



# **SERVIDORES E SERVIÇOS DE INTERCONNECTIVIDADE LINUX** LINUX - POR ONDE COMEÇAR

Autor: Me Haroldo da Silva Ferreira e Me. Ramiro  
Cordova

Revisor: Alexandre Denicol

INICIAR



# introdução

## Introdução

Esta unidade foi concebida para de forma objetiva e clara apresentar a base de estudos do Sistema Linux, para ser um compêndio sobre temas iniciais e relevantes que servirão como base de estudos desta disciplina. O texto aqui disposto gira em torno do surgimento do Linux e dos cenários indicados para sua aplicação, além de identificar os princípios teóricos e apresentar os principais comandos para utilização do sistema.

Esta unidade foi concebida em quatro tópicos, de forma a oferecer uma organização didática que facilite a compreensão dos conteúdos. No tópico 1, aborda sobre o surgimento do Linux, descrevendo a maneira que surge o Linux. Na sequência, o tópico 2, trata de licenciamento e cenários de implantação do Linux. O tópico 3, é dedicado aos princípios teóricos do Linux e ao interpretador de comandos. Por fim, tratamos sobre a lógica e os comandos básico do Linux no tópico 4.

# O Surgimento do Linux

O Sistema Linux é um Sistema Operacional de código aberto sob licença GPL, baseado em Sistemas Unix, criado por Linus Torvalds em 1991, mas com kernel próprio apenas em 1994. Para entendermos melhor sobre o contexto histórico onde o Linux está incluído, precisamos nos aprofundar nos elementos importantes dessa história.

## Sistema Operacional

Como já dito, o Linux é um Sistema Operacional baseado no Unix. Mas o que é exatamente um Sistema Operacional? Trata-se da coletividade de programas que fazem a interface do usuário e seus programas com o computador, desde o arrastar do mouse até a execução mais complexa de um programa de computador.

Também é responsável pelo gerenciamento de periféricos, recursos (memórias, HD's, impressoras, CD/DVD, etc.) e interpretação de mensagens. No Linux a junção do *kernel* com o conjunto de ferramentas GNU formam o Sistema Operacional Linux, por isso é comum o Sistema Linux ser chamado

de GNU/Linux, mas chamaremos neste material apenas de Linux.

O *kernel* é a base principal de um Sistema Operacional, que pode ser construído ou otimizado de acordo com as especificações de computador e periféricos que possui. Um pacote, é formado por um arquivo, ou um conjunto de arquivos, contendo tudo o que é necessário para instalação de um programa (ou software). Um sistema operacional utilizável é composto por diferentes pacotes, *daemons* e outros programas. Esse conjunto composto por kernel mais os pacotes que compõem o sistema operacional é comumente chamado de distribuição pela comunidade Linux.

A mesma linhagem do *kernel* é compartilhada por todas distribuições Linux, porém as coleções de *software* livres que vêm junto com esse *kernel* podem variar de forma significativa entre as distribuições. Essas coleções de *software* livres começaram a ser desenvolvidas a mais de 30 anos por milhares de pessoas, e são conhecidas como conjunto de ferramentas GNU.

O conjunto de ferramentas GNU foram desenvolvidas através do projeto GNU (GNU's Not Unix), projeto criado em 1983 por Richard Stallman e atualmente é mantido pela Free Software Foundation (FSF). A distribuição Linux oficial do projeto GNU é a Debian, pois foi desenvolvida e é atualizada através de esforços de voluntários espalhados pelo mundo todo, seguindo o mesmo estilo de desenvolvimento do projeto GNU/Linux.

## Unix

Tratando do ponto de vista histórico, o Unix foi criado em 1969 como um projeto de pesquisa no AT&T Bell Labs.

O Unix começou a ser distribuído de maneira gratuita nas universidades em 1976, porém 1977 houve licenciamento do código pela AT&T. À medida que crescia comercialmente o Unix, o preço das licenças de código fonte cresciam na mesma proporção

O Linux é uma forma reelaborada do Unix, que difere basicamente do Unix por ser gratuito, com código aberto e desenvolvido de maneira cooperativa

entre pessoas e organizações.

## Linus Torvalds

Linus Benedict Torvalds, é um Engenheiro de Software, nascido na Finlândia, criador e desenvolvedor do Linux. Criou em 1991 na universidade de Helsinki na Finlândia o primeiro Sistema Linux, uma ramificação do Minix, um Sistema Operacional ,modelo criado por Andrew S. Tannembaum. O Linux gerou grande interesse pela comunidade de computação no mundo todo, por isso seu criador Linus Torvalds viu a necessidade de desenvolver um kernel próprio, isso foi possível seguindo o padrão POSIX, que é o mesmo usado por sistemas Unix.

## Software Livre e Código aberto

O termo código aberto ( *open source* ) em princípio representa o mesmo que *software* livre ( *free software* ), porém, são movimentos distintos que possuem base de valores diferentes.

