

# Unidade III

## 7 GERÊNCIA DE DISPOSITIVOS DE ENTRADA E SAÍDA (E/S)

Além de controlar os processos, a memória principal e o sistema de arquivos, o sistema operacional controla todos os dispositivos de E/S, também conhecidos como periféricos. Eles possibilitam a interação de computador com o mundo externo.

Por exemplo, a interação com o usuário pode ocorrer através de teclado, mouse, tela de toque, joystick. Outra comunicação é através da escrita e leitura de dados em dispositivos de armazenamento como discos rígidos, DVD-ROMs, pen drives. Também há interação através da captura e reprodução de áudios e vídeos, microfone e alto-falantes. Por meio de impressoras e plotadoras ocorre a impressão de informações. Adicionalmente, existe a troca de informações com outros computadores através de diversos tipos de rede, como redes locais (LAN), wi-fi, bluetooth, radiofrequência, telefonia celular e internet das coisas (IoT).

Em função da diversidade dos dispositivos de E/S, o sistema operacional deve implementar uma camada, denominada de subsistema de E/S, com o objetivo de isolar a complexidade dos dispositivos físicos. Com isso, o sistema operacional pode se tornar flexível, de modo a permitir a comunicação das diversas aplicações com qualquer equipamento de E/S. Características técnicas como velocidade de operação, unidade de transferência, forma de representar os dados, operações que podem ser realizadas são abordadas pela camada de device driver, visando oferecer uma interface uniforme e padronizada entre o subsistema de E/S e todos os periféricos.

A gerência de dispositivos de E/S é uma das mais complexas funções do sistema operacional, sendo responsável por gerar a abstração, uma das funções essenciais do sistema operacional. Por meio da abstração de recursos, a interação do programador com a máquina se torna facilitada e permite que os programas e os hardwares evoluam de forma independente. A implementação de gerência de dispositivos é estruturada através de camadas, sendo que uma delas visualiza os diversos tipos de dispositivos do sistema de um modo único. A outra, denominada subsistema de E/S, é específica para cada dispositivo.

O sistema operacional envia comandos para periféricos para emitir instruções (read, write, entre outras), interceptar interrupções e tratar erros de leitura de dados.



### Observação

Atualmente existe uma variedade de dispositivos de E/S que podem estar conectados a um computador. Essa diversidade de periféricos é um dos maiores desafios presentes na construção e manutenção de um sistema operacional, porque cada um possui especificidades e exige distintos mecanismos de acesso.

## 7.1 Subsistema de entrada/saída (E/S)

Um dos objetivos do sistema operacional é tornar as operações de E/S mais simples para o usuário e suas aplicações. Para isso, foi criado o subsistema de E/S, que é responsável por isolar a complexidade de operações específicas para cada tipo de dispositivo da camada de sistema de arquivo, do sistema gerenciador de banco de dados (SGBD) ou diretamente da aplicação. Com isso, torna-se possível para as aplicações a manipulação de qualquer tipo de dispositivo com mais simplicidade.

O subsistema de E/S é formado por um conjunto de rotinas que possibilita a comunicação com qualquer dispositivo que possa ser interligado ao computador. As rotinas de entrada/saída representam chamada de sistema ou system calls e permitem ao usuário realizar operações de E/S sem se preocupar com especificidades do dispositivo com o qual está trabalhando. Nesse caso, quando um usuário salva um arquivo em disco, ele não está interessado em conhecer detalhes como a formatação do disco ou em que trilha/setor do disco o arquivo será armazenado. A figura a seguir apresenta a estrutura de camada da gerência de dispositivos.

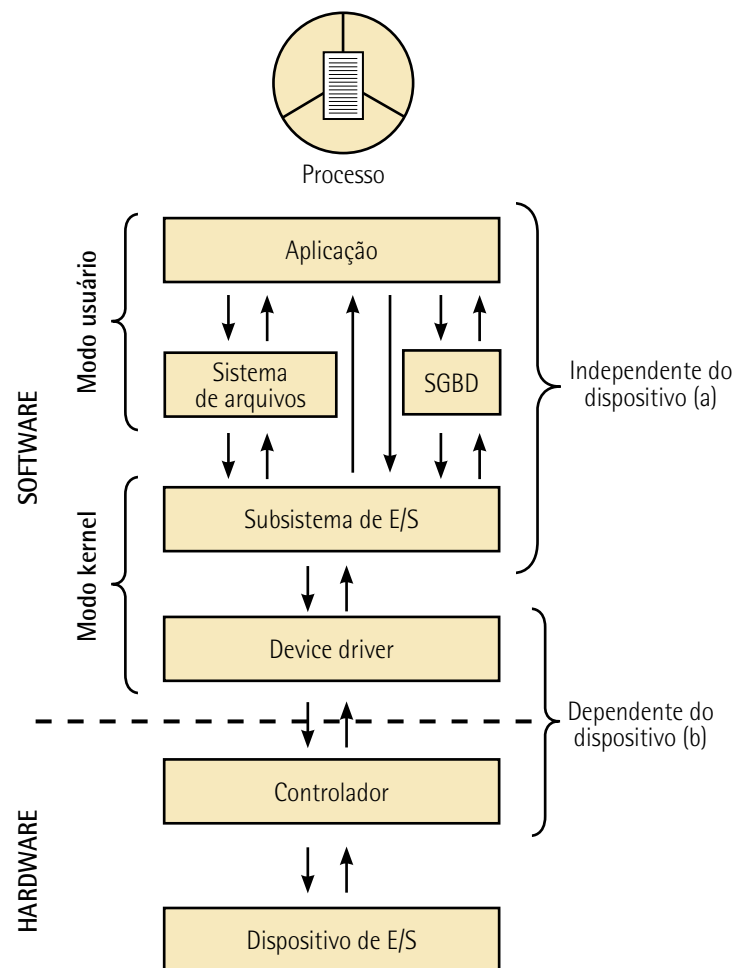


Figura 60 – Arquitetura de camada da gerência de dispositivos

Fonte: Machado e Maia (2013, p. 209).

### 7.2 Device drivers

O device driver, ou somente driver, é um programa ou rotina utilizado para criar uma interface e gerenciar um dispositivo de entrada/saída ou outros periféricos, e essa comunicação do subsistema de E/S com os dispositivos é realizada por intermédio dos controladores de E/S. Ele trata de funções ligadas a todos os dispositivos, enquanto os drivers lidam somente dos seus aspectos específicos.

Os drivers atuam recebendo comandos gerais sobre acessos aos dispositivos e traduzindo-os para comandos específicos para um dispositivo, que serão executados pelos controladores de dispositivo. Cada device driver manipula apenas um tipo de dispositivo ou grupo de dispositivos semelhantes.

Eles são parte integrante do núcleo do sistema operacional ou kernel, e escritos geralmente em Assembly, uma linguagem de máquina e de baixo nível. Como existe dependência entre os drivers e o núcleo do sistema, os fabricantes de dispositivos desenvolvem device drivers diferentes para cada sistema operacional em um mesmo dispositivo.

### 7.3 Controlador de entrada/saída (E/S)

Os controladores de E/S são elementos de hardware que manipulam diretamente os dispositivos de E/S. A comunicação entre o sistema operacional, mais exatamente o device driver, com os dispositivos ocorre através dos controladores. A figura a seguir apresenta um computador com UCP, memória e dispositivos de E/S, e a interligação entre o processador, a memória principal e os controladores de E/S se dá através de barramentos.

