

# Automirror

## Um Sistema Paralelo para Navegação em Filesystems Automontados

Rosiane de Freitas Rodrigues

Cristina Célia de Melo Barros

Paulo Lício de Geus

Instituto de Ciência da Computação

Universidade Estadual de Campinas

CEP: 13081-970 Campinas-SP-Brasil

Tel: +55-192-39-8442/3115 FAX: +55-192-39-7470

e-mail: {rosiane|cris|paulo}@dcc.unicamp.br

### Abstract

This work presents a parallel system for automounted filesystem navigation in a UNIX environment, whose main idea is to report on all automountable files and directories by browsing through NIS and local maps available to a system. In others words, nothing is really mounted as a consequence of the tool's use, allowing for quick access and minimizing cross-mountings among NFS servers, as well as the side effects deriving from this interdependency.

### Resumo

Este trabalho apresenta um sistema paralelo para navegação de *filesystems* automontados em ambiente UNIX, cuja idéia básica é reportar sobre todos os arquivos e diretórios montáveis pelo *automount* e não apenas os já montados, através de uma simples manipulação dos mapas NIS e dos mapas locais do sistema. Isto significa dizer que nada é realmente montado como consequência do uso da ferramenta, possibilitando acesso rápido e minimizando montagens cruzadas entre servidores NFS, bem como os efeitos colaterais decorrentes dessa interdependência.

## 1. Introdução

A idéia para o desenvolvimento do automirror surgiu a partir da análise do princípio de automontagem, onde se verificou a necessidade de um procedimento auxiliar que fornecesse os possíveis pontos de montagem de um arquivo ou diretório procurado sem a necessidade dos mesmos serem montados (pelo automounter). Desta forma, a automontagem pode ficar restrita às hierarquias que efetivamente necessitam ser analisadas.

*Por que se deseja minimizar o uso da automontagem?*

O NFS automounter foi introduzido como procedimento auxiliar do processo de montagem tradicional (uso do mount), possibilitando a montagem de diretórios remotos sem que este procedimento fosse realizado durante o processo de boot do sistema. A montagem pode agora ser feita de maneira dinâmica, sob demanda, onde os diretórios permanecem montados tanto tempo quanto necessário, sendo desmontados depois de passarem um certo período sem que nenhum acesso a eles tenha sido realizado. O processo de automontagem é bem mais flexível do que a montagem. Os diretórios não mais acessados ainda permanecem montados por um determinado tempo e, portanto, permanece a possibilidade de problemas em um servidor causarem efeitos em outros. Montagens desnecessárias podem decorrer do uso excessivo da automontagem: a exploração de hierarquias ou busca de informações pode levar o usuário mais experiente a forçar a montagem de todas as sub-hierarquias de um ponto de automontagem.

Dado este contexto, constatou-se a necessidade de habilitar o usuário a descobrir o que pode ser montado, sem que para isto tenha que ser realmente montada a hierarquia de diretórios de interesse. Como consequência imediata, o usuário obteria a informação desejada em menor tempo e com menos esforço. Mais importante, contudo, seria a sensação de segurança conseguida pelo usuário ao ter sempre uma visão completa da hierarquia; é fato comprovado (por administradores de sistemas) que o funcionamento do *automounter* chega a causar um forte impacto nos usuários menos experientes, quando um determinado diretório não é listado por *ls* no ponto esperado da hierarquia.

Com estes problemas identificados e benefícios almejados, surgiu a idéia do **sistema automirror**, que permite a consulta aos possíveis pontos de montagens do *automount* sem que nenhum deles seja montado. O utilitário que executa tal procedimento foi denominado de *mfs* e funciona semelhantemente ao comando *ls* (lista todos os arquivos do diretório corrente), exceto pela capacidade de listar também todos os arquivos e diretórios ainda não montados pelo *automount*.

