Compartilhamento de Arquivos no Linux

Luiz Arthur

O compartilhamento de arquivos em uma rede de computadores é sem duvida um dos serviços mais importantes, pois é basicamente através dos arquivos que usuários e máquinas trocam informações pela rede.

Assim, a habilidade de compartilhar arquivos é um serviço fundamental de qualquer rede departamental, tornando indispensável para todos os sistema operacionais de rede o suporte ao compartilhamento de arquivos pela rede.

O Linux é um sistema perfeito para este serviço, porque fornece uma gama extensa de mecanismos de compartilhamento de arquivos, que integram desde os clientes Microsoft Windows, clientes Unix e outros clientes que não são compatíveis com qualquer um destes em uma rede única e coesa.

Desta forma, iremos abordar inicialmente como deve funcionar basicamente um servidor de arquivos em uma rede e posteriormente estudar como funcionar o NFS (Network File System), que um sistema de compartilhamento de arquivos via rede de sistemas UNIX.

O Linux possibilita o compartilhamento de arquivos de três formas:

Técnicas de mainframe – Permite aos clientes se registrarem no servidor e compartilhar arquivos diretamente pelo sistema de arquivo Linux. Este modelo funciona com qualquer sistema de cliente que pode emular um terminal Linux, como por exemplo: TELNET, SSH, RLOGIN, etc.

Técnica de rede Unix – Permite aos clientes compartilhar arquivos pela rede com o *Network File System* (sistema de arquivo de rede – NFS). NFS é o software de compartilhamento de arquivo mais popular em redes Unix.

Técnica de rede Microsoft – Permite aos clientes usar o protocolo *Server Message Block* – SMB, para compartilhar arquivos pela rede. SMB é o protocolo NetBIOS usado por sistemas LanManager e Windows NT/2000 da Microsoft para fornecer serviços de compartilhamento de arquivo a clientes Microsoft Windows.

Entendendo o NFS

O NFS, originalmente desenvolvido pela Sun Microsystems, permite compartilhar diretórios e arquivos através de uma rede.

Através do NFS os usuários e programas acessam arquivos localizados em sistemas remotos como se fossem arquivos locais.

O NFS é um sistema cliente/servidor. O cliente usa os diretórios remotos como se eles fizessem parte de seu sistema de arquivo local. O servidor torna os diretórios disponíveis para uso.

Anexar um diretório remoto ao sistema de arquivo local é chamado de **montar um diretório**. O ato de oferecer compartilhamento de diretório é chamado de **exportar um diretório**.

O NFS é um protocolo de chamada a procedimento remoto (**RPC** – Remote Procedure Calls) que é executado em cima dos protocolos UDP e IP. Uma chamada de procedimento remoto simplesmente é uma chamada de sistema que é processada por um servidor remoto.

Quando um programa fizer uma chamada de I/O (Entrada/Saída) para um arquivo NFS, a chamada é interceptada pelo sistema de arquivos NFS, e é enviada através da rede ao servidor remoto para processamento.

Números de Portas do NFS

Os daemons que processam as solicitações do NFS no servidor não tem números de porta UDP padrão. Ao invés disto, a eles são atribuídos dinamicamente um número de porta pelo **portmapper** do RPC. Em alguns sistemas, o programa portmapper é chamado de rpc.portmap ou simplesmente **portmap**.

Então, o portmap é um servidor que converte chamadas RPC em números de portas de rede. Assim, quando um servidor RPC é iniciado ele abre a porta 111, que é a padrão do portmappers, desta forma, quando um cliente quer utilizar algum serviço RPC ele contata o portmapper e descobre qual é a porta atribuida ao serviço desejado.

