

Linux e serviços de rede

[Conceitos sobre linux](#)

[Diretórios do linux](#)

[Caminhos absolutos e caminhos relativos](#)

[Comandos linux](#)

[Arquivos](#)

[comandos 2](#)

[Um pouco sobre home do usuário](#)

[O caractere coringa: *](#)

[O caractere ponto: .](#)

[Comandos 3: Cat, Tac, More, Head e Tal](#)

[Uso do Pipe \(| \) como um combinador de comandos](#)

Conceitos e comandos no linux

Conceitos sobre linux

Linux é o nome que se dá para o Kernel ou núcleo de um sistema operacional, o kernel é responsável por conectar o hardware com o software.

Foi criado por Linus Torvalds em 1991 e é mantido por uma comunidade e empresas que criam sua própria versão do Linux.

Linux adota a licença livre GPL gnu.

Uma distribuição linux é o nome que se dá para um conjunto de software que contém o kernel linux mais as personalizações de cada distribuidor. Por exemplo, no site distrowatch existe uma lista que mostra quais as distribuições existem. Existem muitas distribuições!

Existem distribuições para usuários comuns e distros para servidores!

Diretórios do linux

/bin - Arquivos binários de comandos essenciais do sistema.

/boot - Arquivos de boot (inicialização; boot-loader; Grub); kernel do Linux.

/dev - Dispositivos (devices) de entrada/saída: floppy, hardisk, cdrom, modem.

/etc - Scripts de configuração e inicialização.

/home - Diretório de usuários do sistema!

/root - Diretório local do superusuário (root)

/usr - Arquivos de usuários nativos da distribuição.

/usr/local - Para instalação de programas não oficiais da distribuição.

/usr/src - Arquivos fontes do sistema necessários para compilar o kernel

/sbin - Arquivos de sistema essenciais (binários do superusuário)

/lib - Biblioteca e módulos do sistema.

/mnt - Diretório de montagem de dispositivos, sistemas de arquivos e partição.

/opt - Para instalação de programas não oficiais da distribuição.

/proc - Diretório virtual (RAM) onde rodam os processos ativos.

/tmp - Arquivos temporários gerados por alguns utilitários.

/var - Arquivos de log e outros arquivos variáveis.

Caminhos absolutos e caminhos relativos

Quando não colocamos barra na frente para indicar um diretório, estamos indicando o caminho relativo ao diretório atual. Isso é o **caminho relativo**. ex: `cd home` (relativo ao caminho em que o usuário se encontra)

Quando colocamos barra na frente do diretório, estamos indicando o SO que queremos partir da raiz, ou seja, do diretório barra. Ex: `cd /home` (refere-se à pasta home lá no diretório raiz)

Comandos linux

pwd - Exibe o diretório atual (print working directory)

cd \$nome_do_diretorio - Migra para o diretório (change directory)

cd - Migra para o diretório do home do usuário.

cd .. - Migra para o diretório anterior.

man \$nome_do_comando - Exibe as opções de utilização de um comando.

Ex: `man ls`, `man cd`, `man pwd`

ls - Lista o conteúdo do diretório atual

ls \$caminho_diretorio - lista os arquivos do diretório do caminho

ls -a - Lista todos os arquivos, incluindo os arquivos ocultos.

ls -l - Lista com mais detalhes

ls -lh - Lista em formato humano, nesse caso o lh é uma combinação de dois modos de usar o ls, com o l e com o h juntos, **é possível combinar**

argumentos!

mkdir \$nome_da_pasta - cria uma pasta

mkdir -p \$nome_da_pasta/nome_da_outra - Cria diretórios seguindo o caminho indicado!

rmdir \$nome_do_diretorio - remove o diretório

rmdir -p \$nome1/nome2/nome3 - Remove todas as pastas do caminho

sudo su - Para acessar o usuário root, usuário root tem permissão máxima.

Esse comando só ira ser possível para usuários que tem a permissão para ser um usuário root.

exit - sai do usuário root

sudo passwd root - Se desejar criar senha de root no Ubuntu

Arquivos

No linux, tudo é arquivo. Ou seja, tudo é representado no sistema de arquivos do linux.

Os arquivos são case sensitive.

comandos 2

file \$nome_do_arquivo - determina o tipo daquele arquivo, se é diretório, arquivo binário, arquivo de texto etc.

touch \$nome_do_arquivo - Cria um arquivo.

rm \$nome_arquivo - Remove um arquivo.

rm -i \$nome_arquivo - remoção segura.

rm -rf \$nome_arquivo_ou_diretório - remove tudo de maneira recursiva, **r** de recursive **f** de force.

cp \$arquivo_origem \$arquivo_destino - Faz uma cópia do arquivo de origem com o nome do arquivo de destino

cp -r \$dir_origem \$dir_destino - faz uma cópia recursiva do diretório_origem com nome diretório_destino.

cp -i \$arq_origem \$arq_destino - Cópia em modo seguro

mv \$arq1 \$arq2 - Move o arquivo 1 com o nome do arquivo dois, **se isso acontecer no mesmo diretório, o arquivo 1 será substituído pelo arquivo 2**, esse comando é usado muitas vezes para renomear um arquivo.