A idéia de código aberto valorizar a vantagem prática de desenvolver um *software* , de forma mais prática, tornando mais poderoso e confiável. Mas não se preocupa com o aspecto social, ou a liberdade para os usuários de computador, que é o ponto principal abordado pelo movimento de *software* livre.

O *software* livre é um movimento criado para respeitar a liberdade e senso de comunidade dos usuários, o termo livre não quer dizer *software* grátis, na verdade corresponde ao direito de executar, copiar, modificar e melhorar o *software* . Sendo assim, na visão desse movimento, o software não livre é um problema social e a solução é o uso de um *software* livre.

## Licença GPL

A licença GPL ou GNU GPL (GNU General Public License) está na versão 3, é

uma licença gratuita para softwares e outros tipos de trabalho, visa garantir a liberdade para criar, compartilhar e alterar todas versões de um software, garantindo que permaneça livre para todos os usuários.

Do ponto de vista dos desenvolvedores, a GPL permite a reivindicação de direitos autorais e cobrar pela distribuição se achar necessário, e oferecer uma licença com permissão legal para cópia ou modificação.

## saiba mais

### Saiba mais

A licença GPL tem como objetivo preservar as condições legais relacionadas a liberdade de softwares, garantindo que as implementações futuras realizadas no software também sejam livres. O kernel do Linux está licenciado de acordo com as regras da licença GPL v2. Para saber mais detalhes sobre a licença GPL V3, que foi publicada no ano de 2007, com o objetivo de resguardar os direitos éticos dos desenvolvedores de software livre, acesse a página oficial mantida pela *Free Software Foundation* em:

ACESSAR

Um aspecto importante da GPL é que se você distribuir um software com características livres, os destinatários devem receber os mesmos direitos e você deve demonstrar os termos para que eles conheçam esses direitos que estão escritos na licença GPL.

## O Linux

Caros estudantes, chegamos até esse tópico sabendo como o Linux foi criado e no que se baseou Linus Torvalds para realizar sua criação, além disso

aspectos que levaram milhares de pessoas a entrar no movimento de software livre para criação e distribuição de várias versões desse Sistema Operacional.

Agora utilizando uma visão mais técnica sobre o Sistema Operacional Linux, veremos alguns aspectos relevantes:

- Convivem sem conflitos com outros Sistemas Operacionais no mesmo computador.
- Multitarefa real.
- Multiusuário.
- Suporte a nomes extensos para diretórios e arquivos.
- Conectividade com outros tipos de plataformas como Unix, Windows e Apple.
- Modularização.
- Suporte nativo a balanceamento de carga.
- Não requer reinicialização de sistema após modificação de configuração.
- Menor necessidade de *hardware* potente.
- Suporte a mais de 63 consoles.
- Devido a separação de privilégios não é vulnerável a vírus.
- Distribuído livremente e licenciado pelos termos da GPL.

Apesar das características já conhecidas, cada distribuição possui singularidades. Como o sistema de instalação, interface gráfica ou até mesmo a localização de arquivos. A escolha da distribuição deve ser feita de acordo com a necessidade, Debian, Ubuntu, Fedora / Red Hat e Suse estão entre as mais conhecidas, possuindo versões para computadores pessoais e servidores.

# praticar

## Vamos Praticar

Desenvolvedores que trabalham em projetos de *software* livre usam a licença GPL para reivindicar direitos autorais e manter características de software livre ao seu projeto e suas futuras versões. Pensando em características de *software* livre, qual das alternativas abaixo está de acordo com a licença GPL V3 ?

- ☐ **a)** Não permite a modificação de um software livre.
- ☐ **b)** Proíbe a cobrança pela distribuição de um *software* livre.
- ☐ **c)** Permite executar um software livre, apenas para fins comerciais.
- ☐ **d)** Permite a cobrança pela distribuição de um *software* livre.
- ☐ **e)** Permite a instalação de um software somente mediante o pagamento da licença.



# Cenários de Implantação

Para utilização de qualquer Sistema Operacional, independente de ser Linux ou não, é importante saber as necessidades requeridas pelo cenário onde será implantado, além de conhecer as características do Sistema Operacional que pretende utilizar. Sendo assim, apresentamos alguns aspectos das principais distribuições Linux e em qual cenário de implantação é indicada.

## Debian

O Debian é um Sistema Operacional utilizado no mundo inteiro por instituições comerciais, educacionais e por usuários domésticos. É a distribuição oficial do projeto GNU, é mantida por esforço voluntário de milhares de pessoas. Pode utilizar o kernel do Linux ou do FreeBSD e contém mais de 59000 mil pacotes de softwares que podem ser utilizados.

Debian é um sistema indicado para computadores com *hardware* de todos os tipos, desde o mais modesto até o mais avançado. Também é muito utilizado por desenvolvedores, possui uma plataforma sólida e com muitos pacotes.

Como não existe suporte oficial ao Debian, apenas comunidades ou listas de ajuda, não é indicado para ser usado como um servidor empresarial.

