

Curso 450

Linux Fundamentals in Cloud

Versão 2015_3.0



Editor Vim

O "**Vi**" é o editor básico do GNU/Linux, e está disponível em grande parte das distribuições Linux, o vim é uma versão mais completa e com mais recursos do que o "Vi", cujo seu significado é "**Vim = VI iMproved**".

Outros Editores de Texto

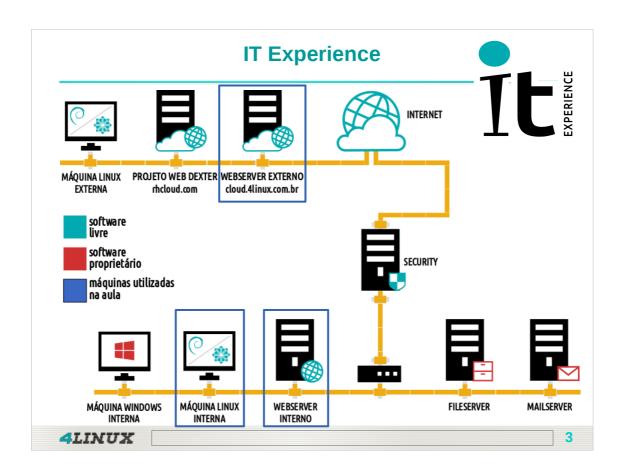
Vi – Sem dúvida nenhuma o editor mais famoso de todos os tempos, presente em quase todas as distribuições.

Nano – Editor padrão de muitas distribuições como Debian, CentOS, esse editor é diferente do "vim" e é muito fácil de ser usado.

Pico – Muito parecido com o "nano", este está presente nas distribuições Slackware e Gentoo.

Mcedit – Editor muito fácil e completo. Seu grande diferencial é a possibilidade da utilização do mouse, mesmo no ambiente textual.

Ed – O editor de textos mais simples no mundo Unix, o "ed" é um editor de linha para terminais aonde não é possível abrir uma janela de edição.



Anotaçoes:		

Objetivos da Aula

Aula 03 (parte 1/2)

- Conhecer o Editor de Texto Vim:
 - ➤ Principais funcionalidades do Vim;
 - ➤ Personalizando o Vim (vimrc);
 - ➤ Personalizando as mensagens do sistema:
 - /etc/issue e /etc/issue.net
 - /etc/motd



4LINUX

_ _

Anotações:			
	 · · · · · · · · · · · · · · · · · · ·	 _	
	 	 	1 1

Objetivos da Aula

Aula 03 (parte 2/2)

- Customizar o servidor para melhor administração:
 - ➤ O que são Alias;
 - Variáveis no Shell;
 - ➤ Arquivos de Login:
 - /etc/bashrc (~/.bashrc)
 - /etc/profile (~/.bash_profile e .profile)



4LINUX

Anotações:			
			
			

Editores de Texto do GNU/Linux

Para nós administradores de sistema, ter o domínio dos editores de textos via linha de comando é imprescindível, pois constantemente temos a necessidade de alterar arquivos, visualizar o conteúdo e etc.

Neste capítulo, veremos os principais editores de texto no mundo **GNU/Linux** e suas principais funcionalidades.



4LINUX

-6

Anotações:		

Editor de Texto: Pico

4LINUX

```
UW PICO(tm) 4.6 File: index.html

SIDOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

WW PICO(tm) 4.6 File: index.html

SIDOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html
<html
<hr/>
<html
<h
```

Anotações:

Editor de Texto: Pico

- O mais clássico e histórico editor de texto chama-se "Pico" ("Pine Composer"). É a sessão de edição do texto de um antigo cliente para e-mail (originalmente em Unix) chamado "Pine";
- ➤ O Pine e o Pico, na realidade, não são softwares livres e foram desenvolvidos pelo Depto. de Computação Aplicada à Comunicação da Universidade de Washington;
- No início da Internet, quando a Web ainda não existia, o Pine era o principal cliente para mandar e receber e-mails.



4LINUX

Anotações:		

Editores de Texto Servidor: Máquina Linux Interna Criando arquivos com Pico: Vamos criar um arquivo com o Pico: 1# pico GNU/LINUX is OpenSource Menu do Pico: GAjuda Sair Justificar Onde está? Próx Pág Colar Txt Terro Atual Colar Txt Terro Spell 2# CTRL + O (^O) 3# pico.txt <enter> 4# CTRL + X (^X)

Anotações:		

Editor de Texto: Nano

```
iLE88Dj. :jD88888Dj:
.LGitE888D.f8GjjjL8888E;
                                   .d8888b. 888b 888 888
d88P Y88b 8888b 888 888
888 88888b 888 888
iE :8888Et. .G8888.
                                    D888,
                     :8888:
       D888,
       D888,
                       :8888:
       D888,
888W,
                       :8888:
                       :8888:
       W88W,
                       :8888:
                                    88888b. 8888b. 88888b. .d88b.
888 "88b "88b 888 "88b d88""88b
888 888 .d888888 888 888 888
       W88W:
                       :8888:
       DGGD:
                       :8888:
                        :8888:
                                     888 888 888 888 888 888 Y88..88P
888 888 "Y888888 888 "Y88P"
                        :W888:
                        :8888:
                        E888i
                         tW88D
```

4LINUX

Anotações:	

Editor de Texto: Nano

- Nano é um editor que deve ser executado a partir de um terminal e se concentra em simplicidade;
- ➤ É um clone do antigo editor de texto **Pico**, o editor para o cliente de e-mail Pine, que foi muito popular nos anos 1990, em UNIX e sistemas do tipo UNIX;
- O Pine foi substituído pelo Alpine e o Pico pelo Nano, mas algumas coisas não mudaram assim como a simplicidade de edição com o Nano.