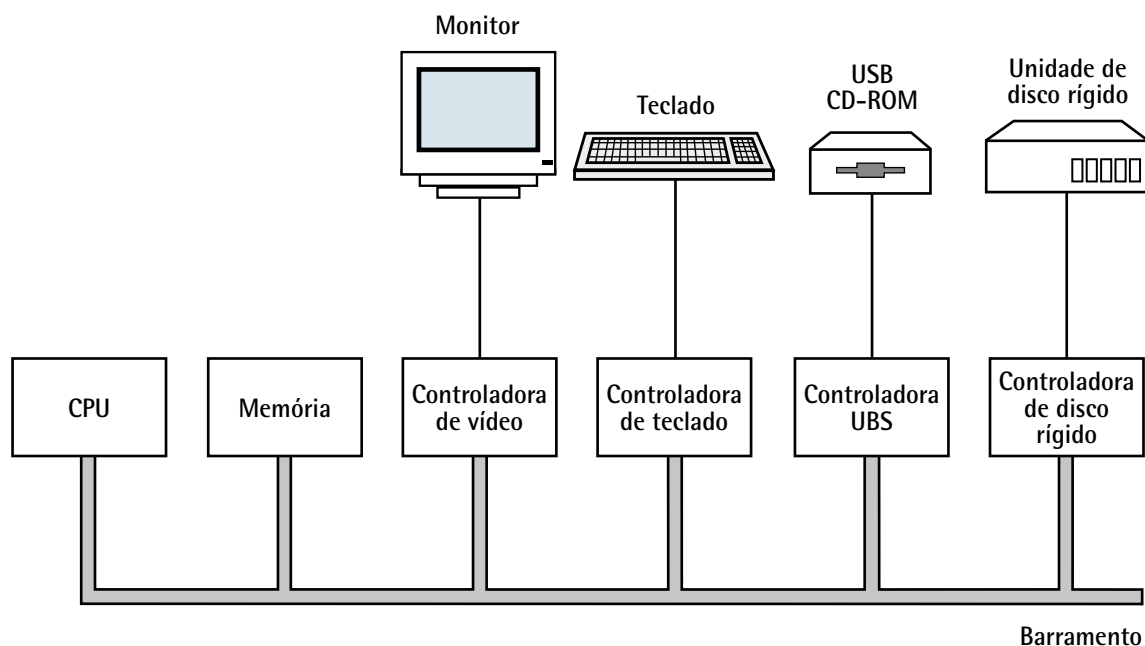


Figura 61 – Diferentes controladores de dispositivos

Fonte: Tanenbaum e Woodhull (2008, p. 215).

O controlador possui memória e registradores exclusivos utilizados na execução de instruções enviadas pelo device driver. Essas instruções de baixo nível são responsáveis pela comunicação entre o controlador e o dispositivo de E/S. Em operações de leitura, o controlador deve armazenar em seu buffer interno uma sequência de bits proveniente do dispositivo até formar um bloco. Após verificar a ocorrência de erros, o bloco pode ser transferido para um buffer de E/S na memória principal.

Os controladores são utilizados na execução de instruções enviadas pelo driver. Em operações de leitura, armazena em seu buffer interno uma sequência de bits provenientes até formar um bloco. Após verificar se no bloco existem erros, é transferido para um buffer de E/S na memória principal.

A transferência do bloco para o buffer pode ser realizada por um controlador de acesso direto na memória ou *direct memory access* (DMA).



### Lembrete

A utilização de barramentos com controladores específicos para os dispositivos de entrada e saída é bastante frequente e visa prioritariamente otimizar a utilização do processador (UCP).

## 7.4 Característica de dispositivos de entrada/saída (E/S)

Para se adequar às muitas possibilidades de interação do computador com o mundo externo, os dispositivos de entrada/saída possuem características diversas de velocidade de transferência, forma de transferência dos dados e método de acesso. A velocidade de transferência de dados de um dispositivo pode variar de alguns bytes por segundo, no caso de dispositivos simples como teclados e mouses, a gigabytes por segundo, para algumas placas de interface gráfica ou de acesso a discos de alto desempenho. A tabela a seguir apresenta alguns exemplos de dispositivos de E/S com suas velocidades típicas de transferência de dados.

**Tabela 2 – Velocidades típicas de dispositivos de entrada e saída**

Dispositivo	Velocidade
Teclado	10 B/s
Mouse ótico	100 B/s
Interface paralela padrão	125 KB/s
Interface de rede fast ethernet	11,6 MB/s
Pen drive ou disco USB 2.0	60 MB/s
Interface de rede gigabit ethernet	116 MB/s
Disco rígido Sata 2	300 MB/s
Interface gráfica high-end	4,2 GB/s

Adaptada de: Maziero (2019, p. 238).

Os dispositivos de entrada/saída podem ser classificados de diferentes formas. São categorizados dependendo de seu fluxo de dados para o computador:

- **Entrada de dados:** CD-ROM, teclado etc.
- **Saída de dados:** impressoras, projetores.
- **Entrada/saída de dados:** modems, discos rígidos, telas touch screen.

Adicionalmente, os periféricos podem ser separados em função da forma com que os dados são armazenados. Nesse caso, os dispositivos podem ser classificados como dispositivos estruturados ou de bloco e dispositivos não estruturados ou de caractere. Os dispositivos de bloco armazenam os dados em bloco que possuem tamanho fixo e endereço próprio. Nesses dispositivos, cada bloco pode ter escrita ou leitura independentemente dos outros. O disco rígido, o pen drive, o DVD-ROM e os demais dispositivos de armazenamento são exemplos dessa categoria.

Por outro lado, nos dispositivos não estruturados ou de caractere há envio e recebimento de caracteres. Entretanto, neles não há endereçamento, bem como não possuem funcionalidades de posicionamento. A impressora, o mouse e a placa de rede são exemplos de dispositivos de caractere.

Outra classificação possível está relacionada com o acesso a posições de memória no dispositivo de E/S. Um dispositivo é de acesso direto quando um bloco pode ser recuperado através de seu endereço. Por outro lado, é de acesso sequencial quando para acessar um bloco o dispositivo percorre sequencialmente os demais blocos.

### 7.5 Princípios de hardware

As unidades de E/S possuem componentes mecânicos e a controladora do dispositivo. O componente mecânico é o próprio dispositivo, que fica aparente para o usuário final. Já a controladora do dispositivo é um componente eletrônico do dispositivo e pode ser programável. No caso dos PCs, a controladora normalmente é um circuito integrado.

Normalmente os fabricantes de controladores de dispositivos desenvolvem as interfaces seguindo um padrão oficial, que é fornecido por organizações reconhecidas internacionalmente como ANSI, IEEE, ISO e outras. Os fabricantes utilizarão os padrões definidos como os tipos de conectores, as tensões elétricas e as propriedades mecânicas. Tal fato contribui para gerar interoperabilidade entre os dispositivos.



### Saiba mais

Para mais informações em relação aos órgãos de padronização, consulte:

ANSI: American National Standards Institute (Instituto Nacional Americano de Padronização)

Disponível em: <https://www.ansi.org/>. Acesso em: 14 mar. 2023.

IEEE: Institute of Electrical and Electronics Engineers (Instituto de Engenheiros Eletricistas e Eletrônicos)

Disponível em: <https://www.ieee.org/>. Acesso em: 14 mar. 2023.

ISO: International Organization for Standardization (Organização Internacional de Normalização)

Disponível em: <https://www.iso.org/home.html>. Acesso em: 14 mar. 2023.

As system calls responsáveis por essa comunicação são chamadas de system calls de entrada/saída, e um de seus objetivos é simplificar a interface entre as aplicações e os dispositivos.

As operações de E/S podem ser classificadas em síncrona ou assíncrona. A operação síncrona ocorre quando o processo que realizou a operação fica aguardando em estado de espera por seu término. Já a operação assíncrona se dá quando o processo que realizou a operação não aguarda pelo seu término e continua pronto para ser executado. Nesse caso, deve existir uma sinalização que indique o término da operação.

O driver comunica-se com os dispositivos através dos controladores. Ele pode ser uma placa independente conectada a um slot do computador ou implementada na mesma placa do processador. O controlador tem memória e registradores próprios.

## 7.6 Interface de acesso

Analisando do ponto de vista do sistema operacional, a característica mais relevante dos dispositivos de entrada/saída é a interface de acesso, isto é, a abordagem a ser utilizada para o acesso ao periférico, para a configuração e o envio e recebimento dos seus dados. Geralmente cada dispositivo disponibiliza um conjunto de registradores denominados **portas de entrada/saída**, que podem ser acessados pelo barramento e são utilizados para comunicação com o processador.

Essas portas podem ser divididas em quatro grupos: portas de entrada, portas de saída, portas de status e portas de controle. Nas portas de entrada ou data-in ports, são enviados dados para o processador, enquanto nas portas de saída (data-out ports) são recebidas informações do processador. Através das portas de status, o processador consulta o estado interno do dispositivo, verificando se as operações foram executadas sem erro. Por fim, as portas de controle são utilizadas pelos processador para o envio de comandos ao dispositivo e para modificar parâmetros de configuração.

