



Certificações **NET.BR**

Pocket Book - 2ª Edição

E vamos que vamos para nossa segunda edição, com dicas de como passar nos exames 101 e 102 da certificação Linux Professional Institute. Todos os meses, conteúdos novos e inéditos.

made with
Beacon

Introdução

Ninguém aprende Linux se não o utilizá-lo. Não basta ler este livro, ou qualquer outro, o que vai diferenciar você é a quantidade de exercícios que você realiza diariamente com o sistema do pinguim.

Prof. Juliano Ramos

Dedicação, inspiração. São estados momentâneos; no final, é sua persistência que vai definir o profissional que você será. Nesta edição, vamos dar sequência ao nosso conteúdo sobre o exame 101 da LPIC-1 prova 4.0.

Isto mesmo, vamos direcionar a atenção para o exame atual.

Atualização de conteúdo

Existe uma vastidão de conteúdo na WEB sobre os exames da LPI (Linux Professional Institute), porém, poucos são os que revisam e atualizam o conteúdo.

A partir desta publicação, nosso foco será na prova 4.0, então, vamos que vamos e boa sorte!

Parceria com a Certificações NET.BR

O Certificações NET.BR é uma empresa Brasileira que oferece exame de certificação para profissionais de TI com expertise em Linux. Até então, os exames eram restritos para empresas, no entanto, a partir de maio o site vai começar a ofertar o exame **LTC - Linux True Certificate** que é um exame com foco na administração de sistemas linux.

Saiba mais sobre esta certificação:

www.certificacoes.net.br

Sobre o Autor

Juliano Ramos, conhecido também como grubelilo ou Prof. Juliano Ramos, é engenheiro da computação, autor de diversos blogs e e-books, consultor de linux em duas editoras, professor na Linux Force e certificado LPI além de outras certificações como Red Hat, Suse, Comptia e Cisco.

profjulianoramos@gmail.com



Desde pequeno eu já sabia o que iria ser: Professor, nas manhãs dominicais eu ensinava passagens bíblicas aos meus amigos, isto com uns oito anos, sempre gostava de estar a frente passando conhecimento.

Humildade

Ser humilde é ter personalidade. Acredito que é isto que faz as minhas aulas serem diferenciadas, eu sou humilde para ouvir.

"Ninguém é tão grande que não possa aprender, ou tão pequeno, que não possa ensinar."

Conto com seu feedback para melhorarmos este trabalho - Vamos que vamos!

Comando split e comando tac

O comando split

O comando split é utilizado para dividir arquivos de texto em arquivos menores. Seu padrão é dividir os arquivos a cada 1000 linhas. Os nomes dos arquivos de saída seguem o padrão.

Exemplo:

Crie um arquivo com 40 linhas e divida-o em 4 arquivos com 10 linhas:

```
split -4 arquivo1.txt arquivosaida
```

O comando tac

O comando tac é o inverso do comando cat, ele mostra o conteúdo de um arquivo de texto de trás para frente.

Exemplo:

```
tac arquivo.txt
```

Comando Tail e comando TR

O comando tail

O comando tail visualiza as 10 últimas linhas de um arquivo. Funciona como o oposto do comando head. As duas opções mais utilizadas do comando tail são:

- n Especifica o número de linhas finais que o tail irá mostrar de um arquivo
- f Mas as últimas linhas de um arquivo continuamente. Útil para arquivo de LOG.

Exemplo:

```
tail -n 50 /var/log/meulog
```

```
tail -f /var/log/meulog
```

O comando tr

O comando tr faz a troca de uma variável especificada. Ele não trabalha diretamente com arquivos, mas você pode utilizá-lo usando a saída de outro comando.

Exemplo:

```
cat arquivo1 | tr a-z A-Z
```

No exemplo acima o comando tr troca todas as letras de a à z para maiúsculas.

```
cat arquivo | tr -d a
```

Neste exemplo o comando tr apaga a letra a.

Comando WC e Xargs

O comando wc

O comando wc conta as linhas, palavras e caracteres de um arquivo.

Exemplo:

```
wc teste.txt
```

O comando xargs

O comando xargs utiliza a saída padrão de um comando como argumento para outro comando, exemplo:

```
cat texto.txt | xargs echo
```

Neste exemplo, cada linha do arquivo texto.txt é passada como argumento para o comando echo.

Editor VIM

O editor VIM tanto no UNIX como no Linux serve para criar arquivos de texto e scripts shells, assim como editá-los. O editor VIM possui dois modos: Edição, quando você aperta a tecla (I) ou (insert). E o modo de comando, quando você aperta a tecla (ESC).

