

Configuring and Managing the Network File System

▲ Chapter Syllabus

- 33.1** NFS Concepts
- 33.2** Configuring an NFS Server
- 33.3** Configuring an NFS Client
- 33.4** NFS Daemons
- 33.5** Troubleshooting

The Network File System (NFS) is used to share files and directories over a network. It is an RPC-based system and works in the client-server model. The NFS server makes some of its file systems shareable with other systems on a network. The NFS client mounts shared file systems from the server to its local mount points. Users of the client machines can use NFS-mounted file systems just like the local file systems. NFS support is available on many platforms including personal computers. On UNIX systems, this has become a de facto standard and provides a transparent way for sharing files. When used in combination with NIS, it facilitates users' logging in from any machine on the network and finding the same environment, home directories, and mailboxes.

NFS is used to save disk space in a number of ways. All files that are static in nature can be shared on a network. These files include application directories under `/opt` and operating system files under the `/usr` directory. The files that are

640 Chapter 33 • Configuring and Managing the Network File System

shared by individual users are users' home directories and mailboxes. The directories that contain configuration files or log files are usually not shared.

NFS version 2 is supported on HP-UX versions 10.20 and earlier. Version 3 is supported on HP-UX 10.30 and later. The new version provides some extra benefits, like larger file sizes and improved performance. NFS version 3 is used by default on HP-UX 10.30 and later.

This chapter begins with an introduction to NFS, where NFS concepts and terms are presented. After that you will move to the configuration of an NFS server. Here you will learn how to start NFS services and how to export file systems to be mounted remotely by NFS clients. In the NFS client configuration process, you will start NFS client services and mount file systems exported by a particular NFS server. In this process, the `/etc/fstab` file is updated. After that a list of NFS daemons with their specific responsibilities in the NFS system is presented. In the last part of the chapter, you will learn some utilities and commands that can be used to monitor and configure NFS activity. Using these utilities, you will be able to see what file systems are exported and what clients are accessing them.

33.1 NFS Concepts

When an NFS server makes one or more of its files and directories shareable, this process is called *exporting*. The exported files or directories are placed in the `/etc/exports` file. Only the exported files and directories can be accessed by an NFS client. The rules that govern communication between a client and server are as follows.

- A system can act as an NFS client and server at the same time.
- A server can export a file, a directory, or a complete file system.
- A server can export only local file systems. File systems mounted from other servers can't be re-exported.
- If a directory is exported, all of the directory tree under this directory are automatically exported.
- A client can mount the exported directories or a subdirectory in the directory tree under the exported directory.

- User and group IDs must match on the client and server machines for proper permissions.

Remote Mount Process

The process of mounting a remote file system is complete when the following steps are completed on the NFS server and client.

- The NFS server processes are started on the server machine. This is done by running the `/sbin/init.d/nfs.core` and `/sbin/init.d/nfs.server` scripts at boot time.
- The shared directories are located in the `/etc/exports` file and are exported using the `exportfs` command. This command is executed at boot time by the `nfs.server` script. However, if you make any change to the `/etc/exports` file, you must use this command to re-export shared directories.
- NFS client processes are running on the client machine. These are started with the help of the `/sbin/init.d/nfs.core` and `/sbin/init.d/nfs.client` scripts. Both of these scripts are invoked at run level 2.
- The remote file system names and mount points are present in the `/etc/fstab` file.

Remote file systems can also be mounted at any time using the `mount` command. However, to have a remote file system mount automatically each time you reboot the system, it must be present in the `/etc/fstab` file.

