



Título do Curso:

Administração de Sistemas
GNU/Linux



Conteúdo do curso

ADMINISTRAÇÃO DE SISTEMAS GNU/LINUX

✓ Tópico 1: Introdução ao sistema operacional GNU/Linux.

✓ Tópico 2: Introdução ao Shell e comandos básicos.

• **Tópico 3: Manipulação de conteúdos com comandos no Shell.**

• Tópico 4: Comandos para gerenciamento do sistema e do Hardware.

• Tópico 5: Editor de Texto VI.

• Tópico 6: Administração de usuários e grupos.

• Tópico 7: Gerenciamento de permissões.

• Tópico 8: Gerenciamento de processos.

• Tópico 9: Sistemas de arquivos e particionamento.

• Tópico 10: Expressões regulares.

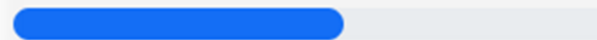
• Tópico 11: Introdução ao Shell Script.

• Tópico 12: Gerenciamento de Pacotes.

• Tópico 13: Agendamento de tarefas (cron) e Backup.

On Going

55%





DGP

Tecnologia da Informação

Tópico 3

Manipulação de conteúdos com
comandos Shell – Parte 2

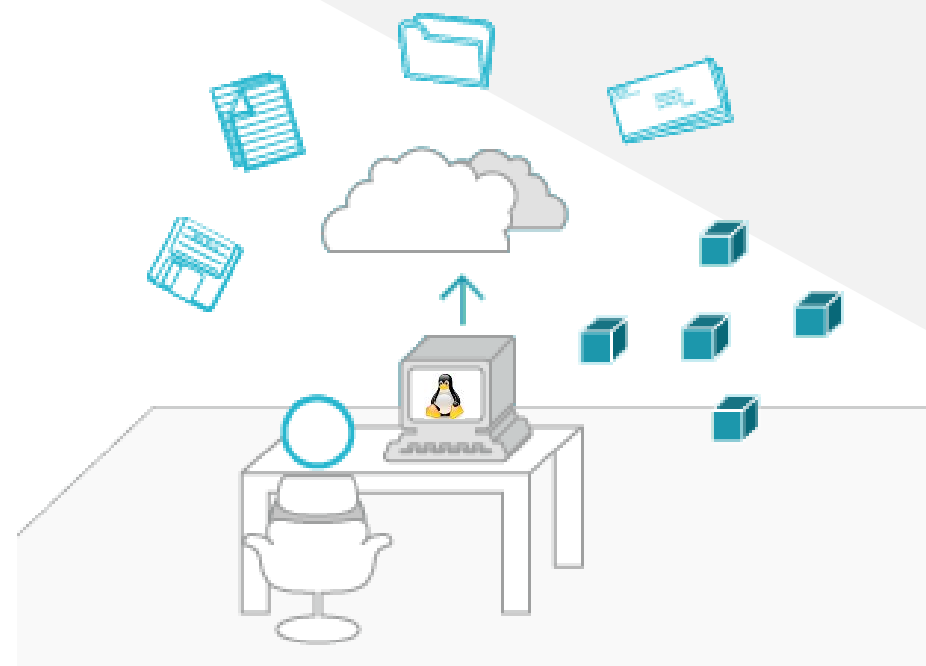


```
[root@localhost ~]# tail -n 50 /var/log/messages | grep kernel >> log-kernel.txt  
[root@localhost ~]#
```

Manipulação de conteúdos

com comandos no Shell – Parte 2

- Neste slide teremos comandos de compactação de arquivos (extensões “zip”, “tar.gz”, outros...), bem como comandos para filtrar dados, conforme listado abaixo:
 - **Filtros de conteúdo;**
 - **Empacotadores e Compactadores;**



Filtros de conteúdo

Tópico 3: Manipulação de conteúdos com comandos Shell.



Filtros

- Os filtros são extremamente importantes para administradores de sistemas GNU/Linux, possibilitando a “interceptação” e filtragem de informações que seriam enviadas para a saída padrão (**stdout**), fornecendo apenas as informações relevantes no momento.
- Geralmente o uso de filtros e o “pipe” (caractere de concatenação) são realizados em conjunto.



