

PROGRAMAÇÃO EM SHELL SCRIPT

UNIDADE 1 - INTRODUÇÃO AO LINUX

Emmanuel Joca Nolêto

Introdução

Caro, estudante! Seja bem-vindo a esta unidade de **Programação em Shell Script!**

Iniciaremos com uma apresentação dos sistemas operacionais Linux, que possibilitam o uso de outras ferramentas fora do *software* proprietário, e que contemplarão conhecimentos sobre a realização de tarefas de forma mais eficiente, utilizando apenas comandos. Você sabia que os sistemas operacionais Linux possuem ferramentas de compatibilidade na execução de tarefas, como escrita de texto, criação de *slides*, entre outros? Sabe o que muda de um sistema Linux para um sistema Windows? Conhece as características do sistema Linux e como ele funciona enquanto servidor de uma empresa?

Diante dessas indagações, serão conceituados aspectos que permitirão a você utilizar o sistema operacional Linux na empresa ou em casa. Destacaremos também as diferenças entre os sistemas voltados para servidores e para estações de trabalho.

Então, fique atento, pois aqui abordaremos os principais temas relacionados ao assunto, como aspectos que norteiam seu uso e como o Shell Script tem um papel crucial para o processo de configuração e manutenção de um sistema Linux, *desktop* ou *server*.

Ao final de seus estudos, você dominará alguns conceitos de quando usar o quê em relação aos sistemas operacionais. Nem precisamos mencionar a importância de conhecer essa ferramenta para o profissional da área de tecnologia, certo?

Leia com atenção para ter um bom aproveitamento do material. Vamos lá?

1.1 Introdução ao sistema operacional

Os sistemas operacionais modernos, como você pode adivinhar, evoluem conforme a necessidade ditada pelo *hardware* existente. Diante disso, o conhecimento de seu funcionamento, bem como de suas particularidades, vantagens e desvantagens, pode ser um fator decisivo num processo de mudança, sendo de grande valia para o profissional. Conheça melhor o Linux a seguir!

1.1.1 O que é o Sistema Operacional Linux?

Atualmente, o sistema operacional Linux é utilizado por diversas empresas devido ao fato de ser gratuito. Além disso, o Linux permite que programas sejam executados de forma semelhante ao Windows e ao Mac Os. A denominação de “Linux”, entenda, é atribuída não ao sistema, mas ao núcleo que está contido internamente, sendo sua denominação técnica “kernel”.

Pois bem, saiba que o kernel é responsável por fazer a conversação entre o *hardware* e os *softwares*. Ele realiza o controle do processador, da memória, do disco e de qualquer outro periférico que esteja conectado de alguma forma ao computador.

VOCÊ O CONHECE?



Você sabia que a criação do sistema operacional Linux deve-se a Linus Torvalds, que também é um dos principais desenvolvedores do sistema? O finlandês programa desde sua infância e não sabia que sua criação iria revolucionar a computação, conectando o esforço de tantos desenvolvedores e entusiastas. Saber mais sobre ele é essencial para compreender o funcionamento do sistema. Confira: <https://www.techtudo.com.br/artigos/noticia/2011/11/conheca-linus-torvalds-o-criador-do-linux.html>.

Tenha em mente que o desenvolvimento do Linux foi colaborativo, ou seja, de forma voluntária, muitos desenvolvedores contribuíram para que o Linux se tornasse o sistema operacional que temos no mercado atual. Hoje, diversos equipamentos eletrônicos que fazem parte de nosso dia a dia usam sistemas operacionais Linux, como roteadores, relógios e televisores.

Muitos dos servidores que funcionam na Internet também têm como sistemas operacionais os sistemas Linux. Devido ao fato de seu documento de licença estabelecer sua gratuidade, seus códigos fonte são abertos para quem desejar. Sua licença por ser do tipo GLP, GNU Public License, o que permite total liberdade na hora de executar um programa, independentemente de sua finalidade. De fato, o sistema é totalmente receptivo para qualquer execução e permite que qualquer programa seja adaptado, contudo, para isso, é necessário que o código fonte do programa também seja livre, compartilhando a mesma licença do Linux.

Diversas corporações no mundo todo, e até mesmo órgãos governamentais, utilizam o sistema Linux para algumas de suas tarefas. Os sistemas bancários, por meio de seus terminais de autoatendimento, assim como os diversos serviços que temos no *smartphone*, como o sistema Android, são baseados no Linux. Outro aspecto que é de grande peso quando falamos nesse sistema é sua capacidade de funcionar em diversos tipos de *hardware*, principalmente em computadores antigos que geralmente descartamos por se tornarem lentos.

Uma das perguntas que nos fazemos nesse cenário é: o que levou tantos indivíduos e empresas a adotarem o sistema Linux? Bem, podemos citar o crescimento da quantidade de *softwares* que existem e sua compatibilidade com os demais sistemas operacionais. Além disso, a possibilidade de manipular um arquivo gerado em outro sistema operacional no Linux despertou o interesse por parte dos usuários domésticos e corporativos.