Para ver o número de portas que o portmapper atribuiu aos programas use o comando rpcinfo:

```
#rpcinfo -p
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100011    1   udp    918  rquotad
    100011    2   udp    918  rquotad
    100011    1   tcp    921  rquotad
    100011    2   tcp    921  rquotad
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100021    1   udp  32795  nlockmgr
    100021    3   udp  32795  nlockmgr
    100021    4   udp  32795  nlockmgr
    100005    1   udp    933  mountd
    100005    1   tcp    936  mountd
    100005    2   udp    933  mountd
    100005    2   tcp    936  mountd
    100005    3   udp    933  mountd
    100005    3   tcp    936  mountd
    100024    1   udp    939  status
    100024    1   tcp    942  status
```

O número de porta 111 é o próprio número de porta do portmapper, que é um número bem conhecido, atribuído no arquivo */etc/services*. Então, sistemas remotos podem contatar o *portmapper* porque está em uma porta padrão. A partir do *portmapper,* os sistemas remotos descobrem as portas usadas por outros serviços RCP.

Os outros serviços executados em conjunto com o NFS são:

status – O status monitora relatórios de quebras, e reinicia o gerenciamento de bloqueio do servidor NFS. Assim estes bloqueios podem ser reajustados corretamente se um cliente NFS reiniciar sem terminar de maneira correta a sua conexão NFS. rpc.statd monitora tráfego TCP e UDP.

rquotad – O servidor de cota remoto, rpc.rquotad, obriga que as cotas do sistema de arquivos, sejam válidas para sistemas de arquivos NFS montados. Cotas de sistema de arquivo controlam a quantidade de armazenamento em disco que um usuário individualmente pode consumir. Este daemon estende este recurso a usuários NFS.

mountd – O daemon mount processa as solicitações de montagem de sistema de arquivo do cliente. O rpc.mountd é o programa que confere se um sistema de arquivo está sendo exportado ou não, e se o cliente que faz a solicitação está autorizado ou não a montar o sistema de arquivo solicitado.

nfs – O programa nfsd manipula a interface em nível de usuário para o módulo de kernel NFS (`nfsd.o`). O arquivo NFS I/O é controlado no módulo de kernel.

nlockmrg – O gerenciador de bloqueio NFS manipula solicitações de bloqueio de arquivo de clientes. O bloqueio de arquivo é usado para evitar corrupção de arquivo quando for possível que múltiplas fontes possam tentar gravar um arquivo ao mesmo tempo. Arquivos somente leitura não requerem bloqueio de arquivo.

Configurando um servidor NFS

O arquivo *`/etc/exports`* é o arquivo de configuração do servidor NFS. Ele controla quais arquivos e diretórios são exportados (serão compartilhados), quais hosts pode acessá-los, e que tipo de acesso é permitido.

O formato geral de entradas no arquivo *`/etc/exports`* é:

`diretório  hosts(opções)`

Diretório: é o nome do caminho completo do diretório ou arquivo sendo exportado. **Atenção!** Se o diretório não for seguido por um host ou uma opção, todos os hosts tem acesso de leitura/escrita ao diretório.

Host: é o nome do cliente ao qual foi dado acesso ao diretório exportado. Valores de host válidos são os seguintes:

- Nomes de hosts individuais, tal como:
  - `meumicro.dominio.com.br`
  - `micro`
- Coringas de domínio como: `*.dominio.com.br`. Ou seja, todos do `dominio.com.br`

- Pares de endereços IP/máscara de endereço como `192.168.100.0/255.255.255.0` para hosts da rede 192.168.100.0, ou seja, todos os hosts que iniciam o IP 192.168.100.

- Grupos de rede com `@grupo`. Este tipo de permissão se da quando é utilizado um servidor NIS.

Opção: define o tipo de acesso que é dado. Se a opção for especificada sem um nome de host, o acesso é dado a todos os clientes. Caso contrário, o acesso só é dado ao host que está nomeado. As duas opções mais comuns são:

`ro` – Especifica que os clientes só podem ler do diretório. Escrever no diretório não é permitido.

`rw` – Dá acesso total de leitura e escrita ao diretório

Um exemplo de arquivo */etc/exports* é:

```
/usr  172.16.5.0/255.255.255.0(ro)
/home 10.0.0.0/255.0.0.0(rw)
/etc/hosts micro1(rw) microgw(ro)
/home/aula *.redes.com.br(rw)
```

Observação: Só utilize espaços depois do caminho do compartilhamento (ex, `/usr`), após isto não utilize mais espaços em branco nem deixe linhas em branco, isto pode causar alguns problemas, como lentidão durante a inicialização do servidor NFS.

