SelfLinux-0.13.1



Automount und Autofs



Autor: Florian Frank (florian@pingos.org)
Autor: Guido Socher (guido@linuxfocus.org)
Autor: Frédéric Raynal (pappy@users.sourceforge.net)
Formatierung: Florian Frank (florian@pingos.org)
Lizenz: GFDL

automount und autofs sind sehr leistungsfähige Werkzeuge, um das Management von Dateisystemen zu erleichtern. Sie erlauben es dem Benutzer eines Rechners, automatisch verschiedene Dateisysteme zu mounten in dem Moment, wo sie gebraucht werden.

Inhaltsverzeichnis

- 1 Einführung
- 2 Beschreibung
 - 2.1 autofs
 - 2.2 automount
- 3 Konfiguration
 - 3.1 /etc/auto.master Datei
 - 3.2 Automount maps
 - 3.3 autofs und NIS
 - 3.4 autofs und LDAP
- 4 Die letzten Details
- 5 Zusammenfassung

1 Einführung

In Abhängigkeit von den physikalischen Gegebenheiten (Festplatten, Disketten, CD-ROMs, ...) und dem Betriebssystem ist der Vorgang des Schreibens von Daten unterschiedlich: Man nennt dies **Dateisystem**. Dies ist zwar eine sehr vereinfachte Darstellung, aber hier ausreichend. Die Datei /etc/fstab enthält die festen mount-Punkte, die beim Starten gemountet werden. Jeder mount-Punkt entspricht einem Verzeichnis im Verzeichnisbaum. Später, wenn man auf andere Punkte im Dateisystem zugreifen möchte, dann kann nur root das Dateisystem mounten. Eine Ausnahme sind Einträge, die die Option users" in /etc/fstab enthalten. Abgesehen davon kann ein normaler Benutzer nicht einfach auf alle Daten und Dateisysteme zugreifen.

Die Manpages von mount und fstab erklären diese Befehle und Konzepte genauer.

Beide (automount und autofs) erlauben es dem Systemadministrator alle Dateisysteme, auf die eine Maschine zugreifen kann, zu konfigurieren in einer ähnlichen Art und Weise, wie dies bei dem Befehl mount passieren würde. Für den Benutzer ist das völlig transparent.

2 Beschreibung

Das Paar (automount und autofs) kann als ein client/server System verstanden werden. Ein Server läuft und wartet auf eine Anfrage. Wenn die Anfrage kommt, dupliziert sich der Server. Der duplizierte Teil beantwortet die Anfrage und der andere wartet auf weitere Anfragen.

Hier spielt autofs die Rolle des wartenden Servers und automount die des duplizierten Servers. Die Anfragen kommen aus einer Konfigurationsdatei.

2.1 autofs

autofs wird im allgemeinen beim Booten gestartet, aber der Systemadministrator (root) kann es natürlich manuell starten und stoppen.

autofs hat 4 Optionen:

start	Wie der Name schon sagt, startet diese Option den Prozess. Beim Start sucht autofs nach
	maps (maps spezifizieren die mount-Punkte) in der Konfigurationsdatei /etc/auto.master.
	Danach startet es automount für jeden einzelnen mount-Punkt. Im Anschluss daran sucht
	autofs nach NIS maps.
stop	Stoppt autofs und alle automount-Instanzen
status	Zeigt die augenblickliche Konfiguration und alle laufenden automount-Instanzen an
reload	Liest die map auto.master ein und beendet die nicht mehr gebrauchten
	automount-Instanzen. Es startet neue automount-Instanzen für neu hinzugekommene
	mount-Punkte. Veränderungen der maps werden beim nächsten Neustart berücksichtigt.

Kurzgesagt ist also autofs nichts als ein Skript, das auto.master befragt, bevor es die einzelnen automount-Instanzen für jeden der mount-Punkte in auto.master startet.

2.2 automount

automount arbeitet nur mit einem mount-Punkt (der von autofs gefunden wurde und automount beim

Start übergeben wurde). Eine **map** beschreibt alle Eigenschaften dieses mount Punktes, die nötig sind, um ein Dateisystem automatisch zu mounten. Dieses automatische Mounten wird durchgeführt, sobald jemand versucht, auf etwas zuzugreifen, das im Verzeichnisbaum unterhalb des mount-Punktes liegt.