Existe uma forma de uso pela qual muitos optam: trabalhar com o Linux instalado em paralelo ao sistema atual do computador, utilizando ora um, ora outro, conforme o desejo do usuário. Por fim, um aspecto positivo dos sistemas Linux é a diversidade de distribuições que existem no mercado e a possibilidade de criação de um sistema operacional próprio.

1.1.2 Distribuições Linux

O surgimento das distribuições ocorreu com a necessidade de criar uma forma mais prática de unir em um só lugar todos os pacotes de códigos fontes que estavam espalhados pela Internet. Usuários mais tímidos ou iniciantes tinham dificuldade em montar seu sistema Linux devido à excessiva distribuição dos pacotes por diversas partes.

VOCÊ QUER LER?



Recomendamos a leitura do artigo sobre Shell Script, de Fábio Correia, disponível no link: <https://www.revista-programar.info/artigos/Shell-Script/>. O artigo proporcionará um maior aprofundamento de seus conhecimentos da linguagem.

Usuários mais avançados do sistema, vendo a importância do sistema e a curiosidade de outros para utilizá-lo, resolveram criar distribuições que já viessem com os códigos necessários para o funcionamento do sistema no computador. Confira na imagem a seguir!

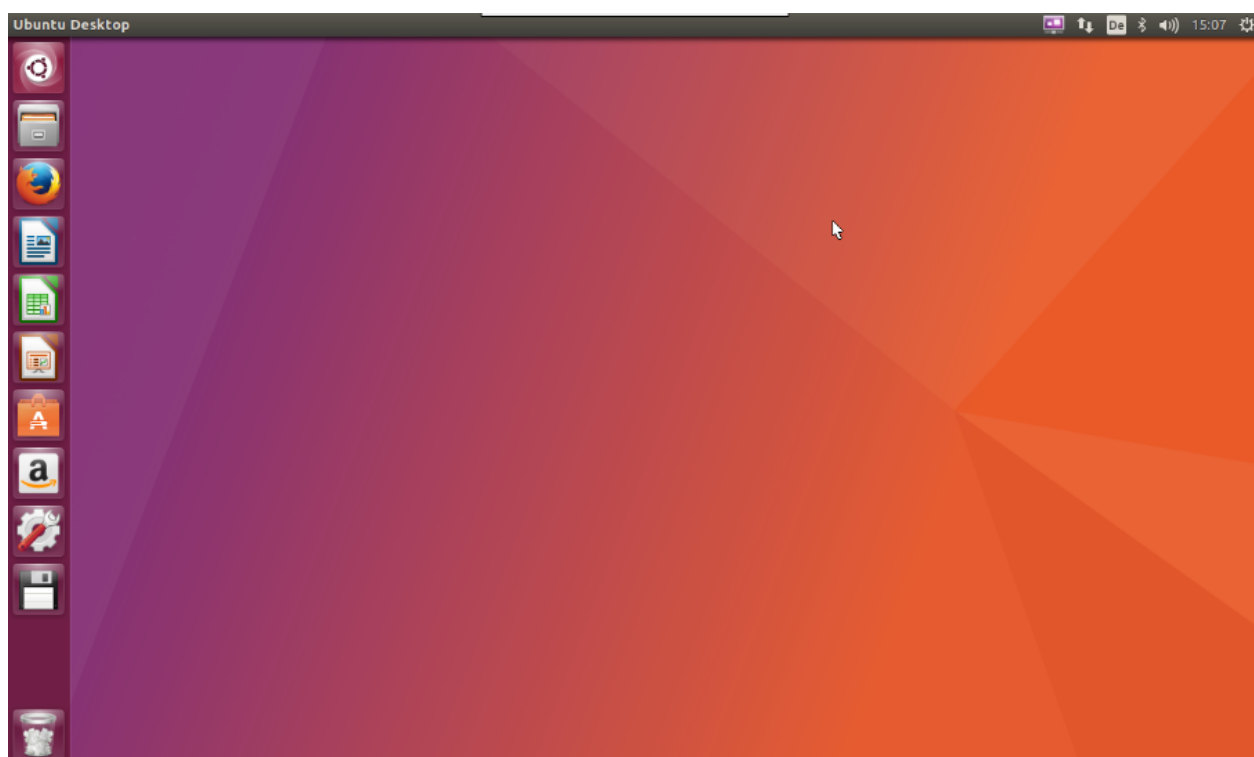


Figura 1 - Interface inicial da distribuição Ubuntu.

Fonte: Elaborada pelo autor, 2019.

Essa mudança na forma de apresentar o sistema fez com que algumas distribuições surgissem. Clique e confira, a seguir, algumas delas!

Slackware

É considerada uma das mais antigas, e seu propósito é atender a usuários menos técnicos, pois ela disponibiliza tudo em pacotes, ou seja, conjunto de códigos já compilados e testados.

Red Hat

É a distribuição mais conhecida. Utiliza gerenciamento de pacotes em formato RPM, tem uma simplicidade em realizar qualquer instalação, possui uma administração em formato gráfico. Atualmente, servidores com

sistemas Linux e distribuição Red Hat funcionam em diversas aplicações corporativas e governamentais. É, contudo, uma distribuição paga.