Iniciando o VIM

Digite no bash: vim ou vim nome_de_arquivo aperte i para escrever e esc para sair e executar os comandos:

Comando	Explicação
:q (enter)	Sair do arquivo sem salvar
:q! (enter)	Sair sem salvar forçando
:wq (enter)	Salva o arquivo e sai do editor
:w (enter)	Salva o arquivo atual
:e arquivo1 (enter)	Abre o arquivo1
:r arquivo1 (enter)	Insere o arquivo no ponto onde esta o cursor
:u (enter)	Desfaz a ultima ação
:d ou dd (enter)	Apaga a linha atual
:yy	Copia a linha onde esta o cursor
:p	Cola o texto
:dd	Apaga a linha do cursor atual
:i	Inserir um texto
:v	Abrir o modo visual

Mais opções do vim

Substituição de texto no vim

Comando	Explicação
<code>:s/velho/novo</code>	Substitui a palavra velho pela palavra novo.
<code>% s/velho/novo/g</code>	substitua em todo o arquivo (%) , todas (g) as ocorrências de velho por novo

Opção do set – vim

Comando	Explicação
<code>:set aw</code>	Salva a cada alteração
<code>:set nu</code>	Mostra o número das linhas
<code>:set ff</code>	Converte o arquivo para DOS
<code>:set et</code>	Troca tab por espaços

Permissões de arquivos e diretórios

Neste capítulo estudaremos o sistema de permissões do linux. O sistema de permissões permite ao administrador do sistema (root) definir o nível de acesso ao usuários, grupos e outros arquivos e diretórios além de programas executáveis.

As permissões de acesso protege o sistema de algumas invasões e respectivos programas não autorizados. Por exemplo, no linux, podemos impedir que algum programa malicioso se instale na máquina, delete algum arquivo ou que seja transferido para outras pessoas via rede, de modo a invadir o sistema.

3 níveis de permissão

O linux gerencia o sistema de privilégios em 3 tipos:

1. Privilégio do dono
2. Privilégio do Grupo
3. Privilégio de outros

Cada um destes tipos é dividido em 3 níveis de permissões de acesso:

1. Permissão de leitura (r)
2. Permissão de escrita (w)
3. Permissão de execução (x)

Permissões de arquivos e diretórios II

As definições de propriedade do usuário e do grupo fazem parte do inode e ambas são atribuídas quando um arquivo é criado. Geralmente, o proprietário é o usuário que criou o arquivo. O grupo do arquivo normalmente é definido como padrão do seu criador. A propriedade de grupo adiciona flexibilidade em situações nas quais uma equipe compartilha arquivos. Os “outros” usuários são aqueles que não são membros do grupo do arquivo e também não são os proprietários. Para cada uma dessas três classes de usuários, o modo de acesso define três tipos de permissões, que se aplicam diferentemente para arquivos e diretórios. As permissões encontram-se listadas na tabela abaixo:

Permissão	Abreviatura	Permissão de arquivo	Permissão de diretório
Leitura	R	Examinar o conteúdo do arquivo	Listar o conteúdo do diretório
Escrita	W	Escrever no arquivo ou modificar o arquivo	Criar e remover arquivos no diretório
Execução	X	Rodar o arquivo como um programa	Acessar o diretório

Permissões de arquivos e diretórios III

Essas três permissões se aplicam as três classes diferentes de usuários: usuários, grupo e outros. Cada uma tem permissão de leitura, escrita e execução.

Visualizando privilégios e permissões

Exemplo:

```
ls -l /home/usuario
```

(Visualiza privilégios e permissões de arquivos)

```
ls -ld /home/usuario
```

(Visualiza privilégios e permissões de diretórios)

Ao executar o comando `ls -l` você terá como retorno, algo semelhante a :

```
drwx----- ... 2 wester ..... 512 Jan ... 29 23:30 .. Arquivos/
```

```
-rw-rw-r-- ... 1 wester ..... 280232 Dec .. 16 22:41... notas.txt
```

Permissões de arquivos e diretórios IV

O primeiro caractere das linhas acima indicam o tipo de arquivo:

d Diretório

b Arquivo de bloco

c Arquivo especial de caractere

p Canal

s Socket

- Arquivo "normal".

Repare agora que no restante da string ainda há 9 caracteres. Que são os 3 níveis de permissões (Dono, grupo e outros). usando o exemplo da segunda linha, arquivo (notas.txt):

rw- Permissões do proprietários

rw- Permissão do grupo

r-- Permissão para outros

Permissões de arquivos e diretórios V

Tabela de permissões comuns:

Comando	Descrição
---	Nenhuma permissão
r--	Permissão de leitura
r-x	Leitura e execução
rw-	Leitura e gravação
rwX	Leitura, gravação e execução

Alterando diversas permissões de uma só vez

Para diretório:

```
find /oracle -type d | xargs chmod 775
```

Para arquivos:

```
find /oracle -type f | xargs chmod 775
```

Mais comandos de busca

Comando updatedb

O comando updatedb atualiza a base de dados com o caminho e nome dos arquivos e diretórios do sistema de arquivos.

A base de dados geralmente fica em `/var/lib/slocate/slocate.db` e é executado no cron (sistema de agendamento de tarefas).

