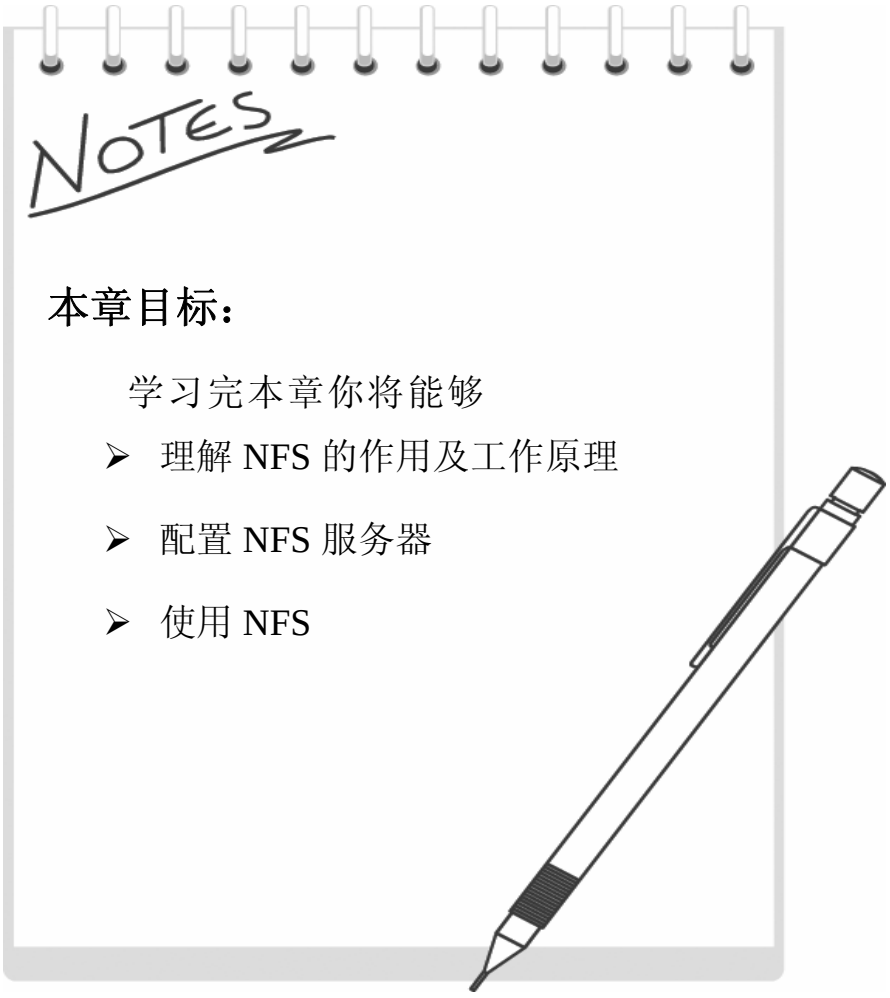
第五章　NFS 服务器的配置与使用

本章导读

NFS（Network Files system）是网络文件系统，它能够在不同的 Linux/UNIX 系统上使用，以达到文件的共享。本章将介绍有关网络文件系统 NFS 的知识。



1. NFS 简介

什么是 NFS 呢？它是 Network File system 的缩写，即网络文件系统。

NFS 是由 SUN 公司开发，并于 1984 年推出的一个 RPC 服务系统，它使我们能够达到文件的共享，在不同的系统间使用，所以它的通信协议设计与主机及操作系统无关。当用户想使用远程文件时只要用“mount”命令就可把远程文件系统挂接在自己的文件系统之下，使远程的文件与使用本地计算机上的文件一样。

例如在计算机 A 上，要把计算机 B 上的/usr/man 挂接到 A 的/usr/man 只需执行如下命令即可：

mount B:/usr/man /usr/man

用户不但可以 mount（挂接）目录，而且可以挂接一个文件。在挂接之后用户只能对文件做读取（或者写入）的操作，而不能在远程计算机上把此文件或目录移动或删除，但是如果挂接 /usr/man 后，则不能再挂接 /usr/man 底下的目录，否则会发生错误。