A figura a seguir representa a interface de acesso do dispositivo de E/S, e a quantidade de portas e as funções delas dependem do dispositivo selecionado.

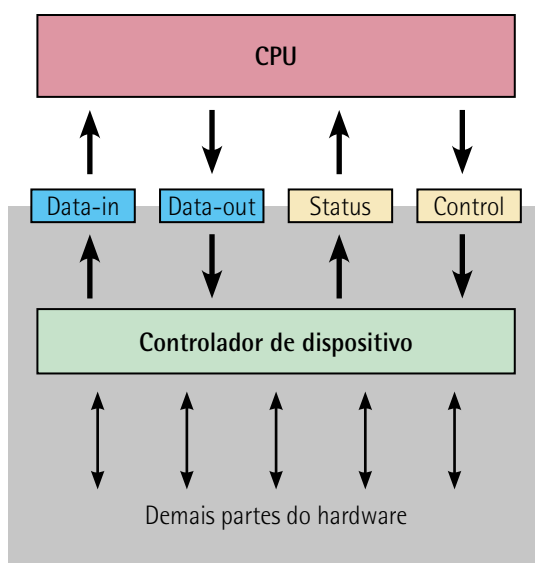


Figura 62 – Interfaces de acesso de dispositivos de E/S

Fonte: Maziero (2019, p. 239).

## 8 SISTEMA OPERACIONAL LINUX

Neste capítulo, será apresentado o sistema operacional Linux, analisando seu histórico e motivação para o desenvolvimento, o tipo de licença utilizada, as distribuições, as questões de segurança e os passos necessários para sua instalação. Linux é um sistema operacional de código aberto, sendo uma variante do sistema operacional Unix. Ele é utilizado em dispositivos de diferentes portes e por muitos desenvolvedores para gerenciar seus ambientes e desenvolver novos códigos para o mercado.

### 8.1 Origens do GNU/Linux

A primeira versão do sistema operacional Unix foi desenvolvida em 1969 no Grupo de Pesquisas da Bell Laboratories, com a liderança de Ken Thompson. Dennis Ritchie participou da etapa de melhoria do sistema, que foi escrito na linguagem de programação C. Além da utilização no Bell Laboratories, o Unix começou a ser adotado em diversas universidades ao redor do mundo.

Segundo Ferreira (2003), Richard Stallman e Linus Torvalds são responsáveis pela criação do sistema em meados de 1984. No ano de 1983, Richard Stallman iniciou o projeto que ficou conhecido como GNU, cujo objetivo era o desenvolvimento de um sistema operacional baseado no Unix, mas sem cobrança de licenças de uso e com permissão de copiar, estudar, modificar ou até distribuir seu código. O nome do projeto é acrônimo para "GNU's not Unix!", em português "GNU não é Unix!".

Desde a sua concepção, o Linux foi desenvolvido para ser um sistema operacional multitarefa e multiusuário. Algumas ferramentas foram sendo desenvolvidas, por exemplo, editor de texto e compiladores de linguagem C. A partir de 1992, aproximou-se de sua fase quase adulta, entretanto, faltava o núcleo ou kernel do sistema operacional.

O time liderado por Stallman estava envolvido em um projeto de desenvolvimento de um código chamado Hurd, que nada mais era do que justamente um núcleo. Esse projeto já era mantido por empresas que incentivavam e apoiavam a ideia de software livre e foi aproveitado no desenvolvimento do novo sistema operacional. Richard Stallman fundou em 1985 a Fundação para Software Livre, em inglês Free Software Foundation (FSF). Essa organização sem fins lucrativos se dedica à eliminação de restrições sobre a cópia, o estudo e a modificação de programas de computador, que são bandeiras do movimento do software livre.

A fundação foi importante para ajudar na concepção do novo software. Ainda segundo Ferreira (2003), a FSF também possui a responsabilidade de manter a General Public License (GPL). Essa licença defende juridicamente um software livre e mantém as quatro liberdades de uso do sistema operacional:

- **Liberdade 0:** executar o programa para qualquer finalidade.
- **Liberdade 1:** estudar o funcionamento do programa, podendo fazer alterações e adaptá-lo conforme a necessidade. É preciso ter acesso ao código-fonte.
- **Liberdade 2:** redistribuir cópias do software, de modo que ajude o próximo.
- **Liberdade 3:** modificar o programa e disponibilizar as modificações.

### 8.1.1 Histórico do Linux

Em 1991, Linus Torvalds desenvolveu um novo kernel chamado Linux. Este se uniu às ferramentas desenvolvidas pelo projeto GNU, surgindo o que é conhecido atualmente como GNU/Linux. Seu desenvolvimento teve como base o sistema operacional Minix, criado em 1987 por Andrew Tanenbaum. O Minix baseava-se nos padrões do Unix e, na época, era usado apenas na área acadêmica para estudos e desenvolvimentos específicos.

O Linux é muito parecido com qualquer outro sistema Unix; na verdade, a compatibilidade com o Unix era um objetivo importante do seu projeto. No entanto, ele é muito mais jovem do que a maioria dos sistemas Unix. Seu desenvolvimento começou em 1991, quando um estudante universitário finlandês, Linus Torvalds, passou a desenvolver um kernel pequeno, mas autossuficiente, para o processador 80386, o primeiro processador verdadeiro de 32 bits no conjunto de CPUs da Intel compatíveis com PCs.



O Linux foi batizado assim pois trata-se da junção do nome de seu desenvolvedor, Linus Torvalds, com Unix, o sistema operacional tomado como base para o desenvolvimento do kernel do Linux. Entretanto, o Unix é desenvolvido para servidores de grande porte.

No início de seu desenvolvimento, o código-fonte do Linux foi disponibilizado livremente pela internet, sem custos e com restrições mínimas de distribuição. Como resultado, a sua história tem sido a da colaboração de muitos desenvolvedores de todo o mundo, correspondendo-se quase exclusivamente pela internet. A partir de um kernel inicial que implementava parcialmente um pequeno subconjunto de serviços do sistema Unix, o Linux cresceu para incluir toda a funcionalidade esperada de um sistema Unix moderno.

Em seus primórdios, o desenvolvimento do Linux evoluiu em grande parte em torno do kernel central do sistema operacional – o núcleo, executivo privilegiado que gerencia todos os recursos do sistema e que interage diretamente com o hardware do computador. Precisamos, naturalmente, de muito mais do que esse kernel para produzir um sistema operacional completo.

Assim, temos que diferenciar o kernel do Linux de um sistema Linux completo. O kernel do Linux é um componente de software original desenvolvido a partir do zero pela comunidade Linux. O sistema Linux, conforme o conhecemos hoje, inclui vários componentes, alguns escritos do zero, outros emprestados de outros projetos de desenvolvimento, e ainda outros criados em colaboração com outras equipes.

O sistema Linux básico é um ambiente-padrão para aplicações e programação de usuários, mas não impõe nenhum meio-padrão de gerenciamento da funcionalidade disponível como um todo. À medida que o Linux amadurecia, surgiu a necessidade de outra camada de funcionalidade no topo do sistema. Essa necessidade tem sido atendida por várias distribuições do Linux. Uma distribuição inclui todos os componentes-padrão do sistema Linux, mais um conjunto de ferramentas administrativas para simplificar a instalação inicial e a atualização subsequente do Linux para gerenciar a instalação e a remoção de outros pacotes do sistema. Uma distribuição moderna também contém, tipicamente, ferramentas para o gerenciamento de sistemas de arquivos, criação e gerenciamento de contas de usuário, administração de redes, navegadores da web, editores de texto, e assim por diante.



Figura 63 – Tux, mascote oficial do Linux

## 8.2 Distribuições do Linux

Baseado nas quatro definições de liberdade estabelecidas para o uso do GNU/Linux, os usuários iniciaram um processo de personalização do sistema, programando-o de acordo com as necessidades individuais e dando início às distribuições.

Segundo Ferreira (2003), uma distribuição é um conjunto de vários softwares agrupados em mídias. Com esses instaladores customizados, é possível facilitar o trabalho do usuário e dos administradores. As distribuições Linux começaram a ficar mais populares a partir do final dos anos 1990, quando se tornaram uma alternativa livre aos sistemas operacionais que existiam na época. Esses inicialmente eram muito populares na linha de servidores; posteriormente, começaram a participar ainda do mercado de desktops.