4LINUX

Anotações:	

Editores de Texto Servidor: Máquina Linux Interna Criando arquivo com Nano: A criação de arquivos com o Nano, é exatamente igual ao editor de texto Pico: 1# nano GNU/LINUX is OpenSource Menu do Nano: GAjuda GG Gravar CR Ler o Arq M Pág Anter CK Recort Txt C Pos Atual Nano: 2# CTRL + O (^O) 3# nano.txt <enter> 4# CTRL + X (^X) GAIUMX 12

Anotações:			



Anotações:		

Editor de Texto: Vim

- ➤ O Vi é um dos mais antigos (1976) editores de texto em linha de comando, para Unix e Linux;
- É derivado do Ed, do Ex, Joe e de outros editores "de linha" rudimentares;
- O Vim (Vi Improved) é uma versão mais poderosa e maior em termos de espaço em disco e requisitos de memória do editor de texto vi.



4LINUX

Anotações:	

Primeiro contato com o Vim na máquina Webserver Interno: Modo comando: espera um comando a ser executado; Modo inserção: ativa edição de texto; the cd /root the vim primeiro.txt Tecle I para inserir o Texto Abaixo > GNU/Linux is Open Source Tecle ESC para sair do Modo Inserção e voltar para o Modo de Comando > Tecle: wq para Salvar e Sair >

Editor Vim

Ao invocar o "vim", este entra direto para o modo de "Visualização". Para modificar o arquivo, usam-se os modos de inserção, deleção e de substituição.

Para voltar ao modo de visualização, sempre se usa a tecla "ESC", de forma que para entrar no modo de inserção, use a tecla "i".

Criando um arquivo

A sintaxe para se criar um arquivo no vim é bem simples, sendo apenas o comando vim e o nome do arquivo a ser criado, onde o mesmo será salvo no diretório corrente a sua criação.

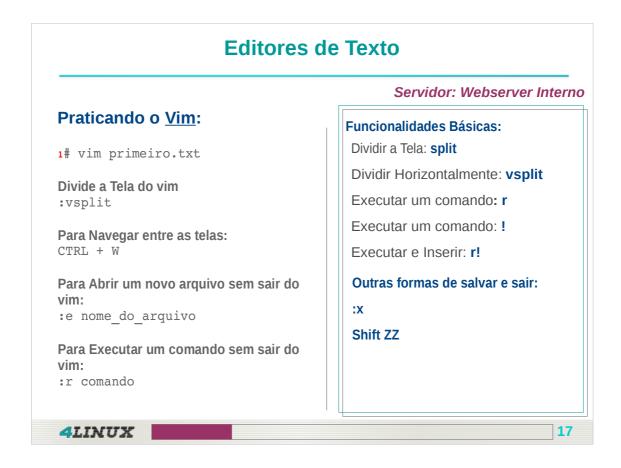
Após digitar o texto, você precisará sair do modo de inserção para realizar qualquer outra ação, para sair do modo de inserção digite **ESC**, assim como para salvar :w, salvar e sair :wq, e caso queira forçar uma ação, adiciona um ponto de exclamação no final :wq! e :q! .



Funcionalidades do Vim

As funcionalidades do Vim é extremamente útil para a praticidade de um SysAdmin, tendo como objetivo a agilidade para a edição de arquivos.

Caso queira conhecer mais funcionalidades do vim, você pode acessar o site: http://code.google.com/p/vimbook/downloads/list



Funcionalidades do Vim

As funcionalidades do Vim é extremamente útil para a praticidade de um SysAdmin, tendo como objetivo a agilidade para a edição de arquivos.

Caso queira conhecer mais funcionalidades do vim, você pode acessar o site: http://code.google.com/p/vimbook/downloads/list



Personalizando o Vim:

A grande maioria dos serviços em "Unix" são configurados através de arquivos de configuração, o "vim" não seria diferente. Seu arquivo de configuração é o "**vimrc**".

Para configurar o seu editor de texto, basta adicionar as funcionalidades desejadas no arquivo de configuração, encontrado em *letc/vim/vimrc* no Debian e *letc/vimrc* em distribuições Red Hat.



Arquivos de Mensagens

Os arquivos "/etc/issue", /etc/issue.net e "/etc/motd" são usados para mostrar mensagens para os usuários e não interferem na parte operacional do sistema.

A diferença entre o arquivo "/etc/motd" para os outros arquivos de mensagem, é ele exibe uma mensagem após o usuário se "logar" no sistema.

Enquanto o "/etc/issue" e "/etc/issue.net" exibe uma mensagem para o usuário antes que o mesmo faça "login" no sistema, sendo que o "/etc/issue.net" é destinado apenas para logins remotos.



