



Criar e alterar links físicos e simbólicos

Sumário

Capítulo 1

Criar e alterar links físicos e simbólicos	3
1.1. Mãos a obra.....	4

Capítulo 2

Gerenciando	12
2.1. Objetivos.....	12
2.1. Troubleshooting.....	12

Índice de tabelas

Índice de Figuras

Capítulo 1

Criar e alterar links físicos e simbólicos

- Criar links;
- Identificar rígido e / ou links lógicos;
- Copiar arquivos "versus" links;
- Usar links para apoiar as tarefas de administração do sistema.

1.1. Mãos a obra

O acesso a arquivos e diretórios no sistema, pode ser otimizado na criação dos chamados “links”. Essas informações estão gravadas em uma coleção de blocos, e tudo o que diz respeito a um arquivo, como o dono do arquivo, data de acesso e permissões estão gravados em um inode.



O que são inodes?

É um tipo especial de estrutura de dados, onde contem informações básicas sobre seus arquivos e pastas, como permissões de acesso, identificação dos donos dos arquivos, data e hora do último acesso, tamanho e os ponteiros para o arquivo. Sendo assim uma identificação única do arquivo. Vamos a exemplo prático:

Exibir qual inode de um arquivo:



```
# stat /etc/passwd
```

```
File: '/etc/passwd'
Size: 1544          Blocks: 8          IO Block: 4096   arquivo comum
Device: 802h/2050d Inode: 145092       Links: 1
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2010-09-10 07:34:16.000000000 -0300
Modify: 2010-09-09 22:18:46.000000000 -0300
Change: 2010-09-09 22:18:46.000000000 -0300
```

Em nosso exemplo o inode 145092 contem informações sobre o arquivo /etc/passwd

Agora vamos exibir os inodes das partições



```
# df -i
```

```
Sist. Arq.      Inodes    IUsados  ILivr  IUsos% Montado em
/dev/sda2      183264    11568    171696    7% /
tmpfs          48281         4    48277    1% /lib/init/rw
udev           48281    1009    47272    3% /dev
tmpfs          48281         1    48280    1% /dev/shm
/dev/sda1       62248         29    62219    1% /boot
/dev/sda3      183264        277    182987    1% /home
/dev/sda7       62336         48    62288    1% /tmp
/dev/sda5      214272    71323    142949   34% /usr
/dev/sda6      214272    9395    204877    5% /var
/dev/sda8      214272        139    214133    1% /var/log
```



Qual a relação de inodes com links?

Quando você cria um link, é como criar um apelido para a localização de um arquivo ou diretório em um sistema de arquivos. Você tem duas maneiras de criar esses links, e a diferença entre eles está na ligação com os inodes.

Link Simbólico

O link simbólico é um arquivo especial de disco do tipo link, que tem como conteúdo, o caminho para chegar até o arquivo original tendo o inode diferente.

Link Físico

O link físico (hardlink) é apontado para o mesmo inode do arquivo original.

Diferenças entre os tipos de links

Link Simbólico

- É possível criar links para arquivos e diretórios;
- O arquivo original não precisa estar na mesma partição de disco;
- Caso o link simbólico for apagado e/ou movido, não afeta o original;
- Qualquer usuário pode criar e/ou remover seu link simbólico.

Link Físico

- Não é possível criar hardlink para diretórios;
- O arquivo original precisa estar na mesma partição de disco;
- Caso o link físico for apagado e/ou movido, irá afetar o original;
- Somente o usuário root pode criar e/ou remover link físico.

Criar links

Vamos a prática na criação e identificação dos links, através do comando `ln`. O que difere os dois tipos é a flag `-s` na criação dos links.

`ln <flags> <caminho do arquivo original> <caminho do arquivo do link>`



```
# ln -s /etc/passwd /root/passwd
```

Em nosso exemplo foi criado um link simbólico no diretório /root, que aponta para o arquivo /etc/passwd. Veja abaixo maneiras de identificar um link.

