The Automounter, and automating its Installation

1 Aim

After successfully working through this exercise, You will:

- understand what the automounter does
- be able to set up the automounter manually
- be able to write a shell script suitable for use with `kickstart` to automatically configure the automounter as part of an unattended installation.

2 Background

The automounter provides a convenient way of accessing network drives. You use it when you log into your network account to access your home directory. It is often used with NFS (Network File System), but can also be used to mount local disks such as floppy disks or removable media such as Zip disks and CDROMs.

Currently your home directory is mounted by the automounter when you log into your netork account. The configuration parameters for the automounter are provided by our LDAP directory server. You may see some information about the current automonter configuration by typing:

```
$ service autofs status
```

2.1 Advantages of the Automounter compared with putting entries in /etc/fstab

You can always put an entry into `/etc/fstab` that permanently mounts a network partition on boot. But `autofs` is much more flexible.

- Any user can mount the drive automatically when they need it
- Does not use the system resources when it does not need to be mounted

2.2 The way it is configured

There are two sets of configuration files used by `autofs`: the master file, `/etc/auto.master`, described by `man auto.master`, and the *map* files used to define what can be mounted on individual entries listed in `/etc/auto.master`. These are described in `man 5 autofs`.

The `/etc/auto.master` file has the format:

```
directory   mapfile options
```

The options are optional.

The directory is the top-level directory where automatic mounts occur. This entry points to another file (your map), which takes care of individually defining these mounts and their file system types. See `/etc/auto.misc` as an example of a map file.

3 Procedure

1. Identify the documentation that comes with the `autofs` package with the RPM command:

   ```
   $ rpm -qld autofs
   ```

To read the manual pages for the /etc/auto.master configuration file:

```
$ man auto.master
```

and for the other /etc/auto.* configuration files:

```
$ man 5 autofs
```

Read the documentation.

2. Change to the root directory of your file system, and create a directory called .auto there:

```
$ sudo mkdir /.auto
```

3. create a dangling symbolic link;

```
$ sudo ln -s .auto/ftp /ftp
```

If you type ls here, you will see that the link is shown in red, perhaps blinking, to indicate that the link does not point anywhere. The automounter will actually mount the network drive on this.

4. edit the master automounter file, /etc/auto.master:

```
$ xhost +localhost
$ sudo -v
$ sudo emacs /etc/auto.master
```

5. Edit a line there to read:

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

and make sure there is a newline at the end, and save it.

6. Create a file /etc/auto.ftp that contains this line:

```
ftp     -ro,soft,intr   nicku.org:/var/ftp/pub
```

and make sure there is a newline at the end, and save it.

7. Restart the automounter, and change to that directory:

   ```
   $ sudo service autofs restart
   $ cd /ftp/redhat-7.2
   $ ls -l
   ```

8. If the automounter is not working properly, try seeing what the status of it is. Here is the output from a machine which is working correctly:

   ```
   $ service autofs status
   Configured Mount Points:
   ------------------------
   /usr/sbin/automount --timeout 60 /.auto file /etc/auto.ftp
   /usr/sbin/automount --timeout 60 /home ldap nicku.org:ou=auto.home,
   dc=tyict,dc=vtc,dc=edu,dc=hk
   /usr/sbin/automount --timeout 60 /usr/local ldap nicku.org:
   ou=auto.practical,dc=tyict,dc=vtc,dc=edu,dc=hk

   Active Mount Points:
   --------------------
   /usr/sbin/automount --timeout 60 /.auto file /etc/auto.ftp
   /usr/sbin/automount --timeout 60 /home ldap nicku.org:ou=auto.home,
   dc=tyict,dc=vtc,dc=edu,dc=hk
   /usr/sbin/automount --timeout 60 /usr/local ldap nicku.org:
   ou=auto.practical,dc=tyict,dc=vtc,dc=edu,dc=hk
   ```

Please note that I have broken the long lines here into two so that they can fit on the page.

Note too that anything which prevents NFS, such as a firewall, will stop you from mounting a network drive. This should not be the case with you. Here is how to test if you have a firewall.

```
$ sudo ipchains -L -n
```

Output that looks like this:

```
ipchains: Incompatible with this kernel
```

or that looks like this:

```
$ sudo ipchains -L -n
Chain input (policy ACCEPT):
Chain forward (policy ACCEPT):
Chain output (policy ACCEPT):
```

show that your firewall is turned off.

4 Automating this with a shell script

The next step is to automate this. The aim is to use this script with `kickstart`[*] in a future lab session so that you can automatically configure the automounter when you perform an unattended installation.

4.1 Procedure for Scripting the automounter setup

Do this *only* after you are certain that your automounter is working properly. You should buld your shell script up piece by piece, not try to write it all at once.

1. undo the setup completely:

- **(a)** delete the directory `/.auto`
- **(b)** delete the file `/etc/auto.ftp`
- **(c)** delete the line

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

  from `/etc/auto.master`.

2. Now write the code to create the directory and the symbolic link. The code should work no matter where you run the script.

You may find it helpful to refer to the notes about symbolic links on our subject web site.

Make sure that this works before proceding.

3. The pseudo code for the whole system is:

```
if there is no line containing /etc/auto.ftp in /etc/aut
    add the appropriate line to /etc/auto.master
```

[*] Kickstart is a system for performing automated system installation.

```
overwrite any file /etc/auto.ftp with the appropriate co
make the directory /.auto
if any file or directory exists with the name /ftp
    delete it.
create the symbolic link
turn on the autofs service with chkconfig in runlevels 3
```

4. Show your lecturer your working shell script.