

# Unidade VIII

## 8 INSTALAÇÃO DO LINUX

Antes de efetuar a instalação do Sistema Operacional, é importante saber quais serão os serviços que o servidor proverá. Com isso, será possível realizar uma instalação personalizada, considerando, por exemplo, a definição do tamanho dos discos e suas respectivas partições. Em uma situação real, esse exemplo seria apenas um dos fatores importantes que deveriam ser planejados com a equipe de TI.

### 8.1 Distribuição CentOS

Nesse momento não será uma preocupação a configuração do *layout* do disco, pois existem assuntos que ainda não foram estudados e que são necessários para definir essa configuração.

Para essa primeira instalação, o particionamento será realizado de forma automática.

É importante destacar que todas as telas e exemplos seguintes foram criados com base na distribuição CentOS.



#### Saiba mais

Conheça mais sobre a distro CentOS na página oficial da distribuição, na guia *About*.

[<www.centos.org/about/>](http://www.centos.org/about/).

#### 8.1.1 Passo a passo sobre a instalação do *software*

- **Passo 1:** acesse o *site* [<www.centos.org>](http://www.centos.org) (CentOS Project) e clique no ícone **Get CentOS Now**, destacado em laranja, conforme exibido na figura seguinte.

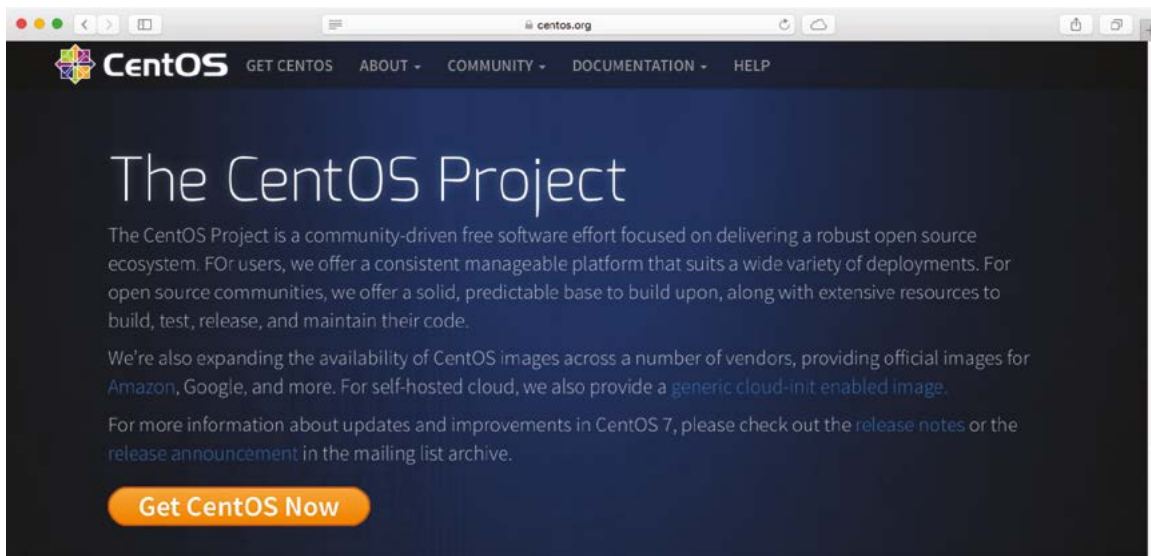


Figura 171 – Site do CentOS

- **Passo 2:** clique no ícone **Minimal ISO**, destacado em laranja, conforme exibido na figura a seguir.



Figura 172 – Download CentOS

- **Passo 3:** selecione um dos endereços ou **mirrors** disponíveis para a realização do *download* da imagem com o nome de CentOS-7-x86\_64-Minimal-1511.iso, conforme exibido na próxima figura.



Figura 173 – Download da imagem

Após fazer o *download* da imagem e gravá-la em um DVD, deve-se começar a instalação do sistema.

Insira o DVD na unidade de disco do servidor e ligue-o para dar início à instalação. Em seguida, selecione a opção **Install CentOS**, conforme exibido na figura a seguir.

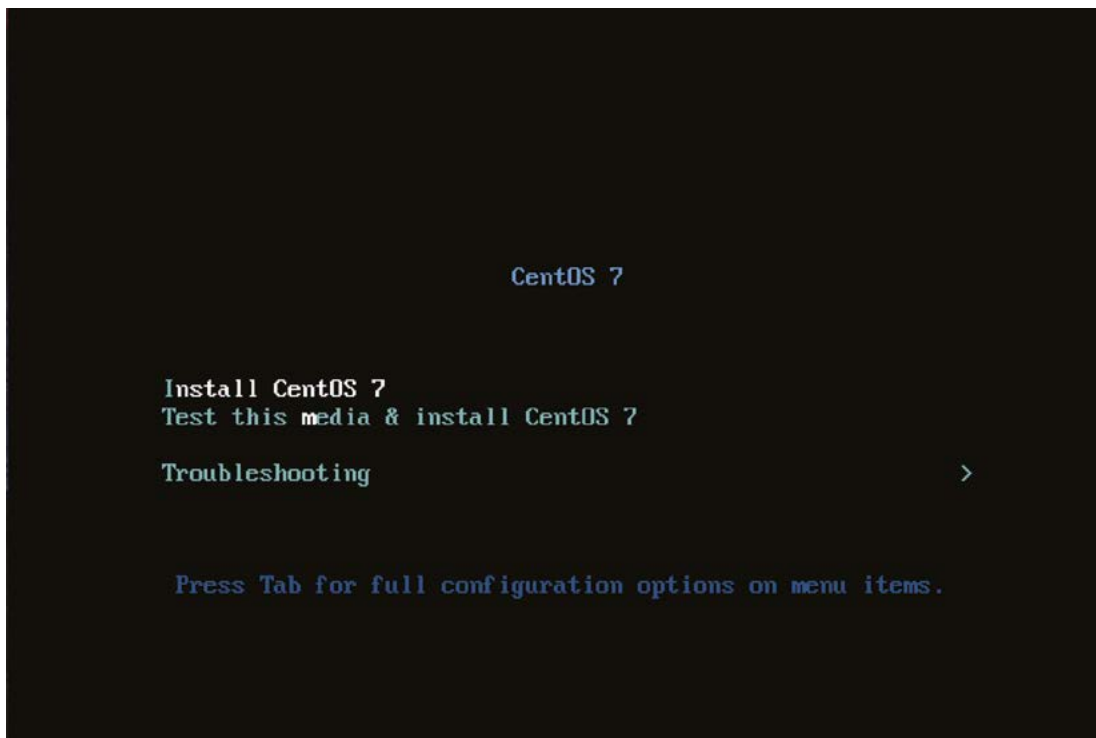


Figura 174 – Menu de seleção da instalação

Selecione a primeira opção e aguarde o carregamento da janela de seleção do idioma. Em seguida, selecione a opção Português no campo disponível e clique no botão **Continuar**, conforme exibido na figura seguinte.

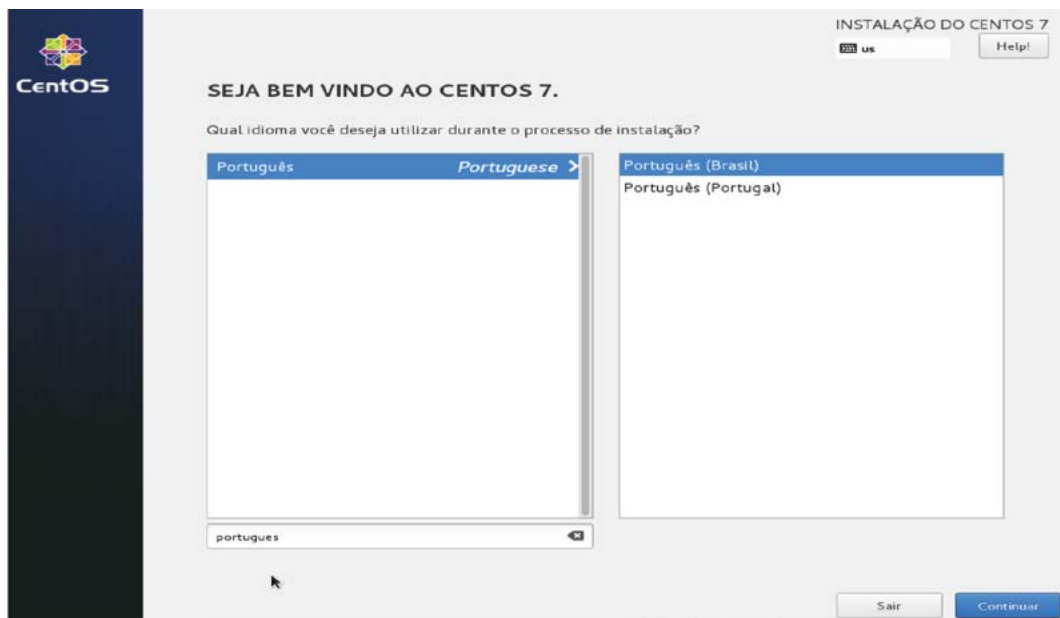


Figura 175 – Janela de escolha do idioma para a instalação

Na janela de Resumo da Instalação, clique na opção **Destino Instalação** e depois no botão **Finalizado**, localizado no canto superior esquerdo, conforme exibido na figura seguinte.