NFS 就是一种促使 servers（服务器）上的文件能被其他的计算机挂接而达到资源共享的网络文件系统，使用这些文件的计算机就可称为 Client（客户机），一个客户机可以从服务器上挂接一个文件或者一个层次的目录。然而，事实上任何一台计算机都可以是 NFS 服务器或 NFS 客户机，甚至同时为 NFS 服务器和 NFS 客户机。

NFS 服务器所共享出来的文件或目录都记录在/etc/exports 文件中，当启动 NFS 服务器时，脚本/etc/rc.d/rc 会自动启动 exportfs 程序，搜索/etc/exports 这一个文件是否存在，并且赋予正确的权限给所有共享出去的文件或目录。

但需要注意的是，只有服务器所共享出去的文件或目录，NFS 客户机才能够挂接。同样，当启动客户机时，系统会自动去挂接所有服务器共享的目录或文件，而挂接到的所有路径都会记录在/etc/fstab 下。

当客户机挂接一个目录或文件时，并不是说复制服务器上的这一个目录或文件到本地的计算机上，而是在使用时从服务器上读取文件到本地的内存中，因此，可以用 cd 进入这一个挂接到的目录，就如同进入本地的目录一样

2. NFS 对软件的需求

NFS 目前已经发展到了第 4 个版本（NFSv4），但是第 4 个版本目前还没有广泛

应用。在 Linux 系统中，要使用 NFS，首先必须要内核支持。目前的发行版本缺省内核都支持 NFS。但是如果要作下列事情之一，Linux 的内核版本要 2.2.18 以上：

- 在 Linux 系统和其它 Unix 系统之间混合使用 NFS。
- 使用 NFS 的安全文件锁
- 使用 NFSv3

除了需要内核支持之外，还需要 nfs-utils 软件，如果使用 NFSv3，nfs-utils 的版本号要在 0.1.6 以上，RedHat Enterprise Linux 带的 nfs-utils 的版本是 nfs-utils-1.0.6，并且用来加载 NFS 文件系统的 mount 命令的版本要 2.10m 以上：

```
[root@koorka ~]# rpm -q nfs-utils
nfs-utils-1.0.6-65.EL4
[root@koorka ~]# mount --version
mount: mount-2.12a
```

3. 配置 NFS 服务器

在准备好软件之后，接下来就可以配置 NFS 服务器了。 配置 NFS 服务器的一般步骤：

（1）确定计算机为 NFS 文件系统的服务器。

（2）对服务器上的硬盘进行分区，确定哪一些分区是要用来作为客户机所共享的文件系统。

（3）确定每一台客户机的访问参数（即读写的权限）。

（4）创建/etc/exports 文件（一般系统都有一个缺省的 exports 文件，可以直接修改；如果没有，自己创建一个）

（5）重新启动 NFS 服务器或者用命令 exportfs -a 输出所有的目录，并且用 nfsd &启动 nfsd 守护进程。

3.1 配置文件/etc/exports

要配置 NFS 服务器，首先就是编辑/etc/exports 文件。在该文件中，每一行代表一个共享目录，并且描述了该目录如何被共享。下面是一个典型的 exports 文件：

```
/projects       proj*.local.domain(rw)
/usr              *.local.domain(ro) @trusted(rw)
/home/joe       pc001(rw,all_squash,anonuid=150,anongid=100)
/pub             (ro,insecure,all_squash)
```

上面的配置说明在这台服务器上共享了 4 个目录，其中的参数下面会进行详细说明。从这个配置例子可以看出，exports 文件的格式为条目格式为：

```
共享目录              允许访问的主机（选项）
```