Arquivo de Mensagem

Por padrão, as distribuições traz no arquivo de mensagem /etc/issue a versão do Kernel e a Arquitetura do Sistema.

As informações fornecidas por padrão pelas distribuições não é uma boa prática de segurança, portanto o ideal é alterar a informação padrão por uma mensagem personalizada.



Personalizando arquivos de mensagem:

Alterar os banners padrão das distribuições é uma boa prática de segurança, portanto nesse laboratório iremos adicionar um banner da empresa Dexter Courier.



Anotações:		
	 	 · · · · · · · · · · · · · · · · · · ·
	 	



Copiando o issue:

- 1# cd /etc
- 2# cp issue issue.net
- > Arquivo issue:
- 4# logout

DEU TUDO CERTO!

Servidor: Webserver Interno / Webserver Externo

Particularidades da issue:

Devido às variáveis do getty, será necessário proteger todas as \ do desenho em Ascii.

O arquivo issue.net, pelo contrário, é um arquivo texto normal, portanto não precisa de exceção na barra invertida.

Veja um exemplo de proteção:

- 1# echo L!nux
 - -bash: !nux: event not found
- 2# echo L\!nux

L!nux

4LINUX

Anotações:		



Servidor: Webserver Interno / Webserver Externo

Redirecionadores:

No Linux existem alguns redirecionadores para facilitar a manipulação de Arquivos no terminal:

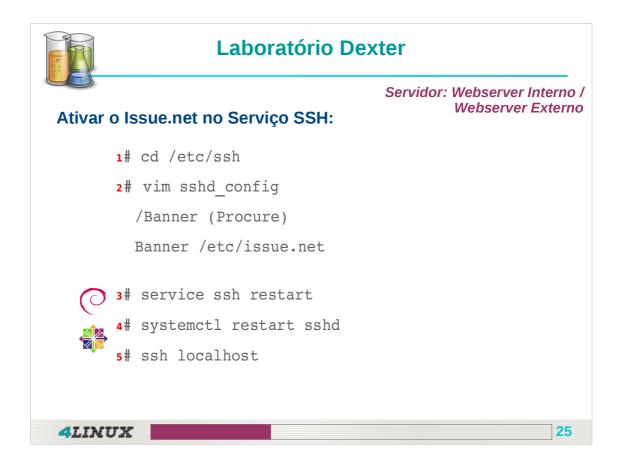
- > Redireciona o conteúdo apagando o arquivo de destino.
- >> Redireciona o conteúdo acrescentando ao final do arquivo de destino.

Veja um exemplo:

- 1# echo "Linux is Free" > /tmp/teste.txt
- 2# cat /tmp/teste.txt
- 3# echo "Linux is Free" >> /tmp/teste.txt
- 4# cat /tmp/teste.txt
- 5# echo "Linux is GPL" > /tmp/teste.txt

	T.	Τ	a	ſ	IJ	X	
_			•		•		ľ

Anotações:	



Arquivo de Mensagem /etc/issue.net

Como a mensagem contida no arquivo /etc/issue.net é exibida somente no acesso remoto, é necessário "ativar" esse banner contido no arquivo de configuração do servidor SSH.



Servidor: Webserver Interno / Webserver Externo

Colocar no arquivo motd um aviso aos analistas:

1# vim /etc/motd

Seja Cauteloso!

Lembre-se de sempre realizar

copias de segurança ao

editar arquivos em producao.

2# ssh localhost (Remoto)

3# logout (Local)

Motd - Message of the Day

Diferente do issue, o arquivo motd é único, a mensagem irá aparecer tanto para quem logar no terminal, quanto para um usuário que logar via SSH.

Ajuste o letc/motd nos dois Servidores da Dexter!

4LINUX

Anotações:		





Retirar os Banners que as distribuições colocam por padrão no arquivo issue e issue.net, informando a versão da distribuição e do kernel, é uma boa prática de segurança.

Nesse momento, os servidores da Dexter possuem seus arquivos de login personalizados, seguindo as boas práticas de segurança.

4LINUX 27

Anotações:		



Desafio: Migração do Site da Dexter

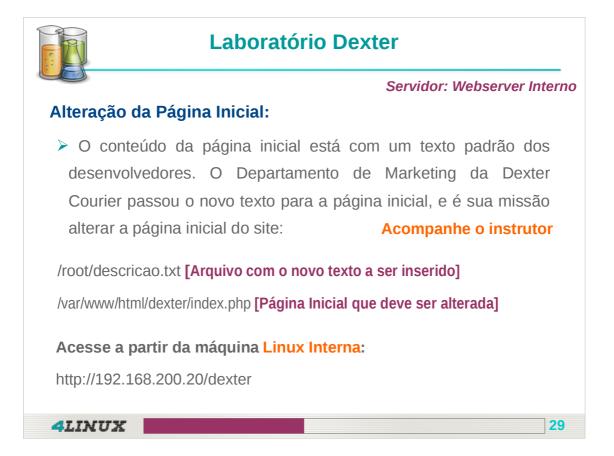
Antes de migrarmos o site da Dexter Courier para a Cloud, precisamos fazer algumas alterações no site, pois o mesmo encontra-se na reta final para a migração, restando apenas a alteração de alguns textos na página inicial (index.php).