Fedora

É um sistema que tem sua comunidade mantida pela Red Hat, seu foco é teste de soluções recentes no mercado. Isso proporciona uma certa instabilidade devido ao fato de seus códigos serem recentes. Toda e qualquer novidade desenvolvida pela Red Hat é testada primeiro em uma distribuição Fedora.

Ubuntu

É considerada na atualidade a distribuição mais utilizada para *desktop* e usuários iniciantes. É baseada na distribuição Debian, a qual, por sua vez, é famosa por ser estável, já que todo pacote ou atualização lançada é incansavelmente testada antes de ser liberada para os usuários finais. O Ubuntu foi uma distribuição que inovou a forma de instalar o sistema operacional no computador, possibilitando a realização dessa tarefa pelo sistema Live CD. Nesses CDs, havia tudo que era necessário para o funcionamento do sistema operacional, além de outros programas, como navegadores, editores de texto etc.

Todo esse arcabouço de programas e as facilidades que eles proporcionaram foram de grande importância para a transição dos usuários que desejavam utilizar o sistema Linux.

VAMOS PRATICAR?



Diante de tantas opções para se ter uma distribuição Linux, qual das distribuições citadas anteriormente você escolheria e por qual motivo?

1.2 Linux e suas características

Você sabe de que forma o sistema operacional Linux se diferencia do sistema operacional Windows? Tenha em mente que esse despertar para compreender como é o funcionamento de cada sistema permite que uma análise comparativa seja feita e o entendimento de qual é a melhor solução. Vamos lá?

1.2.1 Introdução às características dos sistemas operacionais

Para começar, tenha em mente que os sistemas operacionais trabalham de forma semelhante. De fato, existem características que geram resultados iguais, ainda que o caminho seja diferente. Estamos falando de características como janelas gráficas, efeitos de animação, ícones diversos e integração com mouse e teclado; recursos que as distribuições Linux têm nas versões voltadas para *desktop*. As distribuições servidoras não disponibilizam a parte visual e gráfica devido ao fato de sua utilização impactar na performance do sistema.

As características peculiares de cada sistema operacional, ou seja, formas de executar um comando, forma de personalização, não existem para criar dificuldades para o usuário. Apenas com a utilização de diferentes sistemas operacionais é possível realizar o comparativo entre eles, e afirmar que um sistema é melhor que outro em determinada situação. Contudo, não podemos esquecer que essa afirmativa é bem distinta de acordo com o público que vai utilizar o sistema, e isso impacta no que é entendido, como pontos positivos e pontos negativos.

1.2.2 Diretórios

Entenda que o Linux tem um formato de diretórios diferente do sistema Windows, sendo a primeira diferença a forma com que esses diretórios aparecem para o usuário. Não há mais a distinção dos diretórios por letras, sendo que todas as pastas ficam num local denominado “diretório raiz”, cuja representação é feita apenas por meio de uma barra. Uma outra diferença está nas unidades de disco que são colocadas no computador; unidades como pen drives, CDs ou DVDs, discos externos etc. também ficam no diretório raiz.

Confira essas distinções na imagem a seguir!

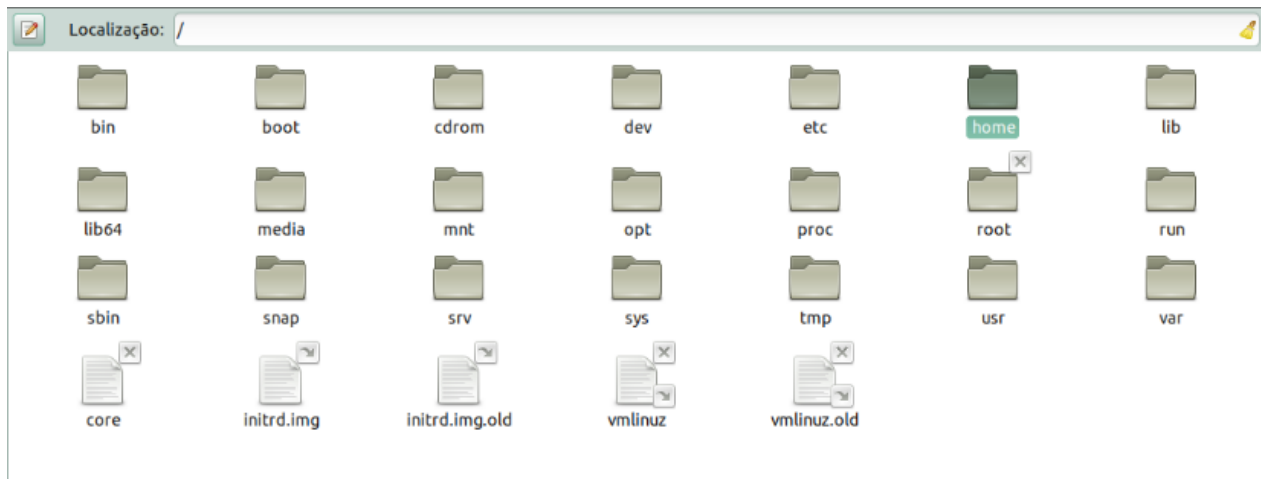


Figura 2 - Lista de diretórios de um sistema Linux.

Fonte: Elaborada pelo autor, 2019.