Para se iniciar o servidor NFS no Slackware execute o comando:
#/etc/rc.d/rc.nfsd restart

Confira o status para ver se os daemons necessários estão sendo executados:

```
#ps -Cnfsd
  PID TTY          TIME CMD
 8802 ?        00:00:00 nfsd
 8805 ?        00:00:00 nfsd
 8806 ?        00:00:00 nfsd
 8807 ?        00:00:00 nfsd
 8808 ?        00:00:00 nfsd
 8809 ?        00:00:00 nfsd
 8810 ?        00:00:00 nfsd
 8811 ?        00:00:00 nfsd

#ps -Crpc.mountd
  PID TTY          TIME CMD
 8813 ?        00:00:00 rpc.mountd
```

Atenção! Talvez, seja necessário antes de ligar o `nfsd`, executar os aplicativos `rcp.portmap`, `rcp.lockd` e `rpc.statd`. Isto pode ser feito no Slackware com a execução do seguinte script:

```
# /etc/rc.d/rc.rpc restart
```

Mapeando IDs de usuário e IDs de grupo (outras opções)

Identificação de usuários (IDs) e de grupo (GID) são tão fundamentais ao NFS quanto são a qualquer outra parte do sistema de arquivo Linux (são estas por exemplo, que tornam o Linux menos vulnerável a vírus).

Mas diferente do UID e do GID que você atribui quando cria contas de usuários novos, você pode não tem qualquer controle sobre os UIDs e GIDs atribuídos pelos clientes de seus servidores NFS.

Assim, o NFS fornece várias ferramentas para lhe ajudar a lidar com os possíveis problemas que surgem por causa disto.

Um dos problemas mais óbvio é como um modelo de segurança de host confiado á lidar com a **conta root**. É muito improvável que você queira que as pessoas com acesso de root tenham o mesmo nível de acesso ao seu servidor.

Por padrão, o NFS evita isto com a configuração ***root_squash***, que mapeia solicitações que contém o UID root e o GID para a UID e GID **nobody**. Que é uma espécie de usuário visitante sem mais vantagens.

Assim, se alguém estiver se registrando para um cliente como root, a ele só são dadas permissões mundiais no servidor (do usuário nobody).

Mas se for necessário **dar permissões de root** é possível utilizar a opção **no_root_squash**, mas é preciso ter cuidado com esta opção.

Já a opção ***all_squash*** mapeia todo usuário de um sistema cliente ao usuário nobody.

Por exemplo, a entrada seguinte exporta o diretório /pub para todo cliente:

```
/pub (ro,all_squash)
```

Ele dá acesso de somente leitura e limita todos usuários a terem permissões do usuário nobody.

Também é possível mapear todos os usuários para um ID de usuário especifico.

```
/home/aula
micro1(all_squash,anonuid=1001,anongid=1001)
```

Aqui, o host `micro1` tem permissões de não root (`all_squash`) e todos os usuários são mapeados para o ID e GID `1001`.

Comando exportfs

Depois de definir os diretórios para exportar no arquivo */etc/exports*, é possível executar o comando `exportfs` para processar o arquivo */etc/exports* e construir o */var/lib/nfs/xtab*.

O arquivo */var/lib/nfs/xtab* contém informações sobre os diretórios atualmente exportados, e é o arquivo que o comando *mountd* lê ao processar solicitações de montagem de cliente.

Para processar todas as entradas no arquivo */etc/exports*, basta executar o seguinte comando:

```
# exportfs -a
```

Este comando constrói um arquivo *xtab* completamente novo, baseado no conteúdo do arquivo */etc/exports*.

Também é possível usar o comando *exportfs* para apenas atualizar um arquivo *xtab*, com o comando:

```
# exportfs -r
```

É possível também usar o comando *exportfs* para exportar um diretório que não está listado no arquivo */etc/exports*. Um exemplo seria o seguinte comando:

```
# exportfs micro1:/usr/local -o rw
```

Note que a opção *-o* é usada para especificar opções de exportação para o sistema de arquivo exportado.