Weiterhin ist automount für das automatische unmount der Dateisysteme zuständig. Dieses geschieht nach einiger Zeit, wenn auf ein Dateisystem nicht mehr zugegriffen wird. Die Voreinstellung ist 5 Minuten.

3 Konfiguration

Die Konfiguration wird mit Hilfe von 2 Dateien durchgeführt. Die Datei auto.master, enthält alle mount-Punkte, und eine mount-Punkt-Datei, die die Optionen für einen speziellen mount-Punkt beschreibt.

3.1 /etc/auto.master Datei

Hier werden die mount-Punkte im sogenannten Sun maps Format beschrieben.

/etc/auto.master: Jede Zeile beschreibt genau einen mount-Punkt und zeigt zu einer Datei, die die weiteren Optionen für diesen mount-Punkt enthält. Die dritte Spalte erlaubt es spezielle Optionen an mount zu übergeben.

Die Syntax einer Zeile ist:

```
/etc/auto.master

mount-point map-for-the-associated-automount [ -mount-options-separated-by-comma ]
```

Beispiel:

```
/home ldap 10.1.7.7:ou=home,ou=autofs,dc=foo,dc=bar
/misc /etc/auto.misc --timeout 60
/mnt yp:mnt.map -intr,nosuid,nodev
```

Dies konfiguriert 3 mount-Punkte, /home, /misc und /mnt. Um auf Dateien in /misc zuzugreifen, wird automount die Datei /etc/auto.misc lesen und dort eine Beschreibung des Dateisystems finden. In der letzten Zeile sieht man Optionen. Sie werden in den Manpages zu mount beschrieben.

3.2 Automount maps

Die Syntax einer automount map ist fast die gleiche, wie von auto.master:

```
Schlüssel [ -mount-optionen-seperiert-durch-Komma ] Ort-des-Dateisystems
```

Der Schlüssel ist ein symbolischer Name für das Dateisystem unter dem mount-Punkt.

Hier ist die Datei /etc/auto.misc aus dem obigen Beispiel:

```
/etc/auto.misc

kernel -ro,soft,intr ftp.kernel.org:/pub/linux
cdrom -fstype=iso9660,ro :/dev/cdrom
floppy -fstype=auto :/dev/fd0
windoz -fstype=vfat :/dev/hda1
```

Der absolute Pfad zu einer Datei ist dann:

/mount-Punkt/Schlüssel/Pfad/Datei

Man sieht auch, dass die erste Zeile auf ein über NFS exportiertes Dateisystem verweist (nur um zu zeigen, wie flexibel automount und autofs sind).

3.3 autofs und NIS

In einem kleinen Netzwerk zu Hause wird man wahrscheinlich in der Lage sein, ohne NIS zu leben. NIS Network Information Service ist ein Weg, um Konfigurationsdateien (z. B. in /etc) auf andere Maschinen zu veteilen.

Der Hauptserver in unserem Beispielnetzwerk ist **sol**. Drei andere Maschinen des Netzwerkes sind **telesto**, **mars** und **venus**. **sol** ist als NIS Server für die Domain foobar konfiguriert. Die anderen Maschinen sind nur NIS Clients von **sol**.

Zunächst betrachten wir die Konfiguration des Servers sol. Wir beginnen mit dem Definieren einiger NIS maps, die alle notwendigen Informationen enthalten.

Soweit es NFS betrifft ist die Konfiguration sehr eingeschränkt. Die Datei /etc/exports von sol enthält:

```
/etc/exports
/usr/local *
```

An dieser Stelle sollte man auf automount setzen, um auf das exportierte Verzeichnis /usr/local zuzugreifen. Anstatt dieses Verzeichnis zur Boot-Zeit zu mounten, geschieht es automatisch, sobald ein Benutzer auf eine Datei in diesem Verzeichnis zugreift. Man erzeugt die Datei /etc/auto.map, um zu definieren, was zugreifbar sein wird, sowohl durch automount als auch durch NIS:

```
/etc/auto.map

sol sol:/usr/local
```

