

Aula 00

*Prefeitura Municipal de Vitória-ES
(Guarda Civil Municipal) Passo
Estratégico de Informática*

Autor:

Thiago Rodrigues Cavalcanti

10 de Fevereiro de 2024

SISTEMA OPERACIONAL LINUX. CONCEITOS DE ORGANIZAÇÃO E GERENCIAMENTO DE ARQUIVOS E PASTAS. EXECUÇÃO DE PROGRAMAS. PERMISSÕES DE ACESSO. IMPRESSÃO

Sumário

Análise Estatística	2
Roteiro de revisão e pontos do assunto que merecem destaque	2
Conceitos	2
Linux.....	3
Principais características	3
Distribuição X Versão.....	4
Interface gráfica (GUI).....	5
Linha de comando	6
Gerenciamento de pacotes.....	8
Estrutura de pastas.....	8
Comandos	11
Permissões de arquivos.....	14
Aposta estratégica	16
Questões estratégicas	18
Questionário de revisão e aperfeiçoamento.....	23
Perguntas	23
Perguntas com respostas.....	24



ANÁLISE ESTATÍSTICA

A análise estatística estará disponível a partir da próxima aula.

ROTEIRO DE REVISÃO E PONTOS DO ASSUNTO QUE MERECEM DESTAQUE

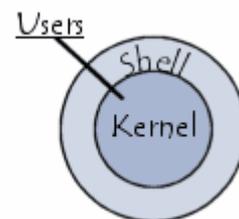
A ideia desta seção é apresentar um roteiro para que você realize uma revisão completa do assunto e, ao mesmo tempo, destacar aspectos do conteúdo que merecem atenção.

Para revisar e ficar bem preparado no assunto, você precisa, basicamente, seguir os passos a seguir:

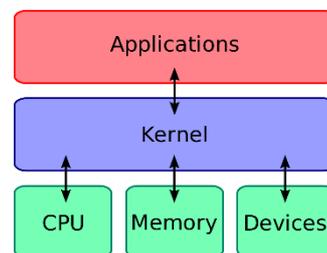
Conceitos

Antes de começarmos especificamente sobre Linux, vamos aprender algumas definições que o ajudarão a compreender melhor a aula. É importante prestar bastante atenção neste início, pois como o Linux pode não ser familiar, esses conceitos podem ajudar a esclarecer algumas dúvidas.

- Software Livre é uma expressão utilizada para definir qualquer programa que pode ser executado, copiado, modificado e redistribuído gratuitamente. Ou seja, o usuário possui livre acesso ao código-fonte para realizar alterações de acordo com suas necessidades. Cuidado para não confundir Software Livre e software gratuito. O software gratuito é distribuído em sua forma executável, não permitindo acesso ao código-fonte.
- Open Source (código aberto), apesar de ser uma expressão muito confundida com Software Livre, possui uma diferença sutil em relação a características como cópia, modificação e distribuição. Apesar de ser Open Source um programa pode não autorizar sua modificação, o que contraria o conceito de software livre.
- Shell é a interface que funciona como intermediário entre o usuário e o sistema operacional. A sua função é ler a linha de comando digitada pelo usuário, interpretar seu significado, executar o comando e devolver o resultado pelas saídas. Na verdade, a interface Shell é um arquivo executável, encarregado de interpretar comandos, transmiti-los ao sistema e devolver resultados. Por este motivo foi escolhido o nome Shell, que é um termo em inglês e significa concha.



- Kernel é um termo em inglês que traduzido significa núcleo. Ele é o componente central de um sistema operacional e funciona como ponte entre os aplicativos e o processamento real de dados feito a nível de hardware. As responsabilidades do núcleo incluem gerenciar os recursos do sistema (a comunicação entre componentes do hardware e software).



Linux

Linux, apesar de popularmente ser usado como termo para se referir a um sistema operacional, na verdade é o núcleo (**Kernel**) do sistema operacional. O termo correto para indicar o sistema operacional é **GNU/Linux**. Ele foi desenvolvido inicialmente pelo finlandês Linus Torvalds e lançado em 1991, baseado em outro sistema operacional, o **Unix**.

O Linux é **Software Livre** e **Open Source**. Apesar de não ser tão usado entre usuários domésticos, o Linux é muito popular entre os profissionais de T.I. e muito usado como sistema operacional para Servidores, onde faz frente ao seu principal concorrente, o Windows que é um software proprietário. O símbolo do Linux é este simpático pinguim que podemos observar na imagem ao lado.



Principais características

Entre as características do Linux temos:

- Licença de uso **GPL/GNU** (*General Public License* ou Licença Pública Geral).
- É um sistema **Case Sensitive** (onde o sistema diferencia letras maiúsculas de minúsculas).
- É um sistema **Multitarefa** (executa mais de um aplicativo ao mesmo tempo).
- É um sistema **Multiusuário** (vários usuários podem acessá-lo ao mesmo tempo).
- É um sistema **Preemptivo** (quando temos vários programas em execução e ao alternar entre os programas, o processo de um programa é interrompido para executar outro processo).
- Suporta **Multiprocessamento** (onde é possível ter mais de um processador).
- Suporta o compartilhamento de bibliotecas (onde os programas compartilham recursos do sistema operacional).
- Suporta processamento 32 e 64 bits.
- Possui tanto interface gráfica, quanto interface de linha de comando (Shell).
- Possui um superusuário (usuário administrador, com controle total sobre o sistema, chamado **root** e representado no Shell pelo símbolo #).
- Possui pelo menos um Usuário Comum (durante a instalação é necessário ser configurada pelo menos uma conta de usuário comum. Este usuário é representado no Shell pelo símbolo \$).



- Possui várias versões.

Das características apontadas acima, apenas as duas primeiras não são características comuns com o Windows.