Também conhecidas como distros, cada distribuição possui suas características muitas vezes exclusivas, por exemplo, a maneira de realizar algum tipo de configuração no sistema, instalação de programas, automatização de tarefas, recuperação de sistemas danificados e monitoramento de redes de computadores. Em geral, apesar de suas diferenças, a essência de todas elas é similar.

Assim, aprendendo a usar o GNU/Linux em determinada distribuição, não haverá dificuldades para trabalhar com outras, porque em geral todas possuem a mesma sintaxe de comandos e funcionalidades.

## 8.3 Licença GPL

A licença pública geral (GPL) garante liberdades como a execução do programa a qualquer propósito, o estudo do seu funcionamento e adaptação às suas necessidades, a redistribuição de cópias de modo a auxiliar os próximos usuários e a possibilidade de aperfeiçoar o programa e poder liberar suas modificações, de modo a permitir que toda a comunidade se beneficie com as alterações.



### Lembrete

GPL é um dos tipos de licenças públicas mais comuns para programa de código aberto e requer que evoluções baseadas no programa original sejam autorizadas sob a mesma licença.

As distribuições livres do Linux são mantidas por comunidades de colaboradores que não objetivam o lucro econômico com o desenvolvimento do software.

### 8.4 Buscando documentação no sistema

Uma parte importante de qualquer sistema operacional são a documentação, os manuais técnicos que descrevem o uso e o funcionamento dos programas. O sistema GNU/Linux possui uma ampla documentação técnica oficial. Os próprios comandos já possuem suas respectivas documentações, e o mais comum é se utilizar do programa MAN (manual), seguido pelo comando desejado que se quer aprimorar. Serão apresentados os seguintes assuntos:

- tipos de documentação;
- comandos de documentação;
- diretórios de documentação;
- sites oficiais.

#### 8.4.1 Tipos de documentação

Existem diversas maneiras para consulta da documentação do sistema GNU/Linux. Além da documentação relacionada aos comandos e seus parâmetros, pode-se consultar how tos e páginas de manuais, que ajudarão o profissional a realizar determinado procedimento, por exemplo, a instalação de um servidor de compartilhamento de arquivos, um servidor para hospedagem de sites, um servidor de banco de dados etc.

- **How to:** refere-se a um procedimento detalhado para a realização de determinada atividade. Esse tipo de documentação pode ser encontrado nos diversos sites oficiais ou não oficiais, como blogs.
- **Páginas de manuais:** esse tipo de documentação já se encontra instalado no próprio sistema. A cada novo comando instalado, sua respectiva documentação também será. As man pages são organizadas por sessões que começam do valor 1 até 9.
- **GuiaFoca:** o GuiaFoca GNU/Linux apresenta de forma didática explicações sobre computadores e o sistema GNU/Linux para usuários iniciantes, além de abordar formas de administração e segurança do sistema para usuários avançados.

### 8.5 Modelo de segurança no Linux

De acordo com Silberschatz, Galvin e Gagne (2015), o modelo de segurança do Linux está intimamente relacionado com os mecanismos de segurança típicos do Unix. As preocupações de segurança podem ser classificadas em dois grupos: autenticação e controle de acesso.

A **autenticação** representa a garantia de que ninguém possa acessar o sistema sem primeiro provar que tem direitos de entrada.

O **controle de acesso** se refere ao fornecimento de um mecanismo que verifique se um usuário tem o direito para acessar determinado objeto e impedir o acesso a objetos quando necessário.

### 8.5.1 Autenticação

A autenticação no Unix tem sido tipicamente executada por meio do uso de um arquivo de senhas publicamente legível. Uma senha de usuário é combinada com um valor salt aleatório, e o resultado é codificado com uma função de transformação unidirecional e armazenado no arquivo de senhas. O uso da função unidirecional significa que a senha original não pode ser deduzida do arquivo de senhas, exceto por tentativa e erro. Quando um usuário apresenta uma senha ao sistema, ela é recombinaada com o valor salt armazenado no arquivo de senhas e passa pela mesma transformação unidirecional. Se o resultado coincidir com o conteúdo do arquivo de senhas, a senha será aceita.

Historicamente, as implementações desse mecanismo no Unix têm apresentado vários problemas. As senhas eram, com frequência, limitadas a oito caracteres, e o número de valores salt possíveis era tão baixo que um invasor poderia facilmente combinar um dicionário de senhas comumente utilizadas com cada valor salt e ter uma boa chance de obter uma ou mais senhas no arquivo, ganhando acesso não autorizado a quaisquer contas, que ficariam então comprometidas. Extensões ao mecanismo de senhas têm sido introduzidas de modo a manter a senha criptografada secreta em um arquivo que não seja legível publicamente, permita senhas mais longas ou use métodos mais seguros de codificação.

Outros mecanismos de autenticação têm sido introduzidos para limitar os períodos durante os quais um usuário é autorizado a se conectar ao sistema. Além disso, existem mecanismos para distribuir informações de autenticação a todos os sistemas relacionados em uma rede.

Um novo mecanismo de segurança foi desenvolvido por fornecedores do Unix para resolver problemas de autenticação. O sistema de módulos de autenticação conectáveis (*pluggable authentication modules* – PAM) é baseado em uma biblioteca compartilhada que pode ser usada por qualquer componente do sistema que precise autenticar usuários. Uma implementação está disponível no Linux. O PAM permite que módulos de autenticação sejam carregados sob demanda, como especificado em um arquivo de configurações com abrangência em todo o sistema. Se um novo mecanismo de autenticação for adicionado em uma data posterior, ele poderá ser adicionado ao arquivo de configurações, e todos os componentes do sistema poderão usá-lo imediatamente. Os módulos do PAM podem especificar métodos de autenticação, restrições de contas, funções de configuração de sessões e funções de mudança de senhas (de modo que, quando os usuários modificarem suas senhas, todos os mecanismos de autenticação necessários possam ser atualizados imediatamente).

### 8.5.2 Controle de acesso

O controle de acesso em sistemas Unix, inclusive no Linux, é executado por meio do uso de identificadores numéricos exclusivos. Um identificador de usuário (UID) identifica um único usuário ou um único conjunto de direitos de acesso. Um identificador de grupo (GID) é um identificador adicional que pode ser usado para reconhecer direitos pertencentes a mais de um usuário.

O controle de acesso é aplicado a vários objetos no sistema. Cada arquivo disponível é protegido pelo mecanismo de controle de acesso padrão. Além disso, outros objetos compartilhados, tais como seções de memória compartilhada e semáforos, empregam o mesmo sistema de acesso.

Cada objeto em um sistema Unix, sob controle de acesso de usuários e grupos, tem um UID exclusivo e um GID exclusivo associados a ele. Processos de usuário também têm um UID exclusivo, mas podem possuir mais de um deles. Se o UID de um processo coincidir com o UID de um objeto, então o processo tem direitos de usuário ou direitos de proprietário sobre ele. Se os UIDs não coincidirem, mas algum GID do processo coincidir com o GID do objeto, então os direitos de grupo serão conferidos; caso contrário, o processo terá direitos universais sobre o objeto.

O Linux executa o controle de acesso atribuindo aos objetos uma máscara de proteção que especifica as modalidades de acesso, tais como leitura, gravação ou execução, que devem ser concedidas a processos com acesso de proprietário, de grupo ou universal. Assim, o proprietário de um objeto pode ter acesso total de leitura, gravação e execução para um arquivo; outros usuários de determinado grupo podem receber acesso de leitura, mas ter negado o acesso de gravação; e o restante das pessoas pode não receber nenhum tipo de acesso.

A única exceção é o UID raiz privilegiado. Um processo com esse UID especial recebe acesso automático a qualquer objeto no sistema, ignorando verificações de acesso normais. Tais processos também recebem permissão para executar operações privilegiadas, por exemplo, a leitura de qualquer memória física ou a abertura de sockets de rede reservados. Esse mecanismo permite que o kernel impeça usuários comuns de acessar esses recursos: a maioria dos recursos internos importantes do kernel é implicitamente de propriedade do UID raiz. O Linux implementa o mecanismo `setuid` padrão do Unix.