NFS and RPC

Remote Procedure Calls (RPCs) allow procedures on the server machine to be called by programs on a client machine. All NFS communication between client and server takes place through RPCs. Since RPC (and NFS) is supported on a number of operating systems and platforms, a common data format must be used between client and server for passing data to RPCs. RPCs are made through program numbers. These program numbers are listed in the `/etc/rpc` file, which is displayed here.

```
##
# pragma VERSIONID "@(#)rpc:    11R2-4"
#      file of rpc program name to number mappings
##
rpcbind      100000  portmap sunrpc rpcbind
rstatd       100001  rstat rup perfmeter
rusersd      100002  rusers
```

642 Chapter 33 • Configuring and Managing the Network File System

```
nfs          100003  nfsprog
ypserv       100004  ypprog
mountd       100005  mount showmount
ypbind       100007
walld        100008  rwall shutdown
yppasswd     100009  yppasswd
etherstatd   100010  etherstat
rquotad      100011  rquotaprogram quota rquota
sprayd       100012  spray
selection_svc 100015  selnsvc
#
pcnfsd       150001  pcnfs
#
# NEW SERVICES ADDED AT 6.5
#
rex          100017  rex
llockmgr     100020
nlockmgr     100021
status       100024
#
# SUN SUPPORTS THE FOLLOWING THAT HP DOES NOT @ release 6.5
#
3270_mapper  100013
rje_mapper   100014
database_svc 100016
alis         100018
sched        100019
x25.inr      100022
statmon      100023
bootparam    100026
ypupdated    100028  ypupdate
keyserver    100029  keyserver
tfsd         100037
nsed         100038
nsemntd      100039
ypxfrd       100069
nisd         100300  rpc.nisd
nispasswd    100303  rpc.nispasswd
nis_cachemgr 100301
nisd_resolv  100302  rpc.nisd_resolv
automountd   100099
ttldbserver  100083
cmsd         100068  dtcalendar
#
```

Each line in this file starts with the RPC server name followed by a program number. Aliases can be listed after a program number. A line that starts with the # character is a comment.

The portmap and rpcbind Daemons

Contact between any RPC client and server starts with the help of the `portmap` or `rpcbind` daemon. Up to version 10.20, the `portmap` daemon was used, which was replaced by `rpcbind` in the later versions. The process of establishing a connection between client and server is carried out through the following steps. When an RPC server starts, it registers its program number with `portmap` and the TCP or UDP ports to which it is listening.

1. The client contacts `portmap` using port number 111 on the server machine requesting the port number used by the RPC server. The client sends the RPC program number of the server program as defined in the `/etc/rpc` file.
2. The `portmap` returns the port number used by a server using that program number.
3. The client connects to the server at the port number returned by `rpcbind`.

The `rpcbind` daemon facilitates dynamic binding of program numbers to ports.

NFS Versions

Starting with HP-UX version 10.30, NFS version 3 has been used by default. Advantages of version 3 over NFS version 2 are:

- NFS version 3 supports 64-bit file offset, whereas version 2 is limited to 32 bits. A file offset of 32 bits can address a maximum file size of 2 GBytes. NFS version 3 supports a maximum file size of 128 GBytes.
- Version 3 supports asynchronous I/O using kernel buffer cache management that provides data safety.
- File handle size up to 64 bits can be used.
- Cache management is improved.

A detailed discussion of the features supported by different NFS versions is beyond the scope of this book.

644 Chapter 33 • Configuring and Managing the Network File System

33.2 Configuring an NFS Server

The networking subsystem and NFS subsystem must be present in the kernel configuration to use NFS. The configuration process of an NFS server consists of multiple steps which will be discussed shortly. You may want to set up one or more NFS servers depending on the requirements and nature of the use of shared files and directories. For powerful NFS servers, the ratio of clients per server may be increased compared with a slow server. Similarly, you may consider changing this ratio depending on the type of applications used. For applications that are disk intensive, a lower ratio may be more efficient. Similarly, the network also plays an important role in the speed of data transfer. For high-volume data transfer, you may plan to have a high-capacity network or segment your network to divide network traffic.

After the planning process, you can follow the following steps to configure an NFS server. For proper access permissions to files shared through NFS, keep the UID and GID of all users the same on the NFS server and client machines. This can be done by copying the `/etc/passwd` and `/etc/group` files from the NFS server onto all client machines. If you are using NIS, all NFS server and client machines should be NIS clients in the same NIS domain.

Creating the /etc/exports File

This file consists of a list of file systems that are made available or *exported* to NFS clients. The following major rules apply to this file.