28

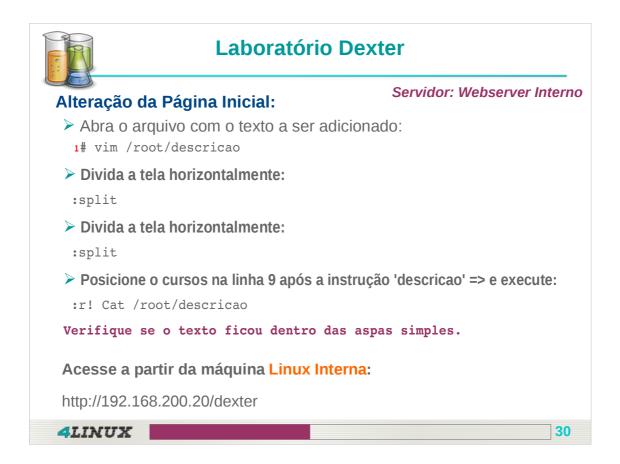
Configuração Estática de Rede

Tudo que vimos até agora, são configurações que podem ser atribuídas através de linha de comando (configurações dinâmicas).



Configuração Estática de Rede

Tudo que vimos até agora, são configurações que podem ser atribuídas através de linha de comando (configurações dinâmicas).



Configuração Estática de Rede

Tudo que vimos até agora, são configurações que podem ser atribuídas através de linha de comando (configurações dinâmicas).



Configuração Estática de Rede

Tudo que vimos até agora, são configurações que podem ser atribuídas através de linha de comando (configurações dinâmicas).

Pergunta LPI

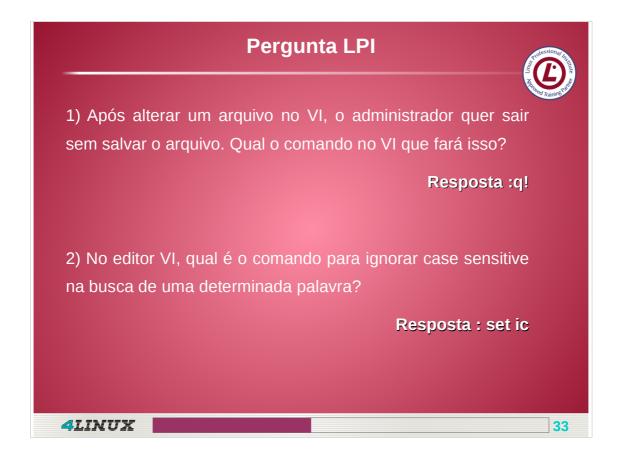


1) Após alterar um arquivo no VI, o administrador quer sair sem salvar o arquivo. Qual o comando no VI que fará isso?

2) No editor VI, qual é o comando para ignorar case sensitive na busca de uma determinada palavra?

4LINUX

Anotações:			



RESPOSTA CORRETA: :q!

O comando :q permite sair de um arquivo, sendo que para sair sem salvar basta adicionar o "!" (ponto de exclamação) para forçar a saída.

RESPOSTA CORRETA: :set ic

A opção **set ic** do vim permite ignorar o case-sensitive que por padrão é aplicado sobre buscas efetuadas dentro do editor de textos.



Anotações:		

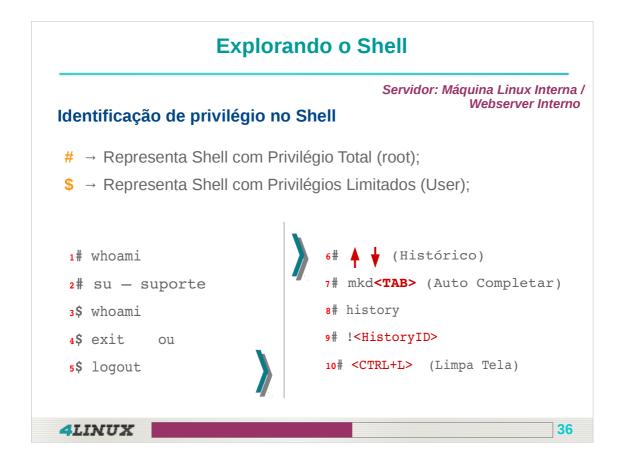
Administração do Shell Shell A "shell" é uma camada de acesso ao sistema básico, o sistema operacional do computador, que pode ser acessada tanto pelo modo gráfico quanto em modo texto. Aplicações Shell Kernel Hardware

O que é uma shell?

Podemos definir uma "shell" como a camada de acesso ao sistema básico no sistema operacional do computador.

Uma "shell" pode ser personalizada para atender as necessidades do administrador. Como exemplos de personalização podemos citar a definição de um idioma padrão, personalização e automatização de processos entre outras coisas.