Gerenciador de boot (inicialização)

É possível, em um mesmo, computador que o usuário tenha instalado dois sistemas operacionais instalados. O recurso é possível dividindo (particionando) o HD. Basta instalar o Linux em uma partição e o Windows na outra. Para isso existem softwares, gerenciadores de boot, que terão a função de gerenciar o carregamento do sistema que o usuário escolher ao ligar o computador. Os gerenciadores de boot mais usados no Linux são: LILO e o GRUB.

Distribuição X Versão

No Linux é usado o termo distribuição (distro) para indicar uma versão. É importante entender que toda distribuição é uma versão, mas nem toda versão é uma distribuição, pois existem versões das distribuições. Por exemplo: Ubuntu 20.04 (onde a distribuição Ubuntu está na versão 20.04).

Vamos entender melhor a diferença entre distribuição e versão. Existem várias distribuições: Debian, Red Hat, Ubuntu... Essas distribuições são versões que possuem como sistema base o Linux. Por isso, essas distribuições são diferentes versões do mesmo sistema base. Cada uma com seu formato, aparência, estilo e configurações.

Agora nem toda versão é uma distribuição, porque por exemplo o Debian está na versão 11.2, já o Ubuntu está na versão 20.041, ou seja, estas versões não são distribuições do Linux, mas sim versões de um mesmo sistema. Neste caso, podemos comparar a versão a uma atualização para melhorias do sistema.

Atualmente existem mais de mil distribuições do Linux, cada uma com seu conjunto de softwares e configurações. As distribuições podem ser comerciais, onde o usuário paga pelo sistema e recebe um suporte técnico; ou não comerciais, onde não há qualquer cobrança pelo sistema, basta o usuário realizar o download na Internet. Na imagem abaixo temos algumas dessas distribuições e suas imagens registradas.

¹ Informações atualizadas em março/2022.





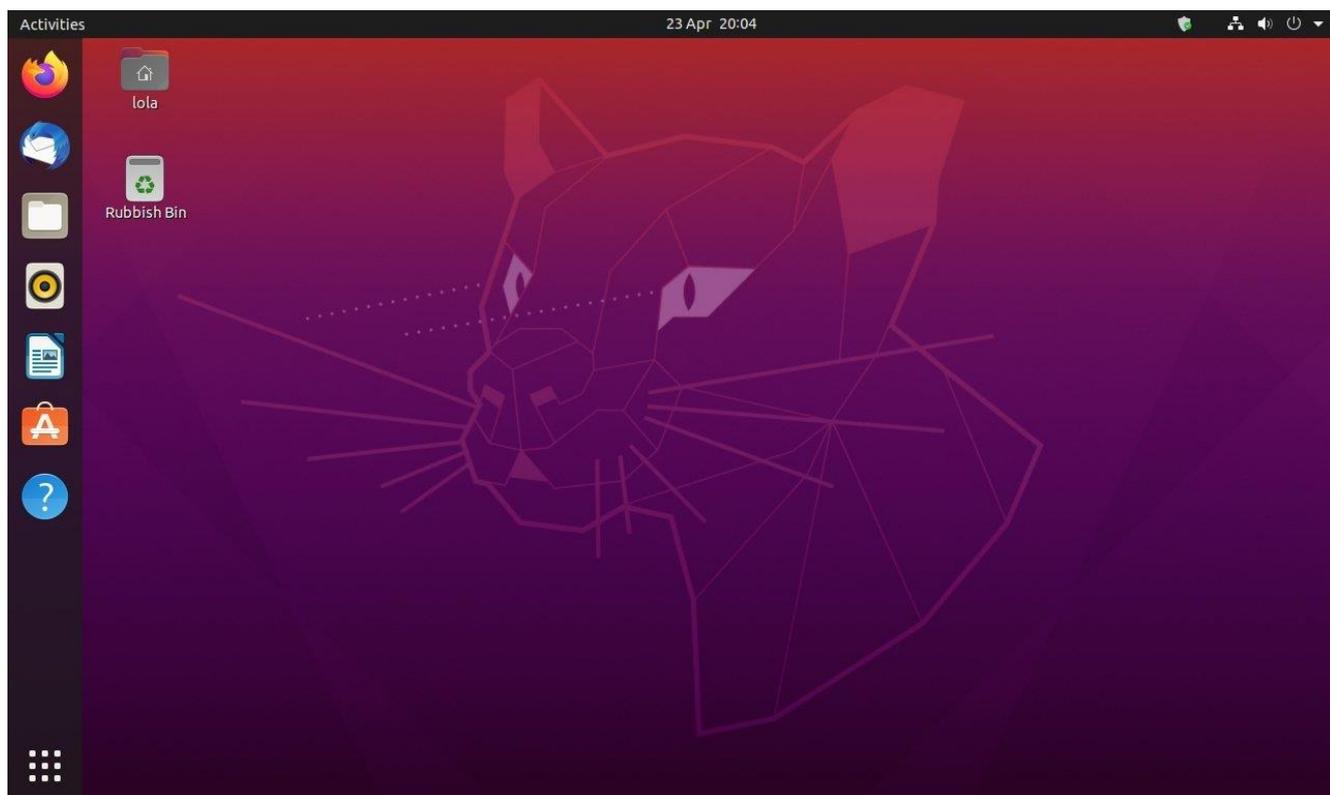
Interface gráfica (GUI)

Vimos nas características que o Linux possui duas formas de interação com o usuário, através da interface gráfica e através da linha de comando. A interface gráfica é a forma mais atrativa como o Linux se apresenta.

Além das várias distribuições do sistema operacional, existem várias interfaces gráficas que podem ser utilizadas nas versões. Por exemplo: o Ubuntu possui as interfaces Gnome e KDE, o Debian possui as interfaces Gnome, KDE e LXDE, e assim por diante. Os ambientes gráficos costumam ser totalmente diferentes uns dos outros, como podemos observar na imagem abaixo o Ubuntu, com a interface KDE. O termo “interface gráfica” pode ser substituído por “ambiente gráfico” ou “ambiente de trabalho”.