Comando whatis

Procura no banco de dados whatis por uma palavra exata, exemplo:

```
whatis ls
```

Comando apropos

Procura no banco de dados whatis por partes de uma palavra.

```
apropos mkdir
```

Sistema de arquivos

O sistema de arquivos do linux é tão robusto e flexível que possibilita ao usuário criar e manter arquivos em diferentes partições, diferentes discos e até em computadores remotos.

nomenclatura de discos no linux

Por padrão, os discos IDE são nomeados da seguinte forma no linux:

`/dev/hda` Disco IDE conectado na interface primária Mestre

`/dev/hdb` Disco IDE conectado na interface primária Escravo

`/dev/hdc` Disco IDE conectado na interface secundária Mestre

`/dev/hdd` Disco IDE conectado na interface secundária escravo

Podemos usar a mesma nomenclatura para discos do padrão SCSI ou SATA, bastando mudar de hda para sda. Exemplo:

`/dev/sda` Disco SCSI conectado no primeiro canal

`/dev/sdb` Disco SCSI conectado no segundo canal

`/dev/sdc` Disco SCSI conectado no terceiro canal

Limitação SCSI

O padrão SCSI não tem a limitação de 4 discos como o IDE podendo chegar até pelo menos 15 discos. Cada disco pode ter de uma a dezesseis partições. No linux cada partição é representada por um número inteiro.

Exemplo:

A primeira partição de um disco IDE conectado na interface primária mestre vai ser nomeada como `/dev/hda1`. Se neste disco existir uma segunda partição, o nome será `/dev/hda2` e assim por diante.

Partições primárias

Cada disco pode conter no máximo quatro partições primárias. Estas partições contêm necessariamente um sistema de arquivos e pelo menos uma deve ser criada.

As partições primárias são nomeadas:

`/dev/hda1`, `/dev/hda2`, `/dev/hda3` e `/dev/hda4`

* Para que a carga do sistema operacional seja possível, uma destas partições deve ser marcada como ativa

Limitação SCSI II

partições extendidas

Cada disco pode conter no máximo quatro partições primárias. Estas partições contêm necessariamente um sistema de arquivos e pelo menos uma deve ser criada.

As partições primárias são nomeadas:

`/dev/hda1`, `/dev/hda2`, `/dev/hda3` e `/dev/hda4`

* Para que a carga do sistema operacional seja possível, uma destas partições deve ser marcada como ativa

partições lógicas

As partições lógicas existem em conjunto com uma partição estendida e podemos ter de uma a doze partições deste tipo. Elas são nomeadas à partir do número cinco até dezesseis.

Exemplo:

`/dev/hda1` (Partição primária), `/dev/hda2` (Partição estendida), `/dev/hda5` (Partição lógica), `/dev/hda6` (Partição lógica)

(...)

`/dev/hda16` (Partição lógica)

Sobre a LPI

O Linux Professional Institute - LPI - é uma organização sem fins lucrativos, sediada na Califórnia - USA e constituída em 1999 pela comunidade Linux, e, desde então, desenvolve de forma acessível um programa de certificação em sistemas GNU/Linux reconhecido internacionalmente por empresas, empregadores e profissionais de TI. A certificação LPI está entre as mais procuradas do mundo por profissionais da área de Tecnologia da Informação. Certificar-se é uma forma de atestar conhecimentos profissionais, ou seja, validar o conhecimento e a eficiência de alguém em determinado assunto.

A principal vantagem da LPI sobre outras certificações Linux é a neutralidade de distribuição, pois as provas do LPI são baseadas no Linux Standard Base, um conjunto de normas que mantém a compatibilidade entre as diferentes versões e distribuições do sistema operacional. A certificação LPI é, portanto, independente da distribuição. O LPI é reconhecido como a primeira organização do mundo a defender e ajudar o uso profissional do Linux, Open Source e Free Software através de certificação profissional. Os exames de certificação do LPI são aplicados em milhares de lugares no mundo, em vários idiomas e com o apoio de empresários, fabricantes e instrutores.

- Formulado por uma comunidade de profissionais Linux, voluntários, fornecedores e educadores.
- Desafiador: se você não souber o assunto, não será aprovado.
- Neutro quanto a distribuição: Verifica o conhecimento em qualquer sistema Linux padrão. Relevante: Consulta milhares para determinar as habilidades que precisam ser testadas.
- Apoiado e patrocinado por um grande número de empresas e projetos Linux.
- Uma certificação que a comunidade Linux pode respeitar e se orgulhar.

Visite o site do LPI Mundial em: www.lpi.org

Conclusão

"Não se foque na quantidade do conteúdo e sim na sua capacidade de executá-los".

Prof. Juliano Ramos

Recebi um bom feedback do pocket book - primeira edição. O que mais me chamou a atenção foi um e-mail que me informou que mais de 20 páginas já não era mais um pocket book!

Concordo plenamente; não se foque na quantidade de conteúdo e sim na sua capacidade de executá-los.

Vou limitar as edições em no máximo 20 páginas, assim, é garantido que você irá testar todos os comandos e estudar mais sobre eles em outras documentações até o lançamento da próxima edição.

Lembre-se, livros de TI não são livros de romance, para você deitar no sofá e ler. Você precisa estar a frente do linux, digitando cada comando e buscando mais informações sobre eles na internet ou em outros livros.

Qualquer dúvida adicional, contem comigo.

Prof. Juliano Ramos - profjulianoramos@gmail.com