Os arquivos do usuário do sistema, por outro lado, vão para dentro da pasta `/home`. Isso impede que arquivos fiquem espalhados por diversas partes do sistema operacional. Para possibilitar um melhor entendimento, disponibilizamos, a seguir, uma tabela com o descritivo dos diretórios principais. Acompanhe!

- `/bin`

Responsável por armazenar executáveis do sistema; todos os arquivos de programas usados com certa frequência ficam nesse diretório. Esses arquivos também são chamados de “binários essenciais”.

- `/usr`

Contém a maioria dos programas utilizados pelo usuário. Nele também estão contidas as bibliotecas necessárias para o funcionamento dos programas do sistema. Muitos desses arquivos são somente leitura, para evitar alterações que prejudiquem o sistema.

- `/usr/bin`

Diretório que armazena os arquivos binários que não são essenciais para o funcionamento do sistema.

- `/usr/lib`

Onde ficam armazenadas bibliotecas usadas pelos programas. O conteúdo desse diretório é bem semelhante aos arquivos `.dll` no Windows.

- `/boot`

Diretório onde fica armazenado o kernel do sistema operacional e também os arquivos responsáveis pelo *grub*, que gerencia as partições do sistema de *boot* do computador. Existem sistemas operacionais mais antigos que criam uma partição para os arquivos do *boot* separadamente.

- `/dev`

Esse diretório é responsável por permitir o acesso aos dispositivos externos que venham a ser conectados ao computador. O mais interessante é que esses arquivos não são arquivos e nem pastas, mas sim *links* para acesso aos dispositivos que estão conectados ao computador.

- /etc

São o equivalente ao registro do Windows, ou seja, os arquivos de configuração do computador ficam todos nesse diretório. Todos esses arquivos são configurados de forma manual em algumas distribuições do Linux, mas já existem outras que possibilitam uma forma visual de configurar.

CASO



Você sabia que o primeiro longa-metragem de sucesso produzido em servidores Linux foi Titanic, em 1997? Nele, o RMS Titanic, um gigante navio de passageiros que naufragou em 15 de abril de 1912 teve sua história contada no longa-metragem dirigido por James Cameron. Muitas das cenas do filme foram feitas por meio de computação gráfica, nas quais foram empregados recursos de renderização.

Todo o processo do naufrágio está disponível em *sites* especializados no assunto. O diretor criou um filme dramático usando recursos bons e baratos, mas sem deixar a qualidade de lado. Muitas das cenas que necessitavam de efeitos digitais foram feitas intercalando efeitos tradicionais do cinema, sendo, depois, passadas para o computador por meio de quadros e armazenadas em arquivos num espaço de aproximadamente cinco terabytes. Diversas formas foram usadas para as composições de uma cena completa, por exemplo, pessoas em um plano, água e fumaça renderizados digitalmente, a miniatura do navio, cada pedacinho unido como uma grande e bela colcha de retalhos. Para renderizar os efeitos do filme, foram utilizadas aproximadamente 350 CPUs SGI, 200 CPUs DEC Alpha. Estes, por sua vez, estavam todos ligados por uma rede de 100 Mbps. Pesquise na Internet outras produções que possam ter sido feitas usando servidores Linux.

Fora todos os diretórios citados, temos também outros que fazem parte do sistema, sendo que cada um desempenha um papel em seu funcionamento.

1.2.3 Permissões em sistemas Linux

As permissões em sistemas Linux são divididas em três atributos principais: leitura; gravação; e execução, sendo esta última apenas para arquivos. A representação se dá pelas letras “r”, “w”, “x”. Todas essas permissões são atribuídas de acordo com o dono, o grupo e os demais usuários do sistema. Por regra, o dono sempre tem a permissão de fazer tudo que desejar e os demais tipos de acessos são restritos apenas à leitura.

Saiba que essas características são atribuídas não apenas aos arquivos, mas às pastas. Vale ressaltar que “grupo” corresponde a um conjunto de usuários que podem realizar os procedimentos nos arquivos e pastas que lhes foram atribuídas. A representação das permissões por meio de letras é uma característica que muitas vezes não fica bem clara, devido ao fato de essas terem como origem termos em inglês, dos quais apenas a primeira letra é usada. Leia atentamente as definições a seguir e entenda melhor o que cada uma das letras significa.

- “r” é a primeira letra da palavra “read”, cujo significado é “ler” ou “leitura”. Permite que seja feita a leitura em um arquivo ou diretório em que haja essa atribuição.
- “w” é a primeira letra da palavra “write”, que significa “escrever” ou “escrita”. Permite que seja feita a escrita em um arquivo ou diretório ao qual tenha sido atribuído esse tipo de permissão. Vale lembrar que até para apagar ou deletar um arquivo ou diretório é necessário ter permissão.
- “x” foge da regra das duas permissões anteriores, devido ao fato de representar uma palavra que inicia com vogal. Assim, foi atribuída a segunda letra. Essa permissão representa o termo “execute”, no qual podemos realizar o procedimento semelhante à execução de um arquivo executável no sistema Windows.

Fora a possibilidade de execução de arquivo quando essa permissão for atribuída a diretórios, é possível acessar por meio do comando “cd” do sistema.

