

Lab 4: NFS and Automounting

In this lab, you will setup your system for NFS and automounting. Perform the indicated steps, and write the answers to questions in the appropriate spaces on this hand-out. Turn-in one filled in handout per computer system by the beginning of class next week. Be sure to include your name (and your partner's name, if you have one). Note: For this lab, you will only be able to access the machines within your subnet.

Configure the system

Bring the system up single-user and perform the initial network configuration.

- Step 1. Boot your system into single user mode. To accomplish this, reboot the system, and interrupt the Red Hat splash screen at the beginning of the boot process with Control-X. Then type **linux single** at the **boot:** prompt.
- Step 2. Configure your system with the IP address 10.0.0.N/24, where N is your system number. Don't forget about the files **/etc/hosts** and **/etc/sysconfig/network**. Refer to Lab 1 if necessary – by now, you should be able to reliably configure your network in just a few minutes, and understand the relationship between the various configuration files and commands.

Enable the NFS server/client services

Setup the system to run the NFS client and server services when in multi-user mode.

- Step 3. Check your system to see if NFS will be started automatically when you come up to multi-user mode (init level 3, 5). Examine the **/etc/rc3.d** directory to see if a symbolic link exists to the script **/etc/init.d/nfs**. It might be named **KNNnfs** or **SNNnfs**, where NN are numbers. Also check the **/etc/rc5.d** directory, which is the same init level as **/etc/rc3.d**, but adds the graphical login instead of the textual login. If a symbolic link whose name starts with an S does not exist, the service will not start automatically.

Q1. Which symbolic links link to the **/etc/init.d/nfs** file? _____

- Step 4. You can create the link manually, or you can use the Red Hat **chkconfig** program to create the link in all the appropriate directories. Enable the service by creating the link. Be sure that the symbolic link created looks like the others in the directory, and be sure the symbolic link can be used to access the file **/etc/init.d/nfs**. The command line for **chkconfig** is below:

```
# chkconfig nfs on
```

Q2. Now which symbolic links link to the **/etc/init.d/nfs** file? _____

- Step 5. The NFS start script just linked must run after the **portmap** and **nfslock** scripts - check these script numbers too. Remember, the scripts are run in numerical order (**S** = start, **K** = kill).

Q3. What are the names of the links for the **portmap** and **nfslock** startup scripts? _____

Exporting a directory

Export a directory to be available via NFS to NFS clients on the network.

- Step 6. You are going to create a directory that will be exported to members of your network. It is fairly common to use the **/exports** directory as a place holder for directories that you share. This lets the system administrators immediately recognize that everything under **/exports** might be in use by others. Create the directory **/exports/hostname**, where *hostname* is the name of your host.

Q4. What are the permissions on this directory? _____

- Step 7. Create the file **/exports/hostname/test.hostname** - you can use the **touch** command if you want to create the file. This file will be used only to help your net-mates see that they have successfully mounted your shared directory. If they see the file, they have mounted your exported directory.

- Step 8. Export the **/exports/hostname** directory as read-only to all of the members of only this network. This requires adding the appropriate line to the file **/etc/exports**. Like most daemons, the NFS daemon **nfsd** will not know about the changes; you must use the **exportfs** command with the appropriate option to inform **nfsd** of the changes. See the **exportfs** man page for a list of options, or the lecture notes. Hint: the **-v** option is handy too.

- Q5.** What are the differences between the **-r** and **-a** options of **exportfs**? _____
- Step 9. You can validate the list of filesystems actually being exported (that is, those that are in the **/var/lib/nfs/xtab**). Run both **exportfs** with no arguments, and **showmount -e** to see a list of the filesystems exported by your host. Make sure that the correct filesystems are being exported as you expect.
- Q6.** Which of the NFS daemons are running on your system? _____
- Q7.** How many **nfsd** processes are running? _____
- Q8.** What does the **-e** option to **showmount** mean? _____
- Step 10. Bring the system up multi-user (run level 3) and login as root. NFS should now be running. Validate that the appropriate daemons are running.

Mounting NFS filesystems

Mount NFS-exported filesystems onto your system. Validate access (read/write) permissions.