Depois que o cliente completou seu trabalho com o sistema de arquivo temporariamente exportado, o diretório pode ser removido da lista de exportação com a opção *-u*.

```
# exportfs -u micro1:/usr/local
```

A opção *-u* pode ser combinada com a opção *-a* para fechar completamente todas as exportações sem terminar os daemons NFS:

```
# exportfs -ua
```

Depois que os daemons estiverem executando e o arquivo `exports` tiver sido processado, os clientes podem montar e usar os sistemas de arquivos oferecidos por seu servidor.

Configurando um cliente NFS

Para configurar um cliente NFS, é preciso saber o nome de host ou endereço IP do servidor NFS e os diretórios que ele exporta.

O comando ***showmount*** do Linux lista os diretórios que um servidor exporta e os clientes autorizados a montar estes diretórios. Por exemplo:

```
# showmount --export localhost
Export list for localhost:
/tmp        *
/usr/local 192.169.100.0/255.255.255.0
/home      10.0.0.1
```

O comando *showmount* lista então os diretórios NFS exportados por um determinado host (neste caso o `localhost`), e quais clientes podem montar os diretórios. Sendo então possível montar quaisquer dos diretórios oferecidos por `localhost` se você for um usuário em um dos clientes aprovados.

Comando mount para NFS

Antes de usar um diretório NFS, é necessário anexar os diretórios exportados pelo servidor ao sistema de arquivos locais. Isso é feito com o comando ***mount***.

O *mount* pode oferecer desde formas simples para montar o diretório exportado quanto formas mais difíceis e complexas, mas que dão mais suporte ao cliente.

No mais simples, *mount* identifica o sistema de arquivo remoto para acessar o diretório local pelo qual será acessado.

```
# mount servernfs:/tmp            /mnt/exports/tmp
# mount 192.168.100.254:/home     /mnt/exports/home
```

Um comando mount simples funciona sob muitas circunstâncias, mas quando necessário, opções podem ser acrescentadas à linha de comando *mount* com o argumento *-o*.

As opções do *mount* podem ser vistas com o comando: ***man mount***

Comando umount

O oposto do comando *mount* é o comando ***umount***, que é usado para remover um diretório montado do sistema de arquivo local.

Um sistema de arquivo pode ser desmontado usando o nome de sistema de arquivo remoto ou o diretório local.

```
# umount servernfs:/tmp
# umount /mnt/exports/tmp
```

Usando o arquivo /etc/fstab para montar diretórios NFS

Um comando *mount -a* faz o Linux montar todos os sistemas de arquivos listados em */etc/fstab*.

A tabela de sistemas de arquivos, */etc/fstab*, define os dispositivos, partições e sistemas de arquivos remotos (NFS, SMB, etc) que compõem o sistema de arquivo completo de um computador Linux. Cada linha neste no arquivo */etc/fstab* descreve uma parte individual do sistema de arquivo.

Um exemplo, de arquivo */etc/fstab* é:

```
/dev/hda3   swap         swap         defaults          0   0
/dev/hda5   /            reiserfs     defaults          1   1
/dev/hda1   /mnt/win     ntfs         ro                1   0
/dev/hda6   /mnt/hda6    vfat         defaults          1   0
/dev/cdrom  /mnt/cdrom   iso9660      noauto,user,ro    0   0
/dev/fd0    /mnt/floppy  auto         noauto,owner      0   0
proc        /proc         proc         defaults        0   0
/dev/hde1   /mnt/flash    msdos       noauto,user,ro    0   0
servernfs:/tmp  /mnt/exports nfs     rw                0   0
```

Desta forma, o NFS oferece subsídios para o compartilhamento de informações, através de sistemas de arquivos via rede. Possibilitando que um usuário remoto acesse tais arquivos como se estes arquivos estivessem disponíveis localmente para o usuário.

fim