# Linux CENTOS

## Kernel

Kernel não é um sistema operacional, mas sim, um software no qual lhe permite receber instruções de alto nível, e encaminhá-las ao hardware da máquina.

Além disso é responsável por fazer todo gerenciamento de tarefas que se encontram em execução na memória do computador, escalonamento etc.

## Gerenciador de Arquivos

É um sistema usado pelo sistema operacional para gerenciar os arquivos. O sistema controla como os arquivos são salvos ou recuperados.

O sistema operacional irá organizar seus arquivos em diretórios de forma organizada.

Linux File Systems: ext3, ext4, xfs.

Windows File Systems: NTFS, FAT.

## Descrição do gerenciador de arquivos

* **/boot** -> Arquivos de configuração usado quando a máquina é inicializada (grub.cfg).
* **/root** -> usuário root, não é o mesmo que diretório /, embora /, seja chamado de root também.
* **/dev (devices)** -> Disco, CD ROM, teclado e qualquer periférico terá um arquivo de texto nesse diretório.
* **/etc** -> arquivos de configuração.
* **/bin (usr/bin) interpretador** -> Os comandos Linux que os usuários usam a todo momento ficam nesse diretório.
* **/sbin (usr/sbin)** -> Comandos para o gerenciador de arquivos.
* **/opt** -> Diretório para adicionar aplicações de terceiros, como o cliente da Oracle, por exemplo, qualquer aplicação não nativa do Linux, o esperado é estar nesse diretório.
* **/proc** -> Todos os programas em execução, ou seja, em memória possuem arquivos nesse diretório.
* **/lib (usr/lib)** -> Diretório com bibliotecas escritas em linguagem C para a execução de aplicações. Caso queira visualizar qual biblioteca cada comando usa, digite em seu terminal: **strace -e open command**
* **/tmp ->** Diretório para arquivos temporários.
* **/home ->** Diretório para usuários
* **/var ->** Sistema de logs**.**
* **/run ->** Diretório de daemons, programas que iniciam quando o sistema começa a executar, daemons são executados em segundo plano.
* **/mnt ->** Executar sistemas de arquivos externos
* /**media** -> Ao adicionar um CD ROM na bandeja do comutador e executá-lo, o CD ROM será inicializado sob o diretório /media

## Listar e mostrar diretório atual

## Texto Descrição gerada automaticamente com confiança média

## Comandos:

* Ls: listing
* Pwd: Print working directory

A esquerda da imagem, podemos ver algumas sequencias de letras, algumas iniciando com a letra ‘d’, outras com ‘-’. D significa, diretório ou pasta, enquanto -, significa arquivo.

## Propriedades de arquivos ou diretórios Linux

Tabela

Descrição gerada automaticamente

Ao iniciar com ‘L’, significa ‘link’ para outro arquivo.

Tabela

Descrição gerada automaticamente

## O que é *root*?

Existem 3 tipos de *root*.

1. *Root account*: uma conta na qual tem acesso a todos comandos e diretórios na máquina, seria o administrador
2. *Root as /*: O primeiro diretório é chamado de diretório root
3. *Root as home directory*: A conta root no diretório /root, tem um diretório chamado /home, quando ouvir, diretório root home directory é esse.

## Mudar senha

Se você estiver logado em sua conta sem ser o root e quiser mudar a senha basta digitar:

* **passwd**

Se você estiver em sua conta root e quiser mudar a senha de alguma outra conta, basta digitar:

* **passwd username**

E caso queira trocar a senha da conta root, acesse ela primeiro e digite:

* **passwd**

## Criando arquivos e diretórios

* Criando arquivos
  + Touch: cria um arquivo vazio
  + cp: copiando um arquivo existente
  + vi: editor de textos
* Criando diretórios
  + Mkdir: make directory.

Texto

Descrição gerada automaticamente

Criando arquivos vazios.

Texto

Descrição gerada automaticamente

Copiando arquivos.

## Copiando diretórios

Comando a executar:

* Cp -R <pasta origem> <pasta destino>
* -R: copiar de forma recursiva

Texto

Descrição gerada automaticamente

## Encontrar arquivos e diretórios

Existem duas maneiras de se encontrar arquivos e diretórios em um sistema Linux:

* **Find**
* **Locate**

Tabela

Descrição gerada automaticamente

Com o comando *find* e usando a flag -*name*, é necessário passar o nome exato do arquivo buscado e a extensão, olhando o manual desse comando existe uma flag -*regex* e ajuda bastante:

Texto preto sobre fundo branco

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente com confiança média

Usando o comando *locate*, esse comando ao contrário do *find* que faz uma busca em cada diretório em tempo real, ele busca em um arquivo, podendo ser atualizado com o comando ***sudo updatedb***, fazendo com que ele seja mais rápido.