- Step 11. To mount a filesystem, you must have a mount point. It is very common to use directories under **/mnt** as place holders for mount point. The reason for this is that you never want to mount any NFS filesystem onto a mount point in **/**, as this causes system hangs when the NFS server is down. And **/mnt** is a convenient place to do this. Create a mount point directory **/mnt/temp** to be used to mount your own exported filesystem.
- Step 12. Mount your own shared filesystem **/exports/hostname** onto your mount point directory **/mnt/temp**. Trying to mount your own filesystem will validate that you have the filesystem exported correctly, and that the proper daemons are running. If you are having trouble mounting your own exported filesystem, consider the following:
- Did you give the correct **mount** command?
 - Are all the necessary daemons are running? Try: **ps -ef**, and **rpcinfo -p**
 - Is the filesystem properly exported? Try: **showmount -e**
 - Is the loopback interface functional? Try: **ping localhost**
- Q9.** Why is the loopback interface required to be running for this test? _____
- _____
- Step 13. Verify that the file **/mnt/temp/test.hostname** exists. If not, retrace your steps to see what when wrong.
- Step 14. Un-mount the filesystem mounted at **/mnt/temp**.
- Step 15. Find a host in your row, or elsewhere on the network, that has exported a filesystem. Use the command **showmount -e remote-host** (where *remote-host* is the hostname or IP address of one of the hosts on the network) to see which filesystem(s) are being exported from the NFS server *remote-host*.
- Step 16. Now try to mount their exported **/exports/remote-host** filesystem. You can mount it on the already created mount point **/mnt/temp** if you want, or you can create a new host-specific mount point such as **/mnt/remote-host**. The latter will allow you to mount several filesystems at the same time. If you are having trouble mounting the filesystem, be sure the administrator of *remote-host* has correctly exported it.
- Step 17. Test that the mount was successful – use the **mount** command to see a list of the mounted filesystems. Make sure that you see the contents of the filesystem you believe you just mounted. Run **ls -l** on the mount point directory (either **/mnt/temp** or **/mnt/remote-host**, depending on where you mounted the filesystem) to see if the file named **test.remote-host** exists.
- Step 18. Run **ls -ld** on the mount point to see the permissions of the directory itself (instead of its contents).
- Q10.** What are the permissions of the directory? _____
- Q11.** Are these the permissions of your mount point directory, or of the fileserver's directory? _____
- Q12.** Do you have write permissions on this directory? _____
- Step 19. Try to create a file in the mounted directory.

Q13. Explain why you can or cannot create the file. _____

Step 20. Perform Step 15 through Step 19 using some other host(s) in the lab. Be sure not to mount another filesystem onto the already used mount point – create a new mount point instead.

Step 21. Un-mount all the NFS-mounted filesystems with the single command below:

```
# umount -a -t nfs
```

Q14. What are the `-a` and `-t` options? _____

/etc/fstab

Add the convenience and boot-time entries for mounting NFS filesystems.

Step 22. Add entries into your **/etc/fstab** so that you can mount the NFS filesystems more conveniently. You probably do not want them to be mounted at boot time, since the machines on your net may not be up when you are booting.

Q15. What option(s) do you use to ensure filesystems are not mounted at boot time? _____

Q16. What mount options did you specify? _____

Step 23. Re-mount the same filesystems you mounted earlier, this time using only the full path of the mount point directory. Since you have added entries in **/etc/fstab**, you can use this shortcut.

Step 24. Re-edit your **/etc/fstab** and set one of these filesystems to automatically mount at boot time.

Step 25. Verify that others do not have any of your filesystems mounted using **showmount -d**. If they do, ask them to un-mount these filesystems. Do not proceed to the next step until all others have unmounted your export filesystem.

Step 26. To validate that the filesystem mounts when coming up into multi-user mode, reboot your system into multi-user mode. Make sure the system administrator of the remote filesystem knows you are performing this test, so that they keep the system running and the filesystem exported! Otherwise, your system might hang if you did not use the correct options.

Q17. How do you know if the filesystem was mounted when the system came up multi-user? _____

Q18. What mount option keyword was required in **/etc/fstab** for the filesystem to be mounted during multi-user bring up? _____

Q19. What single standard UNIX command is run by the system startup scripts to mount all the NFS filesystems? _____

Step 27. Re-edit **/etc/fstab** and disable the automatic boot-time mounting for all NFS filesystems. Un-mount all NFS filesystems.

Exporting and Permissions

Export a directory to be available to only a specific host.

Step 28. To simulate two filesystems that you might want to share with others, create two directories **/exports/readonly** and **/exports/readwrite** both with permissions 755, owned by root. Configure sharing of these two directories as read-only and read-write, respectively.