Abaixo temos um exemplo da interface gráfica do Ubuntu, Gnome.





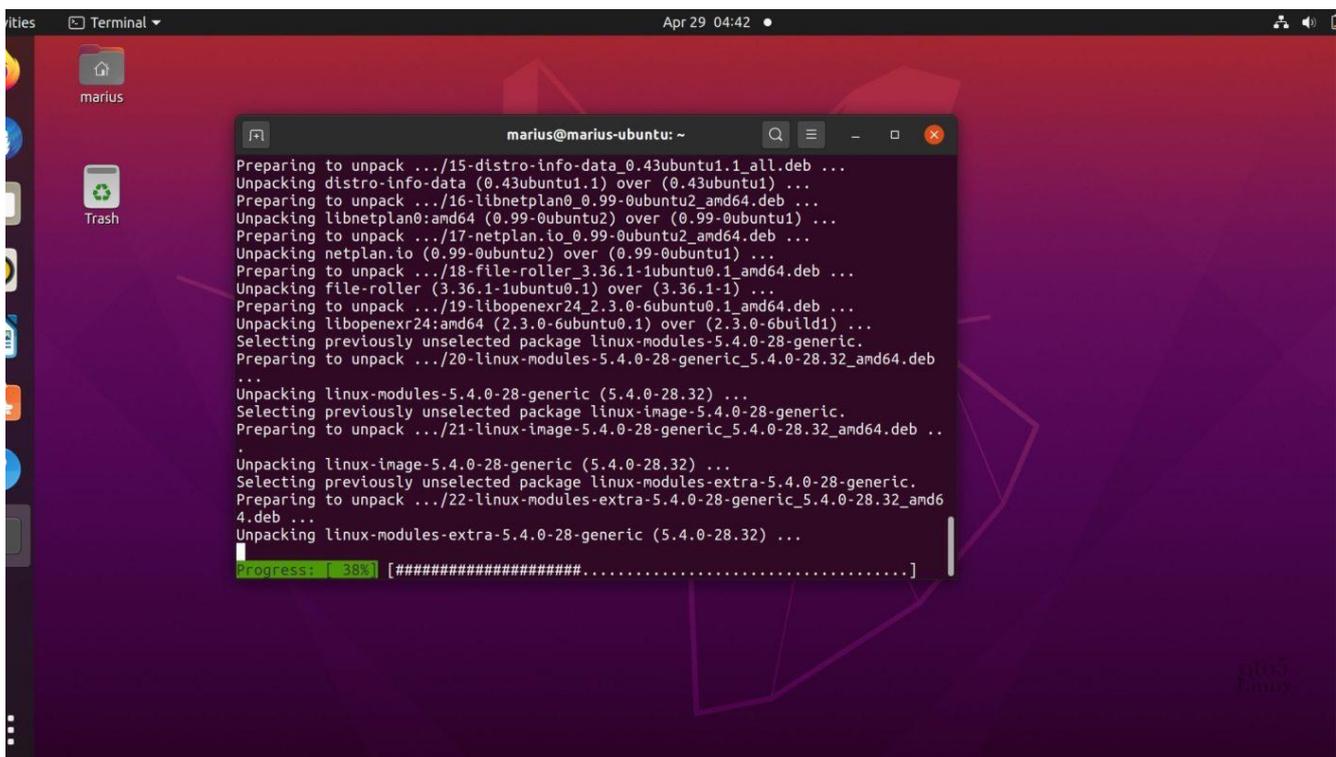
Linha de comando

A interface de linha de comando é a preferida pelos profissionais de T.I. principalmente quando se trata de implantação de servidores, pois como não existem recursos gráficos a serem processados, o uso da memória RAM e do processador é exclusivo para as necessidades dos serviços fornecidos. Abaixo temos uma imagem do modo linha de comando.



```
-rw-r--r-- 1 root root      1260 mar 16 2016 ucf.conf
drwxr-xr-x 4 root root      4096 fev 26 07:49 udev
drwxr-xr-x 2 root root      4096 abr  1 2016 udisks2
drwxr-xr-x 3 root root      4096 out 12 2016 ufw
-rw-r--r-- 1 root root       338 nov 18 2014 updatedb.conf
drwxr-xr-x 3 root root      4096 fev 26 07:50 update-manager
drwxr-xr-x 2 root root      4096 fev 26 07:50 update-motd.d
drwxr-xr-x 2 root root      4096 out  7 2016 update-notifier
drwxr-xr-x 2 root root      4096 out 12 2016 UPower
-rw-r--r-- 1 root root       270 out  6 2016 upstart-xsessions
-rw-r--r-- 1 root root      1018 out  5 2015 usb_modeswitch.conf
drwxr-xr-x 2 root root      4096 ago  8 2016 usb_modeswitch.d
drwxr-xr-x 2 root root      4096 fev 26 07:50 vim
lrwxrwxrwx 1 root root        23 fev 25 21:09 vtrgb -> /etc/alternatives/vtrgb
-rw-r--r-- 1 root root      4942 jul 15 2016 wgetrc
-rw-r--r-- 1 root root        30 abr  4 22:42 whoopsie
drwxr-xr-x 2 root root      4096 fev 25 21:29 wildmidi
drwxr-xr-x 2 root root      4096 out 12 2016 wpa_supplicant
drwxr-xr-x 11 root root      4096 out 12 2016 X11
drwxr-xr-x 6 root root      4096 out 12 2016 xdg
drwxr-xr-x 2 root root      4096 out 12 2016 xml
-rw-r--r-- 1 root root       477 jul 19 2015 zsh_command_not_found
cassio@notebook:~$ pwd
/home/cassio
```

Observe que também é possível acessar a linha de comando em ambiente gráfico, através do Terminal ou Shell. Na imagem abaixo podemos visualizar como é este acesso.