Da man diese Informationen (die neue auto.map und die netgroup Dateien) in die NIS Datenbank integrieren will, muss man das Makefile von NIS anpassen, bevor die NIS Datenbank neu erstellt wird. Was auto.map betrifft, so so wird diese Datei nicht standardmässig definiert. Man muss nur einen neuen Eintrag in das Makefile hinzufügen, mit der assoziierten Regel (benutzen der existierenden als ein Modell):

```
/var/yp/Makefile

#To be added in the Yellow Pages Makefile

AUTO_MAP = $(YPSRCDIR)/auto.map
# ...
#...
auto.map: $(AUTO_MAP) $(YPDIR)/Makefile

@echo "Updating $@..."

-@sed -e "/^#/d" -e s/#.*$$// $(AUTO_MAP) | $(DBLOAD) \
-i $(AUTO_MAP) -o $(YPMAPDIR)/$@ - $@
-@$(NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@
```

Diese Produktionsregel entfernt nur Kommentare, fügt einen neuen Eintrag in die Datenbank hinzu und übermittelt dann die Information zu jedem NIS Server der NIS-Domain foobar.

Man muss hierzu nur make aus dem Verzeichnis /var/yp heraus ausführen.

```
root@linux / # cd /var/yp
root@linux /var/yp/ # make
```

Nun zu den drei Clients **telesto**, **mars** und **venus**. Hier gibt es nicht viel tun. Man muss lediglich autofs mitteilen, dass es eine neue map, die von YP (NIS) übergeben wird, verwalten soll. In jeder Datei /etc/auto.master des Clients informiert die folgende Zeile über die Anwesenheit einer **map** auto.map, verwaltet durch die YP Dienste (NIS).

```
/etc/auto.master

/usr/local yp auto.map --intr,nosuid,nodev
```

Danach muss autofs neu gestartet werden, damit die neue map wirksam wird.

```
root@linux / # /etc/init.d/autofs restart
Stopping automounter:
   Stopped 798()
done.
Starting automounter: /usr/local.
```

Jetzt hat man ein einziges physikalisches Verzeichnis /usr/local auf sol. Wenn spezifische Programme auf sol installiert werden, können alle Maschinen diese sofort nutzen.

3.4 autofs und LDAP

Die Benutzung von LDAP zur Steuerung des Automounters, bietet sich vor allem in Umgebungen an, in denen bereits ein oder auch mehrere LDAP-Server im Einsatz sind.

Auf Serverseite ist es lediglich nötig das entsprechende Schema für den automounter einzubinden. Dieses

Schema wird, obwohl es in den meisten Fällen sehr nützlich ist, leider nicht mit RopenLDAP ausgeliefert.

Im Folgenden findet man das für den automount nötige Schema, welches man am besten als /etc/ldap/schema/automount.schema speichert.

```
attributetype ( 1.3.6.1.1.1.25 NAME 'automountInformation'
    DESC 'Automount information'
    EQUALITY caseExactIA5Match
    SUBSTR caseExactIA5SubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{1024} SINGLE-VALUE )

objectclass ( 1.3.6.1.1.1.2.13 NAME 'automount' SUP top STRUCTURAL
    DESC 'Automount information'
    MUST ( cn $ automountInformation )
    MAY ( description ) )

objectclass ( 1.3.6.1.4.1.2312.4.2.2 NAME 'automountMap' SUP top STRUCTURAL
    DESC 'An group of related automount objects' MUST ( ou ) )
```

Des weiteren muss diese Datei natürlich auch in der Konfigurationsdatei des LDAP-Servers eingebunden werden:

```
/etc/ldap/slapd.conf

# Schema and objectClass definitions
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/automount.schema
```

Nun müssen die gewünschten Einträge noch im LDAP-Server vorgenommen werden. Es gibt zahlreiche graphische Oberflächen, die hierbei hilfreich sein können. Auf Grund der grossen Zahl dieser Programme, werden hier nur die realen Daten im ldif-Format angegeben.

```
dn: ou=autofs,dc=foo,dc=bar
objectClass: top
objectClass: organizationalUnit
ou: autofs
dn: ou=home,ou=autofs,dc=foo,dc=bar
objectClass: top
objectClass: organizationalUnit
ou: home
dn: cn=/,ou=home,ou=autofs,dc=foo,dc=bar
objectClass: top
objectClass: top
objectClass: automount
cn: /
automountInformation: saturn:/home_phys/&
```

Zu obigen Eingaben wäre die äquivalente auto.home:

/etc/auto.home

* saturn:/home_phys/&

Auf den Clients gibt es auch in diesem Fall nicht sehr viel zu tun. Man muss nur autofs mitteilen, dass eine neue map existiert, die über LDAP bezogen und verwaltet werden soll.