## Ubuntu

Talvez a distribuição mais popular entre os usuários finais, devido a sua facilidade de instalação e sua interface amigável. É um variante da distribuição Debian, e possui versões domésticas e versões para servidor, além de versões específicas para *cloud computing* e *IoT ( Internet of Things )* que podem ser adquiridas de forma gratuita. Serviços de *cloud* e desenvolvimento, podem ser adquiridas com licenciamento pago na versão *enterprise* , que contam com suporte aos moldes dos servidores Microsoft.

Com base nisso, é correto afirmar que o Ubuntu é indicado para computadores domésticos mais poderosos e para usuários que buscam maior facilidade de uso.

## Fedora

O Fedora é uma distribuição desenvolvida de forma aberta é a versão para usuário final mantida pela Red Hat, foi criada em 2002, seus desenvolvedores se basearam numa versão antiga do Red Hat para criá-la. Não possui custo para licenciamento e, pelo fato de não contar com suporte oficial da Red Hat, não é indicada para ambiente corporativos.

Existe uma versão para servidores que é indicada para ser usado em serviços de cloud, com modularidade, fácil administração em seu console e domínio empresarial. Por possuir um ciclo de vida curto e ter suporte apenas da comunidade Fedora, é indicado para usuários com experiência elevada.

## Red Hat

O Red Hat Enterprise Linux, com certeza é a distribuição Linux mais utilizada comercialmente. Trata-se de um Sistema Operacional de código aberto que

permite trabalhar com serviços básicos de um servidor, além de possuir ferramentas para *cloud* , virtualização *bare metal* , ou ferramentas para desenvolvimento como o Open Container, por exemplo.

Possui suporte pago disponível de forma ininterrupta, por isso, é o mais indicado para estruturas empresariais. Conta com um sistema de certificação para profissionais, visando garantir a qualidade do profissional que trabalha com esse Sistema Operacional, as famosas certificações RHCA (Red Hat Certified System Administrator) e RHCE (Red Hat Certified Engineer).

## Slackware

Existem distribuições que são indicadas para os usuários com pouca experiência em Linux, com certeza a Slackware não é uma delas. Criada por Patrick Volkerding em 1993, sendo a primeira distribuída no mundo, Slackware é uma distribuição que possui a premissa de manter sua estabilidade e o que é tradicional do Linux.

Possui atualizações de versões de uma forma mais lenta, quando comparado com outras versões de Linux, é um sistema mais simples, indicado para usuários tradicionais que não precisam de suporte pago.

É uma distribuição conhecida como “esquelética” por possuir o básico para utilização, mantendo-se no topo de distribuições mais estáveis.

praticar

## Vamos Praticar

Como foi explicado neste tópico, cada distribuição Linux possui características

próprias, fazendo com que seja mais indicada para um perfil empresarial ou doméstico, para usuários com pouca ou com muita experiência. Por que é correto afirmar que o Slackware é indicado para usuário doméstico mais experiente ?

- ☐ **a)** Por que mantém o que é tradicional no Linux.
- ☐ **b)** Por que possui suporte pago.
- ☐ **c)** Por que sua interface gráfica possui brechas de segurança.
- ☐ **d)** Porque necessita de um gerenciador de pacotes para instalação de programas.
- ☐ **e)** Porque permite realizar todas as configurações por interface gráfica.

# Explicações Básicas do Ambiente

Neste tópico veremos os principais componentes existentes no Sistema Operacional GNU/LINUX, descrevendo detalhes sobre arquivos e explorando a estrutura básica de diretórios do Sistema Linux. Outro ponto importante abordado, é o uso do interpretador de comandos, comumente chamado de terminal ou *shell* e também iremos tratar sobre editores de arquivos.

## Arquivos e Diretórios

Talvez os menos experientes com Sistema Linux, imaginem que as mesmas extensões de arquivos e estruturas de diretórios do sistemas não tenham diferenças entre um Sistema Operacional Windows e Linux.

Assim como no Windows, uma extensão é representada por letras após o "." no nome do arquivo. Porém, para o sistema operacional Linux a extensão funciona apenas como um informativo para os usuários, permitindo que os mesmos identifiquem o tipo de software que pode trabalhar com os arquivos. Segue abaixo alguns exemplos:

- script. **sh** - Arquivo de um script.
- arquivo. **gz** - Arquivo compactado pelo gzip (compactador).
- system. **log** - Arquivo de registro do sistema.

No Linux, os arquivos podem ser do tipo texto ou binário, os arquivos do tipo texto podem ser lidos por pessoas, um código de programação, um arquivo de configuração, etc. Já os arquivos do tipo binário, possuem conteúdo para baixo nível, que somente podem ser entendidos por computadores e são gerados através de compilação.

No Linux, os arquivos e os diretórios são *case sensitive*, ou seja, diferencia letras maiúsculas de minúsculas. Por exemplo, o diretório `/Aula` é totalmente diferente do diretório `/aula`.