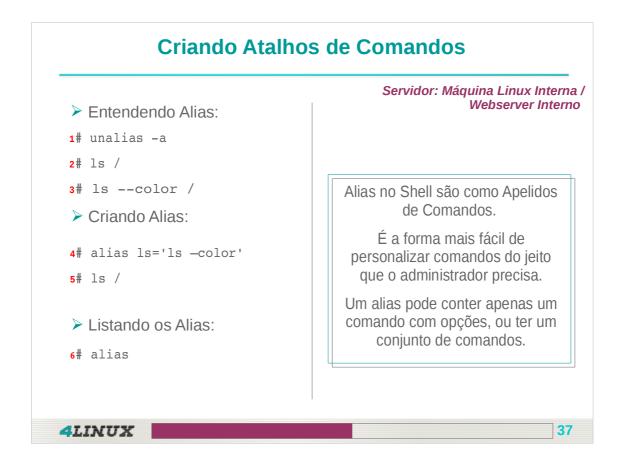
Nos tópicos a seguir, veremos como fazer algumas dessas personalizações nos servidores da empresa Dexter.



History

O comando history é extremamente útil para checar históricos de comandos digitados, além disso o "history" possui a conveniente função de "invocar" comandos digitados através de sua identificação dentro do histórico geral, o chamado "history id".

Os históricos gerados pelo history são armazenados no arquivo .bash_history localizado no diretório home de cada usuário, esta localização pode ser modificada como veremos durante este capítulo.



Alias

Um recurso do "shell" que facilita muito a vida do administrador é a definição de "aliases" para comandos, os "aliases" permitem que um determinando comando seja invocado através de um apelido, seja para facilitar o acesso a sua execução ou para traduzir em formato simples um comando complexo.

Imagine que um determinado usuário gosta de utilizar o comando "Is" sempre com os parâmetros "--color" e "-la". O que seria dele se toda vez que fosse executá-lo tivesse que escrever o comando com todos os parâmetros?!

NOTA:

Muitas distribuições já possuem uma série de alias pré-configurados.

Variáveis de Ambiente: Definição

Todo software que é executado no Linux precisa de várias informações para que possa funcionar corretamente: nome de usuário, tamanho do terminal, tipo do terminal, localização do executável, localização de alguma biblioteca, etc.

Sem este tipo de informação não é possível utilizar o sistema de modo produtivo já que seria necessário passar tais informações a cada programa que for usar – e todas as vezes que for usar o mesmo programa.

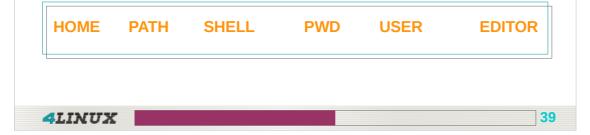


Anotações:		

Servidor: Webserver Interno

Quais são as variáveis mais importantes?

Algumas variáveis você vai encontrar em qualquer sistema Linux que você usar na vida. Elas definem alguns parâmetros importantes para que você consiga usar o sistema tranquilamente, sem ter que ficar lembrando parâmetros antes de começar a usar o sistema. Algumas destas variáveis são:



Anotações:			

Servidor: Webserver Interno

Quais são as variáveis mais importantes?

- ➤ **HOME** → Define o diretório home do usuário logado;
- > PATH → Define os diretórios usados para encontrar os comandos;
- ➤ **SHELL** → Define o shell que está sendo utilizado;
- **PWD** → Define em qual diretório você está no momento;
- **► USER** → Define o usuário que está logado;
- ➤ **EDITOR** → Define o editor de textos padrão para aplicações que invocam editores de texto automaticamente.

4LINUX 4

Anotações:		

Servidor: Webserver Interno

Variáveis de Ambiente: Importância

Se não existisse uma variável de ambiente chamada **PATH** você teria que digitar todo o caminho do comando para listar por exemplo:

1# /bin/ls



Quando você digita o comando **Is**, o sistema busca esse comando em algum diretório que esteja na variável **PATH.**

- ➤ Visualizar o conteúdo de uma variável:
- 2# echo \$PATH
 usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

4LINUX

Anotações:		

Servidor: Webserver Interno

Alterando idioma do sistema:

- O idioma usado em seu sistema pode ser modificado facilmente através das variáveis de ambiente;
- Atualmente a maioria dos programas estão sendo localizados;
- ➤ A localização é um recurso que especifica arquivos que contém as mensagens do programas em outros idiomas;
- Você pode usar o comando locale para listar as variáveis de localização do sistema e seus respectivos valores.
- 1# locale

4LINUX 42

Anotações:		

Servidor: Webserver Interno

Alterando idioma do sistema:

As principais variáveis usadas para determinar qual idioma os programas localizados utilizarão são:

- ➤ LANG → Especifica o idioma_PAIS local. Podem ser especificados mais de um idioma na mesma variável separando-os com :, desta forma caso o primeiro não esteja disponível para o programa o segundo será verificado e assim por diante;
- ➤ LC_MESSAGES → Especifica o idioma que serão mostradas as mensagens dos programas. Seu formato é o mesmo de LANG;
- ▶ LC_ALL → Configura todas as variáveis de localização de uma só vez. Seu formato é o mesmo de LANG.

4LINUX 43

Anotações:	

Servidor: Webserver Interno

Alterando idioma do sistema:

- > Altere as variáveis de idioma:
- 1# export LANG=pt_BR
- 2# export LC ALL=pt BR
- 3# export LC MESSAGES=pt BR
- As variáveis de ambiente podem ser especificadas no arquivo *letc/environment* desta forma as variáveis serão carregadas toda a vez que seu sistema for iniciado.