Figura 176 – Janela de resumo da instalação

Agora, clique na opção **Rede & Nome do Host**, para definir o nome do computador.

No campo **Nome do host**, substitua o nome localhost.localdomain por srvcentos7.localdomain e clique no botão **Finalizado**, no canto superior esquerdo, conforme exibido na figura a seguir.

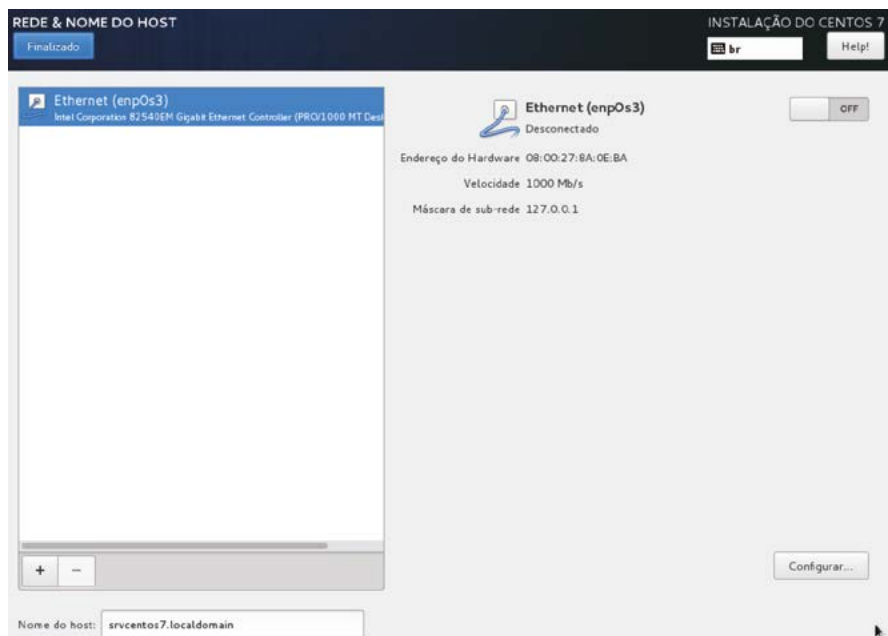


Figura 177 – Janela de Rede & Nome do Host

A configuração de rede será realizada manualmente.

Clique no botão **Iniciar a instalação**, conforme exibido na figura a seguir.



Figura 178 – Janela de Resumo da Instalação

A partir desse momento, o sistema será instalado.



Figura 179 – Janela de Progresso da Instalação

Após a instalação do sistema, será necessário definir a senha do usuário *root* e também criar um usuário comum. Esse processo é exibido na figura anterior.

Clique na opção **Senha Raiz** para cadastrar a senha do usuário *root*.

Caso cadastre uma senha que não satisfaça os requisitos de segurança, uma tarjeta em laranja é apresentada, conforme exibido na figura seguinte. Se mesmo assim quiser continuar com a senha, clique no botão **Finalizado** duas vezes.

Figura 180 – Janela de senha *root*

Será cadastrado um usuário comum, conforme pode ser exibido na figura seguinte.

Clique na opção de **criação de usuário**.

Cadastre o usuário de nome **tux** e informe uma senha.

Se a senha cadastrada for simples, a mensagem de alerta ocorrerá novamente. Clique duas vezes no botão **Finalizado**.

Figura 181 – Janela para criar usuário

Aguarde a instalação do sistema, até aparecer o botão **Finalizar configuração**, conforme exibido na figura a seguir.



Figura 182 – Janela de configuração

Clique no botão e aguarde novamente, esse processo poderá demorar alguns minutos. Finalizada a instalação, clique no botão **Reiniciar**, conforme exibido na figura seguinte.

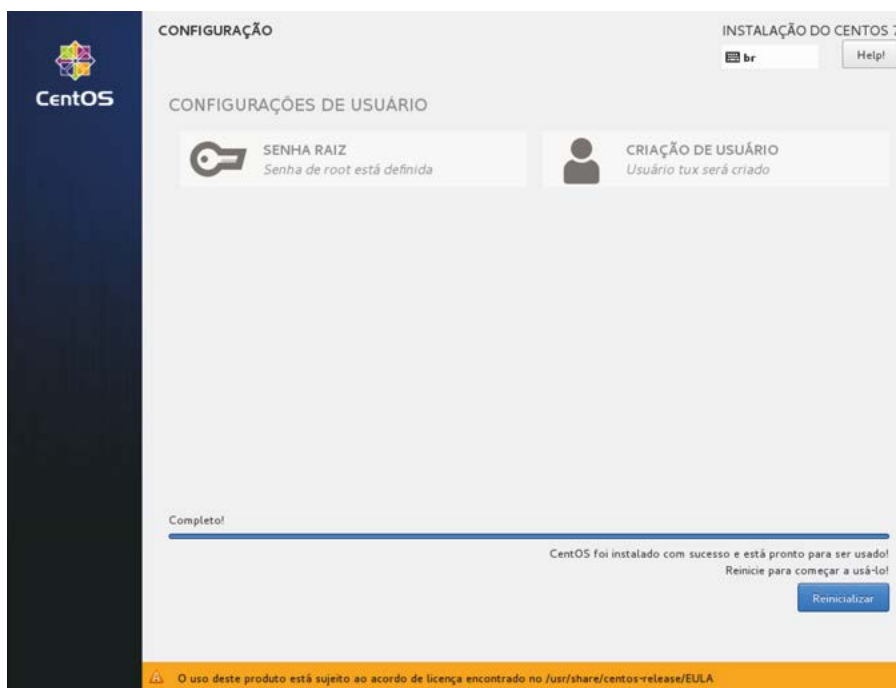


Figura 183 – Janela de configuração

**Pronto!** O Sistema Operacional está instalado e disponível para ser utilizado.



### 8.2 Editor de texto Vim

O Editor Vim, muito poderoso e com diversas funcionalidades indispensáveis no dia a dia, é bastante utilizado por programadores, administradores de sistemas e profissionais da área. Afinal, a maior parte dos serviços é configurada por meio de arquivos armazenados no diretório etc., e como nem todo servidor possui ambiente gráfico e muitos deles são operados remotamente, o Vim se destaca como uma opção leve e ágil.

Neste subtópico serão abordados os seguintes assuntos:

- como esse editor funciona;
- trabalho em modo de edição;
- trabalho em modo de comando;
- operações;
- arquivo de configuração do Vim.

#### 8.2.1 Funcionamento do editor

O programa Vim provavelmente seja o editor de texto mais utilizado pelos profissionais que trabalham com o sistema GNU/Linux. Ele é uma melhoria do antigo editor Vi, muito utilizado em Sistemas Operacionais Unix.

Segundo Ferreira (2003), por padrão, o CentOS já vem com o editor instalado, e para executá-lo basta digitar "vi". Ao ser executado, aparecerá a tela de abertura do editor Vim, conforme exibido na figura seguinte.



```
VIM - Vi IMproved
        version 7.4.168
        by Bram Moolenaar et al.
    Modified by <bugzilla@redhat.com>
Vim is open source and freely distributable

        Help poor children in Uganda!
type  :help iccf<Enter>      for information

type  :q<Enter>              to exit
type  :help<Enter> or <F1>   for on-line help
type  :help version7<Enter> for version info
```

Figura 184 – Tela de abertura do Vim

Esse editor pode funcionar de duas maneiras: modo de edição, em que as teclas possuem suas funcionalidades originais, podendo o texto ser completamente redigido ou alterado no formato desejado; e modo de comando, utilizado para manipular o arquivo, podendo executar tarefas de automação, como copiar, apagar uma sequência de linhas, salvar as alterações do arquivo, realizar pesquisas, entre outras funções.

### 8.2.2 Executando em modo de edição

Ao abrir o editor de texto, por padrão, ele fica em **modo de comando**. Para o acesso ao **modo de edição**, pressione a tecla Esc e depois a tecla I, ou O, ou A, ou Insert.

Realize os testes usando cada opção e perceba a diferença. Sempre use a tecla Esc para alterar como o editor irá trabalhar.

### 8.2.3 Executando em modo de comando

Para acessar o modo de comando, efetue esta sequência:

- Clique na tecla Esc, mantenha pressionada a tecla Shift e clique na tecla :.
- Será apresentado no canto inferior esquerdo o Prompt de Comando :.
- Informe o comando desejado.
- Para sair do editor, digite q.
- Para salvar as alterações e sair do editor, digite wq.
- Se for um arquivo novo, será apresentada uma mensagem informando que o arquivo não tem nome.

Digite o exemplo a seguir:

**:wq teste**

O arquivo será salvo no diretório atual em que o processo foi executado com o nome **teste** e o editor será encerrado.

#### 8.2.3.1 Operações

Nas operações seguintes, serão realizadas algumas operações usando a ferramenta Vim.

Vamos abrir um outro arquivo e realizar algumas pesquisas.

```
[root@centos7 ~]$ vim /etc/services
```

Para pesquisar determinada palavra, em modo de comando, digite a tecla "/" (barra normal) e, em seguida, digite a palavra a ser pesquisada.

Por exemplo:

```
/ssh
```

Outra maneira de realizar uma pesquisa seria utilizando o caractere `?`. Com isso, a pesquisa será realizada de maneira recursiva, em uma sequência de baixo para cima. Caso o editor encontre mais de uma palavra, basta pressionar a tecla `N` para mostrar a próxima ocorrência.