Nos Sistemas Operacionais Linux os diretórios são especificados por `/` diferente do MS-DOS que usa `\`, o **mkdir** é o comando utilizado para criar diretórios no Linux.

O principal diretório em Sistemas GNU/Linux é o diretório raiz representado por `/`, onde estão inseridos os diretórios do sistema. Para acessar o diretório raiz, basta digitar o comando `cd /` e para saber o diretório atual basta digitar **pwd**.

Já o comando **ls** serve para listar o conteúdo de um diretório, assim como em quase todos os comandos em Linux, existem opções para executar um comando, para conhecer estas opções basta usar **--help**, então ficaria **ls --help** para saber as opções para o comando **ls**.

```
root@ramiro-virtual-machine:/home/ramiro# ls --help
Uso: ls [OPÇÃO]... [ARQUIVO]...
Lista informações sobre os ARQUIVOS (no diretório atual por padrão).
Lista as entradas em ordem alfabética se não for usada nenhuma opção -cftuvSUX
nem --sort.

Argumentos obrigatórios para opções longas também o são para opções curtas.
-a, --all                não ignora entradas começando com .
-A, --almost-all        não lista as entradas implícitas . e ..
--author                 com -l, emite o autor de cada arquivo
-b, --escape             emite escapes no estilo C para caracteres não-
                        gráficos
--block-size=TAM         o tamanho considera blocos de TAM bytes; exemplo:
                        "--block-size=M" emite tamanhos em unidades de
                        1.048.576 bytes; veja o formato de TAM abaixo
-B, --ignore-backups     não lista as entradas implícitas terminadas com ~
-c                       com -lt: ordena por, e mostra, ctime (hora da
                        última modificação da informação de estado do
                        arquivo);
                        com -l: mostra o ctime e ordena por nome
                        demais casos: ordena por ctime
-C                       lista as entradas em colunas
--color[=QUANDO]         controla se usa cores para distinguir os tipos de
                        arquivo. QUANDO pode ser "never" (nunca),
                        "always" (sempre) ou "auto" (automaticamente)
-d, --directory         lista os diretório em si, e não seu conteúdo
-D, --dired              gera a saída projetada para modo "dired" do Emacs
```

*Figura 1.1 - Demonstração do comando `ls --help`.*

*Fonte: Elaborada pelo Autor*

Uma outra opção é usar o comando *man*, para ver o manual do comando, por exemplo, **man ls**. Com o comando **man** é possível verificar a descrição do comando, além de todas suas opções.

Voltando ao assunto diretórios, no Linux o diretório anterior é identificado por "-", então quando quiser retornar ao diretório anterior, o comando **cd** - pode ser utilizado. Já o comando **cat** pode ser utilizado para ler um arquivo, contido em qualquer diretório.

Os Sistemas GNU/Linux, segundo a FHS (Filesystem Hierarchy Standard), possui a seguinte estrutura básica de diretórios:

- / - Diretório raiz.
- /bin/ - Local dos programas essenciais do sistema.
- /boot/ - Arquivos do boot.
- /dev/ - Local dos arquivos de acesso aos dispositivos.
- /etc/ - Local de arquivos de configuração.
- /home/ - Diretório do usuário
- /lib/ - Local das bibliotecas essenciais.

- /mnt/ - Local para ponto de montagem temporária.
- /media/ - Local de montagem para mídia removível.
- /opt/ - Local para programas que não são essenciais.
- /proc/ - Sistema de arquivos do kernel, é utilizado apenas por ele.
- /root/ - Diretório home do super usuário, conhecido como root.
- /sbin/ - Diretório de programas root para administração e controle do sistema.
- /tmp/ - Local de arquivos temporários.
- /srv/ - Local para dados específicos, de serviços e servidores.
- /usr/ - Local de programas ou arquivos compartilhados, acesso apenas leitura normalmente.
- /var/ - Local de arquivos que são gravados com frequência, como logs por exemplo.

## Interpretador de Comandos

A terminal ou *shell*, é um interpretador de comandos, que é o responsável em “traduzir” as instruções enviadas pelo usuário através de comandos digitados ou executados por *script*. Os Sistemas Operacionais GNU/Linux possuem vários interpretadores de comandos, por exemplo, bash, sh e ash.

Como já falado, os comandos podem ser enviados pelo usuário de duas formas, uma delas é digitando o comando, essa é a maneira interativa que depende do usuário para executar uma tarefa. Já a outra forma é a não-interativa, onde *scripts* são criados para o computador executar as tarefas na ordem encontrada no arquivo.