Tal mecanismo permite que um programa seja executado com privilégios diferentes daqueles referentes ao usuário que está o executando. Por exemplo, o programa `lpr` (que submete um job a uma fila de impressão) tem acesso às filas de impressão do sistema, mesmo se o usuário que o estiver executando não tenha. A implementação do `setuid` no Unix diferencia os UIDs real e efetivo de um processo. O UID real é o do usuário que está executando o programa; o UID efetivo é o do proprietário do arquivo. No Linux, esse mecanismo é ampliado de duas maneiras.

Em primeiro lugar, o Linux implementa o mecanismo `saved user-id` da especificação Posix, que permite que um processo abandone e readquira seu UID efetivo repetidamente. Por motivos de segurança, um programa pode querer executar a maioria de suas operações de modo seguro, abandonando os privilégios concedidos por seu status `setuid`; mas ele consegue executar operações selecionadas com todos os seus privilégios. As implementações-padrão do Unix disponibilizam esse recurso somente por permuta entre os UIDs real e efetivo. Quando isso é feito, o UID efetivo anterior é lembrado, mas o UID real do programa nem sempre corresponde ao UID do usuário que está o executando. UIDs salvos possibilitam que um processo defina seu UID efetivo como seu UID real e, então, retorne ao valor anterior de seu UID efetivo sem ter que modificar o UID real em momento algum.



## Observação

A interface Posix, sigla de Portable Operating System Interface, representa uma série de padrões implementados principalmente para sistemas operacionais baseados no Unix. O Linux, o Solaris e o Mac OS X são exemplos compatíveis com o Posix.

A segunda melhoria fornecida pelo Linux é a incorporação de uma característica dos processos que concede apenas um subconjunto dos direitos do UID efetivo. As propriedades de processos fsuid e fsgid são usadas no momento da concessão de direitos de acesso a arquivos. A propriedade adequada é estabelecida sempre que o UID ou GID efetivo é definido.

No entanto, o fsuid e o fsgid podem ser definidos independentemente dos ids efetivos, permitindo que um processo acesse arquivos em nome de outro usuário sem, de forma alguma, assumir a sua identidade. Especificamente, processos de servidor conseguem usar esse mecanismo para servir arquivos para determinado usuário sem ficar vulneráveis ao encerramento ou suspensão por esse usuário.

Para concluir, o Linux fornece um mecanismo para a transmissão flexível de direitos de um programa para outro — algo que se tornou comum em versões modernas do Unix. Quando um socket de rede local é estabelecido entre dois processos no sistema, um deles pode enviar ao outro processo o descritor de arquivo de um de seus arquivos abertos; o outro processo recebe um descritor de arquivo de duplicata do mesmo arquivo.

Esse mecanismo permite que um cliente passe, seletivamente, o acesso a um único arquivo para algum processo de servidor sem conceder a ele nenhum outro privilégio. Por exemplo, não é mais necessário que um servidor de impressão tenha que ler todos os arquivos de um usuário que submeta um novo job de impressão. O cliente da impressão pode simplesmente passar ao servidor os descritores de arquivo de quaisquer arquivos a serem impressos, negando ao servidor acesso aos outros arquivos do usuário.

## 8.6 Distribuição CentOS

Antes de efetuar a instalação do sistema operacional, é importante conhecer quais serão os serviços que cada servidor realizará. Com base nesse planejamento, 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.

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 defini-la. 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

O projeto CentOS é um esforço de uma comunidade de software livre focada na entrega de uma plataforma Linux robusta e de código aberto. A distribuição CentOS Linux é consistente, gerenciável e se adequa a uma ampla variedade de implementações.

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

CENTOS. *CentOS Linux*. [s.d.]. Disponível em: <https://bit.ly/3TmVBYi>. Acesso em: 14 mar. 2023.

## 8.7 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 GetCentOS Now, destacado em laranja, conforme exibido na figura seguinte.



Figura 64 – Site do CentOS



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



Figura 65 – Página de download do CentOS

- **Passo 3:** selecione um dos endereços ou mirrors disponíveis para a realização do download da imagem com o nome de CentOS-8.1.1911-x86\_64-dvd1.iso

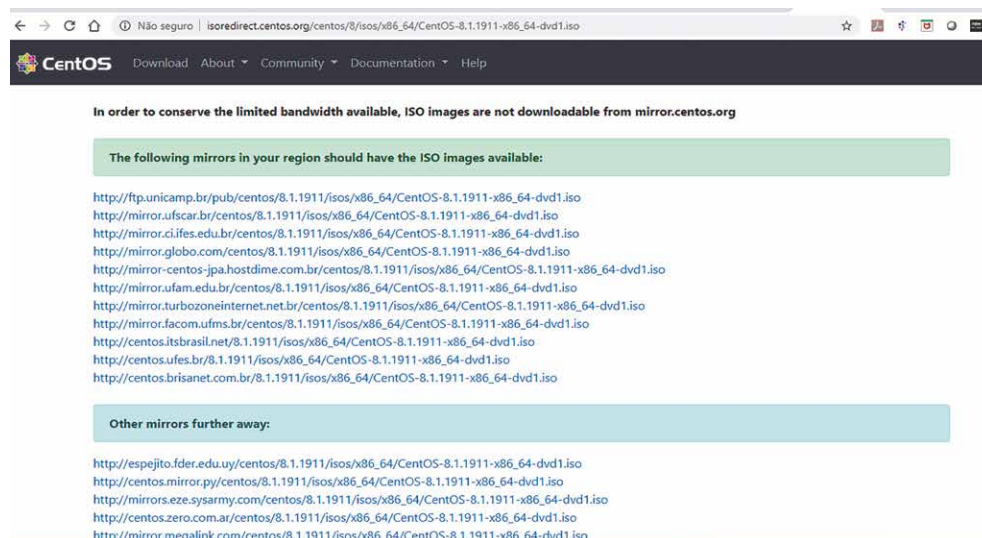


Figura 66 – Seleção entre mirrors disponíveis



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.

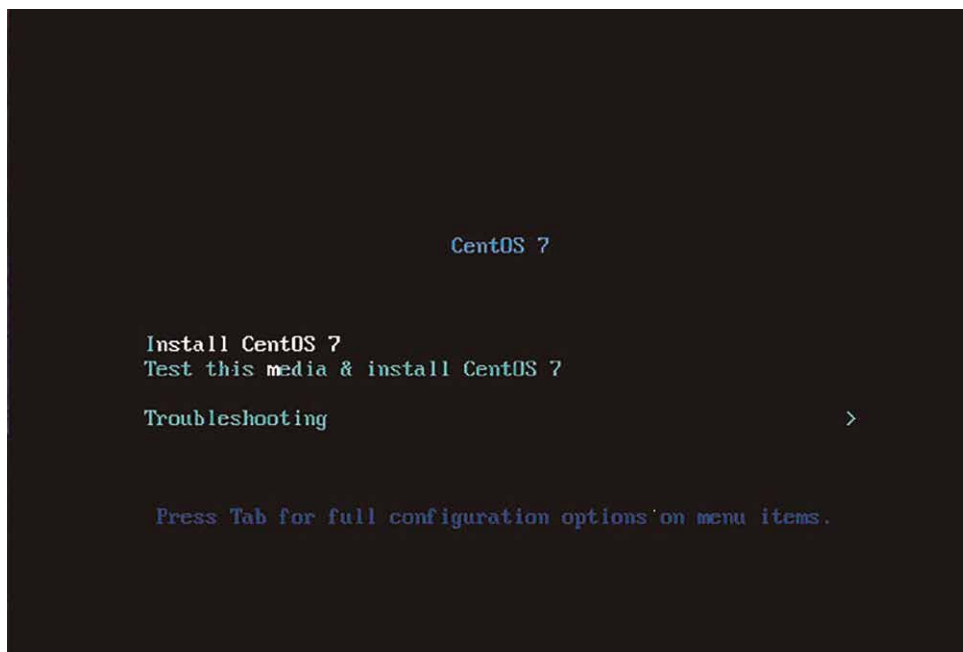


Figura 67 – Menu de seleção da instalação

Nesta etapa, 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.