### 8.2.4 Acessar uma determinada linha do texto

É possível acessar uma determinada linha do arquivo informando ao editor o número da linha desejada. Por exemplo, ainda usando o mesmo arquivo aberto anteriormente, será acionado o Prompt de Editor e então informado o número da linha desejada, por exemplo, `:20`

Nesse caso, o editor irá posicionar o cursor na linha 20 do arquivo. Para ter certeza de que o cursor está realmente na linha 20, basta digitar o comando `:set nu`. O número será exibido na linha atual, e vale lembrar que esse número não ficará gravado no arquivo.

### 8.2.5 Arquivo de configuração Vim

Poderão ser realizadas algumas customizações no editor, por exemplo: numerar as linhas de um texto ou colori-lo, no caso de se estar desenvolvendo um *shell script*.

Edite o arquivo **vimrc** que se encontra no diretório `/etc/` no CentOS ou no diretório `/etc/vim/` no Debian, lembrando que precisa ter permissão do usuário *root*. A tabela a seguir exibe algumas opções que podem ser incluídas no arquivo de configuração:

**Quadro 1 – Opções de configuração do Vim**

Opção	Função
<code>set ve=all</code>	Permite mover o cursor por toda a tela
<code>set bg=dark</code>	Define cores apropriadas ao terminal padrão do modo texto
<code>set hlsearch</code>	Marca todos os textos que casarem com o padrão pesquisado em vídeo reverso
<code>set incsearch</code>	Permite pesquisa incremental
<code>set backspace=indent,eol,start</code>	Permite que a tecla Backspace remova fins de linha
<code>set visualbell</code>	Retira o bipe do Vi
<code>set ignorecase</code>	Permite pesquisar um texto ignorando caixa alta/baixa
<code>cab Wq wq</code> <code>cab WQ wq</code> <code>cab W! w!</code> <code>cab Q! q!</code>	Corrige erros comuns de digitação no modo de comando
<code>set nu</code>	Numera as linhas
<code>set syntax on</code>	Colore o texto

Veja na tabela seguinte as opções mais utilizadas do Vim.

**Quadro 2 – Outras opções do Vim**

Opção	Função
:wq, ZZ, :x	Salva o arquivo e sai do editor
u	Desfaz uma modificação
gg	Avança para a primeira linha do arquivo
G	Avança para a última linha do arquivo
h	Leva o cursor um caractere à esquerda
j	Avança para a linha seguinte
k	Leva o cursor para a linha anterior
l	Avança um caractere à direita
:20	Avança para a linha 20
e: arquivo	Alterna a edição do arquivo atual para outro arquivo informado
!:comando	Executa o comando informado e exibe o resultado. Não será inserido no documento aberto
:r!comando	Executa o comando informado e salva o resultado no arquivo
:set nu	Inserir numeração no texto, apenas para visualização
:set nonu	Desabilita a numeração
:help	Ajuda do editor
n	Repete o comando / ou ?
:sav novoarquivo	Salva o arquivo atual como outro arquivo ("novoarquivo")
Ctrl+e	Mova a tela uma linha acima
Ctrl+y	Mova a tela uma linha abaixo
Ctrl+G	Exibe o nome do arquivo, o número da linha atual e o total de linhas no texto

## 8.3 Sistemas de Arquivos e Diretórios

O sistema de arquivos é responsável pelo gerenciamento das informações que são gravadas em uma determinada partição do disco. O sistema GNU/Linux trabalha com uma grande variedade de sistemas de arquivos. Seguem alguns:

- **EXT2**
  - Conhecido como Second Extended FileSystem.
  - Utiliza blocos do mesmo tamanho para armazenar os arquivos.
  - Trabalha com partições de até 4 terabytes.
  - O tamanho máximo do nome do arquivo é de 255 caracteres.

- Reserva normalmente 5% dos blocos para o superusuário (*root*).
- Não possui *journaling* e foi substituído pelo ext3.

- **EXT3**

- Permite utilizar o sistema de cotas.
- Pode trabalhar com blocos de 1,2 e 4 kilobytes.
- O tamanho máximo da partição é de 16 terabytes.
- O tamanho máximo dos arquivos é de 2 terabytes.

- **EXT4**

- Podem facilmente ser convertidos para o formato ext4.
- Melhorias no desempenho e na capacidade de armazenamento.
- Disponibiliza o recurso de *journaling*, podendo este ser habilitado ou não.
- O tamanho máximo de cada arquivo é de 16 Tb.

- **ReiserFS**

- Possui suporte ao *journaling*.
- Trabalha com blocos de 4 kilobytes
- O tamanho máximo da partição é de 1 ectabyte.
- O tamanho máximo do arquivo é de 1 ectabyte.

- **XFS**

- Permite utilizar o sistema de cotas de usuários e de grupos.
- Trabalha com blocos de até 64 kilobytes.
- O tamanho máximo da partição é de 16 terabytes (32 bits) e 8,58 ectabytes (64 bits).
- O tamanho máximo dos arquivos é de 16 terabytes no Linux 2.4 e de 8,58 ectabytes no Linux 2.6, estando com o endereçamento de 64 bits acionado.



### Saiba mais

Conheça mais sobre os sistemas de arquivo Linux lendo o capítulo 5 do livro:

FERREIRA, R. E. *Linux: guia do administrador do sistema*. 2. ed. São Paulo: Novatec, 2003.

## 8.4 Diretórios do sistema (FHS)

Devido à existência de diversas distribuições disponíveis, foi preciso realizar uma padronização nos diretórios encontrados (cada diretório possui características diferentes). Essa padronização é conhecida como **File System Hierarchy Standard (FHS)**.

Segundo Negus (2014), o **FHS**, que hoje é utilizado por 99% dos sistemas Linux, é mantido pela **Free Standard Groups**, que possui engenheiros de organizações como IBM, Red Hat, Dell e HP.

Essa estrutura está dividida de uma forma hierárquica, em que cada diretório possui uma finalidade diferente.

Dentre as diversas características que um bom técnico Linux deve possuir, pode-se destacar o conhecimento sobre a estrutura de arquivos do Sistema Operacional.

O diretório-raiz do sistema é representado por uma barra normal (/). Abaixo do diretório-raiz, conforme exibido na figura seguinte, são exibidos os subdiretórios que organizam os arquivos no sistema.

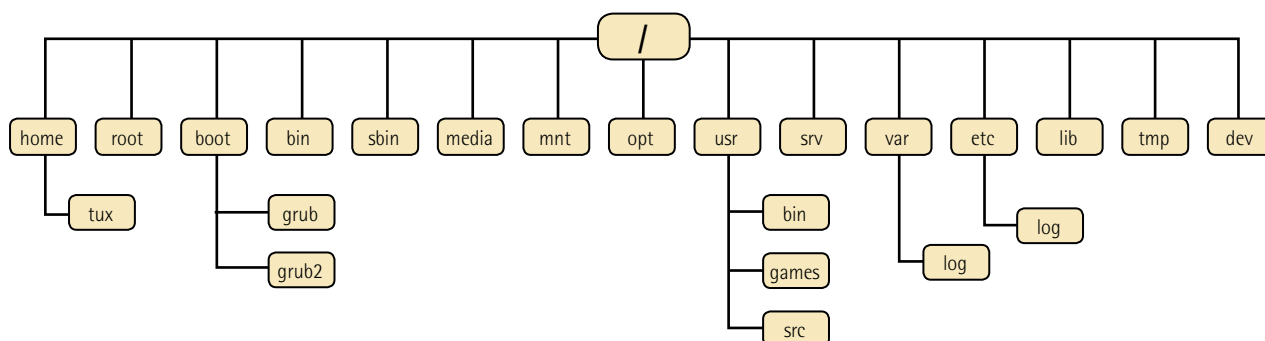


Figura 185 – Estrutura do FHS

A tabela seguinte apresenta a importância de cada um dos diretórios exibidos na figura anterior.

**Quadro 3 – Detalhamento dos diretórios**

Diretório	Descrição
/	Diretório-raiz do sistema GNU/Linux.
Home	Armazena os diretórios e arquivos dos usuários comuns. O nome desse diretório é opcional
Root	Armazena os diretórios e arquivos do usuário <i>root</i> . O nome desse diretório é opcional.
Boot	Contém arquivos necessários para inicialização do sistema, gerenciador de inicialização e imagem do <i>kernel</i>
Bin	Comandos essenciais de uso geral. Todos os usuários podem executar
Sbin	Comandos restritos para o uso do usuário <i>root</i> . O usuário comum poderá executar esses comandos, porém uma configuração no sistema deverá ser realizada
Media	Ponto de montagem para mídias removíveis. Exemplo: CD-ROM, <i>pen drive</i> , cartão de memória.
Mnt	Ponto de montagem para partições de discos locais ou outros diretórios compartilhados em rede
Srv	Dados de serviços providos pelo sistema
Usr	Arquivos e comandos de programas instalados, jogos, serviços etc. Possui a segunda maior árvore de diretórios do sistema
Var	Contém arquivos de dados variáveis, como: <i>logs</i> do sistema, diretórios de <i>spool</i> , caixa de correio etc.
Opt	Utilizado para instalação de programas não essenciais para o funcionamento do sistema. Exemplo: banco de dados, sistema de ERP, navegador etc.
Etc	Arquivos de configuração do sistema do <i>host</i> específico.
Lib	Bibliotecas compartilhadas e módulos do <i>kernel</i>
Tmp	Arquivos temporários
Dev	Arquivos que representam os dispositivos de <i>hardware</i>

## 8.5 Manipulando *hardwares* e dispositivos

A seguir, será descrita a manipulação de *hardwares* e dispositivos em sistemas Linux, já que neste Sistema Operacional eles trabalham de maneira bem diferente da verificada nos sistemas Windows.