Nas próximas seções será descrito mais detalhadamente o princípio de funcionamento da automontagem de hierarquias de diretórios, bem como o comando *ls*, ambos de fundamental importância para um melhor entendimento do sistema proposto. Em seguida será descrito o sistema automirror, detalhes sobre o *mfs* - utilitário que implementa as idéias propostas - e, por fim, algumas considerações finais.

## 2. O Automounter

O princípio de *automontagem* de hierarquias de arquivos remotos viabiliza a montagem e desmontagem de diretórios por usuários do sistema dinamicamente. Na verdade, a montagem é feita automaticamente quando um usuário executa um comando que precise realizar o acesso a diretórios remotos e são *desmontados* automaticamente depois de decorrido um certo tempo sem que tenha sido efetuado algum acesso aos mesmos. O sistema que implementa tal facilidade é chamado *automounter*.

O *NFS automounter* (*Network File System automounter*) foi desenvolvido para substituir, em determinados casos, o uso de NFS tradicional na montagem de *filesystems* via rede. Ao contrário do *mount*, o *daemon automount* permite que a montagem de arquivos e diretórios seja realizada de maneira dinâmica, sob demanda. O *automount* não consulta o arquivo */etc/vfstab* (SysV) ou */etc/fstab* (BSD) [3] [4], e sim, consulta um conjunto de mapas que especificam como a montagem deve ser realizada. Tais mapas podem ser arquivos locais ou mapas *NIS* (*Network Information Service*). Os nomes dos mapas podem ser passados para o *automount* através da linha de comando ou através dos outros mapas.

O procedimento interno do *automount* é o seguinte: o *daemon automount* é iniciado, seja da linha de comando ou de um *script rc* (*boot*), com o intuito de servir os pontos de montagem especificados nos mapas e, desta forma, o processo de montagem pode ser feito como se fosse através do *mount*, ou seja, para o *kernel* as hierarquias estão previamente montadas. O *daemon* fica aguardando que uma requisição ao sistema de arquivos seja feita sobre as hierarquias sob seu cuidado; o *kernel* é bloqueado, a hierarquia é montada e então retorna-se ao *kernel* para ser efetuado o acesso ao arquivo solicitado. A hierarquia de diretórios remotos é desfeita quando nenhum acesso for realizado pelo usuário ou sistema, depois de um certo tempo pré-determinado.

A vantagem no uso do *automount* é que durante o processo de *boot* do sistema, nenhum *filesystem* tratado pelo *automount* é montado desnecessariamente. Isto diminui, em média, o número de montagens cruzadas entre os servidores NFS e, desta forma, diminui também as chances de um servidor ter problemas devido aos problemas ocorridos em algum outro servidor conectado na rede.

### 2.1. O Comando ls

O comando *ls* lista o conteúdo do diretório corrente, exibindo o nome e outras informações adicionais de acordo com a opção solicitada. O *ls* mostra somente os arquivos e os diretórios efetivamente montados no sistema.

Um exemplo de uma execução do comando *ls* seria:

```
dir_corrente% ls -a
```

Tal execução, com a opção *-a*, lista os nomes de todos os arquivos do diretório "dir\_corrente", incluindo os arquivos que começam com "." (ponto) que normalmente não são listados.

### 3. O Automirror

O *automirror* ou *sistema paralelo para navegação em filesystems automontados*, possibilita a visualização de todos os arquivos e diretórios montados que *podem* ainda ser montados num determinado diretório, sem que para isto eles sejam montados.

Um usuário, desta forma, pode verificar a partir do seu diretório corrente *quais* são os arquivos e diretórios que *podem* ser montados pelo *automount*. A verificação dos pontos de montagem de determinado arquivo ou diretório torna-se possível sem que o processo de automontagem seja efetuado. Sendo assim, o usuário primeiramente procura por um diretório ou arquivo usando o *automirror* e quando encontrar os possíveis pontos de montagem é que ele utiliza o processo de automontagem do sistema.