- Each line in the file starts with a file or directory name followed by a set of options related to that file or directory. See Table 33-1 for a list of options.
- Multiple options can be listed by using a comma to separate these options.
- The exported files and directories may have restricted access permissions. For example, some directories may be exported only to particular NFS clients. Similarly, if you don't want any modification to files in some directories, like manual pages, you can export these as read-only.
- All subdirectories are automatically exported when you export a parent directory.
- You can export an individual file or a directory.
- A file system mounted from another NFS server can't be exported.

- You should not export the root file system to all clients. This makes every file and directory visible to all NFS clients.
- If a file system is exported with the `root` option, clients will have `root` user access on that file system. This should be done very carefully. The `root` permission is not granted by default.

A sample `/etc/exports` file is shown next.

```
/home          -access=myhp:hp0
/usr/share/man -ro
/opt/apps      -ro
```

The first line grants access to the `/home` directory to users on hosts `myhp` and `hp0`. The second and third lines grant read-only access to the `/usr/share/man` and `/opt/apps` directories to all hosts.

The options used in this file are listed in Table 33-1.

Table 33-1 Options Used in the `/etc/exports` File

Option	Description
<code>-ro</code>	Used to grant read-only access to clients.
<code>-rw</code>	Used to grant read-write access to clients.
<code>-access=client1[:client2]</code>	Used to grant access to one or more clients only. The list of clients is separated using a colon.
<code>-root=client1[:client2]</code>	Used to grant root access to one or more clients. The list of clients is separated using a colon.
<code>-anon=uid</code>	If a request comes from an unknown user, he or she is granted access using the UID specified here. Usually a UID of <code>-2</code> is used that corresponds to a user nobody with minimum access rights.
<code>-anon=65534</code>	Access to unknown users is not granted.

You can also use SAM to export file systems using the `Network File System` button in the `Networking and Communications` area of SAM.

A file system does not become available automatically for clients until it is exported using the `exportfs` command. File systems in `/etc/exports` are exported at boot time with the help of the `exportfs` command. If you want to export these manually without rebooting after making a change, use the `exportfs -a` command. This command reads the `/etc/exports` file and copies these entries to the `/etc/xtab` file. Use the following command to export all file systems in the `/etc/exports` file.

646 Chapter 33 • Configuring and Managing the Network File System

```
exportfs -a
```

To unexport all exported file systems, use the following command.

```
exportfs -au
```

To unexport a particular file system, e.g., `/usr/share/man`, use this command.

```
exportfs -u /usr/share/man
```

Starting the Server Process at Boot Time

There are three startup scripts for the NFS system in the `/sbin/init.d` directory. The scripts are listed next.

1. The `nfs.core` script is executed at system run level 2 and is used for both NFS client and server machines. It starts `portmap` or `rpcbind` depending on the HP-UX version. The `rpcbind` daemon is used in HP-UX 10.30 and above, while the earlier versions use `portmap`.
2. The `nfs.client` script is executed at system run level 2 on an NFS client.
3. The `nfs.server` script is executed at system run level 3 on an NFS server. Among other tasks, it executes the `exportfs` command mentioned earlier.

These scripts use configuration parameters present in the `/etc/rc.config.d/nfsconf` file at startup time. The following values should be present for a system acting as an NFS server.

```
NFS_CLIENT=0
NFS_SERVER=1
NUM_NFSD=4
NUM_NFSIOD=4
PCNFS_SERVER=0
START_MOUNTD=1
```

Starting an NFS Server and Exporting Directories Manually

If you want to start an NFS server without rebooting the system, follow these steps.

1. Edit the `/etc/exports` file and add entries for exported file systems.
2. Use the `exportfs -a` command to export these file systems.
3. Edit the `/etc/rc.config.d/nfsconf` file and set proper values to different variables as mentioned earlier.