Filtros – Comando “grep”

- grep → Pesquisa em arquivos ou em sua entrada padrão por uma sequência de caracteres informada.
 - `grep <opções> [string_ou_regex_desejada] [arquivo ou caminho]`
 - Ex.: `grep “root” /etc/passwd`
 - Ex.: `cat /etc/passwd | grep “root”`
 - Os dois comandos acima apresentam o mesmo resultado (procura pela *string* “root” dentro do arquivo “/etc/passwd”), porém, no primeiro exemplo especificamos o arquivo e no segundo exemplo filtramos a sua entrada padrão.
- Com o comando “grep” (ou “egrep”), também podemos utilizar expressões regulares, caracteres curinga e “meta caracteres” (expressões regulares), criando inúmeras possibilidades.
 - Meta caracteres: São símbolos e caracteres literais com função específica (mais detalhes no slide sobre expressão regular).



Filtros – Comando “grep”

- Vamos supor que desejamos listar apenas os subdiretórios de um determinado diretório... O que fazer?
 - Ex.: `ls -l /lib | grep ^d`
 - No caso acima, utilizamos um meta caractere (o “^”, que representa o início da linha). Portanto, o “grep” filtrou toda linha que começa com “d”.
 - Ex.: `ls -l /lib | grep d`
 - Já neste exemplo, o “grep” filtraria todas as linhas que contém a letra “d” em qualquer parte da linha, ou seja, o objetivo não seria alcançado.
- Agora vamos supor que desejamos listar todo o conteúdo, exceto subdiretórios... Aí está um grande recurso do “grep”:
 - Ex.: `ls -l /lib | grep -v ^d`
 - O parâmetro “-v” exhibe o inverso do que foi solicitado, ou seja, vamos visualizar todo o conteúdo, exceto diretórios.



Filtros – Comando “grep”

- Uma opção muito utilizada é o “-r”, que nos permite realizar a busca de uma determinada string de forma recursiva:
 - Ex.: `grep -r root /etc`
 - Lembre-se que podemos concatenar comandos, caso a saída seja extensa.
 - Ex.: `grep -r root /etc | more`
 - Ex.: `grep -r root /etc | less`
- Podemos verificar apenas os arquivos e diretórios que foram criados/modificados em 2010:
 - Ex.: `ls -l /etc | grep 2010`
- Agora desejamos visualizar apenas os **arquivos** criados/modificados em 2010:
 - Ex.: `ls -l /etc | grep -v ^d | grep 2010`



Filtros – Comando “grep”

- Conforme citado anteriormente, temos inúmeras possibilidades com o comando “grep”, que vamos explorar apenas nas demandas do dia-a-dia de trabalho.
- Também é importante sabermos que além do “grep”, temos:
 - **egrep** → o mesmo que “grep -E” (Interpreta o padrão de busca como uma expressão regular estendida);
 - **zgrep** → Realiza a busca dentro de arquivos compactados com o padrão “gzip”;
- Em sistemas Microsoft, temos o comando “findstr” com função similar ao “grep”.

Filtros de conteúdo

Tópico 3: Manipulação de conteúdos com comandos Shell.



Filtros – Comando “wc”

- O comando “wc” (acrônimo de “*word count*”) realiza a contagem da quantidade de linhas, palavras ou caracteres de um determinado arquivo ou entrada padrão:
 - Ex.: `ls -l /etc | wc -l`
 - No exemplo acima, o “wc” tem como entrada padrão a listagem do diretório “/etc”. O parâmetro “-l” realiza a contagem de linhas.
 - Ex.: `cat /etc/passwd | wc -l`
 - Ex.: `wc -l /etc/passwd`
 - Ambos acima realizam a contagem de linhas do arquivo “/etc/passwd”.
- Vamos supor que desejamos saber a quantidade de subdiretórios contidos em “/etc”
 - Ex.: `ls -l /etc | grep ^d | wc -l`



Filtros – Comando “wc”

- Dentre as principais opções do comando “wc”, temos:
 - “-l” → Contabiliza a quantidade de linhas;
 - “-w” → Contabiliza a da quantidade de palavras;
 - “-m” → Contabiliza a quantidade de caracteres;
 - “-c” → Contabiliza a quantidade de bytes;
- Para contabilizar em apenas um comando a quantidade de linhas, palavras e caracteres, utilizamos o “wc” sem parâmetros:
 - Ex.: `wc /etc/passwd`

```
root@darkstar:/etc# wc /etc/passwd
24  32 964 /etc/passwd
```