Gerenciamento de pacotes

O gerenciamento de pacotes no Linux é uma excelente maneira de distribuir, instalar e desinstalar softwares nas distros (distribuições). Um pacote é um arquivo que contém as informações necessárias para que o gerenciador de pacotes possa instalar, manter e remover os programas como também os demais arquivos necessários de um determinado aplicativo. Uma característica importante do Linux é a modularização de seus componentes e o uso intensivo de bibliotecas compartilhadas. Isso se aplica também aos pacotes de aplicativos. Cada pacote na distribuição contém programas executáveis, scripts, documentação e informações de configuração. Abaixo temos os principais gerenciadores de pacotes do Linux:

dpkg

O comando dpkg é o utilitário responsável pelo gerenciamento de pacotes em sistemas Debian.

apt-get

O comando apt-get (Advanced Package Tool) é um gerenciador avançado de pacotes.

rpm

O rpm (RPM Package Manager) é um gerenciador de pacotes usado em várias distribuições Linux, como o Red Hat, CentOS e Fedora.

yum

O Yum é um acrônimo para Yellow dog Updater, Modified. É uma ferramenta utilizada para gerenciar a instalação e remoção de pacotes em distribuições Linux, que utilizam o sistema RPM.

Estrutura de pastas

Você notou que o Linux possui diversas e diferentes versões. Porém, todas elas seguem uma estrutura de diretórios muito bem definida que está organizada em termos lógicos. Para manter essa organização, os desenvolvedores das distribuições Linux e softwares diversos seguem o *Filesystem Hierarchy Standard – FHS* (padrão para sistema de arquivos hierárquico), que é uma espécie de referência que padroniza quais pastas do sistema recebem determinados tipos de arquivo.

Diretório raiz (/)



Todos os arquivos e diretórios do Linux tem em comum uma única origem: o diretório raiz. Mesmo que estejam armazenados em outros dispositivos físicos, é a partir do diretório raiz – representado pela barra (/) – que você poderá acessá-los.

É importante destacar que o único usuário do sistema com permissão para criar ou mover arquivos do diretório raiz é o **root**, ou seja, o usuário-administrador.

Binários executáveis: /bin

No diretório /bin estão localizados os binários executáveis que podem ser utilizados por qualquer usuário do sistema. São comandos essenciais, usados para trabalhar com arquivos, textos e alguns recursos básicos de rede, como o cp, mv, ping e grep.

Binários do sistema: /sbin

O diretório /sbin também armazena executáveis, mas com uma diferença: são aplicativos utilizados por administradores de sistema com o objetivo de realizar funções de manutenção e outras tarefas equivalentes. Entre os comandos disponíveis estão o ifconfig, para configurar e controlar interfaces de rede TCP/IP, e o fdisk, que permite particionar discos rígidos, por exemplo.

Programas diversos: /usr

O /usr reúne executáveis, bibliotecas e até documentação de softwares usados pelos usuários ou administradores do sistema. Além disso, sempre que você compilar e instalar um programa a partir do código-fonte, ele será instalado nesse diretório.

Configurações do sistema: /etc

No diretório /etc ficam armazenados os arquivos de configuração que podem ser usados por todos os softwares, além de scripts especiais para iniciar ou interromper módulos e programas diversos.

Bibliotecas: /lib

No diretório /lib ficam localizadas as bibliotecas usadas pelos comandos presentes em /bin e /sbin. Normalmente, os arquivos de bibliotecas começam com os prefixos ld ou lib e possuem extensão .so.

Opcionais: /opt

Aplicativos adicionais, que não são essenciais para o sistema, são armazenados neste diretório.

Arquivos pessoais: /home



No diretório /home ficam os arquivos pessoais dos usuários, sempre dentro de pastas que levam o nome de cada usuário, semelhante ao diretório "Usuários" no Windows. A única exceção é o diretório pessoal do administrador que fica localizado em /root.

Inicialização: /boot

Os arquivos relacionados à inicialização do sistema (processo de boot) Linux, ficam em /boot.

Volumes e mídias: /mnt e /media

Para acessar os arquivos de um CD, pen drive ou disco rígido presente em outra máquina da rede, é necessário "montar" esse conteúdo no sistema de arquivos local, isso é, torná-lo acessível como se fosse apenas mais um diretório no sistema.

Em /media são montadas todas as mídias removíveis, como dispositivos USB. Já o diretório /mnt é reservado aos administradores que precisam montar temporariamente um sistema de arquivos externo.

Serviços: /srv

Dados de servidores e serviços em execução no computador ficam armazenados nesse diretório.

Arquivos de dispositivos: /dev

No Linux, tudo é apresentado na forma de arquivos. Por exemplo, ao conectar um pen drive no computador será criado um arquivo dentro do diretório /dev e ele servirá como interface para acessar ou gerenciar o drive USB. Nesse diretório, você encontra caminhos semelhantes para acessar terminais e qualquer dispositivo conectado ao computador.

Arquivos variáveis: /var

Todo arquivo que aumenta de tamanho ao longo do tempo está no diretório de arquivos variáveis. Por exemplo, os logs do sistema.

Processos do sistema: /proc

Nesse diretório são encontrados arquivos que revelam informações sobre os recursos e processos em execução no sistema.

Arquivos temporários: /tmp

Arquivos e diretórios criados temporariamente tanto pelo sistema quanto pelos usuários devem ficar nesse diretório.



Comandos



Esta é uma parte essencial da aula, pois em mais de 60% das provas de concurso são cobrados os comandos básicos do Linux.

Comando: ls

Este comando lista arquivos e diretórios dentro de uma pasta. Ele pode ser associado a alguns parâmetros para que o retorno seja de acordo com a necessidade do usuário.