4. Execute the `/sbin/init.d/nfs.core start` command.
5. Execute the `/sbin/init.d/nfs.server start` command.

Viewing Exported and Mounted File Systems

Exported file systems can be listed using the `exportfs` command without any arguments. To list file systems remotely mounted by NFS clients, you can use the `showmount` command. The command can also be used to list exported file systems on a particular NFS server using the `-e` command-line option. If a client name is specified on the command line, it only lists file systems mounted by that client. The following command shows a list of hosts who have recently mounted local file systems.

```
# showmount
myhp
hp0
#
```

The following command lists file systems mounted by remote hosts in the `host:filesystem` format.

```
# showmount -a
hp0:/home
myhp:/usr/share/bin
#
```

Entries of all remotely mounted file systems and remote NFS clients are present in the `/etc/rmtab` file. It is a text file and can be viewed using the `cat` command.

33.3 Configuring an NFS Client

The NFS and LAN/9000 subsystems must be configured in the HP-UX kernel to configure a system as an NFS client. During the configuration process, you have to start some daemons and edit the `/etc/fstab` file to mount remote file systems at the system boot time. The client configuration process may be completed with the following steps.

1. Edit the `/etc/rc.config.d/nfsconf` file to enable an NFS client to be started at boot time.
2. Edit the `/etc/fstab` file to mount remote file systems at system boot time.

648 Chapter 33 • Configuring and Managing the Network File System

3. Create mount points for the remote file systems.
4. Ensure that the name of the NFS server is present in the `/etc/hosts` file if NIS or DNS are not being used.
5. Synchronize client and server clocks to ensure that time stamps are correct on modified files.
6. Either reboot the system or start the NFS client and mount remote file systems manually.
7. Verify mounted file systems using the `mount` command.

Let's discuss some of these steps in more detail.

Starting an NFS Client at Boot Time

The NFS client script `/sbin/init.d/nfs.client` is started at run level 2 during the boot process. Before this, `/sbin/init.d/nfs.core` scripts get executed. These scripts get configuration parameters from the `/etc/rc.config.d/nfsconf` file. The following line should be present in this file to start a system as an NFS client.

```
NFS_CLIENT=1
NFS_SERVER=0
NUM_NFSD=0
NUM_NFSIOD=4
PCNFS_SERVER=0
START_MOUNTD=0
```

These configuration parameters are used to start NFS daemons, which will be presented in the next section of this chapter.

Creating Mount Points and Configuring /etc/fstab

To mount any local or remote file system, you need a mount point. The mount point is a directory that is used as a reference to access a mounted file system. Create separate mount points for all remote file systems. Entries of all mount points and related file systems should be present in the `/etc/fstab` file so that remote file systems are mounted automatically at boot time. Sample NFS mount file system entries in the `/etc/fstab` file are:

```
myhp:/home          /home03            nfs   rw   0   0
myhp:/usr/share/man /usr/share/man     nfs   ro   0   0
```

The syntax of these entries is the same as you studied in Chapter 18 except for the following two changes.

1. The first field in each line is changed from a file system to a combination of host name used for the NFS server followed by a colon followed by the pathname of the remote file system that exists on the NFS server.
2. The file system type is `nfs` for remotely mounted file systems.

We have used a read-write option in the first line of the `/etc/fstab` file above and a read-only option for the second line. Table 33-2 lists options that can be used with remote file systems.

Table 33-2 *Options Used for NFS File Systems in the `/etc/fstab` File*

Option	Description
<code>rw</code>	Grants read and write access to the file system
<code>ro</code>	Grants read-only access
<code>fg</code>	Retry the mount operation in the foreground
<code>suid</code>	SUID is allowed on this file system
<code>hard</code>	Retry mount request until the server responds
<code>intr</code>	Permits user interrupt during hard mount retry

Mounting Remote File Systems

The NFS client executes the `mount -aQF nfs` command during execution of the `nfs.client` script at startup. However, if you have recently configured an NFS client and don't want to reboot the system, use the following commands.

```
/sbin/init.d/nfs.core start
/sbin/init.d/nfs.client start
```

If the NIS client is already running and you have made a change in the `/etc/fstab` file, use the following command instead of the above two.

```
mount -aF nfs
```

Any file system that is not listed in the `/etc/fstab` file can be mounted using the `mount` command. The following command mounts a file system `/opt/apps` from an NFS server `myhp` on a local mount point `/apps`.

```
mount myhp:/opt/apps /apps
```

You can unmount an NFS file system in the usual way using the `umount` command. Entries of all mounted file systems are stored in the `/etc/mnttab` file.