Filtros – Comando “tr”

- O comando “tr” traduz e/ou deleta caracteres da entrada padrão e exibe o resultado como saída:
 - Ex.: `cat /etc/passwd | tr : ;`
 - No exemplo acima, o “tr” irá “traduzir” (converter) todos os caracteres “:” para “;”.

```
[root@localhost ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

```
[root@localhost ~]# cat /etc/passwd | tr ":" ";"
root;x:0:0:root:/root:/bin/bash
bin;x:1:1:bin:/bin:/sbin/nologin
daemon;x:2:2:daemon:/sbin:/sbin/nologin
```

- Ex.: `ls -l /etc | tr -s " "`
 - O “tr” também “comprime” (squeeze) caracteres repetidos. No exemplo acima, todos os locais que houver mais de um espaço são removidos (execute em sua VM para testar).

```
[root@localhost ~]# ls -l /
total 68
-rw-r--r--. 1 root root 0 Mar 18 16:33 a-teste.txt
lrwxrwxrwx. 1 root root 7 May 10 2019 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 Mar 3 12:55 boot
drwxr-xr-x. 19 root root 3000 Mar 28 08:34 dev
drwxr-xr-x. 107 root root 12288 Mar 3 12:55 etc
```

```
[root@localhost ~]# ls -l / | tr -s " "
total 68
-rw-r--r--. 1 root root 0 Mar 18 16:33 a-teste.txt
lrwxrwxrwx. 1 root root 7 May 10 2019 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 Mar 3 12:55 boot
drwxr-xr-x. 19 root root 3000 Mar 28 08:34 dev
drwxr-xr-x. 107 root root 12288 Mar 3 12:55 etc
```



Filtros – Comando “sed”

- O comando “sed” filtra e transforma texto:
 - Ex.: `sed '/s/nologin/false/' /etc/passwd`
 - No exemplo acima, o “sed” irá substituir a string “nologin” para “false”.
 - OBS.: O comando “grep” foi utilizado somente para “colorir” a saída e facilitar a visualização do que foi alterado com o comando “sed”.

```
[root@localhost ~]# cat /etc/passwd | grep nologin
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

```
[root@localhost ~]# cat /etc/passwd | sed 's/nologin/false/' | grep false
bin:x:1:1:bin:/bin:/sbin/false
daemon:x:2:2:daemon:/sbin:/sbin/false
adm:x:3:4:adm:/var/adm:/sbin/false
```

- O comando “sed” pode ser muito útil no dia a dia. Recomendo o link a seguir que possui diversos exemplos de uso:
 - Link: <https://linuxconfig.org/learning-linux-commands-sed>



Filtros – Comando “diff”

- diff → Compara o conteúdo de dois arquivos e exibe as diferenças:
 - diff <opções> [arquivo1] [arquivo2]
 - Ex.: `diff /etc/yp.conf /etc/yp.conf-bkp`
- Comando muito útil para compararmos as modificações realizadas em um arquivo de configuração com o arquivo original ou backup, caso o serviço apresente problemas após as modificações.

Conteúdo do arquivo 1

```
root@darkstar:~# cat arq1
Arquivo de texto "arq1"
Linha 2
Linha 3
Linha 4
```

Conteúdo do arquivo 2

```
root@darkstar:~# cat arq2
Arquivo de texto "arq2"
Linha 2
Linha 3
Linha 4
```

Comparando o conteúdo dos arquivos

```
root@darkstar:~# diff arq1 arq2
1c1
< Arquivo de texto "arq1"
---
> Arquivo de texto "arq2"
root@darkstar:~#
```




Filtros – Comando “sort”

- O comando “sort” ordena alfabeticamente um determinado arquivo ou entrada padrão:
 - Ex.: `cat /etc/passwd | sort`
 - Ex.: `sort /etc/passwd`
 - Ambos acima enviam para a saída padrão (tela) o conteúdo do arquivo “/etc/passwd” em ordem alfabética.
- Caso seja feita uma auditoria no setor de TI, solicitando uma lista com todos os usuários que possuem “shell válido” no Linux...
 - Ex.: `cat /etc/passwd | grep “/bin/bash” | cut -d ‘:’ -f 1 | sort`
 - Ex.: `sort /etc/passwd | grep “/bin/bash” | cut -d ‘:’ -f 1`
 - Ambos acima enviam para a saída padrão (tela), apenas o “login” dos usuários que possuem “shell válido”.