4LINUX

Anotações:		

Servidor: Webserver Interno

Tipos de Variáveis:

Existem dois tipos de variáveis:

- Variável LOCAL: disponível apenas no shell corrente;
- Variável GLOBAL:
 disponível em todos os
 "subshells".

Variáveis de ambiente (ou globais) têm seus nomes sempre em maiúsculas;

Variáveis de shell (ou locais) têm seus nomes sempre em maiúsculas;

O shell em si não faz distinção de nomes em maiúsculas ou minúsculas pra ele são a mesma coisa. A diferença é apenas para ajudar na interpretação do usuário.

4LINUX

Anotações:			

Servidor: Webserver Interno

Como criar uma Variável Local?

Para declarar uma variável local, basta definir o seu **NOME** e o **VALOR**:

- 1# LOCAL=valor
- 2# echo \$LOCAL
- 3# bash
- 4# echo \$LOCAL
- 5# exit

Repare que após você abrir um novo bash através do comando bash, o valor da variável LOCAL deixou de existir, pois estamos lidando com uma variável LOCAL, onde a mesma só existirá no shell em que foi declarada.

4LINUX

Anotações:			

Servidor: Webserver Interno

Como criar uma Variável Global?

Para declarar uma variável, basta definir o seu **NOME** eu o **VALOR**:

- 1# export CURSO=Linux
- 2# echo \$CURSO
- 3# bash
- 4# echo \$SHLVL
- 5# echo \$CURSO
- 6# exit

Quando você cria uma variável com o comando **EXPORT**, está declarando que a mesma terá o seu valor exportado **a partir** deste bash para todos os outros que vierem a ser abertos.

A Variável **SHLVL** tem como função o valor de qual bash você está logado. Por exemplo:

Caso você estivesse no seu primeiro bash, o valor da variável SHLVL seria 1, o segundo bash teria o valor 2 e assim sucessivamente.

4LINUX

Anotações:			
			
			



Anotações:		

Variáveis de Ambiente Servidor: Webserver Interno Validando o conceito de Variável Global: Verifique se a variável **GLOBAL** existe no seu 1 bash: 1# exit Variáveis Globais somente são válidas 2# exit do bash corrente para os subshells criadas a 3# echo \$SHLVL partir do shell em que variável foi 4# echo \$GLOBAL declarada. 49 4LINUX

Anotações:		

Servidor: Webserver Interno

Conhecendo a Variável PS1:

Agora vamos editar o nosso prompt de comando que é representado pela variável **PS1**.

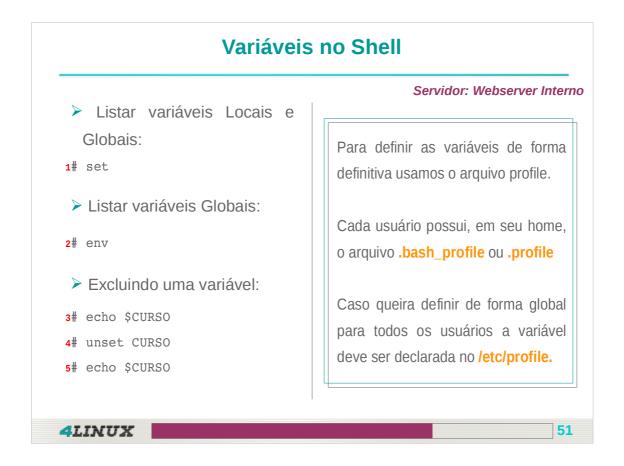
Argumento	Descrição
\VV	Diretório corrente
\vv	Caminho completo do diretório corrente
\u	Nome do usuário
\t	Hora do sistema
\d	Data
\h	Host da máquina

```
1# vim +10 /etc/profile
   export PS1="webserverinterno - [ \w ]# "
```

2# source /etc/profile



Anotações:		
	 	 · · · · · · · · · · · · · · · · · · ·
	 	



Visualizando variáveis com set e env

Variáveis locais e globais podem ser definidas utilizando os comandos "set" e "env" respectivamente.

Quando falamos em visualização de variáveis o comando "env" permite visualizar as variáveis globais, enquanto o comando "set", visualiza ambas as variáveis (Locais e Globais).

O comando "echo" é usado para imprimir algo na tela ou direcionar para um arquivo, na linha de comando o "echo" é muito útil para inspecionar variáveis.

O comando "printenv" também pode ser utilizado para definir variáveis globais.

Servidor: Webserver Interno

Arquivos de Configuração:

- Quando o usuário se loga, o sistema fará a autenticação, configurará o ambiente e iniciará o shell. No caso do bash, o próximo passo é a leitura do arquivo /etc/profile;
- Este arquivo contém comandos que são executados para todos os usuários do sistema no momento do login. Somente o usuário root pode ter permissão para modificar este arquivo;
- Este arquivo é lido antes do arquivo de configuração pessoal de cada usuário (.profile, .bash_profile, .bash_logout e .bash_login).