### 8.5.1 Processo de *boot*

Existem duas etapas na verificação do processo de boot.

- **Primeira etapa**

- Testes de POST (Power on Self Tests).
- Verificar a integridade física do equipamento.
- Detecção de componentes básicos, como: memória, discos, placa de vídeo etc.

- Segunda etapa

- Identificando o dispositivo de inicialização.
- Será verificado, na Master Boot Record (MBR), o Boot Loader:
  - Linux: Lilo, Grub, Grub2;
  - Windows: NT Boot Loader.
- O gerenciador de *boot* indica ao BIOS os arquivos que deverão ser carregados para a memória e os arquivos que devem ser executados para iniciar o equipamento.
- O arquivo `/boot/vmlinuz` (*kernel*) é executado.
- O arquivo `/boot/initrd` (módulos) é carregado na memória para o acesso aos periféricos.

### 8.5.2 Diretórios "virtuais" `/proc` e `/sys` e seus arquivos

Os dispositivos de *hardware*, com exceção da interface de rede, são representados por um arquivo como: HD, terminais de acesso, portas de comunicação, placa de vídeo, placa de som etc.

Esses arquivos ficam armazenados no diretório `/dev/`, onde são criados durante o momento do *boot* do sistema.

Existem dois diretórios "virtuais" que também são criados no momento da inicialização do sistema: o `/proc` e o `/sys/`. Esses dois diretórios não são criados na instalação do sistema.

Para controlar os dispositivos de *hardware* em funcionamento e todos os processos ativos do sistema, o *kernel* utiliza esses dois diretórios.

Esses arquivos são solicitados por alguns comandos do sistema, por exemplo, o comando `lsmod`, que informa os módulos carregados. Exemplos:

- **`/proc/interrupts`:** informa as IRQs em uso por cada periférico.
- **`/proc/bus`:** contém informações específicas do barramento.
- **`/proc/cpuinfo`:** contém informações do processador utilizado (modelo, fabricante etc).
- **`/proc/devices`:** informa os *drivers* de dispositivos configurados no *kernel* que estão em execução.
- **`/proc/dma`:** informa os canais de acesso à memória que estão em uso.



- **/proc/diskstats:** informa as estatísticas de entrada e saída dos dispositivos de bloco.
- **/proc/filesystems:** informa os sistemas de arquivos que possuem suporte do *kernel*.
- **/proc/ioports:** informa as portas de entrada e saída que estão sendo utilizadas.
- **/proc/meminfo:** informa o uso, pelo sistema, de memória RAM, *buffer*, *cache*, entre outros.
- **/proc/modules:** informa os módulos que foram carregados pelo *kernel*.
- **/proc/partitions:** informa os HDs e suas devidas partições reconhecidas pelo sistema.
- **/proc/swaps:** informa o tamanho da memória SWAP utilizada.

No diretório **/sys** também existem outros arquivos referentes ao *hardware*, em que os valores podem ser alterados. Exemplos:

- **/sys/block:** informações dos dispositivos de blocos do HD, CD, DVD, entre outros.
- **/sys/bus:** informações dos barramentos que possuem suporte do *kernel*.
- **/sys/class:** organiza os tipos de dispositivos que possuem suporte do sistema em subdiretórios.
- **/sys/module:** módulos carregados na memória pelo *kernel*. Podemos habilitar/desabilitar esses módulos usando o comando **modprob**. Sempre que um dispositivo é adicionado, um arquivo é criado nesse diretório e gerenciado pelo programa *udev*.

### 8.5.3 Dispositivos de *hardware*

É extremamente necessário saber identificar os arquivos que compõem o *hardware* do computador, para que se tenha um bom resultado ao utilizar o Linux.

#### 8.5.3.1 Barramento

O barramento serve para comunicação do periférico com o processador. Alguns tipos:

- **ISA:** comunicação de 8 a 16 bits.
- **PCI:** comunicação de 32 a 64 bits.
- **AGP:** comunicação de 32 bits (placas gráficas).

Os periféricos são identificados através de arquivos especiais no diretório **/proc**.

### 8.5.3.2 Portas de entrada e saída

Cada periférico necessita de um canal de comunicação específico para a troca de informações com o processador e pode conter mais de um endereço I/O. As portas estão localizadas no arquivo `/proc/ioports`.

### 8.5.3.3 Interrupções (IRQs)

Segundo Negus (2014), toda troca de informação entre os periféricos e o processador deverá ser sinalizada, para que o processador detecte que existem dados a serem lidos.

As mensagens de ativação poderão ser visualizadas via comando **dmesg**. As IRQs associadas aos periféricos estão localizadas no arquivo `/proc/interrupts`.

### 8.5.3.4 Direct Memory Access (DMA)

Faz o controle de determinados dispositivos de um computador para acessarem a memória do sistema de forma independente do Processador.

### 8.5.3.5 Plug and play

Tem o objetivo de fazer com que os dispositivos de um computador sejam automaticamente reconhecidos pelo Sistema Operacional, facilitando desta forma a expansão segura dos computadores e eliminando a configuração manual.

Para consultar os periféricos Plug And Play, pode-se utilizar o comando `lsnp` ou consultar no diretório `/proc/bus/pnp`.

### 8.5.3.6 Portas seriais

Utilizadas para troca de informações e para conexão de periféricos.

Os periféricos seriais foram substituídos pelo Universal Serial Bus (USB), conforme exibido na tabela a seguir, que possui recursos de configuração automática.

**Tabela 1 – Lista de portas seriais**

Portas Windows	Portas Linux	Endereço de I/O	IRQs
COM 1	ttys0	3F8-3FF	4
COM 2	ttys1	2F8-2FF	3
COM 3	ttys2	3E8-3EF	4
COM 4	ttys3	2E8-2EF	3

### 8.5.3.7 Portas paralelas

Conforme exibido na tabela seguinte, as portas de comunicação paralelas estabelecem um canal de comunicação maior que as portas seriais e podem estar disponíveis de LPT1 a LPT4.

**Tabela 2 – Lista de portas paralelas**

Porta	Linux	Endereço de I/O	IRQs
LPT 1	lp0	378-37F	7
LPT 3	lp1	278-27F	5
LPT 2	lp2	3BC-3BE	4

### 8.5.3.8 Placas de som

Em um ambiente computacional, os computadores podem possuir placas de controle de emissão de áudio. Neste caso, se necessário, um *driver* para esse periférico deverá ser carregado, tarefa que pode ser executada através do comando **lspci -v**.

Os módulos de ativação da placa de som estão listados no arquivo **/etc/modules.conf**. Para instalar a placa de som, deve-se carregar o pacote **alsa-utils**.

### 8.5.3.9 Small Computer System Interface (SCSI)

Representa uma tecnologia que permite ao usuário conectar uma vasta lista de dispositivos, como discos rígidos, CD-ROM, *scanners* e impressoras. Dentre os padrões, podem ser destacados:

- SCSI-1: interconexão de até 7 periféricos.
- SCSI-3: interconexão de até 126 periféricos em um único barramento.
- *Fibre channel*: com largura de banda de até 2 Gbps.
- Cabo coaxial: possui uma distância de conexão de até 24 metros.
- Fibra óptica: possui uma distância de conexão de até 10 quilômetros.

Em versões recentes do Linux, encontra-se o modelo SCSI generic chamado de **sg\_map**, e as informações do *driver* encontram-se no diretório **/proc/scsi/sg/**.

### 8.5.3.10 Universal Serial Bus (USB)

Criado para permitir a conexão de periféricos ao computador sem desligar o equipamento e com uma configuração dinâmica de *hardware*.

- **Versões de USB**

- USB 1.0 e 1.1: taxas de transferência de até 12 Mbps.
- USB 2.0: taxas de transferência de até 480 Mbps, utilizado para discos externos, como HDs, CDs e DVDs.

A tabela seguinte exibe uma lista de *drivers* usados pelos dispositivos USB.

**Quadro 4 – Lista de *drivers* USB**

Driver	Chipset
EHCI	Suporte USB 2.0, requer o uso de UHCI, OHCI ou JE
JE	Alternativa ao driver UHCI, utilizado no <i>kernel</i> 2.4
OHCI	Para equipamentos COMPAQ, Power MACs, iMACs, entre outros

### 8.5.4 Hot plugging

Esses recursos são controlados pelos comandos **usbmgr** e **hotplug**, ambos utilizados pelo *kernel* do Linux. As configurações de cada sistema ficam nos diretórios **/etc/usbmgr** e **/etc/hotplug**.

#### 8.5.4.1 Identificando o HD no sistema

Da mesma forma que em outros Sistemas Operacionais, o Linux conta com uma série de discos, dentre os quais podemos citar:

- Integrated Drive Electronics (IDE): discos antigos.
- Serial ATA (SATA): possui maiores capacidades de armazenamento e taxas de transferência mais elevadas.
- Small Computer System Interface (SCSI): para uma maior quantidade de informações, permitindo a conexão de vários periféricos. Possui taxas de transferência superiores às dos padrões IDE e SATA.