Filtros – Comando “cut”

- O comando “cut” exibe partes/seções de cada linha de um determinado arquivo ou entrada padrão:
 - Ex.: `cut -d: -f 1 /etc/passwd`
 - O comando acima separa os dados contidos no arquivo “/etc/passwd”, utilizando como delimitador de cada campo de informação, o caractere ‘:’.
- É importante ressaltarmos que esta sintaxe apenas faz sentido com arquivos que dispõe de um conjunto de dados de forma organizada, como um arquivo “.CSV” (dados separados por vírgula), bem como o “/etc/passwd” que utiliza o “dois pontos” como delimitador.

```
root@darkstar:~# cat /etc/passwd | head -n 3
root:x:0:0::/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
```



Filtros – Comando “cut”

- Obtendo e personalizando informações:
 - Visualizar apenas as três primeiras linhas do arquivo “/etc/passwd”:

```
root@darkstar:~# cat /etc/passwd | head -n 3
root:x:0:0::/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
```

- Visualizar as três primeiras linhas do arquivo “/etc/passwd” e filtrar a sua exibição, possibilitando a visualização apenas do “login” e caminho do Shell:

```
root@darkstar:~# head -n 3 /etc/passwd | cut -d: -f 1,7
root:/bin/bash
bin:/bin/false
daemon:/bin/false
```

- Exibir apenas o “login” e o caminho do Shell substituindo o delimitador com uma mensagem personalizada:

```
root@darkstar:~# head -n 3 /etc/passwd | cut -d: -f 1,7 --output-delimiter="-- Shell Path --> "
root -- Shell Path --> /bin/bash
bin -- Shell Path --> /bin/false
daemon -- Shell Path --> /bin/false
```



Filtros – Comando “awk”

- O comando “awk” possui função semelhante ao “cut”, porém, com muitas possibilidades adicionais:
 - Ex.: `awk -F: '{ print $1 }' /etc/passwd`
 - O comando acima separa os dados contidos no arquivo “/etc/passwd”, utilizando como delimitador de cada campo de informação, o caractere ‘:’ e exibe o primeiro campo de cada linha “\$1”.
 - Ex.: `ls -lh /var/log/ | awk -F " " '{ print "Nome do arquivo: " $9 " - Tamanho: " $5 }'`
 - O comando acima lista o conteúdo do diretório “/var” e envia a saída para o “awk” filtrar e exibir de forma customizada, somente o nome e o tamanho do arquivo. Vide resultado na imagem a seguir.

```
[root@localhost ~]# ls -lh /var/log/ | awk -F " " '{print "Nome do arquivo: " $9 " - Tamanho: " $5}'  
Nome do arquivo: anaconda - Tamanho: 4.0K  
Nome do arquivo: audit - Tamanho: 4.0K  
Nome do arquivo: boot.log - Tamanho: 3.4K  
Nome do arquivo: boot.log-20200322 - Tamanho: 20K
```

- OBS.: Estes são apenas exemplos para compreender o comando, porém, no dia-a-dia, podemos filtrar listas CSV, dados como RG, CPF, entre outras possibilidades.

Empacotadores e Compactadores

Tópico 3: Manipulação de conteúdos com comandos Shell.



Compactadores

- Os compactadores são muito utilizados em rotinas de Backup, onde podemos reduzir o tamanho dos arquivos, proporcionando o uso eficiente de recursos em disco ou mídias de Backup.
- Ao baixar o código fonte de um Software Linux, precisamos descompactar o arquivo antes de realizar a instalação do pacote.
- Além de compactados, os diversos arquivos contidos em um pacote de instalação de Software (código fonte), são “empacotados”, viabilizando a distribuição através de um único arquivo.