/etc/auto.master

/usr/local ldap 10.1.7.7:ou=home,ou=autofs,dc=foo,dc=bar

Danach muss autofs noch neu gestartet werden. Ab sofort stehen nun unter /home die zentral auf saturn liegenden Home-Verzeichnisse zur Verfügung.

4 Die letzten Details

Zunächst wird man feststellen, dass die automatische Vervollständigung des Dateinamens nicht wie üblich funktioniert. Um nicht immer den vollen Pfad angeben zu müssen, benutzt man oft die TAB-Taste. Solange die entsprechende automount-map nicht geladen ist, funktioniert diese TAB Taste nicht.

Dieses Verhalten ist ganz normal. Offensichtlich wird das Verzeichnis durchsucht, wenn man die TAB Taste drückt (globbing), um die verschiedenen Einträge zu lesen. In diesem Fall ist das Verzeichnis jedoch leer, da es das Ziel ist, das Dateisystem nur dann zu mounten, wenn auf eine Datei aus diesem Dateisystem zugegriffen wird.

Betrachtet man die Ausgabe von mount vor und nach dem Zugriff auf eine map, so sieht man die entsprechenden Einträge. Hier wird wieder die vorherige /etc/auto.master mit nur einem mount benutzt:

```
root@linux / # mount

/dev/hda6 on / type ext2 (rw)
none on /proc type proc (rw)
/dev/hda9 on /home type ext2 (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
automount(pid362) on /home type autofs
(rw,fd=5,pgrp=362,minproto=2,maxproto=3)
automount(pid364) on /misc type autofs
(rw,fd=5,pgrp=364,minproto=2,maxproto=3)
automount(pid366) on /mnt type autofs
(rw,fd=5,pgrp=366,minproto=2,maxproto=3)
```

Man sieht, dass es einen Daemon (daemon -- etwa ein Systemprozess) gibt. Weiterhin ist der zugehörige Typ autofs. Nach dem Zugriff sieht dies so aus:

```
root@linux / # mount

/dev/hda6 on / type ext2 (rw)
none on /proc type proc (rw)
/dev/hda9 on /home type ext2 (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
automount(pid362) on /home type autofs
(rw,fd=5,pgrp=362,minproto=2,maxproto=3)
automount(pid364) on /misc type autofs
(rw,fd=5,pgrp=364,minproto=2,maxproto=3)
automount(pid366) on /mnt type autofs
(rw,fd=5,pgrp=366,minproto=2,maxproto=3)
/dev/hda1 on /misc/windoz type vfat (rw)
```

Die letzte Zeile enthält, was man erwartet hat. Wenn man einige Minuten wartet, ohne auf das Dateisystem zuzugreifen, wird der Eintrag wieder verschwinden.

5 Zusammenfassung

Auf einem einzigen alleinstehenden Computer, der nur Linux und Windows beheimatet, sind automount und autofs relativ nutzlos.

Im Zusammenhang mit Netzwerken ist der Nutzen sehr gross, da automount und autofs mit NIS und LDAP sehr gut zusammenarbeitet und man sich keine Gedanken machen muss, wo sich die Dateisysteme physikalisch befinden. Dies wird von den über NIS und/oder LDAP verteilten maps komplett abstrahiert.

Ein großer Vorteil von automount ist, dass ein ausgefallener Dateiserver nur die Maschinen betrifft, die den Server gerade benötigt hatten. Das kann die Ausfallzeit in einer großen Firma (mit vielleicht 10 NFS Servern) signifikant reduzieren. Auch ist es möglich, sofern ein Backup-Server vorhanden ist, zentral über die Maps des automounters auf diesen umzuschalten. So kann nach einem Totalausfall zumindest auf etwas älteren Daten weitergearbeitet werden.