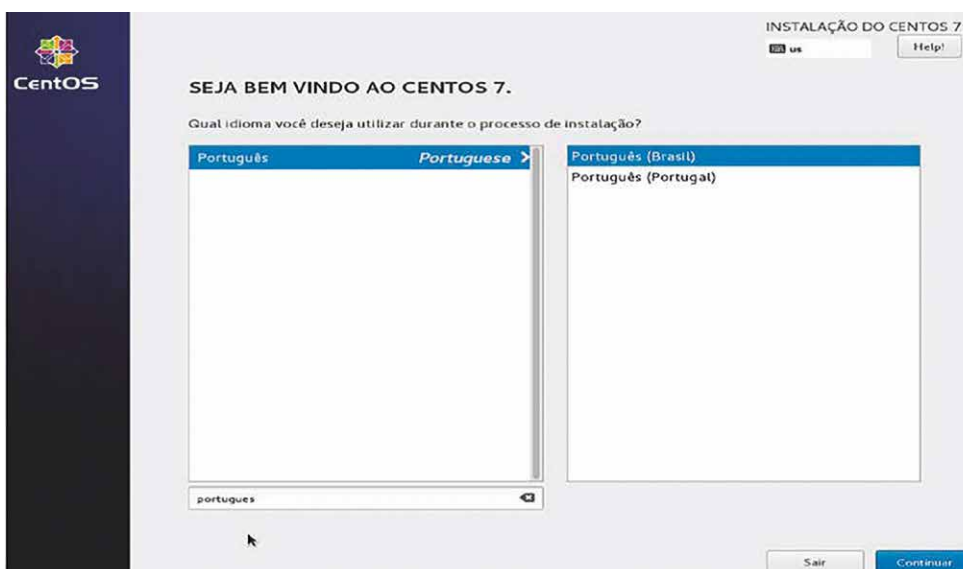


Figura 68 – Escolha do idioma da instalação

Na janela Resumo da Instalação, clique em DESTINOINSTALAÇÃO e, por fim, no botão Finalizado, que está localizado no canto superior esquerdo, conforme apresentado na figura a seguir.



Figura 69 – Tela do resumo da instalação

Após essa etapa, clique na opção REDE & NOME DO HOST para definição do 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. A figura a seguir ilustra este menu.

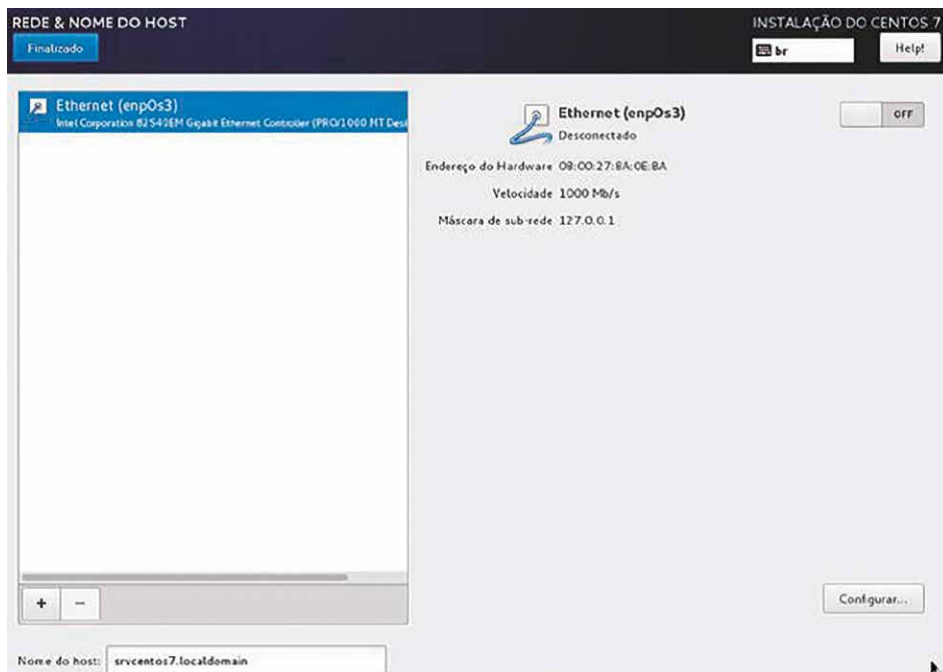


Figura 70 – Janela REDE & NOME DO HOST

A configuração de rede deverá ser realizada de forma manual. Então, clique no botão Iniciar a instalação, no lado direito, na parte inferior. A partir desse instante, começará o processo de instalação, sendo apresentada uma janela para verificação do progresso da instalação.



Figura 71 – Janela do processo de instalação

Finalizada a instalação do sistema operacional, será necessária a definição da senha do usuário root e a criação de um usuário comum. Esse processo é exibido na figura anterior.

Clique na opção Senha Raiz para cadastrar a senha do usuário root. Abaixo do campo Senha Raiz, irá aparecer uma escala de dificuldade para quebra da senha. Caso a senha cadastrada não seja complexa o suficiente para satisfazer os requisitos de segurança, aparece uma tarja laranja com a mensagem: "A senha que você escolheu é fraca. A senha não passa pela verificação do dicionário – é uma palavra da língua portuguesa. Aperte Finalizado duas vezes para confirmá-la", conforme exibido na figura a seguir.

Figura 72 – Definição da senha do usuário root

Após definir a senha do usuário root, será cadastrado um usuário comum, conforme pode ser exibido na figura seguinte. Clique na opção de criação de usuário. Cadastre o nome de usuário, neste exemplo, tux, e informe uma senha para ele e a confirme.

Assim como na definição da senha do root, caso a senha cadastrada seja simples, a mensagem de alerta ocorrerá novamente. Clique duas vezes no botão Finalizado.

Figura 73 – Janela CRIAR USUÁRIO

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



Figura 74 – Janela CONFIGURAÇÃO

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

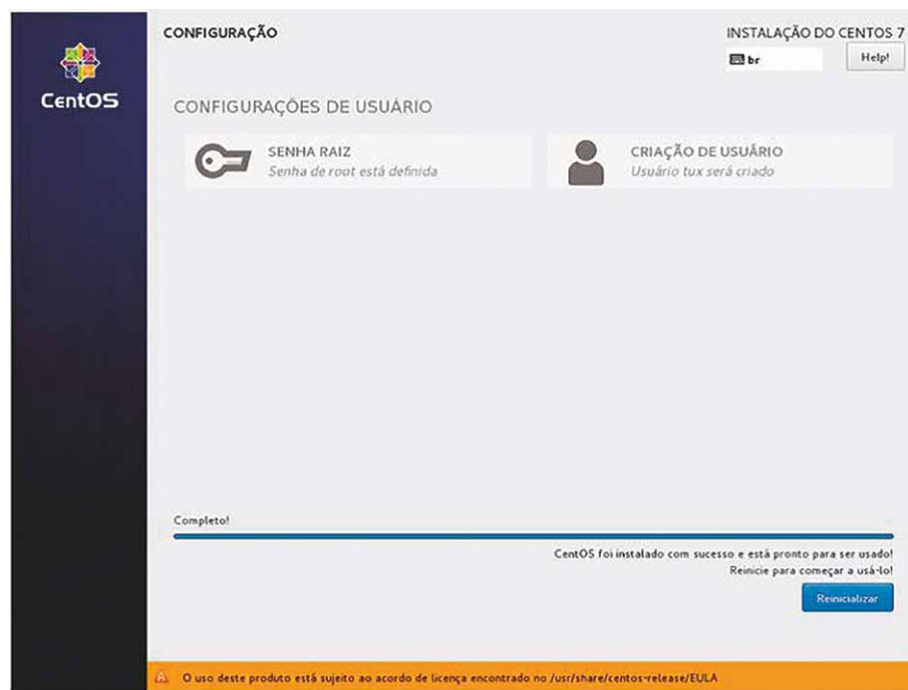


Figura 75 – Janela CONFIGURAÇÃO

Após essas etapas, a instalação estará finalizada e o sistema operacional CentOS estará instalado e disponível para ser utilizado.

### 8.8 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.


Nesta seção, serão abordados os seguintes assuntos:

- funcionamento do editor;
- trabalho em modo de edição;
- trabalho em modo de comando;
- operações;
- arquivo de configuração do Vim.

