Automount mini-Howto

don@sabotage.org

v0.3, 22 ottobre 1998

Questo file descrive il programma autofs, una utility che esegue il mount automaticamente. Viene descritto come configurare questa utility e come risolvere i problemi più comuni. Traduzione di Zaxa Zaxu <bezzolone@yahoo.it>

Indice

1 Introduzione

1.1 Automount - come e perché

L'automount è il processo che permette il mount e l'unmount di alcuni filesystem in maniera automatica. Questa operazione viene eseguita da un demone. Se il filesystem si trova in stato unmounted e l'utente cerca di accedervi, viene automaticamente eseguito il mount. Questo tipo di funzionalità è utile in particolar modo in ambienti di rete molto estesi, oppure quando si ha la necessità di avere numerosi mount incrociati anche tra poche macchine (specialmente se alcune non sono sempre attive). Altri casi in cui è molto utile avere l'automount sono:

- la presenza di unità rimuovibili
- la necessità di avere il mount di filesystem dos in modalità conversione ascii attivata e disattivata contemporaneamente.

1.2 Tipi di automount

Ci sono due diversi tipi di automounter in linux; *AMD* e *autofs*. AMD è il demone automount che funziona in maniera analoga al demone AMD del SunOS. È implementato nello spazio utente e non fa parte del kernel. Non è necessario per il kernel avere la funzionalità di automount integrata se viene eseguito il mount NFS utilizzando il demone AMD, in quanto il traffico diretto ai filesystem automontati vengono passati attraverso il sistema NFS. Autofs è un sitema più recente ed innovativo ed è assistito dal kernel. Questo significa che il codice del kernel che gestisce i filesystem conosce quali sono i punti di mount dei filesystem automontati. Il programma che esegue l'automount si rivolge al kernel per conoscerli. In questo mini-howto verrà descritta solo l'utility autofs.

2 Installazione

L'utility autofs si basa su funzionalità del kernel, perciò il kernel deve essere compilato abilitandone il supporto. Nelle versioni del kernel 2.0.xx questa opzione si presenta come sperimentale, anche se si è dimostrata sufficientemente stabile. Nelle versioni del kernel 2.1.xx (e 2.2.xx) non è più sperimentale.

Per il corretto funzionamento sono necessari anche il programma automount e i file di configurazione. Utilizzando la distribuzione RedHat (in cui il package RPM fa parte della installazione) non dovrebbero sorgere problemi. Il programma automount dovrebbe essere inizializzato da uno script rc che si trova in `/etc/rc.d/init.d`. La corretta installazione del package RPM installa anche questo script, ma bisognerebbe assicurarsi che questo script venga eseguito al boot, creando un link in una delle directory rc?.d, utilizzando il pannello di controllo fornito con RedHat, oppure, se si utilizzano altre distribuzioni, utilizzando un qualunque altro metodo. Non è comunque necessario comprendere a fondo quali operazioni esegue lo script di avvio. Se si sta leggendo questo howto, non dovrebbe comunque interessare.

3 Configurazione

L'installazione del package RPM dovrebbe essere abbastanza agevole, ma ci sono alcuni particolari che è meglio controllare, soprattutto se non lo si è mai fatto prima.

Ci sono due file in /etc, che si chiamano `auto.master` e `auto.misc`. Il contenuto del mio auto.master è il seguente:

```
/auto   /etc/auto.misc  --timeout 60
```

Il primo parametro non è il mount point, ma il percorso base del set di mount point. Il secondo parametro indica dove trovare i mount point. Il terzo parametro indica che i filesystem di cui si è eseguito l'automount possono cominciare a tentare l'unmount 60 secondi dopo l'utilizzo. Non viene ovviamente effettuato l'unmount se i filesystem sono in uso.

Auto.misc è il "map file" predefinito. Possono essere definiti diversi "map file" in auto.master. Questo è il contenuto del mio auto.misc:

```
kernel          -ro,soft,intr           ftp.kernel.org:/pub/linux
cd              -fstype=iso9660,ro      :/dev/cdrom
zip             -fstype=auto            :/dev/hdd4
floppy          -fstype=vfat            :/dev/fd0
```

La prima colonna ("chiave") indica il mount point. In base all'esempio che stiamo considerando diventa /auto/kernel, /auto/cd, /auto/zip ecc. La seconda colonna indica le opzioni di mount. Per l'elenco i opzioni leggere la pagina di manuale relativa a mount (man mount). L'ultima colonna indica la provenienza del filesystem. La prima voce ("kernel") è un mount NFS. Il simbolo : sulle altre righe indica un device locale.