Step 29. So that you can mount the exported directories from your partner's system, create two new mount points **/mnt/readonly** and **/mnt/readwrite**.

Step 30. Add entries into your **/etc/fstab** so that you can mount the two remote filesystems onto these mount points you just created.

Step 31. After your partner is ready, mount the remote shared readonly and readwrite directories onto your mount points.

Step 32. Try to create a file in each the mounted directories with the **touch** command.

Step 33. You should not have been able to write into the read-only filesystem; but you should have also discovered that you were not able to create a file in the read-write directory either. Although the filesystem is mounted read-write, and the remote host has the filesystem exported read-write, the permissions on the directory itself still controls access. If the administrator of the remote host created these directories as root, the permissions will likely be 755 (writable only to root and not group or others). Although you are UID 0 on your system, this is an example of how the default **root_squash** mounting option affects and re-maps UID 0 under NFS. Your UID is 0 locally, but as far as the remote NFS server is concerned, you are UID -2.

Q20. There are three conditions required to be able to write to a mounted, remote NFS directory – what are they? _____

Step 34. Ask the administrator of the remote host to give the appropriate permissions to the exported **/exports/readwrite** directory so that you can write. Once the appropriate access is granted, try again to create a file.

Q21. What are the file's owner / group? _____

Q22. What are the file's numeric UID/GID? (Hint: **ls -ln**) _____

Step 35. Look in the **/etc/passwd** file on your system

Q23. Is there an entry that corresponds to the owner of the file? _____

Step 36. If there was no entry in **/etc/passwd** with UID of 65534 (-2 in an unsigned 16-bit number), create one with the user name **nfsnobody**. The UID and GID should both be 65534, and the login shell should be **/sbin/nologin**. Now re-run the **ls -l** from above and you should find that the **nfsnobody** owns the files. This is the sole purpose of this pseudo-user.

Step 37. Un-mount all NFS mounted filesystems. Use the **mount** command to check that they are all un-mounted.

Configuring the automounter

Setup your system to use the automounting to automatically mount a referenced filesystem

Step 38. If the **automount** daemon is not running, start it with:

```
# /etc/init.d/autofs start
```

Q24. How do you know if the automounter is already running? _____

Q25. How would you setup the automounter to automatically run in multi-user mode? _____

Step 39. Edit the **/etc/auto.master** file and add an entry for an indirect map named **net**, with timeout options of 30 seconds.

Step 40. Create the indirect map file **/etc/auto.net** that contains an entry that maps your partner's **/exports/readwrite** directory into your **/net**. Automounting will occur under **/net**, under the directory name you list as the *key* (the first field in the file). Set the *key* to be the hostname of your partner's machine. Set the options so that mount is performed interruptible, hard, and read-write.

Step 41. The **automount** daemon does not know you have added a new master entry (a new *map* or *mountpoint*) - you need to tell it with the **autofs** script. Use the command below to cause the **autofs** script to force the **automount** daemon to reload the file **/etc/auto.master**.

```
# /etc/init.d/autofs reload
```

Step 42. Use the **status** parameter to the **autofs** command to see the status of configured and active automounted mount points.

```
# /etc/init.d/autofs status
```

Step 43. Run the command below exactly as shown.

```
# ls /net
```

Q26. What mounted filesystems do you see inside **/net**? _____

Step 44. Although the **/net** directory appears empty, the **autofs** filesystem will cause automounting to occur upon referencing anything beneath the mount point. Change directories into the not-yet-mounted directory named **/net/key** where *key* is the name you used above. If you receive an error about a non-existent directory, then the automount failed. Check the end of the **/var/log/messages** log file to see if there are any messages or errors from the automounter (by default, the system is configured to output automounter messages via **syslog** to this file). Resolve any problems before moving on.

Q27. Again do an **ls /net**. What do you see now? _____

Step 45. Now, leave the automounted directory by changing directories to **/net**. Wait 30 seconds, and do another **ls /net** (do not do an **ls** of the mounted directory, or that access will reset the auto-unmount timer). You should find that the directory disappears.

Step 46. Examine the tail of **/var/log/messages**. Notice the automounter related messages.

Step 47. Modify the entry in the **auto.net** indirect map and use wildcards this time so that you can automount the **/exports/remote-host** directory of any of the NFS servers on your network. You should be able to access any other host's **/exports/remote-host** directories by simply changing directories into **/net/remote-host**.

Step 48. Attempt to access several **/net/remote-host** directories.

Q28. What happens when you try to access your own host the same way? _____