Compactadores

Comandos “zip” e “unzip”

- zip → Permite compactar arquivos no formato “zip” (Padrão PKZIP e Winzip utilizado no Windows).
 - zip <opções> [caminho_do_arquivo.zip] [arquivos_a_ser_compactados]
 - Ex.: `zip -r dados.zip /etc/`
 - O comando acima realiza a compactação do diretório “/etc” e todo o seu conteúdo, dentro do novo arquivo de nome “dados.zip”.
 - O parâmetro “-r” permite compactar de forma recursiva.
- unzip → Permite descompactar arquivos no formato “zip” ou apenas listar o conteúdo contido no arquivo compactado (parâmetro “-l”).
 - unzip <opções> [caminho_do_arquivo.zip] <opções> <destino>
 - Ex.: `unzip dados.zip`
 - Descompacta o conteúdo de “dados.zip” no diretório corrente.
 - Ex.: `unzip dados.zip -d /root/bkp-etc/`
 - Descompacta o conteúdo de “dados.zip” no diretório “/root/bkp-etc”.



Compactadores – Comando “tar”

- tar → Permite dois tipos de tarefas:
 - Empacotar dados em um arquivo sem realizar compactação “efetivamente”;
 - Compactar/Descompactar arquivos utilizando o padrão “gzip”.
- O objetivo de empacotar é viabilizar a distribuição de pacotes de instalação de Softwares, publicando apenas um arquivo ao invés dos diversos arquivos necessários para a instalação do Software.
- Ao compactar/descompactar através do comando “tar”, podemos utilizar três tipos de compactadores:
 - “gzip” → extensão “tar.gz” (mais utilizado e eficiente na compressão);
 - “bzip2” → extensão “tar.bz2”
 - “compress” → extensão “tar.Z”



Compactadores – Comando “tar”

- tar → Sintaxe:
 - tar <opções> [arquivo] <opções | | caminho_a_ser_compactado>
 - No próximo slide teremos exemplos de sintaxe. Portanto, vamos conhecer as principais opções e possibilidades de uso:
 - c → Compacta ou empacota dados em um novo arquivo;
 - x → Extrai o conteúdo de um arquivo compactado;
 - t → Lista o conteúdo de um arquivo compactado;
 - v → Exibe na tela o que está sendo compactado ou descompactado;
 - p → Preserva as permissões do arquivo de origem;
 - r → Acrescenta arquivos dentro do pacote “tar”;
 - z → Modo de operação com o comando compactador “gzip”;
 - Z → Modo de operação com o comando compactador “compress”;
 - j → Modo de operação com o comando compactador “bzip2”.
 - f → Modo de operação com arquivos (o padrão é dispositivo de fita);



Compactadores – Comando “tar”

- tar → Sintaxe para compactar (com o compactador “gzip”):
 - Ex.: `tar czf arquivo.tar.gz /etc/`
 - Compacta o diretório “/etc” e seu conteúdo, utilizando o compactador GZIP, dentro de um novo arquivo (`arquivo.tar.gz`).
- tar → Sintaxe para compactar (com o compactador “bzip2”):
 - Ex.: `tar cjf arquivo.tar.bz2 /etc/`
 - Compacta o diretório “/etc” e seu conteúdo, utilizando o compactador BZIP2, dentro de um novo arquivo (`arquivo.tar.bz2`).



Compactadores – Comando “tar”

- tar → Sintaxe para empacotar:
 - Ex.: `tar cf empacotado.tar /etc/`
 - “Agrupa”/“Empacota” o diretório “/etc” e seu conteúdo dentro de um novo arquivo (`empacotado.tar`), sem comprimir os dados.
- tar → Sintaxe para descompactar (com o compactador “gzip”):
 - Ex.: `tar xzf arquivo.tar.gz`
 - Descompacta o conteúdo do “`arquivo.tar.gz`” no diretório corrente.
- tar → Sintaxe para descompactar especificando o diretório:
 - Ex.: `tar xzf arquivo.tar.gz -C /diretorio/destino/desejado/`
 - Descompacta o conteúdo do “`arquivo.tar.gz`” no diretório especificado.



No próximo slide...

- Tópico 4: Comandos para gerenciamento do sistema e do Hardware.
 - Comandos para coletar informações do sistema;
 - Comandos para coletar e definir configurações de hardware.



Referências

- BONAN, Adilson Rodrigues. **LINUX – Fundamentos, Prática & Certificação LPI**. Editora: Alta Books. RJ. 2010;
- PEREIRA, Guilherme Rodrigues. **Slides para aula expositiva**. Centro Universitário UNA.
- SILVA, Gleydson Mazioli. **Guia Foca GNU/Linux**. Disponível em: <https://guiafoca.org/>



DGP

Tecnologia da Informação

Obrigado!



Guilherme Rodrigues