Saiba que as permissões do sistema Linux não estão restritas apenas a letras, mas sim também a valores numéricos, que são baseados no modelo octal. Nesse modelo, em vez de serem utilizadas letras, são empregados valores numéricos para representá-las, conforme listagem apresentada por Monquiro (2008). Observe!

- 0 – Nenhuma permissão de acesso. Equivalente a -rwx.
- 1 – Permissão de execução (x).
- 2 – Permissão de gravação (w).
- 3 – Permissão de gravação e execução (wx). Equivalente à permissão 2+1.
- 4 – Permissão de leitura (r).
- 5 – Permissão de leitura e execução (rx). Equivalente à permissão 4+1.
- 6 – Permissão de leitura e gravação (rw). Equivalente à permissão 4+2.
- 7 – Permissão de leitura, gravação e execução. Equivalente à +rwx (4+2+1).

Para todos os sistemas Linux, essas funcionalidades são fundamentais no momento de criação e execução de algum script, uma vez que elas impactam no resultado final caso não haja a devida permissão no diretório ou arquivo em que se deseja executar ou alterar algo.

VAMOS PRATICAR?



Qual comando devemos escrever para dar permissão de leitura e escrita em um diretório?

1.3 O que é Shell e quais são suas utilizações?

Dentro desse panorama do Linux que acabamos de apresentar, o Shell permite utilizar o sistema por meio de uma interface que não dispõe de um ambiente gráfico. Ela é basicamente um ambiente que funciona exclusivamente em modo texto, ou seja, sem a interação do mouse. Suas funções e tarefas são todas executadas por meio de comandos específicos.

1.3.1 Terminal Shell

Compreender o ambiente onde se executará o Shell Script é de grande importância, da mesma forma que conhecer as possibilidades de execução do Shell Script vai permitir que problemas de um grau de dificuldade maior sejam resolvidos de forma automática. O Shell é de grande valia na busca por esse tipo de solução.

O Shell Script é uma ferramenta muito utilizada por permitir que sistemas operacionais Linux sejam automatizados de forma eficiente, garantindo a velocidade na entrega do resultado de sua execução. Sua forma de funcionamento é muito semelhante aos arquivos em lote do MS-DOS, como você pode observar na imagem a seguir. Dentro dos arquivos de Shell Script, pode haver inúmeras linhas de comando, controle de fluxo de informações complexas e simples, testes com condições complexas, entre outras tarefas que podem ser automatizadas.

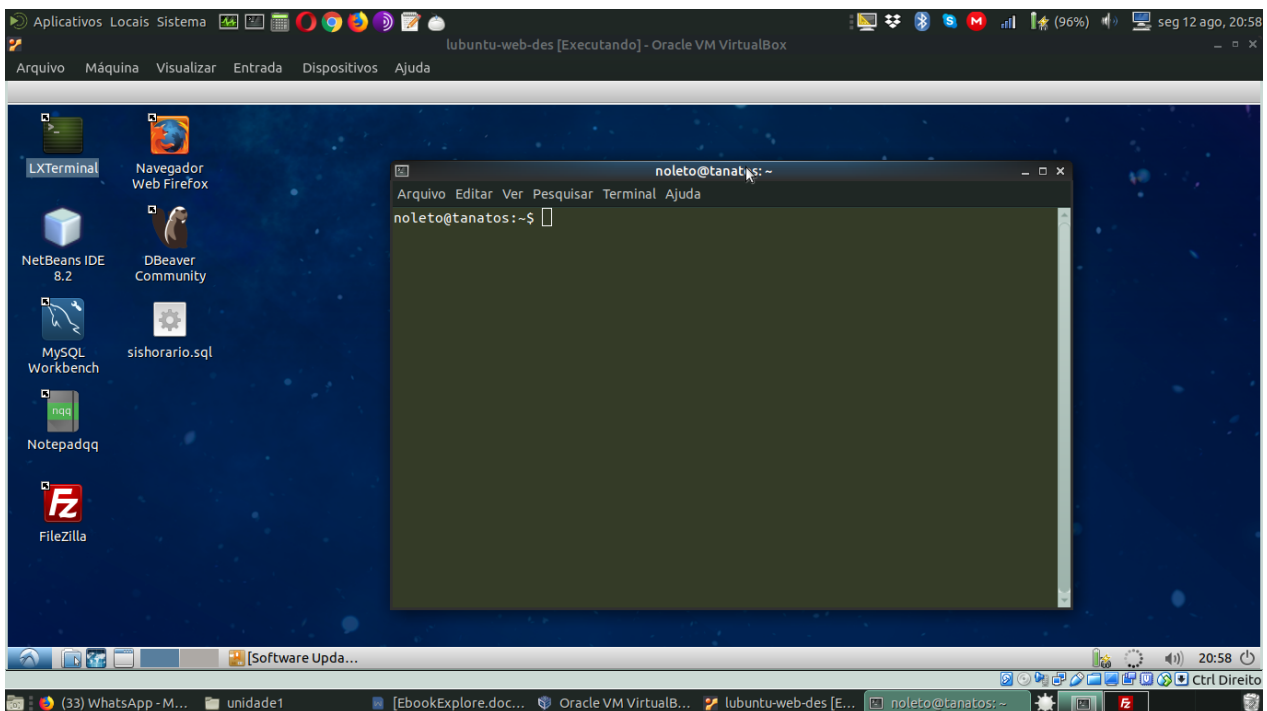


Figura 3 - Terminal Shell.
Fonte: Elaborada pelo autor, 2019.