Um pouco sobre home do usuário

A pasta home contém todos os usuários do sistema, cada usuário tem acesso a sua pasta, mas não à pasta de outros usuários.

Quando acessamos a nossa pasta home um **sinal de ~** aparece no terminal, isso quer dizer que estamos no nosso home do usuário.

Exemplo de home de um usuário que se chama **cleiton**: **~ /home/cleiton/**

Dentro da pasta do cleiton tem todos os arquivos dele.

O caractere coringa: *

O caractere ***** serve como um indicador para todos os arquivos de uma pasta.

Se um usuário quiser copiar todos os arquivos de uma pasta para outro ele faria: `cp teste/* teste2/`

Outra maneira de usar: **`rm -rf *png`** - Remove todos os arquivos png da pasta.

O caractere ponto: .

O caractere ponto (.) diz respeito ao seu diretório atual, isso pode ser usado para copiar arquivos, por exemplo: **`cp /teste2/* .`**

O comando acima copia os arquivos da pasta teste2 para o diretório em que o usuário está

Comandos 3: Cat, Tac, More, Head e Tail

São comandos para manipular o conteúdo dentro dos arquivos.

`cat $nome_do_arquivo` - Exibe o conteúdo dentro do arquivo.

`cat > $nome_do_arquivo` - Se o arquivo não existir, um novo é criado. Se o arquivo já existir, ele é substituído por um novo.

`cat >> $nome_do_arquivo` - Se o arquivo não existir, cria um novo. Se existir, permite que o usuário insira um novo conteúdo no final do arquivo.

`cat $nome_do_arquivo > $nome_de_outro_arquivo` - Copia o conteúdo de um arquivo para outro. Se não existir, é criado. Se existir, o conteúdo é sobrescrito.

cat \$nome_do_arquivo >> \$nome_de_outro_arquivo - Nesse caso, o princípio é o mesmo. Com a diferença que o conteúdo está sendo adicionado no fim, e não sobrescrevendo o arquivo.

tac \$nome_do_arquivo - Mostra o conteúdo do arquivo de trás para frente.

more \$nome_arquivo - Útil para mostrar arquivos muito grandes.

head \$nome_do_arquivo - Mostra as 10 primeiras linhas do arquivo

head -n \$nome_do_arquivo - Mostra a N primeiras linhas do arquivo.

head -c \$n \$nome_do_arquivo - Mostra os n primeiros bytes de arquivo.

tail \$nome_do_arquivo - Mostra as 10 últimas linhas do arquivo.

tail -n \$nome_do_arquivo - Mostra as N últimas linhas do arquivo

tail -f \$nome_do_arquivo - Monitora constantemente as últimas linhas do arquivo.

grep \$string \$arquivo - Filtra as linhas que possuem a string passada. Mostra linha inteira, mas apenas as que possuem essa string que foi passada no comando

sort \$arquivo - Apresenta as linhas do arquivo em ordem alfabética.

wc -l \$nome_arquivo - Conta quantas linhas existem no arquivo e exibe na tela. **O -w conta o número de palavras.**

Uso do Pipe (|) como um combinador de comandos

Podemos utilizar o pipe para utilizar mais de um comando por vez.

Exemplo: ls | sort

Primeiro, o SO executa o comando ls. O resultado do comando ls é mandado para o comando sort, que realiza a ordenação.

Nesse caso o resultado seria um ls ordenado por ordem alfabética.

Comandos relacionados com usuários

adduser - Cria um novo usuário (é preciso ser root).

deluser - Remove um usuário. Pode ou não conter os argumentos :

--remove-home e --remove-all-files.

groupadd \$nome_do_grupo - Cria um grupo

groupdel \$nome_do_grupo - Deleta o grupo

adduser \$usuário \$grupo - Adiciona um usuário a um grupo

groups - Mostra os grupos aos quais o usuário atual pertence.

Comando usermod

O comando usermod serve para alterar informações do usuário.

Normalmente este comando vem acompanhado do comando sudo, para que seja possível realizar alterações.

Existem diversas letras que servem de parâmetros para a alteração das informações.

- -c comentário:
- -d diretório_home
- -e data_de_expiração data, na forma YYYY-MM-DD, que a conta será desativada
- -g grupo primário
- -G grupo grupo(s) suplementar(es)
- -l novo_login login
- -L trava a senha
- -s shell
- -U destrava a senha

Exemplo de utilização para adicionar o usuário caique3 em um grupo:
sudo usermod -g adm caique3.

su \$nome_do_usuario - Muda de usuário.

exit - Volta ao usuário anterior

passwd - Alterar a senha do usuario atual

passwd \$nome_usuario - Altera a senha de um usuario (precisa estar com o usuário root).

whoami - Exibe o nome do meu usuário (quem sou eu?)

who - mostra os usuário logados no sistema

w - mostra quem está logado e o que está fazendo no sistema

Permissionamento

Todo arquivo possui um dono e um grupo. É possível verificar isso com o `ls -l`.

continua..