#### 8.8.1 Funcionamento do editor

O programa Vim provavelmente é o editor de texto mais utilizado pelos profissionais que trabalham com o sistema GNU/Linux, já que é uma versão aperfeiçoada do antigo editor Vi, lançado em 1991. Apesar de ter uma interface simples, sem os recursos dos processadores de texto mais populares, como o Microsoft Word, o Vim é muito poderoso e, por trás de sua aparente simplicidade, esconde funcionalidades indispensáveis no dia a dia.

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 version?<Enter> for version info
```

Figura 76 – Tela de abertura do Vim

Ele funciona 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.

### 8.8.2 Trabalho 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.

### 8.8.3 Trabalho 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
- Com o objetivo de 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.

### 8.8.4 Operações

Em seguida, serão realizadas algumas operações usando a ferramenta Vim. Vamos abrir 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, 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.

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:

```
:40.
```

Nesse caso, o editor irá posicionar o cursor na linha 40 do arquivo. Para ter certeza de que o cursor está realmente na linha 40, basta digitar o comando `:set nu`. O número será exibido na linha atual, e vale lembrar que ele não ficará gravado no arquivo.

### 8.8.5 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`. O quadro a seguir exibe algumas opções que podem ser incluídas no arquivo de configuração: Opção Função.

**Quadro 8 – Opções de configuração do Vim**

Opção	Função
<code>set ve=all</code>	Permite a movimentação do 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</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

### 8.9 Sistemas de arquivos

O sistema de arquivos é responsável pelo gerenciamento das informações 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 terabytes.

### 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.

### 8.9.1 Diretórios do sistema (FHS)

Como há diversas distribuições disponíveis para o Linux, foi necessário realizar uma padronização nos diretórios encontrados. Ela é conhecida como File System Hierarchy Standard (FHS).

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

Essa estrutura está dividida de forma hierárquica, em que cada diretório possui uma finalidade diferente. O diretório-raiz do sistema é representado por uma barra normal (/). Abaixo do diretório-raiz, são apresentados os subdiretórios que organizam os arquivos no sistema, conforme exibido na figura a seguir.

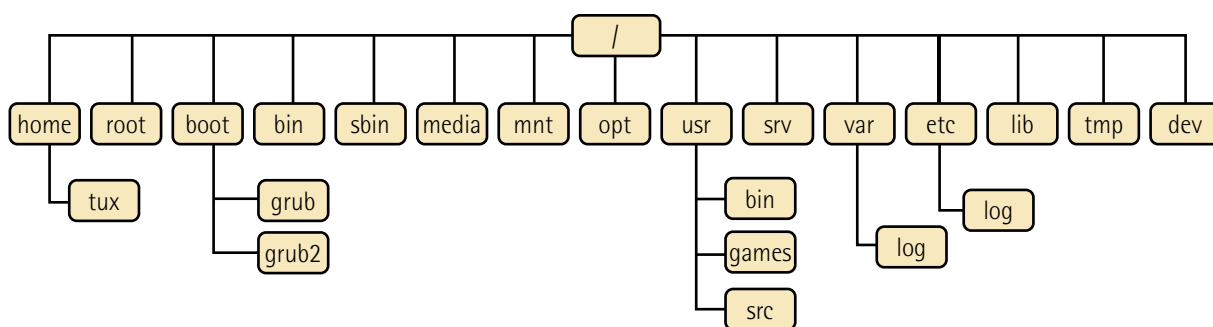


Figura 77 – Estrutura do FHS

Fonte: Alves (2018, p. 185).



## Resumo

Nesta unidade, exibimos a gerência de dispositivos de entrada/saída, que simplifica a utilização de periféricos no ponto de vista de aplicações e do usuário final. Detalhamos o subsistema de entrada e saída e classificamos os seus diversos dispositivos.

Na sequência, introduzimos o sistema operacional GNU/Linux. Contemplamos o projeto GNU, fizemos um breve histórico do Linux e mostramos as diversas distribuições do GNU/Linux.

Abordamos as buscas de documentação no próprio sistema (on-line), discorremos a respeito de alguns aspectos do modelo de segurança do GNU/Linux e analisamos os mecanismos de autenticação e de controle de acesso.

Depois, exibimos a distribuição GNU/Linux CentOS e explicamos o passo a passo de como fazer a sua instalação.

Por fim, investigamos o editor de texto Vim e exploramos alguns aspectos básicos do seu funcionamento, além de apresentarmos conceitos relativos aos sistemas de arquivos e de diretórios, o shell e alguns comandos básicos úteis na administração e no uso de máquinas GNU/Linux.



### Exercícios

**Questão 1.** (Fundatec 2022, adaptada) Leia o texto a seguir, a respeito de periféricos de entrada e de saída.

A classe de dispositivos utilizados atualmente para a realização de operações de entrada e saída é bastante ampla e diversa, apresentando inúmeras características e operações. Entre elas, estão as elencadas a seguir.

- **Impressoras:** criam cópia em papel, ou similar, de textos e figuras. Elas variam em atributos, tendo como forma mais primitiva dispositivos que se assemelham a máquinas de escrever. Atualmente, as mais modernas utilizam como tecnologia a impressão a laser e podem imprimir até milhares de linhas por minuto.
- **Monitores (ou telas):** mostram textos e figuras, ambos podendo ser formatados de diversas maneiras e utilizando várias cores. O tipo mais clássico usava um tubo não muito diferente de um tubo de televisão (tubo de raios catódicos). Atualmente, utilizam-se também visores de cristal líquido, que consomem menos energia.
- **Dispositivos gráficos (ou plotters):** criam cópias impressas de gráficos e curvas em papel. Com o avanço tecnológico, atualmente têm capacidade de produzir material gráfico de ótima qualidade. Os traçadores, dependendo das suas características, podem produzir desenhos com mais de um metro de largura e comprimento limitado pela bobina de papel.
- **Dispositivos de exploração ótica:** podem ler diretamente documentos. Entre os mais primitivos, pode-se citar as leitoras óticas de cartões ou folhas de marcas (como as usadas no vestibular para registro das respostas). Atualmente, existem dispositivos bem mais complexos, como os scanners, que podem transferir para o computador imagens obtidas em figuras, textos e fotografias.
- **Dispositivos de apontamento:** podem indicar informações ao computador por meio do posicionamento de um cursor sobre a tela, como é o caso do mouse e da caneta luminosa (ou light pen).

Adaptado de: WEBER, R. F. *Fundamentos de arquitetura de computadores*.  
Porto Alegre: Bookman, 2012.

Nesse contexto, avalie os periféricos de um sistema computacional dispostos a seguir.

I – Disco rígido

II – Mouse

III – Modem

IV – Pen drive

V – Teclado

Quantos podem ser utilizados tanto para entrada quanto para saída de dados?

A) 1

B) 2

C) 3

D) 4

E) 5

Resposta correta: alternativa C.

### Análise da questão

Um dispositivo (ou periférico) de entrada é capaz de enviar informações do mundo exterior ao sistema computacional. Como exemplo, podemos citar o scanner. Um dispositivo de saída, por sua vez, é capaz de externalizar informações oriundas do sistema computacional. Como exemplo, temos uma impressora comum. Alguns dispositivos, como uma impressora multifuncional, são considerados como dispositivos híbridos, dotados tanto de recursos de entrada (como o scanner) quanto de saída (a impressora em si).

Discos rígidos e pen drives podem ser classificados tanto como periféricos quanto como um tipo de memória. Por meio deles, o sistema computacional é capaz de ler e de gravar informações, o que faz com que sejam considerados como periféricos híbridos.

O modem é um periférico que permite a conexão entre computadores por meio de uma rede para, por exemplo, possibilitar o acesso à internet. Trata-se, também, de um dispositivo híbrido, já que, por meio dele, o sistema computacional pode tanto enviar quanto receber informações.

Mouses e teclados, por sua vez, permitem apenas que usuários enviem dados ao computador, sendo classificados, portanto, como periféricos de entrada.