O *automirror* consulta o mesmo conjunto de mapas consultados pelo *NFS automounter*, ou seja, tanto arquivos locais quanto mapas NIS. Os **mapas** que podem ser acessados são classificados em três grandes grupos: **mapa master**, **mapas diretos** e **mapas indiretos**.

#### MAPA MASTER

Contém a configuração básica do *automounter*. Seu conteúdo permite o acesso aos demais mapas que realmente contém as informações para montagem. Geralmente é o arquivo */etc/auto.master*, mas também pode ser o mapa NIS *auto.master*.

#### SINTAXE:

**mount\_point**            **map\_name**            [**mount\_options**]

- **mount\_point (ponto de montagem)** → contém o nome completo de um diretório a partir de onde vai ser montada a hierarquia do mapa descrito no segundo campo (*map\_name*).
- **map\_name (nome do mapa)** → contém o nome do mapa onde estão relacionados os pontos de montagem e hierarquias de diretórios.
- **mount\_options (opções de montagem)** → contém uma lista de opções de montagem, idêntica a existente em */etc/fstab* ou */etc/vfstab*, que será aplicada como *default* para as entradas do mapa mencionado, a menos que o mapa especifique outras opções.

Ao contrário do NFS (montagem permanente), o *automounter* cria o diretório especificado em *mount\_point* sem auxílio adicional, sempre que possível. Porém, semelhantemente ao NFS, se o diretório já existe e não está vazio, seu conteúdo fica inacessível enquanto o *automounter* continuar ativo.

Se *mount\_point* contiver os símbolos “/”, significa que o mapa especificado é direto, ou seja, não há um ponto de montagem especificado no mapa *auto.master*, e sim o nome de um outro mapa que contém a localização da hierarquia de diretórios.

**EXEMPLO:**

```

auto.master
#mount_point      map              mount_options
/n                /etc/auto.ns    -rw,intr
/home            auto.home       -rw,intr
/-              /etc/auto.direct -rw,intr

```

O símbolo # indica uma linha de comentários. Na configuração acima, o mapa *master* utiliza três mapas auxiliares para a realização da automontagem. Em todos eles as opções de montagem *default* são *rw* e *intr*. Então:

- **auto.ns** → é um mapa local (em */etc*) que, neste exemplo, monta partições de recursos locais (softwares de domínio público, documentos de interesse geral, entre outros);
- **auto.home** → é um mapa NIS (pois o segundo campo não especifica uma hierarquia de diretórios) que monta *home directories* de usuários;
- **auto-direct** → é um mapa local (em */etc*) direto (pois sua entrada no mapa *master* é “/-”), que monta hierarquias em localização padrão do sistema, mas que não são vitais como “/” e “/usr”, por exemplo.

**MAPAS DIRETOS E INDIETOS**

**Mapas Indiretos** → Contém instruções de montagem que possibilitam a construção de uma hierarquia de diretórios como desejado, com bastante flexibilidade. Seus pontos de montagem são relativos a algum diretório.

**Mapas Diretos** → Também contém instruções de montagem, sendo que tais mapas são bem menos flexíveis do que os mapas indiretos. Seus pontos de montagem são absolutos, obtidos diretamente dos mapas e se destinam àqueles filesystems (ou diretórios) que possuem um lugar fixo na hierarquia, independentemente da máquina.

**SINTAXE:**

```

      key           [mount_options]    location

```

- **key (chave)** → contém o *pathname* (hierarquia de diretórios) completo do ponto de montagem, ou seja, a hierarquia sendo montada..
- **mount\_options (opções de montagem)** → contém uma lista de opções de montagem que será aplicada para a hierarquia de diretórios especificada, em detrimento do que possa estar definido nas opções do mapa *auto.master*.
- **location (origem)** → contém a localização de origem da hierarquia de diretórios que está sendo importada, na forma “*servidor:pathname*”.

A diferença entre uma mapa direto e um indireto é que no primeiro a chave é um *pathname* completo, ao passo que num mapa indireto a chave é apenas um nome.