其中，选项只对括号前的主机生效，主机的格式为可以是：

（1）单台主机。例如：pc001。

（2）使用了通配符的主机名。例如：proj*.local.domain，local.domain 域的所有以 proj 开头的主机

（3）IP 地址。例如：192.168.100.99。

（4）地址段。例如：192.168.100.0/255.255.255.0

常用的选项有，默认选项表示如果不使用与之相对的选项，则使用该选项：

ro 该主机有只读的权限

rw 该主机对该共享目录有可读可写的权限

root_squash 客户机用 root 用户访问该共享文件夹时，将 root 用户映射成 nobody 用户。（默认选项）

no_root_squash 与 root_squash 相对。客户机用 root 用户访问该共享文件夹时，不映射 root 用户。

no_all_squash 客户机上的任何用户访问该共享目录时都映射成 nobody 用户。（默认选项）

anonuid 将客户机上的用户映射成指定的 uid 的用户（没有该选项则为 nobody 用户）。

anongid 将客户机上的组用户映射成指定的 gid 的用户组。

sync 所有数据在请求时写入共享（默认选项）

async 与 sync 相对。NFS 在写入数据前可以响应请求

secure NFS 通过 1024 以下的安全 TCP/IP 端口发送

insecure 与 secure 相对。NFS 通过 1024 以上的端口发送 （默认选项）

wdelay 如果多个用户要写入 NFS 目录，则归组写入（默认）

no_wdelay 如果多个用户要写入 NFS 目录，则立即写入，当使用 async 时，无需此设置。

hide 在 NFS 共享目录中不共享其子目录

no_hide 共享 NFS 目录的子目录

subtree_check 如果共享/usr/bin 之类的子目录时，强制 NFS 检查父目录的权限（默认）

no_subtree_check 和上面相对，不检查父目录权限

下面列举几个实例进行说明：

```
/                    zhang(rw)    wang(rw,no_root_squash)
```

该命令行表示共享服务器上的 / 目录，只有 zhang 和 wang 两台主机可以访问，并且两台主机对该共享目录都有可读可写的权限；zhang 主机在用 root 身份访问时，将客户机的 root 用户映射成服务器上的 nobody 用户（root_squash 参数，该参数为缺省参数），相当于在服务器上使用 nobody 用户访问该目录；wang 主机在用 root

用户访问该共享目录时，不映射 root 用户（no_root_squash 参数），即相当于在服务器上用 root 身份访问该目录。

```
/projects        proj*.local.domain(rw)
```

该命令行表示共享/projects 目录，local.domain 域中所有以 proj 开头的主机都可以访问该目录，并且都有读写的权限，客户机上的任何用户在访问时都映射成 nobody 用户（all_squash 参数，该参数为缺省参数）。这里需要特别说明的是，如果客户机要在该共享目录上保存文件，则服务器上的 nobody 用户对/projects 目录必须要有写的权限。

```
/home/joe                                    192.168.100.0/255.255.255.0
(rw,all_squash,anonuid=150,anongid=100) 192.168.200.0/255.255.255.0(ro)
```

该命令行表示共享/home/joe 目录，192.168.100.0/24 网段的所有主机都可以访问该目录，它们对该目录有读写的权限，并且所有的用户在访问时都映射成服务器上的 uid 为 150、gid 为 100 的用户；192.168.200.0/24 网段的所有主机对该目录有只读访问权限，并且在访问时所有的用户都影射成 nobody 用户。

3.2 启动服务

配置好服务器之后，要能够使客户端能够使用 NFS，必须要先启动服务。NFS 需要以下几个服务的支持：

（1）portmap ：当客户端请求 NFS 服务时，首先由该服务响应，然后由它去寻找其他的相关 NFS 服务。在 RedHat Linux 中，如果使用 RPM 包安装的 portmap 软件，执行/etc/init.d/portmap start 即可。启动后，应该看到服务器监听了 tcp 和 udp 的 111 端口。（使用 netstat –ln 命令查看），进程中多了 portmap 进程。

```
[root@koorka ~]# netstat –ln
[root@koorka ~]# ps ax|grep portmap
```

（2）nfs 服务：nfs 服务由 5 个后台进程组成，分别是 rpc.nfsd、rpc.lockd、rpc.statd、rpc.mountd、rpc.rquotad。rpc.nfsd 负责主要的工作；rpc.lockd 和 rpc.statd 负责抓取文件锁；rpc.mountd 负责初始化客户端的 mount 请求；rpc.rquotad 负责对客户文件的磁盘配额限制。这些后台程序是 nfs-utils 的一部，如果是使用的 RPM 包，它们存放在/usr/sbin 目录下。大多数的发行版本都会带有 NFS 服务的启动脚本。在 Redhat Linux 中，要启动 NFS 服务，执行/etc/init.d/nfs start 即可。