Comando ls



```
# ls -l --color /root/passwd
```

```
lrwxrwxrwx 1 root root 11 Set 10 07:51 /root/passwd -> /etc/passwd
```

Veja em nosso exemplo o uso do comando ls com -l para detalhes e --color para exibir cores. A letra “l” no lado esquerdo nos detalhes do arquivo, indica que é um tipo link, e a cor e seta aponta para o arquivo original.

Comando stat



```
# stat /root/passwd
```

```
File: '/root/passwd' -> '/etc/passwd'
Size: 11          Blocks: 0          IO Block: 4096   link simbólico
Device: 802h/2050d Inode: 135464       Links: 1
Access: (0777/lrwxrwxrwx)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2010-09-10 08:22:16.000000000 -0300
Modify: 2010-09-10 07:51:57.000000000 -0300
Change: 2010-09-10 07:51:57.000000000 -0300
```

O resultado do comando stat é mais simples, pois ao lado esquerdo em “File” exibe o caminho do arquivo original e ao lado direito a descrição “link simbólico”



Quando você altera o arquivo original o link é alterado e vice-versa!

Link Físico (hardlink)

Agora vamos criar um hardlink e verificar a informação sobre inodes. O comando para criar é o `ln` sem a flag `-s`.



```
# ln /etc/fstab /root/fstab
```

Em nosso exemplo foi criado um link físico no diretório `/root`, que aponta para o arquivo `/etc/fstab`. Veja abaixo maneiras de identificar um link.

Comando `ls`



```
# ls -l /root/fstab
```

```
-rw-r--r-- 2 root root 786 Ago 20 11:23 /root/fstab
```

Veja em nosso exemplo o uso do comando `ls` com `-l` para detalhes. O número 2 antes de “root” indica que no sistema este arquivo aparece duas vezes. Um é o original e o outro é um link.

Comparação entre os arquivos com o comando `stat`



```
# stat /etc/fstab
```

```
File: '/etc/fstab'
Size: 786          Blocks: 8          IO Block: 4096   arquivo comum
Device: 802h/2050d Inode: 143427       Links: 2
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2010-09-10 07:07:19.000000000 -0300
Modify: 2010-08-20 11:23:00.000000000 -0300
Change: 2010-09-10 08:32:23.000000000 -0300
```




```
# stat /root/fstab
```

```
File: '/root/fstab'
Size: 786          Blocks: 8          IO Block: 4096   arquivo comum
Device: 802h/2050d Inode: 143427       Links: 2
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2010-09-10 07:07:19.000000000 -0300
Modify: 2010-08-20 11:23:00.000000000 -0300
Change: 2010-09-10 08:32:23.000000000 -0300
```

Em nosso exemplo não vemos nenhuma diferença ao criar hardlink, então não seria melhor copiar o arquivo ao invés de criar links?

A diferença está no que você precisa! Um exemplo seria em uma operação de backup, que o ideal é criar uma cópia de seus dados. Para facilitar o acesso rápido a diretórios e atualização de conteúdos de arquivos, você cria links.



Como posso verificar a existência de links no sistema?

Em nosso sistema existem diversos links simbólicos, que apontam para scripts de inicialização, bibliotecas e binários. Esses links ajudam na administração como por exemplo uma única biblioteca ser utilizada por vários binários.