#### EXEMPLO:

##### **/etc/auto.direct**

#key	mount_options	location
/usr/local	-ro,intr	máquina:&
/usr/spool/mail	-nosuid	máquina1,máquina2:&

Este é um exemplo de um mapa direto, pois as chaves são *pathnames* completos. O símbolo "&" é um *alias* para a chave que inicia a linha correspondente.

##### **/etc/auto.ns**

#key	mount_options	location
gnu		máquina:/n/gnu/bin/solaris
dtp		máquina:/l/dtp/s11

##### **auto.home**

#key	mount_options	location
adm		máquina1:/l/proj01/&
sys		máquina1:/l/proj01/&
*		máquina2:/l/home06/&

Estes são exemplos de mapas indiretos. O primeiro exemplo (*/etc/auto.ns*) é um mapa local do sistema e o segundo (*auto.home*) é um mapa NIS. A concatenação do *mount\_point* especificado no mapa master com as chaves em *auto.home* (ou em */etc/auto.ns*) definem os paths das hierarquias a serem montadas.

#### Exemplo:

- *mount-point* em *auto.master*: /home
- chave em *auto.home*: adm

⇒ O diretório *máquina:/l/proj01/adm* é montado em */home/adm*.

Pode ser observado no exemplo de mapa indireto *auto.home* que na última linha a chave é um asterisco ("\*"). Tal símbolo é um coringa que utiliza todas as chaves que o *automounter* tentar no mapa e, por este motivo, ele deve sempre ser usado na última linha do mapa. O "&" servirá de *alias* no campo origem para o conteúdo capturado pelo asterisco. Percebe-se também que as chaves *adm* e *sys* possuem seus *home directories* na mesma máquina (*máquina1*) e, neste caso, todas as outras opções de chave possuem seus *home directories* numa outra (*máquina2*).

## INCLUINDO MAPAS

Pode-se também incluir o conteúdo de outros mapas num certo ponto de um mapa [2]. Neste caso a chave deve ser substituída por “+nome\_do\_mapa”, conforme mostrado a seguir numa outra especificação do mapa *auto.home*:

### **auto.home**

#key	mount_options	location
adm		máquina1:/l/proj01/&
sys		máquina1:/l/proj01/&
+auto.pós		
+auto.grad		
+/etc/auto.proj		

Neste exemplo, separou-se certos mapas *auto.home* em categorias (pos, grad e proj).

O *automirror* procura o arquivo desejado, tratando todas as possibilidades encontradas nos três grandes grupos de mapas (NIS e locais do sistema) e que foram descritas acima, como os diferentes pontos de montagem e chaves existentes. Para o usuário este processo é transparente, ou seja, não importa se o arquivo procurado já foi montado ou não e sim quais são todos os arquivos existentes num determinado ponto do sistema; por isto nenhuma diferenciação é feita na exibição dos arquivos montados e os ainda não montados.

### 3.1. O Comando *mls*

O comando representando o sistema *automirror* foi denominado de **mls** (*mirror ls*), sendo implementado na linguagem de *scripts* do **Korn Shell (ksh)** [1], disponível em **ambientes UNIX**. O resultado da execução do comando *mls* é obtido pela simples navegação nos mapas locais ou NIS do sistema. Em outras palavras, nenhum arquivo ou diretório é realmente montado.

O comando *mls*, tal como o *ls*, lista todos os arquivos montados (temporária e permanentemente) na hierarquia especificada, porém, além disto, ele mostra todos os arquivos e diretórios que ainda não foram montados pelo *automount*. Desta forma, o usuário pode usar o comando *mls* para procurar um determinado arquivo ou diretório no ponto de montagem, e somente quando encontrar, realizar a automontagem (executar o comando *cd - change directory* ou *ls* de tal arquivo ou diretório).

O comando *mls* implementado lista os arquivos e diretórios do diretório corrente que já foram montados ou não, se houver algum. O resultado da execução é listado em ordem alfabética, um nome por linha, quando nenhuma opção for estipulada. Se a opção [-f] for utilizada, o resultado da execução é mostrado em colunas, ordenadas alfabeticamente na vertical. Se a opção for [-l], os arquivos serão mostrados um por linha, com os pontos onde os mesmos podem ser montados.