Parâmetros:

-l: associado ao comando `ls` exibe mais detalhes sobre os arquivos listados (data de criação, permissões dos usuários, tamanho do arquivo)

-a: associado ao comando `ls` exibe os arquivos ocultos (que tem no início do nome um ".")

- Exemplos: `ls -l`
`ls -a`

Comando: cd

Este comando acessa uma determinada pasta digitada junto com o comando.

- Exemplo: `cd /home/concurseiro`

Onde será acessada a pasta do usuário `concurseiro` dentro da pasta `home`.

Comando: mkdir

Este comando cria um novo diretório. Make directory = `mkdir`.

- Exemplo: `mkdir Estrategia Concursos`

Comando: rm

Este comando é usado para apagar arquivos e também pode ser usado para apagar diretórios e subdiretórios vazios ou que contenham arquivos. Ele possui parâmetros que podem ser associados.

Parâmetros:



-v: mostra os arquivos à medida que são removidos.

-R: remove arquivos em subdiretórios e também pode ser usado para remover subdiretórios.

-f: força a exclusão do arquivo ou diretório.

- Exemplo: `rm -v Estrategia Concursos.txt`
`rm -R Estrategia Concursos/Aula02.pdf`

Entretanto existe uma variação deste comando que para apagar diretórios vazios.

Comando: `rmdir`

Este comando apaga um diretório vazio. Remove directory = `rmdir`.

- Exemplo: `rmdir Estrategia Concursos`

Comando: `cp`

Este comando copia um arquivo ou um diretório para um local determinado. Copy = `cp`.

É obrigatório especificar origem e destino.

- Exemplo: `cp Aula02.pdf /home/concurseiro`

Comando: `mv`

Este comando move um arquivo ou um diretório para um local determinado. Move = `mv`.

- Exemplo: `mv Aula02.pdf /concurso`

Também pode ser utilizado para renomear um arquivo ou diretório.

- Exemplo: `mv Aula02.pdf Aula_Linux.pdf`

E ainda é possível ser utilizado para mover e renomear um arquivo ou diretório.

- Exemplo: `mv Aula02.pdf /concurso/Aula_Linux.pdf`

Comando: `who`

Este comando informa qual o usuário está logado no sistema. Who = quem.

- Exemplo: `who`

Comando: `pwd`



Este comando informa qual o diretório corrente, ou seja, o diretório que está sendo acessado.

- Exemplo: `pwd`

Comando: `passwd`

Este comando irá alterar a senha do usuário. Só é possível alterar a senha de um usuário logado ou a partir do superusuário (root) alterar qualquer senha. Cuidado para não confundir este comando com o anterior `pwd`.

- Exemplo: `passwd concurseiro`

Onde será alterada a senha do usuário `concurseiro`.

Comando: `shutdown`

Este comando desliga ou reinicia o sistema, de acordo com o parâmetro utilizado. É necessário informar o tempo em segundos em que será realizada a ação.

Parâmetros:

-r: reinicia

-h: desliga

- Exemplo: `shutdown -r +30`
`shutdown -h +60`

Comando: `clear`

Este comando limpa a tela do terminal ou do modo linha de comando.

- Exemplo: `clear`

Comando: `top`

Este comando apresenta os processos que estão em execução no sistema com informações como ID do processo, usuário que está executando o processo, a quanto tempo este processo está em execução, quanto de memória e processador o processo está consumindo.

- Exemplo: `top`

Comando: `ps`

Este comando permite obter informações sobre o estado dos processos que estão sendo executados no momento.

- Exemplo: `ps`





É muito importante que todos estes comandos sejam testados para melhor fixação. Existem duas formas para você testar esses comandos sem a necessidade de instalar. Uma é através de um Live CD onde você inicializa (boot) o computador pelo CD e tem acesso ao sistema operacional para realizar qualquer teste. A outra é através de um emulador de Shell Linux para Windows que pode ser obtido via download ou acessado online.

Permissões de arquivos

Depois dos comandos, as permissões são um dos aspectos mais importantes do Linux. Elas são usadas para vários fins, mas servem principalmente para proteger o sistema e os arquivos dos usuários. Conforme mencionei nas características do Linux, o sistema possui apenas um super usuário, o **root**, que pode executar irrestritamente ações no sistema.

A partir do comando `ls` vamos entender como funcionam as permissões no Linux. Quando digitamos o comando `ls -l` são exibidos detalhes sobre os arquivos localizados naquele diretório como data de criação, permissões dos usuários, tamanho do arquivo. Abaixo temos um exemplo dos dados retornados pelo sistema ao digitar o comando `ls -l`.

```
root@genesis:/etc# ls -l
total 1088
-rw-r--r-- 1 root root 2981 nov 29 2017 adduser.conf
```

Na primeira linha do exemplo temos o usuário que está logado no sistema (no caso o **root**) e o nome do computador onde ele está logado (neste caso o computador é chamado **genesis**). Note que é apresentado de forma semelhante ao endereço de um e-mail **root@genesis**. Após isso temos o diretório que está sendo acessado (**/etc**) e por fim o comando que foi digitado `ls -l`. Na segunda linha temos o total de arquivos que estão dentro da pasta **etc**.

Na última linha temos as permissões do arquivo **adduser.conf**. O primeiro item que aparece (**-rw-r--r--**) é a forma como o sistema apresenta as permissões do arquivo. Esse item recebe o nome de **string** e o primeiro caractere da string indica seu tipo: se for **"d"** representa um diretório, se for **"-"** equivale a um arquivo. Entretanto, outros caracteres podem aparecer para indicar outros tipos, conforme mostra a tabela abaixo:

Caractere	Tipo
d	diretório
-	arquivo
l	link
b	arquivo de bloco



c	arquivo especial de caractere
p	canal
s	socket

Note que além do primeiro caractere ainda restam 9 caracteres na string. Esses caracteres são divididos em 3 grupos que representam respectivamente permissões de usuário, permissões de grupo e permissão para outros usuários. Ou seja, o usuário logado tem as suas permissões, um grupo de usuários pode possuir permissões a menos, iguais ou a mais que o usuário logado e os outros usuários também tem essas opções (permissões a menos, iguais ou a mais que o usuário logado).