Para criar um script em Shell, é necessário apenas ter instalado no computador um *software* de editor de texto. *Softwares* que podem ser usados para construção dos scripts são o VIM e o NANO. Ambos são editores de texto que funcionam em modo texto, mas sem o uso de mouse, necessitando que se aprenda os comandos “salvar”, “editar”, “fechar” e “abrir”. Clique nos itens e confira mais sobre o tema.

Outra característica do Shell é a possibilidade de realizar uma personalização dos caracteres exibidos ao se executar uma janela desse terminal. Mas não é só isso: você também pode fazer com que a cada execução, informações específicas sejam apresentadas ao usuário, como o nome do computador local, o diretório que está sendo executado, entre outros.

Quando os scripts são executados, suas opções podem ser escritas na linha de comando, sendo chamadas de “argumentos de permissões” que o processo de execução precisa para poder ser realizado. Os sistemas de empacotamento são grandes utilizadores de Shell Script, pois demandam que, no momento de sua execução, sejam inseridas as devidas permissões nos diretórios em que houver acesso. Mas esse é apenas um dos vários exemplos de utilização do Shell.

Tenha em mente que há uma diferença entre o “Shell” e o “script”. Costumamos falar dos dois juntos e imaginamos que sejam uma coisa só, mas saiba que isso não é verdade. O Shell nada mais é do que um interpretador de comandos que permite ao sistema Linux compreender qual tarefa o usuário quer que seja executada, verificando se ele tem ou não permissões para tal. Ele também é popularmente conhecido como “interpretador de comandos”. Já o “script” corresponde a um conjunto de comandos que, escritos da forma correta, podem executar diversas tarefas de forma automática.

1.3.2 O Shell Script pode ser considerado linguagem?

Sendo utilizando em ambientes Linux, o Shell Script pode ser considerado, sim, uma linguagem de programação. Sua facilidade em ser executado dentro desse tipo de sistema proporciona maior segurança ao usuário, pois ele passa a conhecer as consequências das ações executadas. Outra vantagem que podemos citar é a velocidade de execução de qualquer instrução sem gargalo, evitando, assim, transtornos na espera do resultado.

Clique nas setas e aprenda mais sobre o tema.

Os códigos feitos por meio de scripts e executados no Shell são compreendidos de forma interpretada, em que a sequência de instruções é escrita para ser executada de forma sequencial, podendo ser atualizada ou ter sua manutenção feita sempre que necessário. Vale ressaltar que o código compilado é mais rápido devido ao fato de estar escrito em arquivos binários, o que facilita a execução de grandes cargas.

Muitos dos recursos podem ser classificados como “especiais” no Shell, pois permitem realizar a coleta de toda e qualquer entrada de dados, fazendo o direcionamento entre os dados de saída e os comandos que foram digitados pelo usuário. Existe também o histórico, que armazena os dados digitados e permite realizar uma posterior verificação. A maioria dos usuários do Linux tem preferência por um ou outro tipo de Shell, ainda que todos tenham a mesma finalidade.

Outro recurso que pode ser utilizado para esses terminais é verificar a documentação por meio do comando “man bash”. Outro, ainda, corresponde à possibilidade de verificar um comando que apresenta erro ao ser executado. Por meio do histórico do terminal Shell, podemos fazer uma busca dos comandos que foram digitados e recuperá-los para verificar o que estava errado, corrigir e executar novamente. Esse recurso se dá por meio do comando “history”, que realiza o armazenamento de todos os comandos utilizados. Sua utilização, quando é feita sem um valor numérico ao lado, apresenta todos os comandos que foram digitados, e, caso você deseje limitar os comandos buscados, basta digitar um valor numérico inteiro e executar. Esse valor vai limitar a exibição, que apresentará apenas os “n” últimos comandos digitados no sistema por aquele usuário.

Outras formas de utilizar o histórico de comandos do terminal Shell é por meio das teclas para cima e para baixo. Esse procedimento vai possibilitar que se navegue na lista de comandos e até mesmo faça uma busca por meio de uma “string”, texto, do que se deseja encontrar na listagem de comandos. Existe, ainda, o comando “fc”, que, quando digitado e seguido de um número de linha, vai permitir a abertura de um arquivo em que constarão os comandos digitados, sendo que o cursor estará na linha digitada. Essa forma permite até que se altere o comando e salve para que, em momento posterior, seja exibido apenas na forma alterada.

Todas essas opções são uma pequena possibilidade do que se pode fazer com os comandos do terminal Shell. Muitos dos comandos podem ser compilados e interpretados, fatores que impactam no funcionamento de um comando. Essa variação da carga de dados a serem executados é o fator primordial para decidir se as instruções serão compiladas ou executadas. A intercalação entre os executados sendo acionados pelos interpretados é um dos modelos mais usados atualmente para o desenvolvimento de sistemas mais robustos, que permitam manutenções de problemas na execução de instruções.

