Linux Commands

Tiago Heinrich

UniSociesc Joinville

12/03/2020

Editor

- **vi** editor de texto em modo texto
- **vim** *Vi IMproved*, um editor de texto para programadores
- **nano** editor de texto em modo texto
- **emacs** ...

Commands

- A maioria dos comandos aceita um conjunto de variáveis de entrada.
- **man** uma interface para os manuais de referência do sistema
- $ **Comando -opção -opção -opção**

Commands

- ls
- cd
- mkdir
- cp
- mv
- rm

- echo
- cat
- grep
- wget
- adduser

Commands

- **ls** Mosta o conteúdo da pasta
- ls -a Mostra os arquivos ocultos.
- ls -l Mostra os arquivos e suas permissões, tamanho, etc.
- ls -lh Igual ao -l, porem mostra o tamanho num formato “humano”

Commands

- **cd** Muda o diretório
- cd .. Vai um diretório acima
- cd - Volta pro diretório anterior
- cd .. Retorna para o diretório anterior
- cd . Referencia o diretório atual
- **mkdir nome_da_pasta** Faz um diretório
- mkdir .folder ou **touch** .file.txt Cria um folder/arquivo oculto

Commands

- **cp** Copia arquivos e pastas.
- cp -r Modo recursivo, copia a pasta e seu conteúdo.
- cp -v Verbose, mostra o que esta sendo feito

Commands

- **mv** - Move arquivos e pastas.
- mv -v Verbose, mostra o que esta sendo feito

Commands

- **rm** - Remove arquivos e pastas.
- rm -r Mode recursivo, remove pastas e seu conteúdo.
- rm -v Verbose, mostra o que esta sendo feito

Commands

Redirecionadores:

- Joga a saída para o arquivo. Ex: ls > arquivo.txt
- Joga o arquivo para algum comando. Ex: grep hello < arquivo.txt
- Anexa a saída no final do arquivo. Ex: ls » arquivo.txt

Commands

Redirecionadores:

- | PIPE joga a saída de um comando para o próximo. Exemplo: ls | grep algo.
- Irá mostrar apenas as ocorrências com “algo”.

Commands

- **echo** Mostra um texto na tela(que você digita). Útil para acrescentar informações a um arquivo: Ex: echo "teste" ≫ texto.txt
- **cat** Mostra na tela o conteúdo de um arquivo. Ex: cat texto.txt teste

Commands

- **grep** pesquisa qualquer arquivo de entrada, selecionando linhas que correspondem a um ou mais padrões . Exemplo dmesg | grep Error
- **wget** é um utilitário gratuito para download não interativo de arquivos da Web. Suporta protocolos HTTP, HTTPS e FTP. Exemplo: wget <url>
- **adduser** adicionar usuário ou grupo ao sistema

Operações básicas

- ls *.log lista todos os arquivos do formato .log
- ls */*.mp3 lista todos os arquivos .mp3 encontrados dentro de qualquer folder
- ls file-**{1,2,3}**.txt lista os arquivos file-1.txt, file-2.txt e file-3.txt
- touch file-**{1..10}**.txt cria dez arquivos em brando file-1.txt, file-2.txt, ... e file-10.txt
- ls -l **;** ls -a lista o conteúdo e em seguida lista conteúdos ocultos

SSH

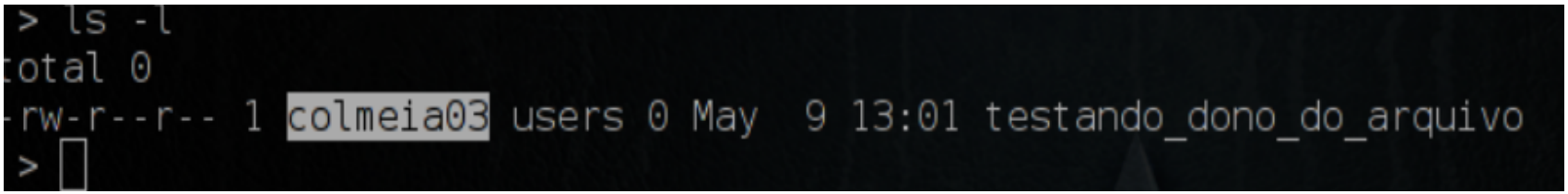
- O **ssh** (Secure Shell) é um programa para efetuar login em uma máquina remota e para executar comandos em uma máquina remota.
- Destina-se a fornecer comunicações criptografadas seguras entre dois hosts não confiáveis em uma rede insegura.
- **ssh nome@address**

Permissão de acesso a arquivos e diretório

- Impede acesso não autorizado de pessoa ou programa
- Manipulando as permissões você pode restringir acesso para:
- **Arquivos**: Leitura, Escrita e Execução
- **Diretórios**: Leitura/Listar e Escrita

Dono, grupos e outros

- No Linux a idéia básica é definir o acesso aos arquivos por: **Dono**, **Grupos** e **Outros usuários**