HDs SATA e SCSI são identificados com o nome do arquivo SD.

Caso seu computador possua somente um HD, será criado um arquivo identificado como SDA.

A cada novo HD adicionado, um novo arquivo é criado, alterando a última letra do nome para b, c, d e assim por diante.



### Lembrete

O arquivo que informa os HDs e suas devidas partições reconhecidas pelo sistema estão em `/proc/partitions`.

Se o computador possuir 1 HD com duas partições, serão encontrados os seguintes arquivos no diretório `/dev`:

- SDA: representa o **primeiro** HD.
- SDA1: representa a **primeira partição** do primeiro HD.
- SDA2: representa a **segunda partição** do primeiro HD.

Adicionando um segundo HD ou um *pen drive*, a regra é a mesma: no caso de um dispositivo de CD/DVD, o arquivo que representa o dispositivo é identificado como SR0.

## 8.6 Administração de usuários

Assim como outros Sistemas Operacionais que usam um esquema de permissão para controle e gerenciamento da manipulação dos arquivos e diretórios, o GNU/Linux não é diferente. Porém, precisamos entender como essas permissões são gerenciadas pelo sistema.

Todas as vezes que um determinado arquivo/diretório é criado, o sistema já associa o nome e o grupo do usuário como responsável pela administração e pelo uso deles, e também a definição do tipo de manipulação, podendo ser leitura, gravação ou execução.

Por padrão, a criação de um arquivo recebe as seguintes permissões:

- Leitura e gravação para o dono do arquivo.
- Leitura para o grupo do arquivo.
- Leitura para outros usuários.

No caso da criação de um diretório, as permissões são:

- Leitura, gravação e execução para o dono.
- Leitura e execução para o grupo.
- Leitura e execução para outros.



### Observação

Somente o dono do arquivo/diretório poderá alterar as permissões, ou trocar o grupo ou o dono.

O usuário *root* pode alterar livremente as configurações do sistema.

#### 8.6.1 Administrando os usuários

O sistema GNU/Linux reconhece três tipos de usuário:

- **Administrador:** identificado como *root*, esse usuário poderá efetuar qualquer tipo de operação no sistema. Possui o poder de superusuário.
- **Sistema:** tem a função de iniciar um determinado serviço, associado ao seu nome de usuário.
- **Comum:** os usuários com essa característica podem gerenciar seus arquivos/diretórios apenas em seu diretório pessoal localizado no */home*.

Para criação dos usuários, utilizamos o comando **adduser**.

Para cadastrar um usuário no sistema, basta utilizar o comando a seguir:

```
[root@srvcentos7 ~]# adduser operador
```

Esse comando, além de criar o usuário no sistema, realiza as seguintes configurações:

- cria o diretório pessoal do usuário operador, com o mesmo nome no */home*;
- copia os arquivos *.bash\_logout*, *.bash\_profile* e *.bashrc* do diretório */etc/skel/* para o diretório pessoal;
- cria o grupo primário do usuário chamado operador e adiciona no arquivo */etc/group* e */etc/gshadow*;
- cadastra as informações pessoais do usuário no arquivo */etc/passwd*;
- bloqueia a conta do usuário no arquivo */etc/shadow*.

As informações dos usuários são mantidas no arquivo */etc/passwd* e são organizadas por um sinal de dois-pontos (:).

Cada linha do arquivo refere-se ao usuário cadastrado, e a edição dessas informações pode ser realizada através de alguns comandos.

Após a criação do usuário, é necessário definir uma senha para o acesso ao sistema.

Para definir a senha, utilizamos o comando **passwd**.

```
[root@srvcentos7 ~]# passwd operador
```

Será necessário informar a senha duas vezes, e se informar uma senha muito simples, o sistema irá apresentar uma mensagem informando que a senha não passa pela verificação dos dicionários, porém a aceitará. Lembrando que no Debian não ocorre essa verificação.

### 8.6.2 Permissão dos arquivos e diretórios

Diz respeito ao controle de acesso de acordo com o usuário em que se está operando o sistema. Sem esse controle, a administração seria impossível, fazendo o sistema perder uma de suas principais características, que é a segurança.

Vamos dar um exemplo:

Verifique a propriedade do arquivo yum.conf.

```
[usuario@srvcentos7 ~]$ ls -l /etc/yum.conf
```

Informações:

```
- rw-r--r--    1    root    root   860   Jan 9 13:50   yum.conf
```

Perceba que no segundo campo (**rw-r--r--**) do resultado temos as permissões que dizem respeito a como o arquivo poderá ser usado. No quarto e no quinto campo (**root root**), está a identificação dos nomes do dono e do grupo.

Segue um breve resumo.

- **Dono**

- Identificado pelo mesmo nome do usuário.
- Define as permissões do arquivo ou diretório.

- **Grupo**

- Permite acesso a determinados arquivos a um conjunto de usuários.
- Possui uma identificação conhecida como *group identification* (GID).

- Outros

- Usuários que não são donos e que não fazem parte de determinado grupo.



### Lembrete

Para criação dos usuários, utilizamos o comando **adduser**.

As permissões são representadas pelas letras **r (read)**, **w (write)** e **x (execution)**.

Essa forma de apresentação é conhecida como formato **literal**; porém, para a alteração dessas permissões, poderemos representar em um outro formato, conhecido como formato octal, em que cada letra é representada por um número.

A letra **r** é representada pelo valor 4, a letra **w** é representada pelo valor 2 e a letra **x** é representada pelo valor 1, ficando da seguinte forma:

- **Read (r em formato literal ou 4 em formato octal)**

- Para arquivos, permite sua leitura.
- Para diretórios, permite que seu conteúdo seja listado.

- **Write (w em formato literal ou 2 em formato octal)**

- Para arquivos, permite alteração de seu conteúdo.
- Para diretórios, permite a criação de novos arquivos.

- **Execution (x em formato literal ou 1 em formato octal)**

- Para arquivos, permite que sejam executados. Por exemplo, um *script* em *shell*.
- Para diretórios, habilita seu acesso por meio do comando **cd**.

- **Hífen (-)**

- Anula uma das três permissões, conforme a posição apresentada.

Crie um arquivo chamado **relatorio** no diretório pessoal do usuário operador.

```
[usuario@srvcentos7 ~]$ touch relatorio
```



Verifique as permissões do arquivo criado.

```
[usuario@srvcentos7 ~]$ ls -l relatorio
```

O arquivo de nome **relatorio** tem como dono e grupo o usuário **usuario**. Esse arquivo poderá ter seu conteúdo lido ou alterado tanto pelo dono quanto pelo grupo, e os demais usuários poderão apenas ler o arquivo.

Repare que as permissões de uso do arquivo são definidas em blocos de três letras.

- **rw-**: para o dono do arquivo.
- **rw-**: para o grupo do arquivo.
- **r--**: para os demais usuários do sistema.

Para alterar as permissões do arquivo, usamos o comando **chmod**.

Podemos informar as novas permissões em formato literal ou em formato octal. Para o formato literal, o comando poderá ser usado assim:

```
[usuario@srvcentos7 ~]$ chmod u+x,g-x,o-r relatorio
```

- A permissão de execução foi adicionada para o dono do arquivo **relatorio**.
- A permissão de execução foi removida para o grupo do arquivo **relatorio**.
- A permissão de leitura foi removida para os demais usuários do sistema.

Essa mesma configuração poderá ser representada em formato octal. Nesse caso, utilizaremos o resultado da soma dos valores das respectivas permissões para informar ao comando **chmod**.

Cada letra é representada por um valor: r = 4, w = 2, e x = 1.

```
[usuario@srvcentos7 ~]$ chmod 750 relatorio
```

O 7 representa as permissões do dono do arquivo. Nesse caso, todas as permissões foram adicionadas, e o resultado da soma de todos os valores é 7.

O 5 representa as permissões do grupo do arquivo.

Apenas as permissões r (4) e x (1) foram somadas, tendo como resultado o valor 5. O valor 0 representa as permissões dos demais usuários do sistema.

Nenhum valor foi somado, portanto foi informado 0.

Verifique a permissão do arquivo **relatorio**.

```
[usuario@srvcentos7 ~]$ ls relatorio
```

```
-rwxr-x---. 1 usuario usuario 0 Dez 9 10:27 relatorio
```

Para o uso em um diretório e seus respectivos arquivos e subdiretórios, a opção **-R** é necessária.

```
[usuario@srvcentos7 ~]$ chmod 755 -R /home/operador/diretorio
```

### 8.7 Administração do *shell*

O *shell* é um módulo que atua como interface usuário-sistema operacional, possuindo diversos comandos internos que permitem ao usuário solicitar serviços do Sistema Operacional. Também implementa uma linguagem simples de programação que permite o desenvolvimento de pequenos programas (os famosos *shell scripts*).

Para acesso local, tem até 8 terminais disponíveis. Os seis primeiros são destinados à operação do sistema em modo texto, e os dois últimos servem para o acesso à interface gráfica, caso esteja instalada.

Esses terminais são acionados através de um conjunto de teclas: **Ctrl + Alt + F1, F2, F3, F4, F5, F6, F7 e F8**.