Texto

Descrição gerada automaticamente

Caso não tenha o locate instalado execute o comando com seu gerenciador de pacotes, em meu caso centos, é o yum.

* **Sudo yum install mlocate**

## WildCards

Wildcard é um caractere que pode ser usado como substituto para qualquer classe de caracteres em uma busca.

* \* -> representa 0 ou mais caracteres
* ? -> representa um único caractere, exemplo -> “te?t” >> “text”
* [] -> representa uma faixa de busca

Por exemplo, vamos criar 9 arquivos e começar a buscar por eles:

Tabela

Descrição gerada automaticamente

Usando ? e \*:

Texto

Descrição gerada automaticamente

Estamos procurando qualquer arquivo que contenha qualquer caractere na primeira posição, depois um bcd e pode haver 0 ou mais caracteres depois.

Usando faixa para buscar:

Texto

Descrição gerada automaticamente

## Soft e hard link

## Antes de explicar sobre soft e hard link, é necessário dizer que ao criar um arquivo, embora visualizemos o nome, para o computador ele é um endereço de memória, e esse ponteiro ou número é chamado de *INODE*.

## **Soft link**: Ao criar um arquivo, em outro diretório podemos criar outro arquivo e fazer um link entre eles, ou seja, fazer com que ambos apontem para o mesmo endereço de memória, ao mudar, deletar, renomear um acontece com o outro, é a passagem de valor conhecido em linguagens de programação.

## Hard Link: Arquivos independentes, ao deletar, renomear, mover, um arquivo não afeta o outro, pois cada um aponta para o seu endereço de memória.

Para criar um soft link basta usar o comando:

* Ln -s

Para criar um hard link basta usar o comando:

* Ln

Para o exemplo, criei um arquivo na home de meu usuário e fiz um soft link em outro diretório para esse arquivo.

**Obs**: Não podermos gerar links no mesmo diretório com o mesmo nome de arquivo.

Texto, Carta

Descrição gerada automaticamente

Podemos ver que é um link, pois, a cadeia de caracteres a esquerda inicia com “l”.

Aqui modifiquei o arquivo em /home/ e afetou o arquivo link em /etc:  
Texto

Descrição gerada automaticamente

Para o disco, são arquivos “distintos”, porém ao realizar a ligação, tudo que fizer em um acontece com o outro. Se remover, o outro arquivo não irá “funcionar” mais, mas se criar um hard link, e o arquivo de link ser removido, o outro arquivo irá continuar existindo.

## Permissões de arquivos e diretórios

## Unix é um sistema multiusuários, logo, cada arquivo e diretório pode ser protegido pelo usuário, ou até mesmo torná-lo acessível para outro usuário.

A restrição de arquivos é baseada por tipos de permissão, existem 3 tipos:

* R – Para leitura
* W – Para escrita
* X – Permissão para executar

Cada permissão pode ser controlada em 3 níveis:

* U – user = você mesmo
* G – group = um grupo de pessoas no mesmo projeto
* O – Other = qualquer um no sistema

Para visualizar as permissões de diretórios e de arquivos, basta digitar o comando ls -l:



Na imagem acima, no primeiro bit, podemos visualizar um “-” indicando arquivo.

* os próximos 3 bits indicam permissões de leitura, escrita e execução para o **usuário**
* os próximos 3 bits indicam permissões de leitura, escrita e execução para o **grupo**
* os próximos 3 bits indicam permissões de leitura, escrita e execução **para qualquer outro no sistema**

Para mudar a permissão existe o comando **chmod:**

Texto

Descrição gerada automaticamente

Texto preto sobre fundo branco

Descrição gerada automaticamente

Esse X nos diretórios indica que posso executá-lo?

Tabela

Descrição gerada automaticamente

Não, indica que podemos entrar nessa pasta, usando por exemplo o comando *cd.*

Chmod usando números:

Diagrama

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

## Proprietário de arquivos

Cada arquivo e diretório possui seus proprietários, usuários que possuem permissões para realizar algo, como ler, escrever e executar algum arquivo, para configurarmos isso, existem dois comandos: ***chown*** e ***chgrp***.

Vou criar um arquivo na pasta de meu usuário e mudar minha permissão usando o usuário root.

## Texto Descrição gerada automaticamente com confiança baixa

Criei um arquivo no diretório do usuário chamado lisa, como podemos ver o proprietário e o usuário no grupo que pode mexer é o meu usuário, posso ler e escrever. Vou mudar isso para permitir somente que o usuário root possa fazer.