As permissões possíveis são: r (read - leitura), w (write - escrita), x (execução) e - (nenhuma). Leitura permite ao usuário ler o conteúdo do arquivo, mas não alterá-lo. Escrita permite que o usuário altere o arquivo. Execução, como o nome diz, permite que o usuário execute o arquivo, quando for o caso.

Existem duas formas de atribuir as permissões. A primeira é utilizando números que equivalem a cada uma delas. R = 4, W = 2, X = 1 e nenhuma = 0.

- Exemplo 1: para permissão de leitura, escrita e execução, devemos somar os três valores (R+W+X = 4+2+1), ou seja, a permissão seria igual a 7.
- Exemplo 2: para permissão de leitura e escrita, devemos somar os três valores (R+W+X = 4+2+0), ou seja, a permissão seria igual a 6. O valor de X é igual a 0 (zero) pois a permissão será apenas de leitura e escrita.
- Exemplo 3: para permissão de leitura e execução, devemos somar os três valores (R+W+X = 4+0+1), ou seja, a permissão seria igual a 5. O valor de W é igual a 0 (zero) pois a permissão será apenas de leitura e execução.
- Exemplo 4: para permissão de escrita e execução, devemos somar os três valores (R+W+X = 0+2+1), ou seja, a permissão seria igual a 3. O valor de R é igual a 0 (zero) pois a permissão será apenas de escrita e execução.

O comando para alterar as permissões é o **chmod**.

Seu parâmetro é: chmod (valor da permissão para usuário) (valor da permissão para grupo) (valor da permissão para outros usuários) nome do arquivo.

- Exemplo: `chmod 744 adduser.conf`

Essa mudança faria com que o usuário logado tivesse todas as permissões possíveis e o grupo e outros usuários tivessem apenas permissão de leitura.

A segunda forma é através de letras, onde seriam usadas U (para usuário), G (para grupo) e O (para outros).

- Exemplo 1: para permissão de leitura, escrita e execução, devemos usar **u=rwx** (para usuário), **g=rwx** (para grupo) e **o=rwx** (para outros).



- Exemplo 2: para apenas uma ou duas permissões, devemos usar **u=rw** (para permissão de leitura e escrita ao usuário) ou **g=r** (para permissão de leitura ao grupo).
- Exemplo: `chmod u=rwx adduser.conf`

Essa mudança faria com que o usuário logado tivesse todas as permissões possíveis.

APOSTA ESTRATÉGICA

A ideia desta seção é apresentar os pontos do conteúdo que mais possuem chances de serem cobrados em prova, considerando o histórico de questões da banca em provas de nível semelhante à nossa, bem como as inovações no conteúdo, na legislação e nos entendimentos doutrinários e jurisprudenciais².



Dentro do assunto “Noções de sistema operacional Linux”, destacamos a barra de ferramentas, suas funções e atalhos. Vamos destacar abaixo algumas que consideramos principais entre as apresentadas na aula.

- **Comando: ls**

Este comando lista arquivos e diretórios dentro de uma pasta. Ele pode ser associado a alguns parâmetros para que o retorno seja de acordo com a necessidade do usuário.

Parâmetros:

- l: associado ao comando ls exibe mais detalhes sobre os arquivos listados (data de criação, permissões dos usuários, tamanho do arquivo)
- a: associado ao comando ls exibe os arquivos ocultos (que tem no início do nome um “.”)

- **Comando: cd**

Este comando acessa uma determinada pasta digitada junto com o comando.

- **Comando: mkdir**

Este comando cria um novo diretório. Make directory = mkdir.

² Vale deixar claro que nem sempre será possível realizar uma aposta estratégica para um determinado assunto, considerando que às vezes não é viável identificar os pontos mais prováveis de serem cobrados a partir de critérios objetivos ou minimamente razoáveis.



- **Comando: rm**

Este comando é usado para apagar arquivos e também pode ser usado para apagar diretórios e subdiretórios vazios ou que contenham arquivos. Ele possui parâmetros que podem ser associados.

- **Comando: rmdir**

Este comando apaga um diretório vazio. Remove directory = rmdir.

- **Comando: cp**

Este comando copia um arquivo ou um diretório para um local determinado. Copy = cp. É obrigatório especificar origem e destino.

- **Comando: mv**

Este comando move um arquivo ou um diretório para um local determinado. Move = mv.

As permissões possíveis são: r (read - leitura), w (write - escrita), x (execução) e - (nenhuma). Leitura permite ao usuário ler o conteúdo do arquivo, mas não alterá-lo. Escrita permite que o usuário altere o arquivo. Execução, como o nome diz, permite que o usuário execute o arquivo, quando for o caso.

Existem duas formas de atribuir as permissões. A primeira é utilizando números que equivalem a cada uma delas. R = 4, W = 2, X = 1 e nenhuma = 0.

Exemplo 1: para permissão de leitura, escrita e execução, devemos somar os três valores (R+W+X = 4+2+1), ou seja, a permissão seria igual a 7.

Exemplo 2: para permissão de leitura e escrita, devemos somar os três valores (R+W+X = 4+2+0), ou seja, a permissão seria igual a 6. O valor de X é igual a 0 (zero) pois a permissão será apenas de leitura e escrita.

Exemplo 3: para permissão de leitura e execução, devemos somar os três valores (R+W+X = 4+0+1), ou seja, a permissão seria igual a 5. O valor de W é igual a 0 (zero) pois a permissão será apenas de leitura e execução.

