

Sintaxes

Antes de abordarmos os primeiros comandos, vamos entender como a sintaxe destes é apresentada nas documentações existentes. Uma sintaxe de comando explica a forma geral de um comando – seu nome, os argumentos que recebe e opções. Vamos analisar então a sintaxe abaixo:

<nome do comando> {a,b,c} [<opções>] [<argumentos>]

Os símbolos < >, [], { } são usados apenas para explicar a sintaxe dos comandos, seguindo uma convenção.

< > - constitui uma descrição do que deve ser digitado;

{ } - As chaves ({ e }) indicam os valores que podem ser usados em determinado campo do comando

[] - Os colchetes ([e]) servem para informar que todo o conteúdo entre eles é opcional, isto é, pode ser omitido.

Conforme os comandos forem sendo apresentados essas regras ficarão mais claras.

Quem sou eu? Onde estou?

O comando `whoami` mostra o nome do usuário associado com a conta em uso:

```
$ whoami
```

O comando `hostname` informa o nome da máquina que o usuário está operando:

```
$ hostname
```

O comando `groups` é usado para mostrar os grupos de usuários aos quais este usuário pertence:

```
$ groups
```

Para obter várias informações sobre a conta de um usuário qualquer – com exceção, evidentemente, da senha – usa-se o comando `finger`. Este comando requer como parâmetro o login de um usuário.

```
$ finger (nome do usuário)
```

O comando `who` mostra todos os usuários que estão usando o sistema no momento:

```
$ who
```

Permissões

Todo arquivo tem, necessariamente, um nome e um conjunto de dados. O Linux associa ainda a cada arquivo algumas outras informações chamadas atributos de arquivo. Entre estes atributos estão o tipo do arquivo e as permissões de acesso a ele. Tais permissões constituem um mecanismo de segurança necessário devido a existência de vários usuários utilizando o mesmo sistema.

O sistema de arquivos do Linux permite restringir a determinados usuários as formas de acesso a arquivos e diretórios. Assim, a cada arquivo ou diretório é associado um conjunto de permissões. Essas permissões determinam quais usuários podem ler, escrever (alterar) ou executar um arquivo (no caso de arquivos executáveis como programas). Se um usuário tem permissão de execução para determinado diretório, significa que ele pode realizar buscas dentro dele, e não executa-lo como se fosse programa.

As Permissões de acesso aos arquivos podem ser vistas através do comando "ls -l". Ao executa-lo, podemos notar várias colunas de informações. Por exemplo:

```
§ ls -l
```

```
drwxrwxr-x 3 joao estudantes 1024 Jul 14 17:59 Documentos
```

O primeiro destes é uma indicação de tipo: se for um "-" indica um arquivo; se for um "d", um diretório; se for um "l", uma ligação (link).

Os nove caracteres em seguida representam as permissões de acesso e são divididos em três grupos com três caracteres consecutivos. O primeiro, segundo e terceiro grupos constituem, respectivamente, as permissões de acesso do dono do arquivo, do grupo ao qual pertence, e de todos os outros usuários do sistema.

O primeiro caractere de cada grupo, indica a presença (r) ou ausência (-) de permissão de leitura.

O segundo caractere de cada grupo, indica a presença (w) ou ausência (-) de permissão de escrita (alteração).

O terceiro caractere de cada grupo (x), indica permissão de execução para o arquivo – nesse caso, um arquivo de programa – ou indica permissão de busca em se tratando de um diretório (permissão de realizar um comando cd para esse diretório).

Alterando as Permissões

Somente o dono e o administrador do sistema podem alterar as permissões de arquivos e diretórios. O

comando `chmod` é usado para isto, `Chmod {7,7,7}<nome do arquivo ou diretório>`.

Este método é baseado em números octais (base 8) que fornecem um código de três dígitos para especificar todas as permissões (dono, grupo e outros). Se representarmos a ausência de uma permissão com um "0"(zero) e a presença desta com um "1".

Permissão	Binario	Octal
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwX	111	7

Manipulando diretórios

O comando pwd sempre informa o nome do diretório de trabalho:

```
$ pwd
```

Para listar o conteúdo de um diretório (mostrar os nomes dos arquivos e diretórios nele contidos) usamos o comando ls (list):

```
$ ls {-l,-a}
```

Para mudar de diretório, movendo-se na árvore de diretórios, utilizamos o comando cd:

```
$ cd <nome do novo diretório de trabalho>
```

A cópia de arquivos é feita usando o comando cp, cuja sintaxe é:

```
$ cp <arquivo original><arquivo copia>
```

O comando mv possui a mesma sintaxe do comando cp mas ao invés de copiar arquivos é utilizado para movê-los de um diretório para outro:

```
$ mv <arquivo original><arquivo copia>
```

Para remover arquivos ou diretórios e utilizado o comando rm:

```
$ rm [opções] [arquivo]
```

Antes de usar esse comando é importante se certificar da operação, ao qual vai ser executada. Sempre deve-se ter bastante cuidado para não fazer a remoção de um arquivo errado. No caso da remoção de um diretório, juntamente com todo seu conteúdo, deve-se utilizar a opção "-r":

```
$ rm -r [diretório]
```

Acesso remoto

Para acesso remoto a contas utiliza-se o comando ssh, esse comando é muito útil para acessar contas cujo o limite de quota foi extrapolada, não permitindo assim o login e quando o usuário retira por engano todas as permissões de sua conta por engano.

```
ssh <nome da conta>@localhost
```