650 Chapter 33 • Configuring and Managing the Network File System

Viewing Mounted File Systems

The `mount` command can be used to list all mounted file systems. To view mounted file systems, use this command without any command-line argument or option.

Viewing Exported File Systems by a Server

You can list file systems exported by a particular NFS server using the `showmount` command. The following command lists file systems exported by server `myhp`.

```
showmount -e myhp
```

Study Break

Study Break: Configuring an NFS Server and Client

A system can be configured as an NFS client and server simultaneously. An NFS server can also mount its own exported file systems. Let's configure a host `myhp` as both an NFS server and client. Create the `/etc/exports` file with a single line in it to export the `/home` file system. Edit the `/etc/rc.config.d/nfsconf` file and set variable values as discussed earlier. Now start the NFS client and server processes using scripts in the `/sbin/init.d` directory. These should be run in the following order.

```
nfs.core  
nfs.client  
nfs.server
```

Create a mount point to mount the exported directory (`/home`). Let's suppose you create directory `/kaka` as a mount point. Use the following command to mount the exported `/home` directory to this mount point.

```
mount myhp:/home /kaka
```

Use the `mount` command without any argument to verify that the directory is mounted.

33.4 NFS Daemons

NFS uses a number of daemon processes to provide reliable file system access over a network. A number of clients may be accessing shared files, and it is the responsibility of the NFS system to guarantee file system integrity. NFS is a stateless server, which means that it does not keep a record of the state of files and clients that are accessing them. To keep consistency of files, a file locking mechanism is used through NFS daemons to ensure that files are properly locked when accessed by a client. Table 33-3 shows a list of NFS daemons and their functions on the client and server.

Table 33-3 *NFS Daemons*

Name	NFS System	Description
portmap	Both client and server	RPC daemon used in HP-UX 10.20 and earlier versions.
rpcbind	Both client and server	RPC daemon used in HP-UX 10.30 and later versions.
rpc.statd	Both client and server	RPC daemon used with the <code>rpc.lockd</code> daemon to provide crash recovery functions.
rpc.lockd	Both client and server	Used for file locking on both NFS servers and clients. Both client and server <code>rpc.lockd</code> daemons interact with each other to provide file locking functions.
nfsd	Server	The NFS server daemon. A number of server daemon processes (by default, 4) are started at startup time. All of these processes register themselves with <code>portmap</code> or <code>rpcbind</code> . A heavily loaded NFS server may start more daemon processes.
rpc.pcnfsd	Server	Used with PCNFS. Not required if all clients are UNIX machines.
rpc.mountd	Server	Keeps a record of mounted and exported file systems. Replies to mount requests.
biod	Client	Used to handle buffer cache in asynchronous I/O on NFS clients.

652 Chapter 33 • Configuring and Managing the Network File System

33.5 Troubleshooting

In any network-related problem, first use the `ping` command to ensure that network is operational. Most NFS problems are related to either the configuration or NFS daemons. The `/etc/exports` file on an NFS server and the `/etc/fstab` file on a client are important to check for any configuration problem. You should also verify that all required daemons are running on the server and client. The order of starting NFS daemons also matters. For example, `rpcbind` or `portmap` should be started before starting the `nfsd` daemon. Similarly, the `inetd` daemon, which serves ARPA/Berkley and RPC services, must be started after NFS daemons have been started. HP-UX provides some commands for NFS troubleshooting. Some of these are discussed here.