Para o acesso ao Terminal 2, por exemplo, basta acionar as teclas **Ctrl + Alt + F2**.

Podemos trabalhar com quantos terminais for necessário, e cada terminal poderá ser acessado com usuários diferentes. Lembrando sempre que o sistema é multiusuário.

Em modo texto, acionado o terminal 7, o sistema irá apresentar a interface gráfica, em que poderá selecionar o usuário previamente cadastrado no momento da instalação e informar a senha dele. Porém, isso somente ocorrerá se a interface gráfica já estiver instalada.

Mesmo trabalhando em modo texto, é possível alterar para o modo gráfico e vice-versa.

Ao acionar um terminal, temos as seguintes informações:

- **No CentOS**

A primeira linha do terminal representa o nome da distribuição, o nome do *kernel* e a versão da distribuição.

### CentOS Linux 7 (core)

A segunda linha informa a versão do *kernel* e em que plataforma está instalada.

**kernel 3.10.0-327.el7.x86\_64 on an x86\_64**

A terceira linha informa o nome do computador, seguido da palavra *login*. É nessa linha que deve ser informado o usuário que deseja acessar o sistema.

**localhost login:**

- **No Debian**

A primeira linha do terminal representa o nome da distribuição, o nome do Sistema Operacional, a versão da distribuição, o nome do *host* e a identificação do terminal.

**Debian GNU/LINUX 8 <nomehost> tty1**

A segunda linha informa o nome do *host*, seguido da palavra *login*. É nessa linha que se deve informar o usuário que deseja acessar o sistema.

**<nomehost> login:**

### 8.7.1 Diferenças de acesso entre os usuários *root* e comum

Após a autenticação com usuário e senha, um Prompt de Comando é apresentado. Por meio dele, ocorre a operação do sistema.

[root@srvcentos7 ~]#

usuario@srvdebian8:~\$

Esse Prompt é composto das seguintes informações:

- **Usuário ou *root*:** nome do usuário logado no sistema.
- **@:** separador do nome do usuário e do nome da máquina.
- **srvcentos7:** nome da máquina.
- **~:** representa o diretório pessoal do usuário.
- **\$ ou #:**

- **\$:** identificação do usuário comum;
- **#:** identificação do usuário *root*.

O comando **pwd** informa o diretório de trabalho atual.

```
[usuario@srvcentos7 ~] $pwd
```

```
/home/usuário
```

Não é uma regra, mas, por uma questão de precaução, sempre opere o sistema com um usuário comum que, quando houver a necessidade de realizar alguma atividade que exija um privilégio de administrador, acessa o sistema como *root*.

Para mudar a conta de um usuário comum para o *root*, utilizando o mesmo terminal, digite o comando **su** com **-** e informe em seguida a senha do *root*.

```
[usuario@srvcentos7 ~]$ su -
```

### 8.8 Comandos úteis da linha de comando do Linux

No Linux, existem diversos comandos e sintaxes que facilitam ou ajudam a operação desse sistema, sendo os listados a seguir alguns dos mais utilizados.

- **ls:** exibe os arquivos que estão dentro da pasta em que o usuário está no momento. Para utilizar, basta digitar **ls** na linha do *shell*. Existem variações, como **ls -l**, com a qual é possível obter informações mais detalhadas sobre os arquivos, como permissões e tamanho. Exemplos:
  - **-l:** lista os arquivos em formato detalhado;
  - **-a:** lista os arquivos ocultos (iniciam com **.**).
  - **-h:** exibe o tamanho num formato legível (combinar com **-l**);
  - **-R:** Diretório-raiz.
- **grep:** basicamente, pode ser usado para pesquisar uma *string* (cadeia de caracteres) em um conjunto de arquivos:

```
grep -r "minha string" *.txt
```

O comando dado pesquisará a cadeia "minha string" em todos os arquivos com extensão **.txt**, recursivamente.

Supondo que se deseje encontrar todas as ocorrências da *string* **mcrypt**, dentro do arquivo **/etc/php5/apache2/php.ini**, uma sugestão é usar esse comando da seguinte forma:

```
cat /etc/php5/apache2/php.ini | grep -i mcrypt
```

O parâmetro **-i** desliga a sensibilidade à caixa das letras –, ou seja, é indiferente estar em minúsculas ou não.

- **cd**: serve para acessar e mudar de diretório corrente. Ele é utilizado para a navegação entre as pastas do computador. Exemplo:

```
cd /home/usuario/Desktop
```

Acessa a pasta correspondente à área de trabalho do usuário **usuario**.

- **shutdown**: comando utilizado para desligar o computador. Exemplos:
  - **shutdown -r now**: reinicia o computador;
  - **shutdown -h now**: desliga o computador (só desligue quando aparecer escrito *system halted* ou algo equivalente).



### Observação

O **now** pode ser mudado. Por exemplo: **shutdown -r +10** fará o sistema reiniciar depois de 10 minutos.

- **pwd** e **passwd**:

**pwd**: comando utilizado para exibir a pasta atual em que o usuário se encontra. Exemplo: se o usuário-teste digitar **cd ~/** e, em seguida, digitar **pwd**, o retorno será **/home/teste**.

**passwd**: altera a senha de um usuário exibindo um Prompt para que a nova senha seja fornecida e logo depois repetida para confirmação. O usuário logado pode alterar a própria senha digitando apenas **passwd**.

- **mv**: tem a função de mover arquivos. Sua utilização é praticamente igual à do comando **cp**, abordado a seguir. Exemplos:

```
mv MV.txt /home/teste/Trabalho/MV.txt
```

O arquivo **MV.txt** foi movido para a pasta **/home/teste/Trabalho** com o mesmo nome.

```
mv MV.txt VM.txt
```

O arquivo MV.txt continuou onde estava, porém agora tem o nome de VM.txt.

- **cp**: comando utilizado para copiar arquivos. Exemplo:

**cp Exemplo.doc /home/teste/Trabalho/EXEMPLO.doc**

O arquivo EXEMPLO.doc foi copiado para a pasta /home/teste/Trabalho com o mesmo nome.

- **rm**: tem a função de remover arquivos. Cuidado ao utilizá-lo, pois caso remova algum arquivo por engano o erro será irreversível. Exemplo:

**rm /home/teste/Arquivo.txt**

O arquivo Arquivo.txt localizado na pasta /home/teste foi deletado.

- **chmod**: altera as permissões de acesso a arquivos/diretórios.

**chmod 777 arquivo**

Dá permissão total de acesso ao arquivo. O primeiro número se refere ao proprietário, o segundo ao grupo e o terceiro aos demais usuários. Lembrando que 1 = executar (x), 2 = escrever (w) e 4 = ler (r), deve ser colocada a soma das opções desejadas.

- **chown**: executado pelo *root*, permite alterar o proprietário ou o grupo do arquivo ou diretório. Alguns exemplos:

**chown usuário arquivo**

**chown usuário diretório**



### Lembrete

Para saber quem é o dono e qual grupo é o proprietário da pasta, basta usar o comando **ls -l /**.

- **ifconfig** e **iwconfig**:

- **ifconfig**: permite configurar as interfaces de rede, sendo o comando utilizado na inicialização do sistema para configuração dessas interfaces. Caso nenhum argumento seja passado junto do comando, este apenas exibirá o estado das interfaces atualmente definidas.

- **ifconfig eth0**: para exibir o estado e as informações da interface de rede eth0;

- **ifconfig eth1 down:** para desativar a interface de rede eth1;
- **ifconfig eth1 up:** para ativar a interface de rede eth1.
- **iwconfig:** similar ao comando **ifconfig**, mas usado para redes Wi-Fi. Permite verificar diversas características das redes *wireless*.
- **ps:** exibe os processos em execução no sistema. Pode ser usado sem parâmetro algum, mas vai produzir uma quantidade muito grande (ou muito pequena) de informações. É costume "filtrar" sua saída com alguns parâmetros e combinando outros comandos, para ver apenas o que interessa. Exemplos:

**ps aux**

Utiliza-se a sintaxe **less** para pausar entre telas:

**ps aux | less**

Usa-se a tecla **q** para sair do **less**.

Para ver apenas os processos referentes ao navegador Chromium, usa-se assim:

**ps aux | grep -i chromium**

Isso é muito útil para descobrir o PID de um processo que se deseja interromper à força, com o comando **kill**.

- **su e sudo:**
  - **su:** passa para o superusuário (no Prompt, onde havia \$, aparecerá #);
  - **sudo:** permite a um usuário autorizado, conforme configurado no arquivo `/etc/sudoers`, executar comandos como se fosse o superusuário (*root*) ou outro usuário qualquer.
- **apt-get:** recurso desenvolvido originalmente para a distribuição Debian, que permite a instalação e a atualização de pacotes (programas, bibliotecas de funções etc.) no Linux, de maneira fácil e precisa. Vale frisar que deve ser utilizado por um usuário com privilégios de administrador.

**dd:** o comando dd é um clássico dos ambientes Unix-Like que permite fazer uma cópia exata de um arquivo, ou seja, uma cópia bit a bit. Sintaxe básica:

dd if=origem of=destino

## 8.9 Instalando, removendo e atualizando programas

De acordo com Ferreira (2003), uma das principais características das distribuições GNU/Linux é o gerenciamento de pacotes.

O programa **rpm** é um gerenciador de pacotes criado pela Red Hat. Tem a função de realizar diversas tarefas administrativas, como instalar, remover, atualizar, consultar etc.