Exemplo 4: para permissão de escrita e execução, devemos somar os três valores (R+W+X = 0+2+1), ou seja, a permissão seria igual a 3. O valor de R é igual a 0 (zero) pois a permissão será apenas de escrita e execução.

O comando para alterar as permissões é o **chmod**.

Seu parâmetro é: chmod (valor da permissão para usuário) (valor da permissão para grupo) (valor da permissão para outros usuários) nome do arquivo.

Exemplo: `chmod 744 adduser.conf`

Essa mudança faria com que o usuário logado tivesse todas as permissões possíveis e o grupo e outros usuários tivessem apenas permissão de leitura.



A segunda forma é através de letras, onde seriam usadas U (para usuário), G (para grupo) e O (para outros).

Exemplo 1: para permissão de leitura, escrita e execução, devemos usar **u=rwx** (para usuário), **g=rwx** (para grupo) e **o=rwx** (para outros).

Exemplo 2: para apenas uma ou duas permissões, devemos usar **u=rw** (para permissão de leitura e escrita ao usuário) ou **g=r** (para permissão de leitura ao grupo).

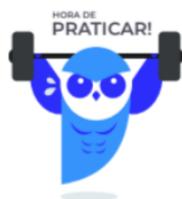
Exemplo: `chmod u=rwx adduser.conf`

Imprima o capítulo Aposta Estratégica separadamente e dedique um tempo para absolver tudo o que está destacado nessas duas páginas. Caso tenha alguma dúvida, volte ao Roteiro de Revisão e Pontos do Assunto que Merecem Destaque. Se ainda assim restar alguma dúvida, não hesite em me perguntar no fórum.

QUESTÕES ESTRATÉGICAS

Nesta seção, apresentamos e comentamos uma amostra de questões objetivas selecionadas estrategicamente: são questões com nível de dificuldade semelhante ao que você deve esperar para a sua prova e que, em conjunto, abordam os principais pontos do assunto.

A ideia, aqui, não é que você fixe o conteúdo por meio de uma bateria extensa de questões, mas que você faça uma boa revisão global do assunto a partir de, relativamente, poucas questões.



1. (FGV / MPE-AL – 2018)

Para baixar arquivos para o sistema operacional Ubuntu, uma das fontes oficiais é mostrada a seguir.

Memorial University of Newfoundland ftp 10 Mbps

O acrônimo ftp refere-se

- ao método de autenticação usado para estabelecer a conexão.
- ao status da conexão.
- a um indicativo da segurança dos dados na conexão.
- a um indicativo da velocidade de transferência de arquivos.



e) a um protocolo para a transferência de arquivo.

Comentários

A transferência de arquivos é realizada através do protocolo **FTP** (abreviação para *File Transfer Protocol* – Protocolo de Transferência de Arquivos) e é uma das mais antigas formas de interação na Internet. O protocolo FTP define a maneira pela qual os dados devem ser transferidos numa rede TCP/IP. Ele tem como objetivos o compartilhamento de arquivos entre dispositivos remotos, a independência dos sistemas de arquivos das máquinas dos clientes e do servidor e a transferência dos dados de maneira eficaz.

Portanto, a alternativa correta é a letra E.

Gabarito: alternativa E.

2. (FGV / MPE-AL – 2018)

Assinale a opção que indica o comando do sistema operacional Linux que pode ser utilizado para exibir o conteúdo de um arquivo texto.

- a) ln
- b) touch
- c) bg
- d) cal
- e) less

Comentários

Vamos entender a ação de cada um dos comandos nas alternativas:

a) ln: Cria um link a um arquivo

b) touch: modifica a data e hora de acesso e modificação de arquivos.

c) bg: significa “Trabalhos” e serve para exibir tudo o que está sendo realizado no shell do GNU/Linux que está em segundo plano.

d) cal: mostra um calendário de um mês e/ou ano em específico.

e) less: pode ser usado como comando para leitura de arquivos que ocupem mais de uma tela.

Gabarito: alternativa E.



3. (FGV / Banestes – 2018)

Ao executar o comando `ls` em um determinado diretório, o seguinte resultado é mostrado no console do sistema operacional Linux:

```
[user@localhost arquivos]# ls text1 text2 text3 text4
```

Se neste mesmo diretório for executado o comando:

```
[user@localhost arquivos]# ls ?[!1-2]*
```

O console mostrará:

- a) text1
- b) text1 text2
- c) text3 text4
- d) text1 text2 text3 text4
- e) nenhum arquivo.

Comentários

O comando `LS` lista arquivos e diretórios dentro de uma pasta. Ele pode ser associado a alguns parâmetros para que o retorno seja de acordo com a necessidade do usuário.

A interrogação (?) tem como finalidade substituir um ou mais caracteres, levando em consideração que eles devem existir.

A exclamação (!) é usada como negativa. Ou seja, para negar os valores em uma lista.

Os valores entre colchetes [] indicam o intervalo de busca. Sintaxe: [início-fim].

O asterisco (*) substitui qualquer quantidade de caractere.

Dessa forma, temos: `ls ?[!1-2]*` → onde `ls` lista os arquivos, `?` lista se tiver 1 ou mais caracteres, `[!1-2]` não lista os que começam com 1 ou 2. A partir daqui temos o ponto “complicado” que determina a resposta da questão: mesmo depois de todas as condições anteriores o que ele diz no final com o asterisco (*) é: liste todos os arquivos.

Gabarito: alternativa D.

4. (FGV / Prefeitura de Niterói - RJ – 2018)

Pedro é o proprietário do arquivo `header.txt` em um sistema Linux e gostaria de assegurar que somente ele tivesse permissão de leitura, gravação e execução a este arquivo, enquanto que todos os demais usuários com acesso ao sistema tivessem somente a permissão de leitura.

Assinale a opção que indica o comando que pode ser usado para conseguir esse objetivo.