O nome terminal também é utilizado para o terminal virtual ou console, que trata-se de uma interface multiusuário. É uma seção de trabalho independente de outras, que pode ser acessada localmente ou de forma remota via telnet, ssh ou rlogin.



```
ramiro@ramiro-virtual-machine:~$ su -  
Senha:  
root@ramiro-virtual-machine:~#
```

*Figura 1.2 - Terminal mudando usuário para root.*

*Fonte: Elaborada pelo Autor*

## Editores de Arquivos

Embora haja ferramentas gráficas para as maiores dos programas que utilizamos em Linux, as vezes elas não são suficientes ou não estão disponíveis. Existem distribuições Linux que utilizam apenas o modo texto, e em empresas para uso mínimo de recurso também são utilizadas, por exemplo, firewall. É comum ver um *firewall* utilizando distribuição Linux para criação de regras com *iptables*.

Portanto é importante aprender a usar a linha de comando, para quem deseja ser no mínimo um usuário avançado em Linux. Como no GNU/Linux a necessidade de editar arquivos do sistema é constante, torna-se importante conhecer os editores de arquivo.

O mais conhecido (não o mais utilizado) é o Vi, criado em 1976, que posteriormente teve uma derivação melhorada em 1991 conhecida por Vim, é um editor modo texto que utiliza comandos para edição ou criação de arquivos.

Já no modo gráfico existem outros vários editores, por exemplo, Gedit, Jedit, Bluefish ou o Kate. Esses possuem mais recurso, possibilidade de usar o *mouse*, além de marcações e cores, que tornam muito mais fácil a experiência do usuário. Porém, como já dito, em Linux nem sempre é viável ou está disponível utilizar uma interface gráfica.

## praticar

# Vamos Praticar

O sistema operacional Linux possui uma estrutura de diretórios hierárquica, com o objetivo de padronizar os tipos de arquivos para cada diretório. Dentre as opções abaixo, qual representa o diretório padrão do Linux utilizado para armazenar os arquivos de configuração dos softwares?

- ☐ a) /system32/
- ☐ b) /dev/
- ☐ c) /tmp/
- ☐ d) /etc/
- ☐ e) /home/

# Principais Comandos Utilizados

Saber os principais comandos do Sistema GNU/Linux é primordial para uma experiência mais eficiente do usuário, não é necessário exatamente decorar, mas compreender a função de cada comando. Utilizar a linha de comando facilita e agiliza o trabalho de qualquer administrador de Sistema Operacional, deixando-o independente da interface gráfica.

Neste tópico abordaremos os principais comandos do Sistema GNU/Linux, que são necessários para navegação do ambiente, leitura de arquivos, busca de arquivos, criação de links e gerenciamento de processos em linha de comando.

## Comandos para Navegação do Ambiente

Para manipulação de diretórios e navegação do ambiente, a seguir os comandos mais úteis, acompanhados de um exemplo. Lembrando que todos os comandos descritos possuem opções, para saber mais use o comando **man** ou o sufixo **--help**.

- **ls** - Listar arquivos de um diretório

Ex. **ls -a** - Lista todos arquivos, incluindo ocultos em um diretório

- **cd** - Entra em um diretório

Ex. **cd /usr/bin** - Entra no diretório /usr/bin

- **pwd** - Mostra o nome e caminho do diretório atual.

Ex. **pwd -P** - Mostra o nome e caminho do diretório evitando links simbólicos.

- **mkdir** - Comando para criar um diretório.

Ex. **mkdir /tmp/teste** - Cria o diretório teste em /tmp.

- **rmdir** - Remove um diretório estando um nível acima.

Ex. **rmdir teste** - Estando em /tmp apaga o diretório teste.

- **cp** - Copia arquivos ou pastas.

Ex. **cp -r teste teste2** - Usando a opção -r clona a pasta teste e cria a pasta teste 2.

- **mv** - Move ou renomeia arquivos e pastas.

Ex. **mv teste teste2** - Renomeia o diretório teste para teste2.

É importante salientar que apesar dos comandos cp e mv serem parecidos, existem diferenças. O comando mv tem a mesma função do comando recortar (ctrl+x) no Windows e o comando cp tem a mesma função do comando copiar (ctrl+c) do Windows.

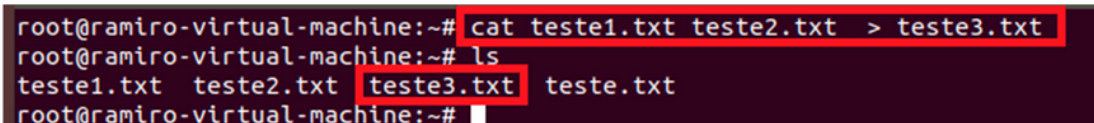
## Comandos para Pesquisa e Visualização de Arquivos

Existem comandos importantes para buscar determinados arquivos ou buscar

dados contidos em determinados arquivos. Na sequência, alguns desses comandos, acompanhados de exemplos.

- **cat** - Mostrar e concatenar arquivos.

Ex. **cat teste1.txt teste2.txt > teste3.txt** - Concatena os arquivos teste1.txt e teste2.txt, gerando o arquivo teste3.txt.