1.3.3 Criação de um Shell Script

Para a criação de um script, existem alguns procedimentos que devem ser feitos. Primeiramente, deve-se saber que os arquivos de script contêm a extensão “sh”. Contudo, vale lembrar que existem diferentes tipos de Shell, alguns deles são o “bash”, o “csh”, o “zsh”, entre outros, sendo bash o mais utilizado. Fora isso, ele também é cobrado na certificação LPI, Linux Professional Institute, empresa certificadora que realiza as certificações referentes ao sistema operacional Linux. Uma vantagem do bash é sua grande semelhança à linguagem C, proporcionando, assim, um maior conforto por parte dos desenvolvedores.

Outro aspecto que deve ser levado em consideração para o processo de programação em Shell Script é a forma como as variáveis devem ser declaradas. Elas seguem algumas regras que são atribuídas para algumas linguagens, mas têm suas particularidades.

Sua maneira de ser declarada é a seguinte: nome_da_variavel=valor_da_variavel, lembrando que não pode haver espaço algum entre o nome e o valor da variável. Elas estão divididas em locais e globais, sendo as globais utilizadas em todo o processo e as locais apenas em partes específicas do código. Para se declarar uma variável local, deve-se seguir algumas regras, como as que listamos a seguir:

- seus nomes devem ser escritos com a letra maiúscula;
- podem ser listadas por meio do comando “set” ou “env”;
- devem ser definidas no *login* do sistema.

Os comandos em Shell possibilitam a colocação de argumentos logo após alguns tipos de inserções, ou no final de uma linha inteira. Ele, o argumento, é composto de um pequeno fragmento de informações que garantem o funcionamento do comando. Ele também pode representar o objetivo do comando para que seu funcionamento seja satisfatório. Existem situações em que argumentos estão associados a uma opção apenas. Caso isso aconteça, eles devem ser digitados logo após o comando, para que sua execução seja imediata.

Outros aspectos relacionados ao tema estão elencados abaixo. Clique e confira!

Um recurso que pode proporcionar maior agilidade na digitação de comandos, e é utilizado regularmente no terminal Shell, é o “alias”, que permite saber todas as formas de digitação do comando que se deseja usar naquele momento. Se não houver, você pode criar o seu “alias” personalizado. Também existem funções que permitem a junção de vários comandos de uma vez só, unificando-os.

Os comandos pré-definidos no Shell são de grande valia e podem resolver os mais diversos problemas que venhamos a enfrentar no sistema operacional Linux. Caso você não saiba como esse ou aquele comando funciona, contudo, pode ativar o comando “type”, que apresentará a forma como o comando funciona.

O Shell cria um armazenamento das informações que serão usadas na sessão. Quando você quiser ver todas as variáveis definidas pelo Shell que estão sendo utilizadas no momento, deve digitar o comando “set”. Também existe o comando “env” que apresenta as variáveis de ambiente, possibilitando a exportação de suas variáveis para todo e qualquer terminal que venha a ser aberto, desde que o novo terminal seja oriundo do terminal-pai que inicializou as variáveis.

Algo que fascina no sistema operacional Linux são as múltiplas maneiras de se fazer qualquer coisa. Digitando o comando “declare”, obtém-se uma lista das variáveis de ambiente atuais e seus valores do sistema que está em funcionamento e, ainda, uma lista de funções do Shell.

As variáveis são definidas por meio do terminal, mas existem algumas dessas variáveis utilizadas no terminal de origem, ou até mesmo acessíveis em outras sessões criadas de forma posterior. Para isso, é necessário utilizar o comando “export”, que irá permitir a outras sessões verem as variáveis. Algo muito interessante a respeito do bash no Linux é a quantidade de comandos que permitem automatizar muitas tarefas.

A seguir, temos um quadro com a lista dos principais comandos Linux. Observe!

Comando	Finalidade
alias	Criar um alias para um comando, nome alternativo.
exec	Um comando invocado com exec substitui a sessão atual do shell.
echo	Replica um texto informado ou conteúdo de variável.
env	Sem argumentos, exibe as variáveis de ambiente e seus conteúdos. Pode executar um comando com variável de ambiente modificada.
export	Define uma variável de ambiente para a sessão e para todas as sessões iniciadas a partir dela.
pwd	Mostra o diretório atual.
set	Define o valor de uma variável. Sem argumentos, mostra as variáveis já definidas.
unset	Remove uma variável na sessão.

Quadro 1 - Listagem de comandos no Linux.

Fonte: Elaborada pelo autor, 2019.