No caso do Debian, o gerenciador de pacotes se chama **dpkg** e possui as mesmas funções descritas para o **rpm**.

Para instalar algum programa utilizando o comando **rpm** ou o **dpkg**, será necessário já ter o pacote no sistema, conforme exibido na tabela seguinte.

**Quadro 5 – Comandos de instalação de pacotes**

Distribuição	Pacote pré-compilado	Repositório
Red Hat	rpm	yum
Suse	rpm	yast
Debian	dpkg	apt-get ou aptitude

### 8.9.1 Instalando um programa

Execute a linha de comando para o acesso aos arquivos da imagem. Montará a unidade em que se encontram os arquivos de instalação:

```
[root@srvcentos7 ~]# mount -t iso9660 /dev/sr0 /media/
```

```
mount: /dev/sr0 is write-protected, mount read-only
```

Os pacotes .rpm estão disponíveis no diretório /media/Packages/.

Acessando o seguinte diretório, você pode instalar o pacote yum:

```
[root@srvcentos7 ~]# cd /media/Packages
```

Após instalar os pacotes **yum-plugin-fastestmirror-1.1.31-34.el7.noarch.rpm** e **yum-3.4.3-132.el7.centos.0.1.noarch.rpm**:

```
[root@srvcentos7 Packages]# rpm -ivh yum-plugin-fastestmirror-1.1.31-34.el7.noarch.rpm  
yum-3.4.3-132.el7.centos.0.1.noarch.rpm
```

Aviso:

```
yum-plugin-fastestmirror-1.1.31-34.el7.noarch.rpm
```



Cabeçalho:

V3 RSA/SHA256 Signature, ID da chave f4a80eb5: NOKEY

Preparando... ##### [100%]

Updating / installing...

1:yum-3.4.3-132.el7.centos.0.1 ##### [ 50%]

2:yum-plugin-fastestmirror-1.1.31-3 ##### [100%]

O parâmetro `-i` instala o pacote informado. Use-o em conjunto com as opções `vh`, em que o `v` mostrará a instalação sendo executada, e o `h`, uma barra de progresso.

### 8.9.2 Atualizando um programa

No **CentOs**, para atualizar determinado pacote, obtenha o arquivo necessário e use a opção **U**:

```
[root@centos7 Packages]# rpm -Uvh yum-3.4.3-132.el7.centos.0.1.noarch.rpm
```

No Debian, obtenha o arquivo atualizado e utilize a opção `-i`:

```
root@srvdebian8 /media/pool/main/a/aptitude# dpkg -i *.deb
```

### 8.9.3 Desinstalando um pacote

Com o comando `rpm`, use o parâmetro `-e` para desinstalar o programa yum:

```
[root@srvcentos7 ~]# rpm -e yum
```

## 8.10 Adicionando *softwares* no *desktop* do Linux

Negus (2014) refere-se a uma distribuição Linux para usuários de computadores pessoais, como interface gráfica e aplicativos de uso pessoal.

O Linux oferece muitas alternativas para *desktop*. As mais populares são **GNOME**, **KDE**, **Xfce** e **LXDE**, que constituem grandes coleções de programas para um ambiente *desktop*, em vez de apenas o "esqueleto".

Esses ambientes apresentam interfaces gráficas que tentam simular um escritório. Todos eles permitem ao usuário configurar várias preferências pessoais e realizar várias tarefas de gerenciamento de seu sistema.

### 8.11 Soluções corporativas

São ferramentas desenvolvidas a fim de otimizar e/ou facilitar o uso do Linux no ambiente corporativo, com empresas especializadas nesse tipo de solução. Algumas soluções corporativas serão descritas a seguir.

#### 8.11.1 OpenStack

Trata-se de um conjunto de projetos de *software* de código aberto usados para configurar e operar infraestrutura de computação e armazenamento em Nuvem, gerenciando os componentes de múltiplas infraestruturas virtualizadas.

Assim como o Sistema Operacional gerencia os componentes de nossos computadores, o OpenStack é chamado de Sistema Operacional da Nuvem por cumprir o mesmo papel em maior escala.

A administração desses recursos é realizada através de um console *web*.

#### 8.11.2 Vagrant

*Software* de computador que cria e configura ambientes virtuais de desenvolvimento. Pode ser visto como um *wrapper* de nível superior em torno de *softwares* de virtualização como VirtualBox, VMware, KVM e Containers Linux (LXC), e de *softwares* de gerenciamento de configuração como Ansible, Chef, Salt e Puppet.

O Vagrant foi originalmente ligado à VirtualBox, mas a versão 1.1 serve como suporte para outros *softwares* de virtualização, como VMware e KVM, e para ambientes de servidor, como Amazon EC2. Embora escrito em Ruby, pode ser usado em projetos escritos em outras linguagens de programação, como PHP, Python, Java, C # e JavaScript.

#### 8.11.3 Puppet

Projetado para gerenciar a configuração de sistemas Unix-like e Microsoft Windows declaradamente. O usuário descreve os recursos do sistema e do seu estado, usando linguagem declarativa do Puppet ou um DSL Ruby (linguagem específica de domínio). Essas informações são armazenadas em arquivos chamados *Puppet manifests*. O Puppet descobre as informações do sistema através de um utilitário chamado *facter* e compila o *Puppet manifest* em um catálogo específico do sistema contendo recursos e dependência de recursos, que são aplicados contra os sistemas de destino. Quaisquer ações tomadas por Puppet são então relatados.

#### 8.11.4 Docker

Programa de código aberto que permite empacotar um aplicativo Linux e suas dependências como um recipiente.

Conhecido como *container*, tem como base a virtualização isolada de aplicativos, uns em relação aos outros, em um Sistema Operacional comum. Essa abordagem padroniza a entrega do programa,

permitindo que aplicativos rodem em qualquer ambiente Linux, físico ou virtual. Como eles compartilham o mesmo Sistema Operacional, os recipientes são portáteis entre distribuições Linux diferentes e são significativamente menores do que as imagens de máquina virtual (VM).

O Docker compete com recipientes de aplicações proprietárias, como o VMware vApp, e com ferramentas de captação de infraestrutura, como o Chef.



### Resumo

Nesta unidade aprendemos a realizar a instalação do Linux com a distro CentOS, que é uma distribuição livre. Após a instalação e seus processos, aprendemos sobre o editor de texto VIM, um dos mais utilizados por profissionais de Linux.

Aprendemos também sobre os sistemas de arquivos que possuem suporte para o Sistema Operacional Linux, como Ext3 e Ext4.

Por fim, aprendemos a administrar *hardwares* e dispositivos no sistema, além de usuários e *shell*, bem como a realizar instalação, atualização e desinstalação de programas, finalizando com algumas soluções utilizadas em ambiente corporativo.



### Exercícios

**Questão 1.** Em muitos sistemas Linux, o shell padrão é o bash shell. Para descobrir qual o shell padrão do sistema Linux, deve-se digitar um comando no terminal:

- A) `$ echo $Shell.`
- B) `$ ECHO $Shell.`
- C) `$ echo $SHELL.`
- D) `$ ECHO $SHELL.`
- E) `$$ echo $$SHELL.`

Resposta correta: alternativa C.

### Análise da resposta

Para saber qual shell você está usando, use um dos comandos a seguir:

\$echo \$0 – retorna o nome do shell, exemplo: bash.

\$echo \$SHELL – retorna o caminho (*path*), exemplo: /bin/bash.

**Questão 2.** (Fau 2016, adaptada) Qual das expressões a seguir é utilizada para se referenciar à interface gráfica do Linux?

A) Kernel.

B) Terminal.

C) Shell.

D) Console.

E) Ambiente X.

**Resolução desta questão na plataforma.**

## FIGURAS E ILUSTRAÇÕES

### Figura 3

AMARAL, H. *Windows 95*. São Paulo: Atlas, 1996, p. 23.

### Figura 4

AMARAL, H. *Windows 95*. São Paulo: Atlas, 1996, p. 28.

### Figura 5

DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. *Sistemas Operacionais*. 3. ed. São Paulo: Pearson, 2010, p. 20.

### Figura 6

DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. *Sistemas Operacionais*. 3. ed. São Paulo: Pearson, 2010, p. 24.

### Figura 7

DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. *Sistemas Operacionais*. 3. ed. São Paulo: Pearson, 2010, p. 27.

### Figura 79

RIR-MAP.SVG. Disponível em: <[https://www.iana.org/\\_img/2013.1/rir-map.svg](https://www.iana.org/_img/2013.1/rir-map.svg)>. Acesso em: 28 nov. 2017.

## REFERÊNCIAS

### Textuais

AMARAL, H. *Windows 95*. São Paulo: Atlas, 1996. p. 23 e 28.

BATTISTI, J.; POPOVICI, E. *Windows Server 2012 R2 e Active Directory*. São Paulo: Instituto Alpha, 2015. p. 47.

COX, J.; PREPPERNAU, J. *Windows 7: passo a passo*. Porto Alegre: Artmed, 2010.

DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. *Sistemas Operacionais*. 3. ed. São Paulo: Pearson, 2010 p. 20, 24 e 27.

FERREIRA, R. E. *Linux: guia do administrador do sistema*. 2. ed. São Paulo: Novatec, 2003.