4LINUX 52

Anotações:			

Servidor: Webserver Interno

Ordem de Carregamento:

Se este arquivo /etc/profile existe o shell processa o conteúdo armazenado nos arquivos: /.bash_profile, /.bash_login e /.bash_logout nessa ordem. Se não existe nenhum o arquivo /etc/bashrc é executado.

~I.bash_profile → executado por shells que usam autenticação;
 ~I.bashrc → executado por shells que não usam autenticação;
 ~I.bash_login → executado quando um usuário efetua o login;
 ~Ibash_logout → executado quando um usuário efetua o logout.

4LINUX

53

Anotações:

Servidor: Webserver Interno

Executar um comando no login e logout:

Para que um comando seja executado quando o usuário efetua LOGIN (iniciar uma sessão) ou LOGOUT (finalizar a sessão), insira o seu comando nos respectivos arquivos que devem estar localizados no HOME de cada usuário:

.bash_login .bash_logout



Caso estes arquivos não existam, você deve criá-los.

- Exibir uma mensagem quando o usuário se logar no sistema.
- 1# vim /home/suporte/.bash_login
 echo Bem vindo ao Linux!

O arquivo ~/.bash_login somente será lido caso o arquivo ~/.bash_profile não exista.

4LINUX

Anotações:			
	 	 	
	 	 	1 1

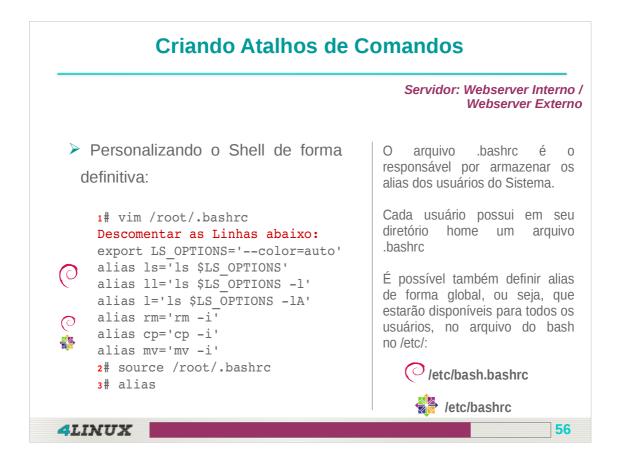
Servidor: Webserver Interno

Executar um comando no login e logout:

- Exibir uma mensagem quando o usuário se deslogar no sistema.
- 1# vim /home/suporte/.bash_logout
 echo "echo Obrigado por utilizar nosso sitema!"
 sleep 3

4LINUX 55

Anotações:			



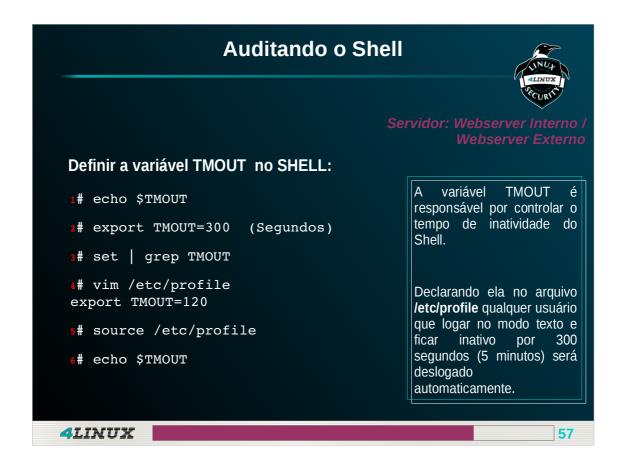
Outros arquivos de personalização:

Os arquivos abaixo podem ser encontrados em algumas distribuições e são opções para criação de variáveis permanentes, ou seja, variáveis que sejam declaradas todas vez que o sistema for iniciado/reiniciado.

/.profile/.bash-profile

Para executar funções durante os processos de login e logut utilize os seguintes arquivos:

/.bash_login /.bash_logout



O arquivo /etc/profile

Quando uma "bash" é executada como uma "shell" de "login" interativo ela lê e executa o arquivo "/etc/profile", se ele existir. Esse arquivo deve conter as configurações gerais que se aplicam a todos os usuários do sistema.

Sendo assim, para criar "aliases" ou definir variáveis ou funções que sejam comuns a todos os usuários, devemos incluí-las no arquivo "/etc/profile".

Caso o usuário necessite é possível adicionar configurações pessoais de personalização utilizando os arquivos "/.bash_profile", "/.bash_login", "/.profile" ou "/.bashrc".

Auditando o Shell Implementar Registro de Data e Hora no History: Para remover os comandos 1# history duplicados usa-se a variável 2# echo \$HISTSIZE HISTCONTROL. 3# echo \$HISTFILE A quantidade de comandos que será armazenado no 4# vim /root/.bashrc history é definido na variável export HISTSIZE=1000 HISTSIZE (Padrão CentOs export HISTCONTROL='erasedups' export HISTTIMEFORMAT='%d-%m-%Y %H:%M - ' 500 e Debian 1000). 5# source /root/.bashrc A personalização é feita na variável HISTTIMEFORMAT. 6# history 4LINUX

Anotações:		

History no Cloud

Servidor: Webserver Externo

Atenção!

