ricorri alle pagine man di bash, o di qualunque altra shell tu stia usando, per questioni concernenti la programmazione in linguaggio di shell.

D21. Ho una domanda su Qmail, SAMBA, Apache, ecc. che non è correlata con la configurazione di virtuald o con l'uso del pacchetto in rapporto a virtuald.

R21. Tutti i pacchetti citati hanno una documentazione completa. Alcuni hanno anche siti web del tipo www.packagename.org a loro dedicati. Puoi consultarli per questioni riguardanti i pacchetti che non abbiano legami con il loro funzionamento in un ambiente virtuale.

D22. Ho parecchi alias di dominio per domain1.com ma la posta continua a rimbalzare dagli alias. Che succede?

R22. Virtmaildelivery utilizza unicamente le variabili di ambiente che gli vengono passate per determinare a quale directory /virtual/domain1.com consegnare i messaggi. Non effettua infatti alcun lookup DNS per determinare l'indirizzo del messaggio. Comunque, se l'indirizzo è submail.mail.domain1.com, virtmaildelivery proverà prima con quell'indirizzo, poi con mail.domain1.com, quindi con domain1.com e poi com in quest'ordine fino a che si abbia un riscontro positivo o non vi sia più un nome di dominio con cui provare.

Nondimeno se si hanno alias di dominio che non sono sottodomini uno dell'altro è necessario creare link simbolici come ad esempio:

```
cd /virtual
ln -s domain1.com domain1alias.com
```

In questo modo virtmaildelivery sarà portato a credere che esistano entrambe le directory, anche se una è solo un link simbolico, e la posta potrà essere consegnata a user@domain1.com o user@domain1alias.com. Si noti che virtexec, ove eseguito, mostrerà entrambi i domini nella casella di dialogo. Si può scegliere uno qualunque dei due, dato che si tratta in realtà dello stesso filesystem virtuale.