Links para scripts de inicialização



```
# ls -l -color /etc/rc2.d
```

```

lrwxrwxrwx 1 root root 17 Fev 24 2010 S10rsyslog -> ../init.d/rsyslog
lrwxrwxrwx 1 root root 15 Fev 24 2010 S12acpid -> ../init.d/acpid
lrwxrwxrwx 1 root root 14 Fev 24 2010 S12dbus -> ../init.d/dbus
lrwxrwxrwx 1 root root 15 Ago 20 21:14 S15bind9 -> ../init.d/bind9
lrwxrwxrwx 1 root root 13 Abr 8 09:47 S16ssh -> ../init.d/ssh
lrwxrwxrwx 1 root root 16 Set 10 04:29 S19autofs -> ../init.d/autofs
lrwxrwxrwx 1 root root 15 Set 9 21:40 S19slapd -> ../init.d/slapd
lrwxrwxrwx 1 root root 14 Fev 24 2010 S20cups -> ../init.d/cups
lrwxrwxrwx 1 root root 13 Set 2 08:25 S20gpm -> ../init.d/gpm
lrwxrwxrwx 1 root root 20 Ago 20 17:03 S20nfs-common -> ../init.d/nfs-common
lrwxrwxrwx 1 root root 27 Ago 20 15:51 S20nfs-kernel-server -> ../init.d/nfs-kernel-server
lrwxrwxrwx 1 root root 14 Set 9 22:00 S20nscd -> ../init.d/nscd
lrwxrwxrwx 1 root root 17 Ago 20 15:52 S20portmap -> ../init.d/portmap
lrwxrwxrwx 1 root root 15 Ago 20 17:14 S20samba -> ../init.d/samba
lrwxrwxrwx 1 root root 16 Ago 20 14:58 S20vsftpd -> ../init.d/vsftpd
lrwxrwxrwx 1 root root 13 Fev 24 2010 S24hal -> ../init.d/hal
lrwxrwxrwx 1 root root 15 Set 9 22:34 S30squid -> ../init.d/squid
lrwxrwxrwx 1 root root 31 Fev 24 2010 S30system-tools-backends -> ../init.d/system-tools-backends
lrwxrwxrwx 1 root root 17 Fev 24 2010 S89anacron -> ../init.d/anacron
lrwxrwxrwx 1 root root 13 Fev 24 2010 S89atd -> ../init.d/atd
lrwxrwxrwx 1 root root 14 Fev 24 2010 S89cron -> ../init.d/cron
lrwxrwxrwx 1 root root 17 Ago 20 22:10 S91apache2 -> ../init.d/apache2
lrwxrwxrwx 1 root root 18 Fev 24 2010 S99rc.local -> ../init.d/rc.local

```

Links para as bibliotecas



```
# ldconfig -p | less
```

```

617 libs found in cache '/etc/ld.so.cache'
libz.so.1 (libc6) => /usr/lib/libz.so.1
libx86.so.1 (libc6) => /lib/libx86.so.1
libx11globalcomm.so.1 (libc6) => /usr/lib/libx11globalcomm.so.1
libxtables.so.0 (libc6) => /lib/libxtables.so.0
libxslt.so.1 (libc6) => /usr/lib/libxslt.so.1
libxml2.so.2 (libc6) => /usr/lib/libxml2.so.2
libxklavier.so.12 (libc6) => /usr/lib/libxklavier.so.12
libxkbfile.so.1 (libc6) => /usr/lib/libxkbfile.so.1
libxcb.so.1 (libc6) => /usr/lib/libxcb.so.1
libxcb-xlib.so.0 (libc6) => /usr/lib/libxcb-xlib.so.0
libxcb-render.so.0 (libc6) => /usr/lib/libxcb-render.so.0
libxcb-render-util.so.0 (libc6) => /usr/lib/libxcb-render-util.so.0
libxapian.so.15 (libc6) => /usr/lib/libxapian.so.15
libwrap.so.0 (libc6) => /lib/libwrap.so.0
libwnck-1.so.22 (libc6) => /usr/lib/libwnck-1.so.22
libwbclient.so.0 (libc6) => /usr/lib/libwbclient.so.0
libwvpack.so.1 (libc6) => /usr/lib/libwvpack.so.1
libvte.so.9 (libc6) => /usr/lib/libvte.so.9
libvorbisfile.so.3 (libc6) => /usr/lib/libvorbisfile.so.3
libvorbisenc.so.2 (libc6) => /usr/lib/libvorbisenc.so.2
libvorbis.so.0 (libc6) => /usr/lib/libvorbis.so.0
libvolume_id.so.0 (libc6) => /lib/libvolume_id.so.0
libvoikko.so.1 (libc6) => /usr/lib/libvoikko.so.1