Tipos de Permissões

- Existem 3 tipos básicos de permissões: **r**, **w** e **x**
  - r - Permissão de leitura para arquivos. Caso for um diretório, permite listar seu conteúdo (através do comando ls, por exemplo).
  - w - Permissão de gravação para arquivos. Caso for um diretório, permite a gravação de arquivos ou outros diretórios dentro dele. Para que um arquivo/diretório possa ser apagado, é necessário o acesso a gravação.
  - x - Permite executar um arquivo (caso seja um programa executável). Caso seja um diretório, permite que seja acessado através do comando cd

Tipos de Permissões

- **-rwxr-xr–** Dono Grupo Outro
- A primeira letra diz qual é o tipo do arquivo:
  - Caso tiver um **d** é um diretório,
  - um **l** um link a um arquivo no sistema (veja comando ln)
  - um - quer dizer que é um arquivo comum

Tipos de Permissões

- **-rwxr-xr–** Dono Grupo Outro
- Da segunda a quarta letra (rwx) dizem qual é a permissão de acesso ao dono do arquivo. Neste caso Dono tem a permissão de:
  - ler (r - read)
  - gravar (w - write)
  - executar (x - execute)

Tipos de Permissões

- **-rwxr-xr–** Dono Grupo Outro
- Da quinta a sétima letra (r-x) diz qual é a permissão de acesso ao grupo do arquivo. Neste caso todos os usuários que pertencem ao grupo Grupo tem a:
  - ler (r - read)
  - executar (x - execute)

Tipos de Permissões

- **-rwxr-xr–** Dono Grupo Outro
- Da oitava a décima letra (r–) diz qual é a permissão de acesso para os outros usuários. Neste caso todos os usuários que não são donos do arquivo teste tem a permissão somente para:
  - ler (r - read)

Alterando o dono

- **chown** [OPTION]... [OWNER][:[GROUP]] FILE
- Pode alterar dono e grupo do arquivo/diretório.
- chown -R userteste.users pastateste
- -R, –recursive : Altera dono e grupo de arquivos no diretório atual e sub-diretórios.

Alterando o grupo

- **chgrp** [OPTION]... GROUP FILE
- Pode alterar grupo do arquivo/diretório.
- chgrp -R users arquivoteste

Alterando as permissões

- **chmod** [OPTION]... MODE[,MODE]... FILE...
- chmod ugoa+-=rwx arquivo/diretório
- Controla que nível de acesso será mudado, especificam em ordem.
  - u: define que as regras serão aplicadas ao usuário
  - g: define que as regras serão aplicadas ao grupo
  - o: define que as regras serão aplicadas aos outros usuários do sistema
  - a: define que as regras serão aplicadas a todos
  - +: adiciona permissão
  - -: remove permissão
  - =: informa que a permissão aplicada deve ser exatamente igual a que será indicada a seguir
  - r: atribui a permissão de leitura
  - w: atribui a permissão de escrita
  - x: atribui a permissão de execução

Exemplos de permissões de acesso

- chmod g+r * Permite que todos os usuários que pertençam ao grupo dos arquivos (g) tenham (+) permissões de leitura (r) em todos os arquivos do diretório atual.
- chmod uo+x teste.txt Inclui (+) a permissão de execução do arquivo teste.txt para o dono e outros usuários do arquivo.
- chmod a=rw teste.txt Define a permissão de todos os usuários exatamente (=) para leitura e gravação do arquivo teste.txt.

Alterando as permissões em modo octal

- Especificar as permissões através de números
- De 0 (zero) à 7 (sete), 1 = Executar, 2 = Gravar e 4 = Ler
  - 0 - Nenhuma permissão de acesso (-).
  - 1 - Permissão de execução (x).
  - 2 - Permissão de gravação (w).
  - 3 - Permissão de gravação e execução (wx). == (2+1)
  - 4 - Permissão de leitura (r).
  - 5 - Permissão de leitura e execução (rx). == (4+1)
  - 6 - Permissão de leitura e gravação (rw). == (4+2)
  - 7 - Permissão de leitura, gravação e execução.
  - Equivalente +rwx e 4+2+1.

Alterando as permissões em modo octal

- chmod 764 teste.txt : dono (7), grupo (6) e outros usuários (4) (-rwxrw-r–)
- chmod 751 teste.txt : dono (7), grupo (5) e outros usuários (1) (-rwxr-x–x)
- Qual o problema em **chmod 777 teste.txt**?

sudo

- **su** executar um comando como outro usuário. Exemplo su - tiago
- Você gosta de ter um sistema seguro? Então evite usar o usuário root
- su - root ou sudo su

Linux

- `https://geekflare.com/run-linux-from-a-web-browser/` (JSLinux ou copy.sh)
- Online Bash `https://repl.it/languages/bash` ou `https://codeinterview.io/`
- Regular expression (RE | regex) e Bash