Outra característica das variáveis é que, antes da escrita, impressão na tela, seu nome deve ser precedido pelo caractere “\$”, para que no momento de execução ela seja diferenciada dos demais comandos, conforme você pode verificar na figura a seguir:

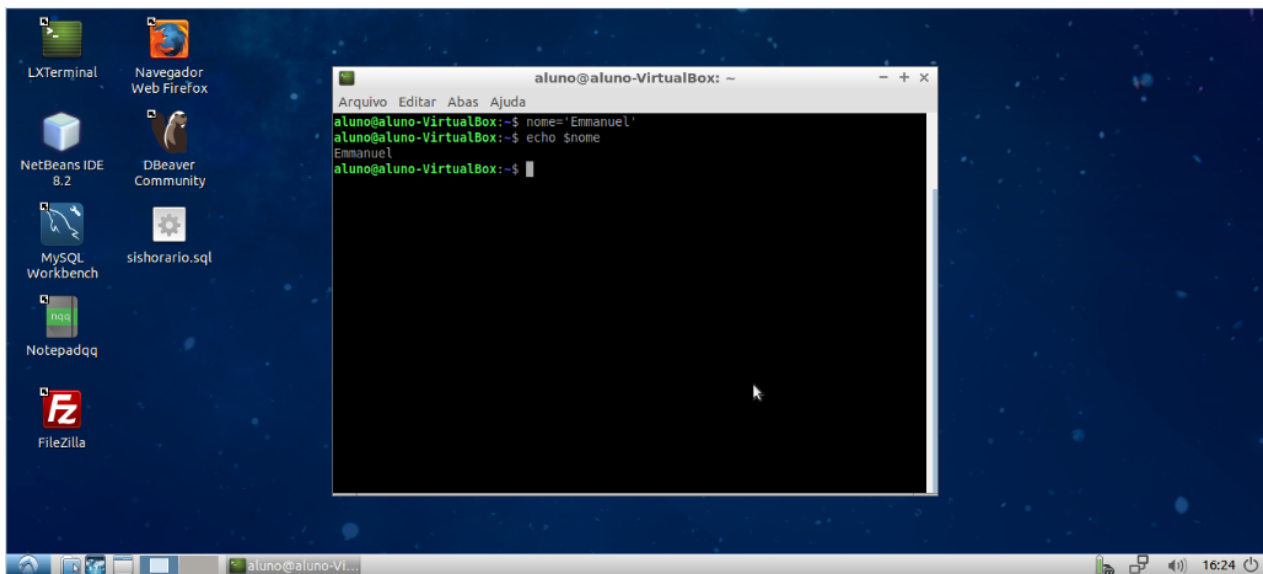


Figura 4 - Uso do comando “echo” e “\$” no terminal.

Fonte: Elaborada pelo autor, 2019.

Bem, são essas as possibilidades iniciais para a diversidade de comandos que existem no Shell. Se fosse para criar uma lista com todos os comandos, seriam necessárias muitas páginas, pois há inúmeras possibilidades.

VOCÊ SABIA?



Uma dica para ampliar seus conhecimentos em comandos Linux é fazer uma busca na “man page bash”, que tem documentação disponível em: <https://www.gnu.org/software/bash/manual/>. Nela, estão os comandos oficiais do próprio bash e um descritivo deles com os respectivos exemplos.

A compreensão do sistema operacional Linux é de grande valia para garantir a performance no processo de criação de scripts mais complexos e que tenham menor possibilidade de erros durante seu processo de execução.

VAMOS PRATICAR?



Qual é o comando que pode ser utilizado no Shell Script para listar todos os arquivos que estão em um diretório? Escreva o comando e exemplifique.

Síntese

Chegamos ao final desta unidade, que abordou aspectos do sistema operacional Linux e as características do Shell Script. Neste momento, você já sabe como se dá o funcionamento de um sistema operacional Linux, bem como suas particularidades e comandos que irão ajudar no processo de criação e execução de scripts em Shell.

Nesta unidade, você teve a oportunidade de:

- compreender como são os sistemas operacionais Linux;
- entender o que são e como funcionam as distribuições Linux;
- conhecer as características que diferenciam o sistema Linux do Windows;
- entender como é a distribuição dos diretórios e dos arquivos em sistemas Linux;
- saber como funcionam as permissões nos diretórios de sistemas Linux;
- entender o que é o Shell em um sistema Linux, e como é seu funcionamento;
- entender a diferença entre Shell e script;
- conhecer os principais comandos ou procedimentos para a criação de um script para executar no Shell;
- saber mais sobre os tipos de Shell que existem.

Bibliografia

CORREIA, F. Shell Script. **Revista Programar**, 27 jan. 2008. Disponível em: <https://www.revista-programar.info/artigos/Shell-Script/>. Acesso em: 23/09/2019.

NEGUS, C. **Linux** – a Bíblia. Rio de Janeiro: Altas Books, 2014.

NEMETH, E. et al. **Manual Completo do Linux**. São Paulo: Pearson Prentice Hall, 2007.

ROMERO, D. **Começando com Linux**: comandos, serviços e administração. São Paulo: Casa do Código, 2014.

SALES, R. Conheça Linus Torvalds, o criador do Linux. **TechTudo**, 17 nov. 2011. Disponível em: <https://www.techtudo.com.br/artigos/noticia/2011/11/conheca-linus-torvalds-o-criador-do-linux.html>. Acesso em: 26/09/2019.

SOARES, W.; FERNANDES, G. **Linux Fundamentos**. 1. ed. São Paulo: Érica, 2010.

STEVENS, W. R. et al. **Programação de rede Unix**. 3. ed. Porto Alegre: Bookman, 2005.