```

Links para as bibliotecas



```
# ls -l $(find /usr/bin -type -l)
```

```
lrwxrwxrwx 1 root root 6 Fev 24 2010 /usr/bin/apropos -> whatis
lrwxrwxrwx 1 root root 5 Fev 24 2010 /usr/bin/arecord -> aplay
lrwxrwxrwx 1 root root 2 Fev 24 2010 /usr/bin/atq -> at
lrwxrwxrwx 1 root root 2 Fev 24 2010 /usr/bin/atrm -> at
lrwxrwxrwx 1 root root 21 Fev 24 2010 /usr/bin/awk -> /etc/alternatives/awk
lrwxrwxrwx 1 root root 4 Fev 24 2010 /usr/bin/cal -> ncal
lrwxrwxrwx 1 root root 3 Fev 24 2010 /usr/bin/captoinfo -> tic
lrwxrwxrwx 1 root root 9 Fev 24 2010 /usr/bin/charmap -> gucharmap
lrwxrwxrwx 1 root root 11 Fev 24 2010 /usr/bin/compose -> run-mailcap
lrwxrwxrwx 1 root root 7 Fev 24 2010 /usr/bin/cpp -> cpp-4.3
lrwxrwxrwx 1 root root 31 Fev 24 2010 /usr/bin/csu2vcard -> ../lib/evolution/2.
22/csu2vcard
lrwxrwxrwx 1 root root 6 Fev 24 2010 /usr/bin/ctstat -> lstat
lrwxrwxrwx 1 root root 14 Fev 24 2010 /usr/bin/cupsdisable -> ../sbin/accept
lrwxrwxrwx 1 root root 14 Fev 24 2010 /usr/bin/cupsenable -> ../sbin/accept
lrwxrwxrwx 1 root root 6 Fev 24 2010 /usr/bin/defoma-app -> defoma
lrwxrwxrwx 1 root root 6 Fev 24 2010 /usr/bin/defoma-font -> defoma
lrwxrwxrwx 1 root root 6 Fev 24 2010 /usr/bin/defoma-id -> defoma
lrwxrwxrwx 1 root root 6 Fev 24 2010 /usr/bin/defoma-subst -> defoma
lrwxrwxrwx 1 root root 6 Fev 24 2010 /usr/bin/defoma-user -> defoma
lrwxrwxrwx 1 root root 11 Fev 24 2010 /usr/bin/edit -> run-mailcap
lrwxrwxrwx 1 root root 24 Fev 24 2010 /usr/bin/editor -> /etc/alternatives/edit
or
lrwxrwxrwx 1 root root 34 Fev 24 2010 /usr/bin/epiphany-browser -> /etc/alterna
```

Capítulo 2

Gerenciando

2.1. Objetivos

- Troubleshooting: Utilização do comando ln.

2.1. Troubleshooting



Quais são as opções de uso do comando ln?

Através do comando ln é possível criar hardlinks e links simbólicos. Seu manual pode ser exibido usando o comando man ln. Vamos a prática:



man ln

Vamos ver na prática algumas opções de uso do comando:

Criar um link simbólico de um diretório de manuais.



```
# ln -s /usr/share/doc doc
```

Em nosso exemplo foi criado um atalho (link) de nome doc que aponta para /usr/share/doc. Para listar o conteúdo do diretório /usr/share/doc, apenas liste o doc em seu diretório atual.



```
# ls --color doc
```

```
libbluetooth2      php5-common
libbonobo2-0       php5-ldap
libbonobo2-common  phpldapadmin
libbonoboui2-0     pkg-config
libbonoboui2-common pmount
libbz2-1.0         pm-utils
libc6              poppler-utils
libc6-i686         portmap
libcaca0           powermgmt-base
libcairo2          procmail
libcairomm-1.0-1   procs
libcairo-perl      psmisc
libcamel1.2-11    pump
libcap1            python
libcap2            python2.5
libcdaudio1        python2.5-minimal
libcdio7           python-cairo
libcdparanoia0     python-central
libcomerr2         python-dbus
libconsole         python-gdata
libcpufreq0        python-glade2
libcroco3          python-gmenu
libcucul0          python-gnome2
libcups2           python-gnome2-desktop
libcupsimage2      python-gobject
```

Criar backup de um arquivo.



```
# ln -b /etc/shadow
```

Criar um link simbólico e força a substituição o link atual.



```
# ln -sf /etc/fstab /root/fstab
```

Criar um link simbólico, mas pergunta que você quer substituir o link atual.



```
# ln -si /etc/fstab /root/fstab
```