- a) `chmod ug+r header.txt`
- b) `chmod 766 header.txt`
- c) `chmod 722 header.txt`
- d) `chmod +r header.txt`
- e) `chmod 744 header.txt`

Comentários

As permissões possíveis são: r (read - leitura), w (write - escrita), x (execução) e - (nenhuma). Leitura permite ao usuário ler o conteúdo do arquivo, mas não alterá-lo. Escrita permite que o usuário altere o arquivo. Execução, como o nome diz, permite que o usuário execute o arquivo, quando for o caso.

Existem duas formas de atribuir as permissões. A primeira é utilizando números que equivalem a cada uma delas. R = 4, W = 2, X = 1 e nenhuma = 0.

- Exemplo 1: para permissão de leitura, escrita e execução, devemos somar os três valores, ou seja, a permissão seria igual a 7.
- Exemplo 2: para permissão de leitura e escrita, devemos somar os três valores, ou seja, a permissão seria igual a 6.
- Exemplo 3: para permissão de leitura e execução, devemos somar os três valores, ou seja, a permissão seria igual a 5.
- Exemplo 4: para permissão de escrita e execução, devemos somar os três valores, ou seja, a permissão seria igual a 3.

O comando para alterar as permissões é o **chmod**.

Seu parâmetro é: `chmod (valor da permissão para usuário)(valor da permissão para grupo)(valor da permissão para outros usuários) nome do arquivo`.

Exemplo: `chmod 744 adduser.conf`

Note que o exemplo acima garante as mesmas permissões solicitadas pela questão.

Portanto, a alternativa correta é a letra E.

Gabarito: alternativa E.

5. (FGV / Prefeitura de Salvador - BA – 2017)

O Linux é um

- a) malware.
- b) sistema operacional.



- c) programa aplicativo.
- d) firmware.
- e) hardware.

Comentários

Linux, apesar de popularmente ser usado como termo para se referir a um sistema operacional, na verdade é o núcleo (Kernel) do sistema operacional. O termo correto para indicar o sistema operacional é GNU/Linux.

Apesar disso, nessa questão temos que considerar que o Linux é um sistema operacional.

Gabarito: alternativa B.

6. (FGV / Prefeitura de Niterói - RJ – 2015)

Windows e Linux têm em comum o fato de serem responsáveis, entre outras funções, pelo gerenciamento:

- a) do acesso às páginas da Internet;
- b) das senhas de login em sites e portais;
- c) da extração e recuperação de backups;
- d) do sistema de arquivos;
- e) das senhas de aplicativos.

Comentários

Questão simples! Apesar das alternativas poderem torna-la confusa. Analisando as alternativas é possível responder essa questão. Mas vamos a teoria:

Sistema operacional é o conjunto de programas que gerenciam recursos, processadores, armazenamento, dispositivos de entrada e saída e dados em uma máquina.

Note que ao gerenciar armazenamento, obrigatoriamente o sistema operacional gerencia o sistema de arquivos.

Gabarito: alternativa D.



QUESTIONÁRIO DE REVISÃO E APERFEIÇOAMENTO

A ideia do questionário é elevar o nível da sua compreensão no assunto e, ao mesmo tempo, proporcionar uma outra forma de revisão de pontos importantes do conteúdo, a partir de perguntas que exigem respostas subjetivas.

São questões um pouco mais desafiadoras, porque a redação de seu enunciado não ajuda na sua resolução, como ocorre nas clássicas questões objetivas.

O objetivo é que você realize uma autoexplicação mental de alguns pontos do conteúdo, para consolidar melhor o que aprendeu ;)

Além disso, as questões objetivas, em regra, abordam pontos isolados de um dado assunto. Assim, ao resolver várias questões objetivas, o candidato acaba memorizando pontos isolados do conteúdo, mas muitas vezes acaba não entendendo como esses pontos se conectam.

Assim, no questionário, buscaremos trazer também situações que ajudem você a conectar melhor os diversos pontos do conteúdo, na medida do possível.

É importante frisar que não estamos adentrando em um nível de profundidade maior que o exigido na sua prova, mas apenas permitindo que você compreenda melhor o assunto de modo a facilitar a resolução de questões objetivas típicas de concursos, ok?

Nosso compromisso é proporcionar a você uma revisão de alto nível!

Vamos ao nosso questionário:

Perguntas

- 1) Qual o comando para saber qual o diretório está sendo acessado?**
- 2) Qual o comando para verificar qual usuário está logado?**
- 3) Qual o comando apresenta os processos que estão em execução?**
- 4) Como funciona as permissões de arquivos no Linux? Quais permissões são possíveis?**



Perguntas com respostas

1) Qual o comando para saber qual o diretório está sendo acessado?

O comando pwd informa qual o diretório corrente, ou seja, o diretório que está sendo acessado.

2) Qual o comando para verificar qual usuário está logado?

O comando who informa qual o usuário está logado no sistema.

3) Qual o comando apresenta os processos que estão em execução?

O comando top apresenta os processos que estão em execução no sistema com informações como ID do processo, usuário que está executando o processo, a quanto tempo este processo está em execução, quanto de memória e processador o processo está consumindo.

4) Como funciona as permissões de arquivos no Linux? Quais permissões são possíveis?

As permissões em Linux são usadas em para vários fins, mas servem principalmente para proteger o sistema e os arquivos dos usuários. São elas: leitura, escrita e execução.

...

Forte abraço e bons estudos!

"Hoje, o 'Eu não sei', se tornou o 'Eu ainda não sei'"

(Bill Gates)

Thiago Cavalcanti





Face: www.facebook.com/profthiagocavalcanti
Insta: www.instagram.com/prof.thiago.cavalcanti
YouTube: youtube.com/profthiagocavalcanti



ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



1 Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



2 Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



3 Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



4 Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



5 Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



6 Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



7 Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



8 O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.