Desse modo, entre os periféricos elencados na questão, aqueles que podem ser utilizados para ambas as situações são três: disco rígido, pen drive e modem.

**Questão 2.** A respeito do sistema operacional Linux, avalie as afirmativas.

I – O usuário de maior poder, que já vem configurado de fábrica, é o root.

II – Há apenas uma distribuição Linux, denominada CentOS.

III – O Linux é um sistema operacional de código aberto.

É correto o que se afirma em:

A) I, apenas.

B) III, apenas.

C) I e III, apenas.

D) II e III, apenas.

E) I, II e III.

Resposta correta: alternativa C.

### Análise das afirmativas

I – Afirmativa correta.

Justificativa: no Linux, o root é o usuário que tem acesso administrativo ao sistema, sendo, desse modo, aquele com maior nível de autorização. Ele tem permissão para criar novos usuários, remover usuários existentes e alterar configurações do sistema, entre outras tarefas gerenciais.

II – Afirmativa incorreta.

Justificativa: uma distribuição Linux, também conhecida como distro, é um sistema operacional criado a partir de uma coleção de softwares, com o uso do kernel Linux. Cada distribuição apresenta características particulares, como a maneira de realizar configurações no sistema e o modo de instalar programas, entre outras. Em geral, apesar de suas diferenças, a essência das distros é similar. Há diversas distribuições Linux, como Ubuntu, Debian, Mint, CentOS ou Fedora.

III – Afirmativa correta.

Justificativa: o Linux é um sistema cujo código-fonte está disponível sob a licença GPL, permitindo que qualquer pessoa o acesse, o modifique ou o distribua, de acordo com os termos da licença. Essa característica faz com que ele seja classificado como um sistema operacional de código aberto (open source).



## REFERÊNCIAS

### Audiovisuais

O FAMOSO problema do jantar dos filósofos! – Sistemas Operacionais. 2018. 1 vídeo (11 min). Publicado pelo canal Luís Paulo Bravin. Disponível em: <https://bit.ly/3YSqFQD>. Acesso em: 14 mar. 2023.

O PROBLEMA dos 5 filósofos. 2015. 1 vídeo (9 min). Publicado pelo canal Valquiria Huttner. Disponível em: <https://bit.ly/3YUjJCA>. Acesso em: 14 mar. 2023.

### Textuais

ALVES, D. *Sistemas operacionais de redes (Windows/Linux)*. São Paulo: Sol, 2018.

AWS. *O que é a computação em nuvem?* [s.d.]. Disponível em: <https://go.aws/3zrlvj>. Acesso em: 14 mar. 2023.

BALL, B.; DUFF, H. *Dominando Linux: Red Hat e Fedora*. São Paulo: Pearson, 2004.

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

BRAGA, S. A.; BEZERRA, H. C.; GARCIA, F. P. Crianças brincando: uma ferramenta para o auxílio na aprendizagem de programação concorrente. In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA (COBENGE), 43., 2015, São Bernardo do Campo. *Anais [...]*. São Bernardo do Campo: Cobenge, 2015. Disponível em: <https://bit.ly/3LoHcsC>. Acesso em: 14 mar. 2023.

BINNIE, C. *Segurança em servidores Linux*. São Paulo: Novatec, 2017.

CENTOS. *CentOS Linux*. [s.d.]. Disponível em: <https://bit.ly/3TmVBYi>. Acesso em: 14 mar. 2023.

COMPUTER MUSEUM HISTORY. *Timeline of computer history*. Califórnia: CHM, [s.d.]. Disponível em: <https://bit.ly/2L452LO>. Acesso em: 14 mar. 2023.

CÓRDOVA JUNIOR, R. S.; LEDUR, C. L.; MORAIS, I. S. *Sistemas operacionais*. Porto Alegre: Grupo A, 2019.

DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. *Sistemas operacionais*. São Paulo: Pearson Prentice-Hall, 2005.

DEMARTINI, F. Linux e Android têm mais falhas críticas que o Windows, aponta estudo. *Canaltech*, 10 mar. 2020. Disponível em: <https://bit.ly/3mPTMHh>. Acesso em: 14 mar. 2023.

DENARDIN, G. W.; BARRIQUELLO, C. H. *Sistemas operacionais de tempo real e sua aplicação em sistemas embarcados*. São Paulo: Blucher, 2019.

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

- FURUKAWA, F.; NUNES, R. *Fundamentos de sistemas operacionais*. São Paulo: Sol, 2011.
- KARGER, P. A.; SCHELL, R. R. *Multics security evaluation: vulnerability analysis*, v. 2. Massachusetts: Electronic Systems Division, 1974. Disponível em: <https://bit.ly/3FmeAMK>. Acesso em: 14 mar. 2023.
- KROPIWIEC, D. D.; GEUS, P. L. *Paradigmas de segurança em sistemas operacionais*. Campinas: Unicamp, 2004. Disponível em: <https://bit.ly/3JiTfVm>. Acesso em: 14 mar. 2023.
- LAUREANO, M. *Máquinas virtuais e emuladores: conceitos, técnicas e aplicações*. São Paulo: Novatec, 2006.
- MACHADO, F. B.; MAIA, L. P. *Arquitetura de sistemas operacionais*. 5. ed. Rio de Janeiro: LTC, 2013.
- MAIA, L. P. *SOsim: simulador para o ensino de sistemas operacionais – Versão 2.0*. Rio de Janeiro: UFRJ, 2007. Disponível em: <https://bit.ly/2Qeb509>. Acesso em: 14 mar. 2023.
- MATEUS. File systems: Comparando os sistemas de arquivos do Linux. *Revista Infra Magazine*, v. 18, 2014. Disponível em: <https://bit.ly/403m8fo>. Acesso em: 14 mar. 2023.
- MAZIERO, C. A. *Sistemas operacionais: conceitos e mecanismos*. Curitiba: DINF – UFPR, 2019. Disponível em: <https://bit.ly/3ZS02Ly>. Acesso em: 14 mar. 2023.
- MEHL, E. L. M. *Do transistor ao microprocessador*. Curitiba: Universidade Federal do Paraná (UFPR), [s.d.]. Disponível em: <https://bit.ly/3LpdZh2>. Acesso em: 14 mar. 2023.
- MICROSOFT. *Windows Server 2012 R2 and Windows Server 2012*. 2016. Disponível em: <https://bit.ly/3mPtBKq>. Acesso em: 14 mar. 2023.
- NEGUS, C. *Linux: a Bíblia – O mais abrangente e definitivo guia sobre Linux*. Rio de Janeiro: Alta Books, 2014.
- NEMETH, E.; SNYDER, G.; HEIN, T. R. *Manual completo de Linux: guia do administrador*. São Paulo: Pearson, 2007.
- NIELES, M.; DEMPSEY, K.; PILLITTERI, V. Y. *An introduction to information security*. NIST special publication 800-12: revision 1. Washington: NIST, 2017. Disponível em: <https://bit.ly/2TumXLx>. Acesso em: 14 mar. 2023.
- POPEK, G. J.; GOLDBERG, R. P. Formal requirements for virtualizable third generation architectures. *Communications of the ACM*, v. 17, n. 7, p. 412-421, 1974.
- SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. *Fundamentos de sistemas operacionais*. 9. ed. Rio de Janeiro: LTC, 2015.

SILVA, M. B. F. *Cibersegurança: visão panorâmica sobre a segurança da informação na internet*. Rio de Janeiro: Freitas Bastos, 2023.

SILVA, M. B. F. *Segurança em sistemas operacionais Windows/Linux*. São Paulo: Sol, 2020.

SMITH, J. E.; NAIR, R. *Virtual machines: architectures, implementations and applications*. San Francisco: Morgan Kaufmann, 2004.

STANEK, W. R. *Windows Server 2012: guia de bolso*. Porto Alegre: Bookman, 2014.

TANENBAUM, A. S.; BOS, H. *Sistemas operacionais modernos*. 4. ed. São Paulo: Pearson: 2016.

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

TANENBAUM, A. S., WOODHULL, A. S. *Sistemas operacionais: projetos e implementação – o livro do Minix*. 3. ed. Porto Alegre: Bookman, 2008.

THOMPSON, M. A. *Windows Server 2012: instalação, configuração e administração de redes*. São Paulo: Érica, 2013.



Handwriting practice lines consisting of 30 horizontal blue lines. The first line is a solid blue line, and the subsequent 29 lines are pairs of dashed blue lines for tracing.



Lined writing area with horizontal lines.





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