No Servidor Cloud, especificamente no **Laboratório da 4Linux**, o history está desativado, pois o aluno não faz acesso via SSH ou Login e sim por meio de um comando específico do **OpenVZ** (vzctl).

Portanto, as configurações do history só funcionarão durante o tempo em que estiver logado.

4LINUX

Anotações:		
	 	 · · · · · · · · · · · · · · · · · · ·
	 	



Bash-completion

O recurso "bash completion" estende a função padrão de autocompletar do bash para atingir linhas de comando complexas com apenas algumas teclas.

Este projeto foi concebido para produzir rotinas de autocompletar programáveis para os comandos GNU/Linux mais comuns, reduzindo a quantidade de teclas que os administradores de sistemas e os programadores precisam apertar diariamente.



BASH Completion

Servidor: Webserver Externo

```
Instalar o pacote bash-completion para agilidade no shell:

1# apt-get install bash-completion

2# vim /etc/bash.bashrc

# enable bash completion in interactive shells

if ! shopt -oq posix; then

if [ -f /usr/share/bash-completion/bash_completion ];
then

. /usr/share/bash-completion/bash_completion

elif [ -f /etc/bash_completion ]; then

. /etc/bash_completion

fi

fi

fi

source /etc/bash.bashrc
```

4LINUX

Anotações:			



BASH Completion

Servidor: Webserver Externo

- ➤ Instalar o pacote bash-completion para agilidade no shell:
- 1# yum install bash-completion
- 2# logout
- 3# yum inst<TAB>

4LINUX

62

Anotações:

Pergunta LPI

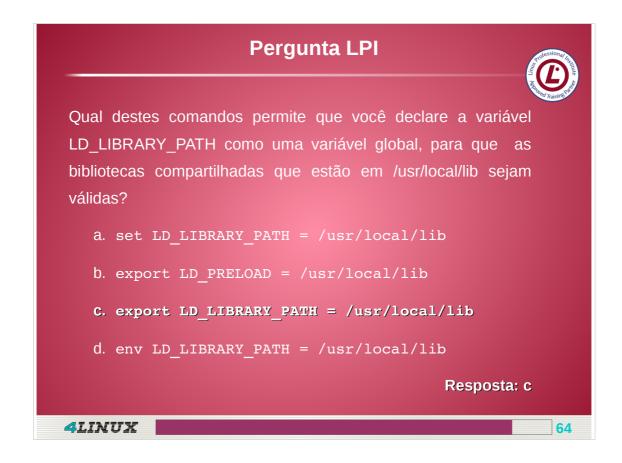


Qual destes comandos permite que você declare a variável LD_LIBRARY_PATH como uma variável global, para que as bibliotecas compartilhadas que estão em /usr/local/lib sejam válidas?

- a. set LD LIBRARY PATH = /usr/local/lib
- b. export LD PRELOAD = /usr/local/lib
- C. export LD_LIBRARY_PATH = /usr/local/lib
- d. env LD LIBRARY PATH = /usr/local/lib

4LINUX

Anotações:			



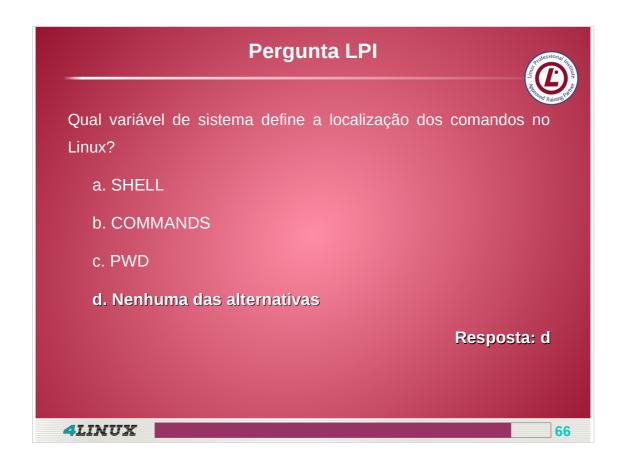
Alternativa C: RESPOSTA CORRETA!

O comando utilizado para exportar uma variável tornando-a global é o comando export, na situação descrita a variável a ser exportada é a variável LD_LIBRARY_PATH logo a alternativa correta é:

export LD_LIBRARY_PATH=/usr/local/lib

Pergunta LPI Qual variável de sistema define a localização dos comandos no Linux? a. SHELL b. COMMANDS c. PWD d. Nenhuma das alternativas

Anotações:		



Alternativa D: RESPOSTA CORRETA!

A variável responsável pela definição da localização de comandos é a variável PATH.

Próximos passos

Para que você tenha um melhor aproveitamento do curso, participes das seguintes atividades disponíveis no Netclass:

- > Executar as tarefas do Practice Lab;
- Resolver o **Desafio Appliance Lab** e postar o resultado no Fórum Temático;
- ➤ Resolver o **Desafio OpenCloud Lab** e postar o resultado no Fórum Temático;
- Responder as questões do **Teste de Conhecimento** sobre o conteúdo visto em aula.

Mãos à obra!

4LINUX