The rpcinfo Command

This command verifies if a service is registered with `portmap` or `rpcbind` and is running. It contacts the `rpcbind` or `portmap` daemon and displays all registered services. A short listing of all the registered services can be displayed using the `-s` command-line switch as shown here.

```
# rpcinfo -s
program version(s) netid(s)          service  owner
100000  2,3,4    udp,tcp,ticlts,ticotsord,ticots  rpcbind  superuser
100024  1        tcp,udp                          status   superuser
100021  2,4,3,1  udp,tcp                          nlockmgr superuser
100020  1        tcp,udp                          llockmgr superuser
100068  5,4,3,2  udp                              cmsd     superuser
100083  1        tcp                              ttldbserver superuser
100005  3,1      tcp,udp                          mountd   superuser
100003  3,2      udp                              nfs      superuser
100069  1        tcp,udp                          ypxfrd   superuser
100028  1        udp,tcp                          ypupdated superuser
100029  2,1      ticots,ticotsord,ticlts          keyserv  superuser
100004  1,2      tcp,udp                          ypserv   superuser
100009  1        udp                              yppasswdd superuser
100007  1,2      udp,tcp                          ypbind   sys
#
```

Here you should find `rpcbind`, `mountd`, and `nfs` registered under the `service` column of the output. You can also find a list of RPC registered services on a remote host using the remote host name as the command-line argument. The following command lists all registered RPC services on host `hp0`.

```
rpcinfo hp0
```

To verify that a service is listening to your requests and responding, make a UDP or TCP call to a particular service. The following command makes a UDP call to the `nfs` service on host `myhp`.

```
# rpcinfo -u myhp nfs
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
#
```

The output shows that the host is listening to NFS requests and accepting version 2 and version 3 requests. Other options used with the `rpcinfo` command are shown in Table 33-4. All of these options can be used with a host name on the command line to find information about a particular host on the network.

Table 33-4 *Options Used with the `rpcinfo` Command*

Option	Description
-d	Deletes the registration of a particular program
-m	Displays statistics of RPC operation
-p	Probes <code>rpcbind</code> using version 2 of <code>rpcbind</code>
-s	Displays a short list
-t	Makes an RPC call using TCP
-u	Makes an RPC call using UDP

Using the `nfsstat` Command

The `nfsstat` command displays NFS activity on a host. By default it displays information about the client, server and RPC. Different command-line options can be used to display information about the NFS client or server only. The following is the output of the command used without any option.

```
# nfsstat

Server rpc:
Connection oriented:
N/A
Connectionless oriented:
calls          badcalls          nullrecv
55769          0                  0
```

654 Chapter 33 • Configuring and Managing the Network File System

```

badlen                xdrCALL                dupchecks
0                     0                     3397
dupreqs
1

Server nfs:
calls                badcalls
55769                0
Version 2: (0 calls)
null                getattr                setattr
0 0%                0 0%                0 0%
root                lookup                readlink
0 0%                0 0%                0 0%
read                wrCache                write
0 0%                0 0%                0 0%
create                remove                rename
0 0%                0 0%                0 0%
link                symlink                mkdir
0 0%                0 0%                0 0%
rmdir                readdir                statfs
0 0%                0 0%                0 0%
Version 3: (55769 calls)
null                getattr                setattr
1 0%                243 0%                2155 3%
lookup                access                readlink
48700 87%            1181 2%                0 0%
read                write                create
5 0%                883 1%                359 0%
mkdir                symlink                mknod
0 0%                0 0%                0 0%
remove                rmdir                rename
0 0%                0 0%                0 0%
link                readdir                readdir+
0 0%                0 0%                1389 2%
fsstat                fsinfo                pathconf
47 0%                1 0%                0 0%
commit
805 1%

Client rpc:
Connection oriented:
N/A
Connectionless oriented:
calls                badcalls                retrans
1529                0                     0
badxids                timeouts                waits

```


Troubleshooting **655**

```

0
newcreds
0
toobig
0
bufulocks
0

Client nfs:
calls
1529
cltoomany
0
Version 2: (503 calls)
null
0 0%
root
0 0%
read
0 0%
create
0 0%
link
0 0%
rmdir
0 0%
Version 3: (1026 calls)
null
0 0%
lookup
0 0%
read
0 0%
mkdir
0 0%
remove
0 0%
link
0 0%
fsstat
1006 98%
commit
0 0%
#

0
badverfs
0
nomem
0

0
timers
16
cantsend
0

badcalls
0

clgets
1529

getattr
3 0%
lookup
0 0%
wrcache
0 0%
remove
0 0%
symlink
0 0%
readdir
0 0%

setattr
0 0%
readlink
0 0%
write
0 0%
rename
0 0%
mkdir
0 0%
statfs
500 99%

setattr
0 0%
readlink
0 0%
create
0 0%
mknod
0 0%
rename
0 0%
readdir+
0 0%
pathconf
0 0%
fsinfo
2 0%

```

656 Chapter 33 • Configuring and Managing the Network File System

Fields like `badcalls`, `nullrecv`, `badlen`, and `timeout` show errors. Pay special attention to these fields. Options used with this command are listed in Table 33-5.

