

Roteiro de Prática: Gerenciamento e Compactação de Arquivos no Linux

Objetivo: Compreender a diferença entre *empacotar* e *compactar*, e aprender a utilizar as ferramentas essenciais do Linux: tar, gzip e bzip2.

1. Introdução Teórica

Antes de começarmos a digitar comandos, é fundamental entender a diferença entre duas ações que muitas vezes são confundidas:

A. Empacotamento (Archiving)

Imagine que você vai viajar e tem 10 camisetas soltas. É difícil carregar todas na mão. Você pega uma **mala** e coloca todas dentro.

- **A ferramenta:** O comando **tar**.
- **O que faz:** Ele agrupa vários arquivos em um único arquivo (chamado de "pacote" ou "archive").
- **Detalhe importante:** A mala não diminui o peso das roupas. O pacote tar tem o tamanho somado de todos os arquivos dentro dele.

B. Compactação (Compression)

Agora imagine aquelas **sacolas a vácuo**. Você coloca as roupas, tira o ar, e elas ocupam muito menos espaço.

- **As ferramentas:** Comandos como **gzip** e **bzip2**.
- **O que fazem:** Usam cálculos matemáticos para reduzir o tamanho do arquivo.
- **Detalhe importante:** Compactadores, por padrão, trabalham em um arquivo por vez.

No mundo real: Para obter organização e economia de espaço simultaneamente, nós usamos os dois juntos (Empacotamos e depois Compactamos).

2. Preparação do Ambiente

Vamos criar um diretório de trabalho e gerar alguns arquivos de teste. Usaremos um arquivo de sistema (/etc/services) e faremos cópias dele, pois é um arquivo de texto grande, ideal para testar compressão.

Execute os comandos abaixo no seu terminal:

Bash

1. Cria a pasta da aula e entra nela

```
mkdir laboratorio_compactacao
```

```
cd laboratorio_compactacao
```

2. Cria 3 cópias de um arquivo grande para nossos testes

```
cp /etc/services arquivo1.txt
```

```
cp /etc/services arquivo2.txt
```

```
cp /etc/services arquivo3.txt
```

3. Liste os arquivos e observe a coluna de tamanho (Size)

```
ls -lh
```

(A opção -h no ls mostra o tamanho em formato legível, como "K" para Kilobytes ou "M" para Megabytes).

3. O Agrupador: Comando tar

Primeiro, vamos apenas agrupar os arquivos, sem diminuir o tamanho.

Comando:

Bash

```
# Sintaxe: tar [opções] [nome_do_pacote] [arquivos]
```

```
# -c: Create (Criar) | -v: Verbose (Mostrar na tela) | -f: File (Arquivo)
```

```
tar -cvf backup_simples.tar arquivo1.txt arquivo2.txt arquivo3.txt
```

Análise:

Execute **ls -lh**.

Observe o tamanho do backup_simples.tar. Ele é, basicamente, a soma dos três arquivos de texto. Não houve economia de espaço, apenas organização.

4. O Compactador Rápido: gzip

O gzip é o padrão da indústria para velocidade. Vamos testá-lo isoladamente.

Comando:

Bash

```
gzip arquivo1.txt
```

Análise:

Execute **ls -lh**.

1. O arquivo arquivo1.txt original **sumiu**. O gzip o substituiu pelo arquivo1.txt.gz.
2. Compare o tamanho dele com os outros arquivos. A redução deve ser drástica.

Como reverter (Descompactar):

Bash

```
gunzip arquivo1.txt.gz
```

5. O Compactador Potente: bzip2

O bzip2 possui um algoritmo mais forte. Ele demora mais para processar, mas geralmente comprime mais que o gzip.

Comando:

Bash

```
bzip2 arquivo2.txt
```

Análise:

Execute `ls -lh`.

Veja o arquivo `arquivo2.txt.bz2`. Em arquivos muito grandes, ele costuma ficar menor que a versão `.gz`.

Como reverter (Descompactar):

Bash

```
bunzip2 arquivo2.txt.bz2
```

6. O Combo Profissional (Tarball)

No dia a dia de um Administrador de Sistemas, não rodamos o tar e depois o gzip separadamente. O comando tar é inteligente e pode chamar os compactadores automaticamente usando "flags" (opções) especiais.

A. Criando um .tar.gz (Padrão Web/Linux)

Usaremos a flag `-z` (de gZip).

Bash

-z: Chama o gzip automaticamente

```
tar -czvf backup_site.tar.gz arquivo1.txt arquivo2.txt
```

B. Criando um .tar.bz2 (Máxima Compressão)

Usaremos a flag `-j` (o bzip2 é associado à letra j).

Bash

-j: Chama o bzip2 automaticamente

```
tar -cjvf backup_logs.tar.bz2 arquivo2.txt arquivo3.txt
```

7. Restaurando Backups (Extração)

Agora vamos simular um desastre (perda de dados) e recuperar nossos arquivos usando o backup que acabamos de criar.

1. O Desastre:

Apague todos os arquivos de texto originais.

Bash

```
rm *.txt
```

```
ls -lh
```

(Agora você só deve ter os arquivos compactados na pasta).

2. A Recuperação:

Para extrair, trocamos a flag -c (Create) pela flag -x (Extract).

Bash

```
# Extraíndo o backup .tar.gz
```

```
tar -xzvf backup_site.tar.gz
```

```
# Verifique se os arquivos voltaram
```

```
ls -lh
```

8. Resumo dos Comandos

Ação Desejada	Ferramenta	Flag do Tar	Exemplo de Comando
Apenas Agrupar	tar	(nenhuma)	tar -cvf arquivo.tar ...
Agrupar + Gzip	tar + gzip	-z	tar -czvf arquivo.tar.gz ...
Agrupar + Bzip2	tar + bzip2	-j	tar -cjvf arquivo.tar.bz2 ...
Extrair (Qualquer)	tar	-x	tar -xvf arquivo_compactado

Desafio Prático

Para fixar o conteúdo, realize a tarefa abaixo sozinho(a):

1. **Crie um diretório chamado meus_documentos.**
2. **Entre nele e crie 3 arquivos vazios (touch doc1 doc2 doc3).**
3. **Volte para a pasta anterior.**
4. **Crie um arquivo compactado .tar.bz2 contendo o diretório meus_documentos inteiro.**
5. **Remova o diretório original meus_documentos.**
6. **Descompacte o arquivo criado para provar que o backup funcionou.**