A terminal window with a dark purple background. The prompt is 'root@ramiro-virtual-machine:~#'. The first command entered is 'cat teste1.txt teste2.txt > teste3.txt', which is highlighted with a red box. The second command is 'ls', also highlighted with a red box. The output of 'ls' is 'teste1.txt teste2.txt teste3.txt teste.txt'. The prompt is then 'root@ramiro-virtual-machine:~#' followed by a cursor.

```
root@ramiro-virtual-machine:~# cat teste1.txt teste2.txt > teste3.txt
root@ramiro-virtual-machine:~# ls
teste1.txt teste2.txt teste3.txt teste.txt
root@ramiro-virtual-machine:~#
```

*Figura 1.3- Utilizando comando cat.*

*Fonte:Elaborada pelo Autor*

- **head** - Mostra as primeiras linhas de um arquivo.

Ex. **head teste.txt** - Exibe as primeiras linhas do arquivo.

- **more** - Mostra um ecrã por vez, muito utilizado com outros comandos.

Ex. **ls | more** - Exibe o conteúdo do diretório possível no primeiro ecrã, aguarda uma tecla para os próximos ecrãs.

- **tail** - É o inverso do comando head, mostra as últimas linhas do arquivo.

Ex. **tail teste.txt** - Exibe as últimas linhas do arquivo.

```
root@ramiro-virtual-machine:~# tail teste3.txt
TESTE1 TESTE1 TESTE1
TESTE2 TESTE2 TESTE2
root@ramiro-virtual-machine:~#
```

*Figura 1.4- Utilizando comando tail.*

*Fonte: Elaborada pelo Autor*

- **grep** - Comando para busca de palavras ou padrões em arquivos.

Ex. **grep alunos teste.txt** - Procura a palavra alunos no arquivo teste.txt.

- **cmp** - Compara dois arquivos byte a byte.

Ex. **cmp teste.txt teste1.txt** - Compara os dois arquivos byte a byte.

- **uniq** - Comando para verificação de linhas repetidas em um arquivo.

Ex. **uniq -d /temp/teste.txt** - Mostra apenas as linhas duplicadas.

Esses comandos são de extrema utilidade, pois permitem analisar mais facilmente arquivos longos de texto. Dependendo da distribuição Linux será necessário instalar o pacote referente ao comando.

## Criação de Links e Gerenciamento de

## Processos

*Link* é um mecanismo utilizado pelo Sistema GNU/Linux que faz referência a outro arquivo ou diretório. O gerenciamento de processos nos Sistemas GNU/Linux funcionam de forma semelhante ao ambiente Windows, a seguir comandos relacionados aos dois temas.

- **ln** - Comando para criar *link* simbólico.

Ex. **ln -s /tmp/teste /home/user** - Cria um link simbólico entre os diretórios.

- **top** - Comando para consulta de processos do sistema.