Os nomes dos arquivos podem ser especificados pelos "coringas" \* e ? e, neste caso, tais caracteres devem ser preservados de interpretação pelo *shell* ("quoted"). Se *mls* for usado sem argumentos, ele listará todos os pontos de montagem do diretório corrente.

As possíveis opções são:

**mls [ -lfv ]**

- l Lista em formato longo, atribuindo para cada arquivo ou diretório a origem do mesmo.
- f Formata o resultado da execução em colunas, ordenando alfabeticamente no sentido horizontal.
- v Mostra informações sobre a versão corrente do *mls*, antes de mostrar o conteúdo do diretório no formato padrão.

**EXEMPLOS:**

```
dir_corrente% ls /home
ic
msc
spec
staff
```

O exemplo acima é de uma execução do *ls* e, abaixo, sobre a mesma hierarquia, é apresentada uma execução do comando *mls*.

```
dir_corrente% mls /home
adm
ahand
ex
grad
ic
lac
msc
phd
ra
spec
staff
sys
```

Observa-se que o exemplo acima, do *mls*, lista os nomes de todos os arquivos montados (arquivos listados pelo *ls*) e os que ainda podem ser montados no diretório */home*.

Novamente, ressalta-se que absolutamente nada é montado como consequência da execução do comando.

O exemplo abaixo apresenta o resultado da execução do mls com a opção “-f”:

```
dir_corrente% mls -f /home
```

```
adm    ahand    ex    grad    ic
lac    msc     phd    ra     spec
staff  sys
```

A seguir, encontra-se um exemplo da execução do mls com a opção “-l”. São listados, juntamente com os nomes dos arquivos montados e que podem ser montados no diretório /home, a origem real dos arquivos e diretórios. A chave *msc* que pode ser montada em /home/msc, por exemplo, está instalada na máquina *grande* sob o diretório /home00/msc.

```
dir_corrente% mls -l /home
```

```
adm    grande:/home00:&
ahand  atibaia:/proj02/ahand
ex     fork:/rep05:&
grad   jaguari:/home05:&
ic     grande:/home00:&
lac    pascal:/home80:&
msc    grande:/home00:&
phd    grande:/home00:&
ra     exec:/home03:&
spec   grande:/home00:&
staff  grande:/home00:&
sys    grande:/rep09:&
```

## 4. Conclusões

O automirror ou sistema paralelo para navegação em filesystems automontados foi elaborado como uma ferramenta de auxílio para o esquema de automontagem em redes UNIX. Ele permite a verificação dos possíveis pontos de montagem, sem que os arquivos ou diretórios procurados sejam montados. Isto faz com que a automontagem só precise ser efetuada quando realmente for necessária, reduzindo automontagens desnecessárias e os possíveis efeitos dessas dependências. Torna ainda o acesso mais rápido, facilitando significativamente a exploração e navegação na hierarquia da rede pelo usuário.

## Referências Bibliográficas

- [1] BOLSKY, Morris J., KORN, David G. *The KornShell: Command and Programming Language*. Prentice Hall: Englewood Cliffs, New Jersey, USA, 1989.
- [2] GEUS, Paulo Lício de. *Administração do Sistema UNIX*. Projeto A-Hand / Departamento de Ciência da Computação, Universidade Estadual de Campinas, Campinas, SP, 1993.
- [3] *Solaris 2.5: System Administration Answerbook*. Sun Microsystems: Califórnia, USA, 1995.
- [4] *SunOS 4.1: System Network Administration*. Sun Microsystems: Califórnia, USA, 1994.

## Agradecimentos

Aos nossos colegas Edson Bacin, Thomaz de Aquino G. Júnior e Wesley Elias Ribeiro, que implementaram a primeira versão do utilitário *mls*.