Table 33-5 *Options Used with the `nfsstat` Command*

Option	Description
-c	Displays client information
-s	Displays server information
-r	Displays information about RPC
-m	Displays information about all mounted file systems
-z	Can be used by the superuser only; displays NFS information and then reinitializes it

■ Chapter Summary

The Network File System (NFS) is used to share file systems over a network. The system works in the client/server model and is based on RPC. An NFS server exports its shared file systems that can be mounted remotely by NFS clients on local mount points. A system may act as an NFS server and client at the same time. The `portmap` RPC daemon is used on HP-UX version 10.20 and earlier, while `rpcbind` is used on later HP-UX versions. The names of exported file systems are listed in the `/etc/exports` file on an NFS server. These file systems are exported at boot time. File systems can also be exported using the `exportfs` command at any later stage. The `showmount` command lists all file systems that are remotely mounted by NFS clients. The `/etc/fstab` file is used on an NFS client to mount remote file systems from a server at boot time. The `mount` command can be used to mount remote file systems at any later stage. The same command is also used to list mounted file systems. Many daemon processes are used on an NFS client and server. These processes help to provide reliable file system access over a network.

The `rpcinfo` command is used for troubleshooting RPC-based services including NFS. It lists all registered RPC services on a local or remote host. The same command can be used to verify that a particular RPC service is operational and is responding to client requests. The `nfsstat` command is another useful tool for NFS troubleshooting.

▲ CHAPTER REVIEW QUESTIONS

1. How can you grant mount access to an exported file system to selected clients?
2. Which subsystems must be present in the kernel to use NFS?
3. Why is it necessary to synchronize time on all NFS clients and servers?

▲ TEST YOUR KNOWLEDGE

1. *Which NFS version is supported on HP-UX 11.00?*
 - A. version 2 only
 - B. version 3 only
 - C. version 4 only
 - D. both versions 2 and 3
2. *Which HP-UX command can be used on an NFS server to list remotely mounted file systems by NFS clients?*
 - A. the `exportfs` command
 - B. the `mount` command without any command-line option
 - C. the `mount` command with the `-a` command-line option
 - D. the `showmount` command
3. *How should the `rpcbind`, `nfsd`, and `inetd` daemons be started?*
 - A. first the `rpcbind` daemon, then `nfsd`, and `inetd` last
 - B. first the `inetd` daemon, then `rpcbind`, and `nfsd` last
 - C. first the `inetd` daemon, then `nfsd`, and `rpcbind` last
 - D. The order does not matter. These can be started in any order.
4. *Which RPC daemon is used on HP-UX 11.00?*
 - A. `rpcinfo`
 - B. `rpcbind`
 - C. `rpcd`
 - D. `portmap`

658 Chapter 33 • Configuring and Managing the Network File System

5. *What is not true about NFS?*
 - A. A system can act as an NFS server and client at the same time.
 - B. Individual files can be exported by an NFS server.
 - C. An NFS server can export file systems mounted from another NFS server.
 - D. A system can mount file systems exported by itself.

6. *What is the advantage of NFS version 3 over version 2?*
 - A. It provides better cache management.
 - B. It provides 64-bit offset for files.
 - C. File sizes up to 128 GBytes are supported.
 - D. All of the above.

7. *Which command is used to manually export file systems for remote mount?*
 - A. `exports`
 - B. `exportfs`
 - C. `showmount`
 - D. `init`

8. *Which command is used to list registered RPC services on a host?*
 - A. `rpcinfo`
 - B. `rpcbind`
 - C. `netstat`
 - D. `nfstat`