

030 - Certificação Linux LPI Essentials Command Line Basics

#F0550 001 - Apresentação

Navegação por diretórios e arquivos

#F0551 002 - Comando locate

O comando `locate` no Linux é usado para encontrar arquivos rapidamente no sistema, consultando um banco de dados previamente atualizado com o comando `updatedb`. Ele é mais rápido que `find`, pois não faz uma busca em tempo real.

Exemplo de uso:

1. Procurar um arquivo específico

```
locate nginx.conf
```

Isso listará todos os caminhos onde o arquivo `nginx.conf` está localizado.

2. Filtrar resultados com `grep`

```
locate .log | grep /var/log
```

Isso exibe apenas os arquivos `.log` dentro do diretório `/var/log`.

3. Atualizar o banco de dados (caso um arquivo recém-criado não seja encontrado)

```
sudo updatedb
```

#F0552 003 - Comando find

O comando `find` no Linux é usado para buscar arquivos e diretórios dentro de um sistema de arquivos, com critérios como nome, tipo, data de modificação, tamanho, entre outros.

Sintaxe básica:

```
find <caminho> <opções> <ação>
```

#F0553 004 - Mais opções para o comando find

Exemplo prático:

1. Buscar por um arquivo específico:

```
find /home -name "documento.txt"
```

◆ Procura por `"documento.txt"` dentro de `/home` e subdiretórios.

2. Encontrar diretórios específicos:

```
find /var/log -type d
```

◆ Lista apenas os diretórios dentro de `/var/log`.

3. Buscar arquivos modificados nos últimos 7 dias:

```
find /etc -type f -mtime -7
```

- ◆ Lista arquivos modificados nos últimos 7 dias dentro de /etc.

4. Localizar arquivos maiores que 100MB:

```
find / -type f -size +100M
```

- ◆ Encontra arquivos maiores que 100MB no sistema.

5. Executar uma ação (exemplo: remover arquivos antigos):

```
find /tmp -type f -mtime +30 -exec rm {} \;
```

- ◆ Remove arquivos em /tmp com mais de 30 dias.

6. Busca até um nível de diretório:

```
find /etc -maxdepth 2 -name passwd
```

- ◆ busca arquivo passwd em até dois níveis de diretório.

O find é poderoso e permite buscas detalhadas combinando múltiplos critérios. 🚀

Parte superior do formulário

Parte inferior do formulário

#F0554 005 - FHS, estrutura do sistema de arquivos

O **FHS (Filesystem Hierarchy Standard)** define a estrutura padrão dos diretórios em sistemas Linux e Unix. Ele organiza os arquivos para manter a consistência e facilitar a administração.

Estrutura básica do FHS:

- / → Diretório raiz, contém todo o sistema.
- /bin → Binários essenciais do sistema (ex.: ls, cp, mv).
- /sbin → Binários administrativos (ex.: fdisk, iptables).
- /etc → Arquivos de configuração do sistema (ex.: /etc/fstab, /etc/passwd).
- /home → Diretórios dos usuários (ex.: /home/user).
- /var → Dados variáveis (logs, cache, filas de impressão, etc.).
- /usr → Programas e bibliotecas de usuários.
 - /usr/bin → Binários não essenciais.
 - /usr/lib → Bibliotecas compartilhadas.
 - /usr/local → Softwares instalados manualmente.
- /tmp → Arquivos temporários.
- /opt → Softwares de terceiros.
- /boot → Arquivos de inicialização do sistema.
- /dev → Dispositivos do sistema (ex.: /dev/sda, /dev/null).
- /mnt e /media → Pontos de montagem temporários e de dispositivos removíveis.

Exemplo prático:

Se você instala um software manualmente, ele pode ser colocado em /opt, enquanto pacotes de sistema instalados pelo gerenciador vão para /usr/bin. Logs gerados pelo sistema vão para /var/log, e os arquivos dos usuários ficam em /home/usuario

#F0555 006 - Navegando na estrutura de diretórios

Se você quer navegar na estrutura de diretórios no Linux, pode usar os seguintes comandos básicos:

Comandos essenciais:

1. Verificar onde está

```
pwd
```

Exibe o diretório atual.

2. Listar arquivos e diretórios

```
ls -l
```

Mostra arquivos e diretórios no formato de lista.

```
ls -la
```

Lista tudo, incluindo arquivos ocultos (. e ..).

3. Mudar de diretório

```
cd /caminho/do/diretorio
```

Acessa um diretório específico.

```
cd ..
```

Volta um nível na estrutura.

```
cd ~
```

Vai para o diretório **home** do usuário.

4. Criar e remover diretórios

```
mkdir novo_diretorio
```

Cria um novo diretório.

```
rmdir diretorio_vazio
```

Remove um diretório vazio.

```
tree -L 1
```

Mostra a estrutura de diretórios em formato de árvore (pode ser necessário instalar: `sudo apt install tree`).

#F0556 007 - Atalhos do CD

Aqui estão os atalhos do comando `cd` no Linux para navegação rápida:

- `cd /` → Vai para a raiz do sistema.
- `cd ~` ou `cd` → Vai para o diretório **home** do usuário.
- `cd ..` → Sobe um nível na hierarquia de diretórios.
- `cd -` → Alterna entre o diretório atual e o anterior.
- `cd ../..` → Sobe dois níveis.
- `cd nome_do_diretorio` → Entra no diretório especificado.
- `cd !$` → Repete o último argumento do comando anterior.

#F0557 008 - Path absoluto e relativo

Em sistemas Linux, os caminhos (paths) são usados para localizar arquivos e diretórios no sistema de arquivos. Existem dois tipos principais de caminhos: absolutos e relativos.

Caminho Absoluto:

- Um caminho absoluto sempre começa com o diretório raiz ("/").
- Ele fornece o caminho completo e inequívoco para um arquivo ou diretório, independentemente de onde você esteja no sistema de arquivos.
- Exemplo: `"/home/usuario/documentos/arquivo.txt"`

Caminho Relativo:

- Um caminho relativo é definido em relação ao diretório de trabalho atual.
- Ele não começa com "/".
- Ele especifica a localização de um arquivo ou diretório com base em sua posição atual no sistema de arquivos.
- Exemplo: Se você estiver no diretório `"/home/usuario/"`, o caminho relativo `"documentos/arquivo.txt"` se refere ao mesmo arquivo que o caminho absoluto `"/home/usuario/documentos/arquivo.txt"`.

Em resumo:

- Caminho absoluto: caminho completo, começando do diretório raiz ("/").
- Caminho relativo: caminho em relação ao diretório atual.

Pontos importantes:

- O uso de caminhos absolutos garante que você sempre acesse o arquivo ou diretório correto, independentemente de onde você esteja no sistema de arquivos.
- Caminhos relativos podem ser mais convenientes quando você está trabalhando dentro de uma estrutura de diretórios específica.