## Se você não for super usuário, não terá permissão para mudar.

## Texto Descrição gerada automaticamente

## Texto preto sobre fundo branco Descrição gerada automaticamente

## Pergunta? Coloquei que somente o usuário root pode usar alterar e ler e executar esse arquivo, caso eu esteja com meu usuário local, posso removê-lo?

Tabela

Descrição gerada automaticamente

Consegui remover. Por quê?

Uma imagem contendo Gráfico

Descrição gerada automaticamente

Porque a pasta de meu usuário, tem permissão total, e essa permissão engloba qualquer outro diretório e arquivo dentro dele. Caso eu tente remover um arquivo, no qual o diretório pertença ao root ai sim não consigo.

Tabela

Descrição gerada automaticamente

Logo precisaria mexer na permissão do diretório para que minhas mudanças surtam efeito, também posso usar a flag -R nesses comandos para mudar o proprietário de forma recursiva.

## Adicionando textos em arquivos

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Eu adicionei uma linha usando o comando echo to “colocar para fora” e um sinal de maior para atribuir “>”. O que aconteceria se eu usasse echo novamente > nome arquivo??? Iria sobrescrever o conteúdo anterior, para que não sobrescreva, preciso usar dois sinais de maior >>.

Tabela

Descrição gerada automaticamente

Como pode ser visto na imagem acima, caso o arquivo não exista, automaticamente ele é criado. Posso guardar em arquivos outputs de comandos, por exemplo guardar em um arquivo o histórico de comandos feitos por mim: history > history\_commands.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Sempre que digitamos algo, esse comando é do tipo standard input com retorno 0, quando realizando algum comando que retorna algo via terminal, isso seria o standard output, com retorno 1, e como já falado anteriormente podemos pegar esse output e coloca-lo em algum arquivo usando os sinais de redirecionamento, >, >>.

## Pipe

Pipe ( | ) é usado pelo shell para conectar o output de um comando com o input de outro comando.



Tabela

Descrição gerada automaticamente

Comandos Para manutenção de arquivos

Texto

Descrição gerada automaticamente com confiança média

Texto

Descrição gerada automaticamente com confiança baixa

Tabela

Descrição gerada automaticamente com confiança baixa

Texto

Descrição gerada automaticamente

## Comandos para processar textos

## Texto, Carta Descrição gerada automaticamente

### Cut

## Cut -c1 arquivo

## Listar a primeira letra de cada linha. É possível mudar o valor após a letra “c” indicando a posição da linha que quer listar.

## Interface gráfica do usuário, Texto, Aplicativo, Email Descrição gerada automaticamente

Listar caracteres definindo um range

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Também é possível listar arquivos baseados em linhas e colunas, usando os delimitadores para selecionar colunas especificas. Vamos usar o arquivo de usuários localizado em /etc/passwd.

Texto

Descrição gerada automaticamente

Cada ‘:’ separa uma coluna, agora vamos listar esse arquivo e mostrar somente a sexta coluna.



Texto

Descrição gerada automaticamente

Listando mais de uma coluna. No caso for adicionada somente mais uma coluna, pois, o arquivo possui somente 7 colunas e não 8 como mandei mostrar.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Cut também funciona em outputs de comandos:

Aplicativo

Descrição gerada automaticamente com confiança baixa

### AWK

AWK é um comando utilizado para extração de dados, na maioria das vezes é utilizado para extrais campos de arquivos ou de outputs.

Primeiro comando é para listar em um arquivo o primeiro campo de todas as linhas:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Usando awk para listar o primeiro e terceiro campo em um pipe:

Interface gráfica do usuário, Texto, Tabela

Descrição gerada automaticamente

Algo que testei, tentei usar o cut para retornar os Fields, porém o delimitador sempre deve ser um caractere e nesse caso é um espaço em branco, cut não funcionou.

Listar a última coluna:

Texto

Descrição gerada automaticamente com confiança média

Precisa ser maiúsculo, lembre-se que Linux é case sensitive.

Buscando uma palavra especifica com comando awk:



Substituindo um campo em uma string



Aqui eu substituo todos os sobrenomes para medina:

Gráfico, Texto

Descrição gerada automaticamente com confiança média

Procurando por um diretório específico em /home, na nona coluna, eu tenho algum diretório chamado seinfeld?

Tabela

Descrição gerada automaticamente

Número de campos em um arquivo, diretório etc.

Interface gráfica do usuário, Tabela

Descrição gerada automaticamente

Bibliografias:

<https://www.godaddy.com/resources/br/artigos/o-que-e-ssh-e-como-funciona>

<https://www.youtube.com/watch?v=sHanr1op-sQ>

<https://www.youtube.com/watch?v=7MQtSJ2TFyc>