

## USO DE NFS

O NFS foi desenvolvido para permitir o compartilhamento de diretórios em rede. Atualmente, encontra-se na versão 3. Essa versão é suportado por *kernel* com versão 2.4 ou superior. Além disso, já encontra-se definido o protocolo do NFS versão 4, mas nenhuma implementação consistente do mesmo.

Os serviços de NFS são implementados ou diretamente no *kernel* ou como módulos. Entretanto, é recomendável instalar o pacote *nfs-utils*, que fornece uma série de ferramentas extras para o NFS, aumentando a performance do servidor NFS de Linux tradicional. Além disso, possui o aplicativo *showmount*, que consulta o daemon de uma máquina remota por informação acerca do servidor NFS (Network File System) dessa máquina. Consulte a página de manual desse comando para mais detalhes.

O processo de uso de NFS é relativamente simples, sendo necessário efetuar configurações na máquina servidora para compartilhar os diretórios. Na máquina cliente, o processo é de montagem de partições, com o uso do *mount* e edição do */etc/fstab*.

### Configurando o Servidor NFS

Em geral, a única configuração a ser feita para compartilhar um diretório é a edição de uma linha no arquivo */etc/exports*. Uma entrada nesse arquivo possui a seguinte sintaxe:

```
diretório maq1(op11,op12) ... maqn(opn1,opn2)
```

Nesse caso, *diretório* é o diretório a ser compartilhado, *maq1* e *maqn* são as máquinas para as quais o diretório está sendo exportado e *opxy* é opção de compartilhamento. As opções de compartilhamento mais importantes são listadas na tabela abaixo. Outras opções podem ser verificadas na página de manual do arquivo */etc/exports* (*man export*).

Opção	Descrição
<i>ro</i>	O diretório é compartilhado apenas para leitura
<i>rw</i>	O diretório é compartilhado para leitura e escrita pela máquina cliente
<i>no_root_squash</i>	Por padrão, qualquer requisição feita pelo usuário <i>root</i> na máquina cliente é convertido para o usuário anônimo (geralmente <i>nobody</i> ou <i>nfsnobody</i> ) na máquina servidora, aumentando o nível de segurança do processo. Caso o usuário <i>root</i> da máquina cliente deva ter acesso irrestrito ao diretório compartilhado, então deve-se usar a opção <i>no_root_squash</i>
<i>all_squash</i>	Mapeia todos usuários e grupos para o usuário anônimo

Um exemplo de configuração do arquivo `/etc/exports` pode ser visualizado na Figura abaixo. Nessa figura, o leitor pode perceber que é possível exportar um diretório para uma rede ou para uma máquina. Além disso, também é possível utilizar máscaras de rede, bem como os caracteres coringas `*` (para representar um conjunto de caracteres) e `?` (para representar um único caracter).

```
/usr 192.168 (ro) madonna (rw) lab*.dominio.com.br (ro) 100.1.0.2 (ro)
/usr/local 192.168 (ro) madonna (rw,no_root_squash) lab*.domain.com (ro)
/home      192.168.0.0/255.255.255.0(rw) *.domain.com (ro)
```

No exemplo da Figura acima, tem-se que o diretório `/usr` encontra-se exportado para as máquinas na rede 192.168, para a máquina de nome `madonna`, para as máquinas do domínio `domain.com` cujos nomes iniciem com `lab` (ex.: `laboratorio`, `lab01`, etc.) e para a máquina 100.1.0.2. Nesse caso ainda, a máquina `madonna` possui poderes de leitura e escrita nesse diretório, ao contrário das outras máquinas. Ainda, no caso do diretório `/usr/local`, o superusuário da máquina `madonna` tem os mesmos poderes do superusuário da máquina que está exportando os arquivos.

Uma observação importante a ser feita sobre exportação de diretórios é que o NFS não exporta subdiretórios que estejam em outras partições. Assim, por exemplo, se `/usr` e `/usr/local` estiverem em partições distintas, eles necessitam ser exportados independentemente, como é o caso da Figura acima.

Configurado o arquivo `/etc/exports`, o próximo passo é carregar o servidor NFS. A bem da verdade, o NFS envolve três servidores: o `rpc.nfsd`, o `rpc.mountd` e o `exportfs`. Como o NFS usa RPC (*Remote Procedure Calls* – chamadas de procedimentos remotos), ele também precisa do `portmap`. Na maioria das distribuições, o processo de inicialização pode ser feito com a chamada

```
service nfs start
```

ou executando-se o comando

```
/etc/rc.d/init.d/nfs start
```

Caso o arquivo `/etc/exports` seja alterado durante a execução do servidor, o processo de sincronização do novo arquivo pode ser feito com o comando

```
exportfs -r
```

Veja a página de manual desse comando para mais detalhes.

## Configuração do Cliente NFS

Como comentado em (SICA; UCHÔA, 2003), o sistema de arquivos adotado no Linux baseia-se numa camada virtual (VFS – *Virtual FileSystem*), construída para suportar múltiplos sistemas de arquivos. Essa estrutura permitiu ao Linux um gerenciamento poderoso desse recurso. Entre os sistemas de arquivos inclui-se o NFS. Assim, supondo-se que uma estação cliente possua os devidos privilégios, ela utilizar uma partição /usr/local no servidor myserv, montando essa partição em um diretório local (por exemplo, /usr/myserv):

```
mount -t nfs myserv:/usr/local /usr/myserv
```

Em várias versões recentes do comando mount, o uso da opção -t nfs é desnecessária. Caso o uso desse ponto de montagem seja freqüente, basta adicionar uma linha no /etc/fstab, utilizando-se o nfs para o tipo de partição:

```
myserv:/usr/local /usr/myserv nfs defaults 0 0
```

Fonte:

Administração de Redes Linux / Fernando Cortez Sica, Joaquim Quinteiro  
Uchôa, Luiz Eduardo Simeone. - - Lavras: UFLA/FAEPE, 2003.