Ex. **top -d 5** - Exibe e atualiza a lista de processos a cada 5 segundos.

```
top - 22:18:34 up 46 min, 1 user, load average: 0,17, 0,12, 0,14
Tarefas: 258 total, 2 em exec., 194 dormindo, 0 parado, 0 zumbi
%CPU(s): 4,2 us, 2,8 sis, 0,0 ni, 93,1 oc, 0,0 ag, 0,0 ih, 0,0 is 0,0 tr
KB mem : 4012832 total, 1493184 livre, 968216 usados, 1551432 buff/cache
KB swap: 969960 total, 969960 livre, 0 usados, 2785112 mem dispon.
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPO+	COMANDO
729	root	20	0	385284	60208	34308	S	3,3	1,5	0:20.32	Xorg
1170	ramiro	20	0	2977636	264080	84108	S	3,0	6,6	0:51.86	gnome-she+
3209	ramiro	20	0	802420	38444	29276	S	2,0	1,0	0:04.03	gnome-ter+
3426	root	20	0	51440	4224	3440	R	0,7	0,1	0:00.95	top
1141	ramiro	20	0	220704	6984	6228	S	0,3	0,2	0:00.12	at-spi2-r+
3351	root	20	0	0	0	0	I	0,3	0,0	0:00.53	kworker/0+
1	root	20	0	159980	9280	6700	S	0,0	0,2	0:03.84	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0+
8	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu+
9	root	20	0	0	0	0	S	0,0	0,0	0:00.57	ksoftirqd+
10	root	20	0	0	0	0	R	0,0	0,0	0:01.11	rcu_sched
11	root	rt	0	0	0	0	S	0,0	0,0	0:00.02	migration+
12	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_inje+
14	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
16	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
17	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks+
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kauditd

*Figura 1.5 - Utilizando o comando top.*

*Fonte: Elaborada pelo Autor*

- **ps** - Comando para consulta de processos do sistema, indicado para usar com o comando **grep**.

Ex. **ps aux** - Lista todos processos em execução, de todos usuários, os nomes dos usuários e que estão em qualquer sessão.

- **kill** - Comando para interromper um processo.

Ex. **kill 1550** - Encerra o processo com PID ( *Process ID* ) 1150.

Esse conjunto de comandos são extremamente úteis para administradores de sistemas linux. Por isso, é extremamente indicado o conhecimento dos comandos e suas opções.

## Comandos Diversos

Os Sistemas GNU/Linux possuem uma gama enorme de comandos, que são úteis para as mais distintas necessidades, a seguir são apresentados comandos diversos que serão utilizados muitas vezes enquanto estiver administrando um sistema Linux.

- **clear** - Limpa a tela e posiciona o cursor para o canto superior esquerdo do vídeo.

Ex. Também é possível utilizar as teclas de atalho **ctrl+L**.

- **date** - Permite visualizar e modificar a hora do sistema.

Ex. **date 11261400** - Muda a data para 26/11 às 14:00 horas.

- **df** - Mostra o espaço livre de cada partição.

Ex. **df -x vfat** - Não lista sistemas de arquivos do tipo vfat.

- **du** - Mostra espaço ocupado por arquivos no diretório atual.

Ex. **du -h** - Mostra o espaço ocupado em *Mega Bytes* .

- **free** - Mostra o uso da memória RAM do sistema.

Ex. **free -m** - Mostra o uso da memória em *Mega Bytes* .

- **time** - Mede o tempo gasto para execução de um processo.



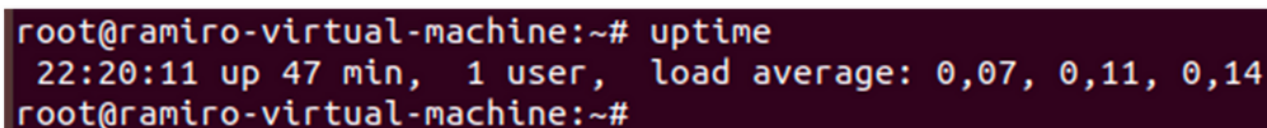
Ex. **time mkdir teste** - Mostra o tempo gasto pelo comando mkdir para criação da pasta teste.

- **touch** - Muda a data e hora que um arquivo foi criado.

Ex. **touch -am teste.txt** - Muda a hora de acesso e modificação do arquivo para o horário atual.

- **uptime** - Mostra o tempo de execução do sistema desde que o computador foi executado.

Ex. **uptame -s** - Mostra o tempo de execução em formato mais organizado.