INTERNET ASSIGNED NUMBERS AUTHORITY (IANA). *Number resources*. 2017. Disponível em: <<https://www.iana.org/numbers>> <[https://www.iana.org/\\_img/2013.1/rir-map.svg](https://www.iana.org/_img/2013.1/rir-map.svg)>. Acesso em: 10 ago. 2017.

MICROSOFT. TechNet Evaluation Center. *Windows Server: avaliações*. 2017. Disponível em: <<https://www.microsoft.com/pt-br/evalcenter/evaluate-windows-server-2012-r2>>. Acesso em: 10 set. 2017.

NEGUS, C. *Linux, a Bíblia: o mais abrangente e definitivo guia sobre Linux*. Rio de Janeiro: Alta Books, 2014.

OLIVEIRA, A. L. de. *Windows Server 2012 R2: uma abordagem prática de Suporte de TI e Redes Corporativas*. Santa Cruz do Rio Pardo: Viena, 2016.

TANENBAUM, A. S.; WETHERALL, D. *Redes de computadores*. 5. ed. São Paulo: Pearson, 2011. p. 31, 48.

ZACKER, C. *Instalação e configuração do Windows Server 2012 R2*. São Paulo: Bookman, 2015. (Série Exam 70-410).

## Sites

<[www.centos.org](http://www.centos.org)>.

<[www.centos.org/about/](http://www.centos.org/about/)>.

## Exercícios

Unidade I – Questão 1: FUNDAÇÃO DE APOIO DA UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL (Faurgs). Concurso público para provimento do cargo de Analista Judiciário 2017: (Ciências Jurídicas e Sociais). Questão 69. Disponível em: <[https://www.qconcursos.com/arquivos/prova/arquivo\\_prova/55564/faurgs-2017-tj-rs-analista-judiciario-area-administrativa-administracao-ciencias-contabeis-economia-prova.pdf](https://www.qconcursos.com/arquivos/prova/arquivo_prova/55564/faurgs-2017-tj-rs-analista-judiciario-area-administrativa-administracao-ciencias-contabeis-economia-prova.pdf)>. Acesso em: 2 out. 2018.

Unidade II – Questão 2: INAZ DO PARÁ. Analista de Informática 2017: Curso Superior. Questão 40. Disponível em: <[https://www.qconcursos.com/arquivos/prova/arquivo\\_prova/54561/inaz-do-para-2017-dpe-pr-analista-de-informatica-prova.pdf](https://www.qconcursos.com/arquivos/prova/arquivo_prova/54561/inaz-do-para-2017-dpe-pr-analista-de-informatica-prova.pdf)>. Acesso em: 2 out. 2018.

Unidade III – Questão 1: FACULDADE GETÚLIO VARGAS (FGV). Tribunal de Contas do Estado de Sergipe (Concurso Público) 2015: Analista de Tecnologia da Informação – Suporte Técnico em Infraestrutura e Redes. Questão 67. Disponível em: <[https://fgvprojetos.fgv.br/sites/fgvprojetos.fgv.br/files/concursos/tcese/TCE\\_SE\\_Analista\\_de\\_Tecnologia\\_da\\_Informacao\\_-\\_Suporte\\_Tecnico\\_em\\_Infraestrutura\\_e\\_Redes\\_\(ANTIREDES\)\\_Tipo\\_1.pdf](https://fgvprojetos.fgv.br/sites/fgvprojetos.fgv.br/files/concursos/tcese/TCE_SE_Analista_de_Tecnologia_da_Informacao_-_Suporte_Tecnico_em_Infraestrutura_e_Redes_(ANTIREDES)_Tipo_1.pdf)>. Acesso em: 2 out. 2018.

Unidade III – Questão 2: INSTITUTO AMERICANO DE DESENVOLVIMENTO (Iades). Empresa Brasileira de Serviços Hospitalares 2013: Analista TI – Sistemas Operacionais. Questão 30. Disponível em: <[https://www.qconcursos.com/arquivos/prova/arquivo\\_prova/29624/iades-2013-ebserh-analista-de-tecnologia-da-informacao-sistemas-operacionais-prova.pdf](https://www.qconcursos.com/arquivos/prova/arquivo_prova/29624/iades-2013-ebserh-analista-de-tecnologia-da-informacao-sistemas-operacionais-prova.pdf)>. Acesso em: 2 out. 2018.

Unidade IV – Questão 1: INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (Inep). Exame Nacional de Desempenho dos Estudantes (Enade) 2011: Engenharia

Grupo III. Questão 26. Disponível em: <[http://download.inep.gov.br/educacao\\_superior/enade/provas/2014/40\\_tecnologia\\_analise\\_desenv\\_sistemas.pdf](http://download.inep.gov.br/educacao_superior/enade/provas/2014/40_tecnologia_analise_desenv_sistemas.pdf)>. Acesso em: 2 out. 2018.

Unidade IV – Questão 2: FUNDAÇÃO CARLOS CHAGAS (FCC). Concurso público para provimento de cargos de Analista Judiciário – Área apoio especializado. Especialidade Tecnologia da Informação 2013. Questão 28. Disponível em: <[http://download.inep.gov.br/educacao\\_superior/enade/provas/2014/40\\_tecnologia\\_analise\\_desenv\\_sistemas.pdf](http://download.inep.gov.br/educacao_superior/enade/provas/2014/40_tecnologia_analise_desenv_sistemas.pdf)>. Acesso em: 2 out. 2018.

Unidade V – Questão 1: CENTRO DE SELEÇÃO E DE PROMOÇÃO DE EVENTOS (Cespe). Analista Judiciário – Tecnologia da Informação 2017. Questão 68. Disponível em: <[https://www.qconcursos.com/arquivos/prova/arquivo\\_prova/55852/cespe-2017-trt-7-regiao-ce-analista-judiciario-tecnologia-da-informacao-prova.pdf](https://www.qconcursos.com/arquivos/prova/arquivo_prova/55852/cespe-2017-trt-7-regiao-ce-analista-judiciario-tecnologia-da-informacao-prova.pdf)>. Acesso em: 4 out. 2018.

Unidade V – Questão 2: INAZ DO PARÁ. Analista de Informática 2017: Curso Superior. Questão 36. Disponível em: <[https://www.qconcursos.com/arquivos/prova/arquivo\\_prova/54561/inaz-do-para-2017-dpe-pr-analista-de-informatica-prova.pdf](https://www.qconcursos.com/arquivos/prova/arquivo_prova/54561/inaz-do-para-2017-dpe-pr-analista-de-informatica-prova.pdf)>. Acesso em: 4 out. 2018.

Unidade VI – Questão 1: FUNDAÇÃO CARLOS CHAGAS (FCC). Técnico Judiciário – Suporte Técnico 2012. Questão 50. Disponível em: <[https://www.qconcursos.com/arquivos/prova/arquivo\\_prova/25311/fcc-2012-tj-pe-tecnico-judiciario-suporte-tecnico-prova.pdf](https://www.qconcursos.com/arquivos/prova/arquivo_prova/25311/fcc-2012-tj-pe-tecnico-judiciario-suporte-tecnico-prova.pdf)>. Acesso em: 4 out. 2018.

Unidade VI – Questão 2: CENTRO DE SELEÇÃO E DE PROMOÇÃO DE EVENTOS (Cespe). Técnico Judiciário – Operação de Computadores 2015. Questão 23. Disponível em: <[https://www.qconcursos.com/arquivos/prova/arquivo\\_prova/46422/cespe-2015-tre-rs-tecnico-judiciario-operacao-de-computadores-prova.pdf](https://www.qconcursos.com/arquivos/prova/arquivo_prova/46422/cespe-2015-tre-rs-tecnico-judiciario-operacao-de-computadores-prova.pdf)>. Acesso em: 4 out. 2018.

Unidade VII – Questão 1: CENTRO DE SELEÇÃO E DE PROMOÇÃO DE EVENTOS (Cespe). Auxiliar Judiciário – Conhecimentos Básicos 2012. Questão 23. Disponível em: <[https://www.qconcursos.com/arquivos/prova/arquivo\\_prova/46422/cespe-2015-tre-rs-tecnico-judiciario-operacao-de-computadores-prova.pdf](https://www.qconcursos.com/arquivos/prova/arquivo_prova/46422/cespe-2015-tre-rs-tecnico-judiciario-operacao-de-computadores-prova.pdf)>. Acesso em: 4 out. 2018.

Unidade VIII – Questão 2: FUNDAÇÃO DE APOIO AO DESENVOLVIMENTO UNICENTRO (FAU). Tecnólogo em Tecnologia da Informação 2016. Questão 68. Disponível em: <[https://www.qconcursos.com/arquivos/prova/arquivo\\_prova/47381/fau-2016-jucepar-pr-tecnologo-em-tecnologia-da-informacao-prova.pdf](https://www.qconcursos.com/arquivos/prova/arquivo_prova/47381/fau-2016-jucepar-pr-tecnologo-em-tecnologia-da-informacao-prova.pdf)>. Acesso em: 4 out. 2018.

---

---

---

---

---

---

---



Handwriting practice lines consisting of 30 horizontal blue lines. Each line is preceded by a small blue dot on the left margin, serving as a guide for letter height and placement.





A series of horizontal lines for writing, consisting of 30 evenly spaced lines across the page.





# Interativa

Informações:  
[www.sepi.unip.br](http://www.sepi.unip.br) ou 0800 010 9000