## Exercício - NFS

1. Estabeleça uma máquina para ser servidor. No diretório /home desta máquina crie alguns arquivos e diretórios com a finalidade de exporta-lo para a rede. Na máquina cliente monte esta partição em /mnt num diretório chamado home\_microxx.
2. Compartilhe o CDROM da máquina servidora com a máquina cliente.

## COMANDOS DIVERSOS DO *CONSOLE*

### Comando `exit`

Encerra a execução de um *shell*. Possui como único parâmetro um inteiro indicando o valor a ser retornado; se omitido, assume o valor de retorno do último comando executado. A Figura abaixo mostra exemplos de execução do `exit`. É utilizado principalmente em *Shell scripts*.

```
Retornando um valor 0 indicando execução bem-sucedida
# exit 0

Retornando um valor de erro
# exit -1
```

### Comando `clear`

O comando `clear` limpa a tela de um terminal. Ele verifica o terminal utilizado em uma variável de ambiente para saber como limpá-lo. Seu uso é ilustrado na Figura abaixo:

```
Limpando a tela do terminal
# clear
```

### Comando `date`

Mostra ou ajusta a data do sistema. A Figura 4.39 mostra exemplos de execução do comando, e a Tabela 4.27 lista seus principais parâmetros. As principais especificações de formato estão na apresentadas na Tabela 4.28. Uma observação a ser feita é que configuração de data e hora no sistema deve ser feita com extrema cautela. O motivo é que vários servidores fazem tarefas periódicas e podem ficar meio “malucos” com uma atualização desse tipo. Pode acontecer de alguns serviços precisarem ser reiniciados para voltarem ao comportamento normal. Sempre que possível, recomenda-se a alteração via BIOS.

Tabela 4.27: Parâmetros do Comando `date`

Parâmetro	Descrição
-d	Exibe a data passada em seguida em vez da data atual
-R	Mostra a descrição da data no formato proposto pela RFC 822 (CROCKER, 1982)
-s	Ajusta a data ou hora para o valor passado em seguida

Tabela 4.28: Especificação de Formatos de Saída do Comando date

Parâmetro	Descrição
date +%a	Nome do dia da semana (3 letras)
date +%A	Nome do dia da semana (completo)
date +%b	Nome do mês (3 letras)
date +%B	Nome do mês (completo)
date +%c	Mostra uma expressão descrevendo a data e hora atuais
date +%d	Dia do mês
date +%D	Data (mês/dia/ano)
date +%j	Dia corrente no ano (1-366)
date +%m	Mês (2 dígitos)
date +%U	Numero da semana no ano (0-53)
date +%y	Ano (2 dígitos)

```
Mostra o dia do mês (entre 1 e 356)
# date +%j

Mostra a data no formato (dd/mm/aa)
# date +%d/%m/%y

Mostra o texto correspondente à data passada
# date -d 8/10/1970
Seg Ago 10 00:00:00 BRT 1981

Alterando a hora do sistema para 21:00:00
(requer privilégios de superusuário)
# date -s 21:00:00

Alterando a data do sistema para 31/12/1999
(requer privilégios de superusuário)
# date -s 12/31/1999
```

Figura 4.39: Exemplos de Uso do date

### Comando who

Mostra informações sobre os usuários que estão usando o sistema. Seus parâmetros principais são mostrados na Tabela 4.29 e alguns exemplos de execução na Figura 4.40.

Tabela 4.29: Parâmetros do Comando `who`

Parâmetro	Descrição
-b	Hora do último <i>boot</i>
-u	Lista os usuários que estão usando o sistema
-p	Mostra os processos ativos
-r	Imprime o nível de execução ( <i>runlevel</i> )
-q	Nome e número de usuários usando o sistema

```
Mostra a hora do último boot do usuário atual:
```

```
# who -b
```

```
Mostra os usuários que estão usando o sistema
```

```
nesse momento:
```

```
# who -q
```

Figura 4.40: Exemplos de Uso do `who`

### Comando `cal`

Exibe o calendário de uma determinada época. Caso não seja passado nenhum mês ou ano, exibe o calendário do mês atual. O mês deve ser especificado com um valor de 1 a 12, e o ano com um valor de 1 a 9999. A Tabela 4.30 mostra alguns dos parâmetros que podem ser passados para o `cal`, enquanto a Figura 4.41 mostra alguns exemplos de sua utilização.

Tabela 4.30: Parâmetros do Comando `cal`

Parâmetro	Descrição
-1	Mostra apenas o mês solicitado (padrão)
-3	Mostra também o mês anterior e posterior ao solicitado
-s	Considera domingo como o primeiro dia da semana
-m	Considera segunda-feira como primeiro dia da semana
-j	Mostra os dias como contínuos no ano (de 1 a 366)
-y	Mostra o calendário do ano atual

```
Mostra o calendário do mês atual
# cal

Mostra o calendário do mês de agosto de 1970, juntamente com
julho e setembro do mesmo ano
# cal 8 1970 -3

Mostra o calendário do ano de 1500 em dias contínuos
# cal 1500 -j
```

**Figura 4.41:** Exemplos de Uso do cal

Fonte:

Linux Intermediário / Joaquim Quinteiro Uchôa, Luiz Eduardo Simeone,  
Matheus Garcia Barbosa Figueiredo, Samuel Pereira Dias. – Lavras:  
UFLA/FAEPE, 2003.