```
root@ramiro-virtual-machine:~# uptime
22:20:11 up 47 min, 1 user, load average: 0,07, 0,11, 0,14
root@ramiro-virtual-machine:~#
```

*Figura 1.6 - Utilizando o comando uptime.*

*Fonte: Elaborada pelo Autor*

- **su** - Permite mudar a identidade do usuário sem fazer *logout*.

Ex. **su user1** - Muda o usuário para user1.

- **chmod** - Muda a permissão de acesso para um grupo ou diretório.

Ex. **chmod uo+x teste.txt** - Da permissão de execução para o dono e os outros usuários do arquivo.

- **chown** - Muda o dono do arquivo ou diretório.

Ex. **chmod user1 teste.txt** - Muda o dono do arquivo teste.txt para user1

Os comandos chown e chmod são bastante utilizados na administração de servidores de arquivos. Nesse tipo de serviço é importante definir adequadamente quem terá permissões de acesso a arquivos e diretórios e esses comandos atuam exatamente nessas definições.

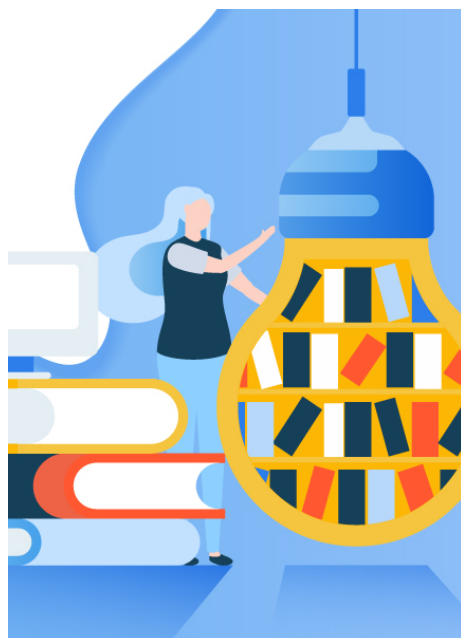
## praticar

# Vamos Praticar

O gerenciamento de processos no Linux é uma tarefa importante para os administradores de sistemas. O conhecimento sobre as possibilidades de comandos utilizados para gerenciamento de processos é uma característica importante para esse tipo de profissional. Dentre os comando apresentados abaixo, qual deles é utilizado para encerrar um processo no sistema operacional Linux?

- ☐ a) ps.
- ☐ b) ln.
- ☐ c) kill.
- ☐ d) top.
- ☐ e) ls.

# indicações Material Complementar



## LIVRO

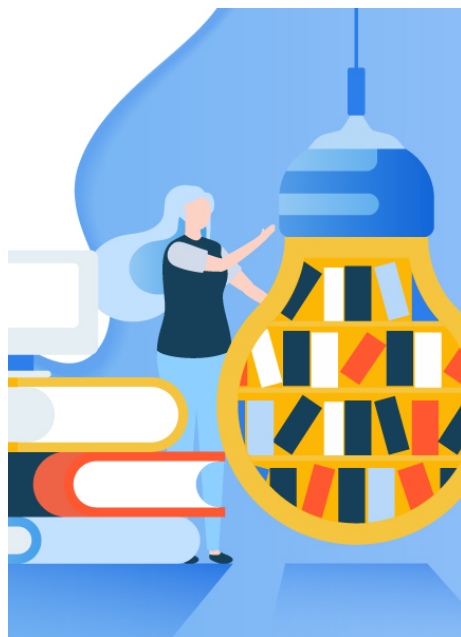
**Nome do livro :** Linux. Guia De Bolso

**Editora :** Alta Books

**Autor :** Daniel J. Barret

**ISBN :** 9788576081043

**Comentário :** Como o próprio nome diz um guia de bolso, muito útil para o dia-a-dia, para ter de forma rápida os principais comandos Linux, com resumos detalhados dos comandos.



## LIVRO

**Nome do livro :** Introdução às Expressões Regulares

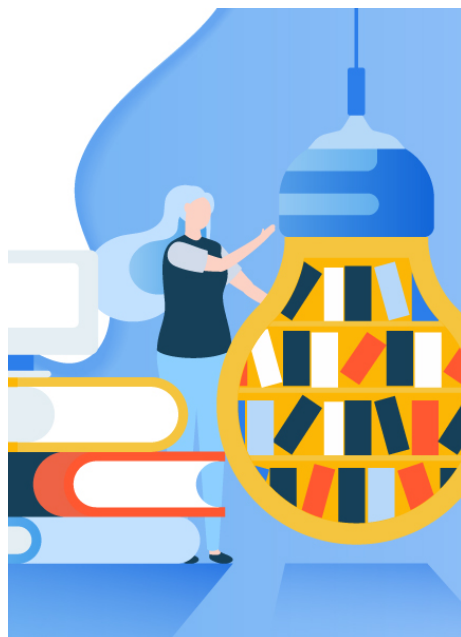
**Editora :** Novatec

**Autor :** Michael Fitzgerald

**ISBN :** 9788575223307

**Comentário :** Expressões regulares é uma forma poderosa de encontrar e trabalhar com padrões de texto. Não apenas o Linux reconhece as expressões regulares, ela pode ser usada também em linguagens de programação, editores de texto ou clientes de email, por exemplo.

---



## LIVRO

**Nome do livro :** Manual Completo de Linux: guia do administrador - 2ª edição

**Editora :** Editora Pearson

**Autor :** Nemeth, Evi; Snyder, Garth; Hein, Trent R.

**ISBN :** 9788576051121

**Comentário :** Um excelente livro sobre Linux, abrange as mais importantes distribuições e as principais ferramentas administrativas utilizadas. Um ótimo guia para quem gostaria de aprender mais sobre o Sistema Operacional Linux.

---

## WEB

**Nome** : CONCEITO, HISTÓRIA E ESTRUTURA DO LINUX

**Ano** : 2016

**Comentário** : Este filme apresenta a história do sistema operacional linux, bem como alguns conceitos importantes que norteiam o assunto.

Você pode conhecer mais sobre o assunto assistindo o vídeo disponível em:

ACESSAR

---

# conclusão

## Conclusão

Esta unidade apresentou as informações necessárias para você, caro estudante, aprender sobre o surgimento do Linux, entender como foi criado e o objetivo da sua criação. Aprendeu o que são distribuições Linux, quais são as mais conhecidas, em quais cenários são indicadas, aprendeu também sobre a GPL, porque foi criada e porque é importante para comunidade de *software livre*.

Aspectos sobre arquivos e seus diretórios foram revelados, interpretadores de textos e editores foram apresentados, assim como os principais comandos utilizados, com intuito de dar uma base de conhecimento sólida para que possa seguir nas próximas etapas de conhecimento desse magnífico Sistema Operacional GNU/Linux.

---

# referências

## Referências Bibliográficas

GUIA foca linux. **Website**. Disponível em: < <https://guiafoca.org/> > Acesso em: 16 dez 2019.

HEIN, trent *et al* . **Manual Completo de Linux: guia do administrador** . Estados Unidos, Pearson, 2007.

LICENÇA GPL. **Website** . Disponível em: < <https://www.gnu.org/licenses/gpl-3.0.html> > Acesso em: 27 dez 2019.

SOFTWARE livre. **Website** . Disponível em: < <https://www.gnu.org/philosophy/free-sw.pt-br.html> > Acesso em: 28 dez 2019.