（3）确认 NFS 是否已经启动。

可以使用 rpcinfo 命令来确认，如果 NFS 服务正常运行，应该有下面的输出：

```
[root@koorka ~]# rpcinfo -p
```

```
   程序  版本  协议     端口
    100000     2    tcp    111   portmapper
    100000     2    udp    111   portmapper
    100003     2    udp   2049   nfs
    100003     3    udp   2049   nfs
    100003     4    udp   2049   nfs
    100003     2    tcp   2049   nfs
    100003     3    tcp   2049   nfs
    100003     4    tcp   2049   nfs
    100021     1    udp  32777   nlockmgr
    100021     3    udp  32777   nlockmgr
    100021     4    udp  32777   nlockmgr
    100021     1    tcp  32781   nlockmgr
    100021     3    tcp  32781   nlockmgr
    100021     4    tcp  32781   nlockmgr
    100011     1    udp    655   rquotad
    100011     2    udp    655   rquotad
    100011     1    tcp    658   rquotad
    100011     2    tcp    658   rquotad
    100005     1    udp    661   mountd
    100005     1    tcp    664   mountd
    100005     2    udp    661   mountd
    100005     2    tcp    664   mountd
    100005     3    udp    661   mountd
    100005     3    tcp    664   mountd
```

4. 配置 NFS 客户端

要在客户端使用 NFS，首先也要先启动 portmap 服务。

设置 NFS 客户机的操作步骤：

（1）编辑好/etc/fstab 这一个文件，确定要挂接的路径都在 fstab 中。

（2）依照 fstab 所设置的内容，在客户机上设置好挂接点（mount point）。（mount_points 就是用 mkdir 设置 exports 所输出的路径）

（3）确定所要挂接的路径，都会出现在/etc/exports 文件中

（4）可以执行 mount 命令连结 server 上的共享目录（mount-a）

如果只是临时使用，可以直接用 mount 命令：

mount servername:共享目录 本地目录，例如：

```
mount 192.168.100.1:/share    /mnt
```

该命令将 192.168.100.1 上的/share 目录挂接到本地的/mnt 目录（当然，服务器端必须要先设置共享该目录）。

/etc/fstab 文件的例子：

```
192.168.100.1:/home/joe        /mnt       nfs    rw     0    0
```

mount 的语法：

```
mount    –t    type[-rv]    -o[option]    server:pathname    mount_point
```

mount 命令的说明：

mount -a 把/etc/fstab 中所列的路径全部挂上。

mount   myhost:/usr/local   /usr/local/myshare

把 myhost 的/usr/local 目录挂接到 client 的 /usr/local/myshare 上并且是 readonly 上。

-t type：用户所要挂接的文件系统类型，如 nfs。

-r：所挂接的路径定为 readonly。

-v：挂接过程的每一个动作，都有消息传回到屏幕上。

umount 命令：

不使用该共享目录时，可以把该目录卸载。

umount   mount_point，例如：

umount   /mnt

umount-a 卸载所有已经挂接上的路径

5. 配置实例

公司需要在网络上共享一个文件夹，所有人都只有只读权限，且只有 192.168.1.0/24 子网的用户可以访问，试通过 NFS 共享该文件夹。NFS 服务器的 IP 地址是 192.168.1.1。

具体操作步骤如下：

（1）以 root 身份登录，在/目录下创建目录 share。

[root@koorka www /]#mkdir /mnt/share

（2）编辑/etc/exports 文件，在该文件中加入下面的命令行。

/mnt/share 192.168.1.0/24（ro,root_squash）

“/mnt/share”表示要共享的目录，192.168.1.0/24 表示允许访问的主机（这里是一个子网的主机），括号内的 ro 表示客户机上的用户对该共享目录只有只读权限，root_squash 表示当客户机上的 root 用户访问该共享目录时，映射该用户为匿名用户，即当客户机上的 root 用户访问该共享目录时相当于服务器上的 anonymous（nobody）用户。

（3）在 192.168.1.0/24 子网上的任意一台客户机上安装共享目录。

[root@www root]mount 192.168.1.1:/mnt/share /www

上面命令行的意思是：将 192.168.1.1 上的/mnt/share 目录作为一个分区挂接到本机的/www 目录下。

（4）测试权限。

进入/mnt 目录建立目录 test，此时会出现下面的提示

mkdir: cannot create directory `mydir': Permission denied

因为在步骤（2）中的命令行中加了参数 ro，任何人对该共享目录都只有只读权限。