4 Aspettando l'unmount

Qualche lettore avrà dato un'occhiata ai 60 secondi necessari per l'unmount automatico e avrà pensato: *Forse è un po' troppo per attendere l'unmount di un dischetto... Forse è sufficiente eseguire un sync e tirar fuori il dischetto, così nessuno se ne accorge!* Vorrei suggerire `una strategia più sicura`. Innanzitutto è possibile modificare il timeout. Ma questo potrebbe non bastare, o diventare poco efficiente in termini di prestazioni. Un'altra possibilità è quella di mandare il segnale `SIGUSR1` al processo automount (utilizzando kill). Ma prima che si inizi a creare bottoni sul desktop per eseguire unmount, è bene chiarire una serie di problemi che potrebbero insorgere.

Il processo automount lanciato dall'utente root e accetta segnali solamente da root. Una delle ragioni che potrebbero spingere l'utente ad utilizzare automunt è al contrario la possibilità di eseguire queste operazioni SENZA essere root. Sarebbe semplice creare un programma C che esegue il "lavoro brutale" come suid-root. Comunque è possibile utilizzare sudo per permettere agli utenti di mandare l'opportuno segnale di kill. L'unico problema è che sudo non permette di utilizzare gli apici (') per processare le opzioni che permettono di individuare il PID corretto del processo a cui mandare il segnale. Si dovrebbe invece disporre del programma killall, che permette di esegure questo (ringrazio per i suggerimenti ricevuti):

```
ALL     ALL=NOPASSWD:/usr/bin/killall -USR1 automount
```

In alternativa si dovrebbe permettere agli utenti di mandare il segnale -SIGUSR1 a tutti i processi. Questo però potrebbe avere effetti collaterali su altri programmi: costringe il riciclo di alcuni window manager e forza la chiusura di xemacs.

5 Domande e risposte

5.1 Non riesco a vedere /auto/floppy, o altri mount point che sto cercando.

Se automount è configurato correttamente, qualunque mount point che si sta ricercando sarà disponibile al momento dell'utilizzo, anche se non lo si vede nel momento in cui non viene utilizzato. Se si sta eseguendo il browse delle directory con un programma grafico potrebbe essere necessario digitare manualmente il percorso. Sicuramente il peggior difetto di autofs è appunto il fatto che non si possa scorrere e scegliere tra i vari mount point (che al momento risultano non utilizzati e quindi invisibili. Se questa particolarità risulta inaccettabile, modificare la configurazione del sistema (per inciso i file che terminano in .c come "configurazione").

5.2 Come faccio a vedere che cosa è in stato di mount?

Usare il comando `df`. `mount` senza opzioni mostra anche le opzioni utilizzate per eseguire il mount.

5.3 Ho inserito ("vfat") che è stato riconosciuto automaticamente come in semplice disco FAT.

Questo non è un problema di automount. Se si specifica, al momento corrente, il paramentro "auto" come tipo di filesystem, non viene testato il mount del filesytem vfat, per cui viene eseguito il mount come msdos.

In effetti VFAT è solamente una variante del filesystem FAT/MSDOS, che permette l'utilizzo dei nomi di file lunghi.

Stando alle osservazioni di uno degli autori di mount (??) il programma mount è soltanto una interfaccia ad una chiamata di sistema, ed è comunque responsabilità dell'utente la specifica del tipo di filesystem. Attualmente viene tentato il mount seguendo un ordine ben preciso nell'elenco dei file system disponibili, piuttosto che un metodo più "euristico", attualmente in via di sviluppo. Nel frattempo è impossibile eseguire il mount di un filesystem vfat a meno che non se ne specifichi esplicitamente il tipo. Molto probabilemnte questo inconveniente verrà risolto a breve.

5.4 Il mio filesystem `/cicciopippo` è in stato di mount e non riesco a fare l'unmount con `kill -SIGUSR1`.

È sicuramente in uso da parte di qualcuno o qualcosa. Probabilmente è impossibile eseguire anche l'unmount manuale. Controllare la presenza di qualche shell posizionata sulla directory, oppure qualche applicazione, directory browser o altro che ha lasciato per così dire la "porta aperta". Se non si riesce ad individuare chi o cosa sta effettuando il blocco, provare ad utilizzare il programma fuser.

5.5 Chi devo ringraziare per la disponibilità di autofs?

Non ringraziatemi. Non ho niente a che fare con questa utility. Ho soltanto voluto evidenziare il grande lavoro che è stato fatto con autofs, e mostrare la sua facilità di utilizzo. Confrontato con il precorsore AMD (che viene tra l'altro venduto ad un prezzo maggiorato perché corredato di una serie di utility a dir poco preistoriche), autofs è molto ben documentato e i suoi sviluppatori hanno tutti i miei ringraziamenti. Il copyright indicato è Transmeta, per cui non ho a disposizione una lista di nomi.

5.6 Dove posso trovare maggiori informazioni sull'automount?

E possibile trovare un tutorial su autofs qui `<http://www.linuxhq.com/lg/issue24/nielsen.html>`. Provare inoltre le am-utils qui `<http://www.cs.columbia.edu/~ezk/am-utils>`.

(Ringrazio per questi URL)