

Guia Foca GNU/Linux

Capítulo 9 - Comandos para manipulação de Arquivos

Abaixo, comandos utilizados para manipulação de arquivos.

9.1 cat

Mostra o conteúdo de um arquivo binário ou texto.

```
cat [opções] [diretório/arquivo] [diretório1/arquivo1]  
diretório/arquivo
```

Localização do arquivo que deseja visualizar o conteúdo.

opções

-n, --number

Mostra o número das linhas enquanto o conteúdo do arquivo é mostrado.

-s, --squeeze-blank

Não mostra mais que uma linha em branco entre um parágrafo e outro.

-

Lê a entrada padrão.

O comando `cat` trabalha com arquivos texto. Use o comando `zcat` para ver diretamente arquivos compactados com `gzip`.

Exemplo: `cat /usr/doc/copyright/GPL`

9.2 tac

Mostra o conteúdo de um arquivo binário ou texto (como o `cat`) só que em ordem inversa.

```
tac [opções] [diretório/arquivo] [diretório1/arquivo1]  
diretório/arquivo
```

Localização do arquivo que deseja visualizar o conteúdo

opções

-s [string]

Usa o [string] como separador de registros.

-

Lê a entrada padrão.

Exemplo: `tac /usr/doc/copyright/GPL.`

9.3 rm

Apaga arquivos. Também pode ser usado para apagar diretórios e sub-diretórios vazios ou que contenham arquivos.

`rm [opções][caminho][arquivo/diretório] [caminho1][arquivo1/diretório1]`

onde:

caminho

Localização do arquivo que deseja apagar. Se omitido, assume que o arquivo esteja no diretório atual.

arquivo/diretório

Arquivo que será apagado.

opções

-i, --interactive

Pergunta antes de remover, esta é ativada por padrão.

-v, --verbose

Mostra os arquivos na medida que são removidos.

-r, --recursive

Usado para remover arquivos em sub-diretórios. Esta opção também pode ser usada para remover sub-diretórios.

-f, --force

Remove os arquivos sem perguntar.

-- arquivo

Remove arquivos/diretórios que contém caracteres especiais. O separador "--" funciona com todos os comandos do shell e permite que os caracteres especiais como "*", "?", "-", etc. sejam interpretados como caracteres comuns.

Use com atenção o comando `rm`, uma vez que os arquivos e diretórios forem apagados, eles não poderão ser mais recuperados.

Exemplos:

- `rm teste.txt` - Apaga o arquivo `teste.txt` no diretório atual.
 - `rm *.txt` - Apaga todos os arquivos do diretório atual que terminam com `.txt`.
 - `rm *.txt teste.novo` - Apaga todos os arquivos do diretório atual que terminam com `.txt` e também o arquivo `teste.novo`.
 - `rm -rf /tmp/teste/*` - Apaga todos os arquivos e sub-diretórios do diretório `/tmp/teste` mas mantém o sub-diretório `/tmp/teste`.
 - `rm -rf /tmp/teste` - Apaga todos os arquivos e sub-diretórios do diretório `/tmp/teste`, inclusive `/tmp/teste`.
 - `rm -f -- --arquivo--` - Remove o arquivo de nome `--arquivo--`.
-

9.4 cp

Copia arquivos.

`cp [opções] [origem] [destino]`

onde:

origem

Arquivo que será copiado. Podem ser especificados mais de um arquivo para ser copiado usando "Curingas" (veja [Curingas, Seção 2.12](#)).

destino

O caminho ou nome de arquivo onde será copiado. Se o destino for um diretório, os arquivos de origem serão copiados para dentro do diretório.

opções

`i, --interactive`

Pergunta antes de substituir um arquivo existente.

`-f, --force`

Não pergunta, substitui todos os arquivos caso já exista.

`-r`

Copia arquivos dos diretórios e subdiretórios da origem para o destino. É recomendável usar -R ao invés de -r.

-R, --recursive

Copia arquivos e sub-diretórios (como a opção -r) e também os arquivos especiais FIFO e dispositivos.

-v, --verbose

Mostra os arquivos enquanto estão sendo copiados.

-s, --symbolic-link

Cria link simbólico ao invés de copiar.

-l, --link

Faz o link no destino ao invés de copiar os arquivos.

-p, --preserve

Preserva atributos do arquivo, se for possível.

-u, --update

Copia somente se o arquivo de origem é mais novo que o arquivo de destino ou quando o arquivo de destino não existe.

-x

Não copia arquivos que estão localizados em um sistema de arquivos diferente de onde a cópia iniciou.

O comando `cp` copia arquivos da ORIGEM para o DESTINO. Ambos origem e destino terão o mesmo conteúdo após a cópia.

Exemplos:

```
cp teste.txt teste1.txt
```

Copia o arquivo teste.txt para teste1.txt.

```
cp teste.txt /tmp
```

Copia o arquivo teste.txt para dentro do diretório /tmp.

```
cp * /tmp
```

Copia todos os arquivos do diretório atual para /tmp.

```
cp /bin/* .
```

Copia todos os arquivos do diretório /bin para o diretório em que nos encontramos no momento.

```
cp -R /bin /tmp
```

Copia o diretório /bin e todos os arquivos/sub-diretórios existentes para o diretório /tmp.

```
cp -R /bin/* /tmp
```

Copia todos os arquivos do diretório `/bin` (exceto o diretório `/bin`) e todos os arquivos/sub-diretórios existentes dentro dele para `/tmp`.

```
cp -R /bin /tmp
```

Copia todos os arquivos e o diretório `/bin` para `/tmp`.

9.5 mv

Move ou renomeia arquivos e diretórios. O processo é semelhante ao do comando `cp` mas o arquivo de origem é apagado após o término da cópia.

```
mv [opções] [origem] [destino]
```

Onde:

origem

Arquivo/diretório de origem.

destino

Local onde será movido ou novo nome do arquivo/diretório.

opções

`-f, --force`

Substitui o arquivo de destino sem perguntar.

`-i, --interactive`

Pergunta antes de substituir. É o padrão.

`-v, --verbose`

Mostra os arquivos que estão sendo movidos.

`-u, --update`

Move somente arquivos antigos, ou novos arquivos.

O comando `mv` copia um arquivo da *ORIGEM* para o *DESTINO* (semelhante ao `cp`), mas após a cópia, o arquivo de *ORIGEM* é apagado.

Exemplos:

```
mv teste.txt teste1.txt
```

Muda o nome do arquivo `teste.txt` para `teste1.txt`.

```
mv teste.txt /tmp
```

Move o arquivo `teste.txt` para `/tmp`. Lembre-se que o arquivo de origem é apagado após ser movido.

```
mv teste.txt teste.new (supondo que teste.new já exista)
```

Copia o arquivo teste.txt por cima de teste.new e apaga teste.txt após terminar a cópia.