



## UNIDADE III

---

### Sistemas Operacionais

Prof. Me. Michel Fernandes

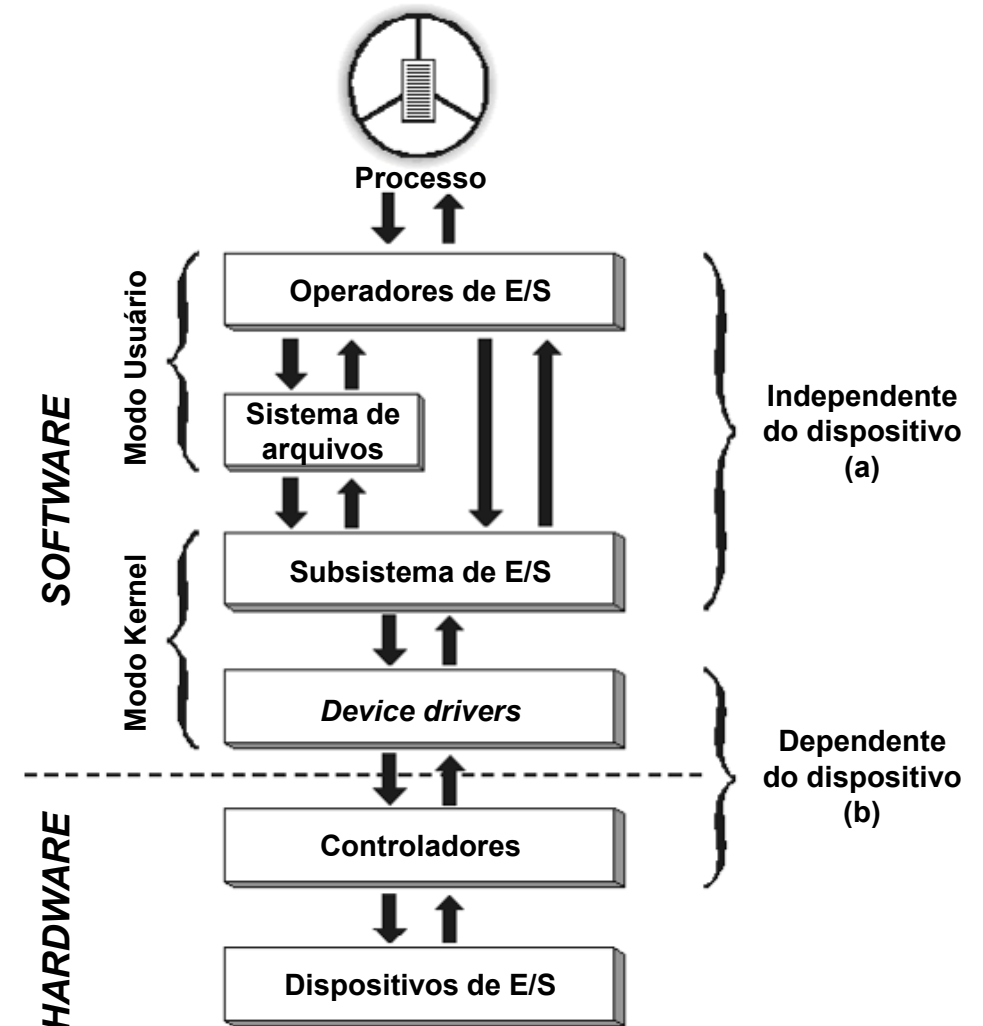
## Gerência de Entrada/Saída (E/S)

- A Gerência de Dispositivos de Entrada/Saída é uma das mais complexas funções do SO e sua implementação é estruturada por camadas e tem como princípio básico a abstração, tornando a interação do programador com a máquina algo muito mais fácil e permitindo que os programas e os *hardwares* evoluam de forma independente, porém estruturada.
- A diversidade dos tipos de E/S exige que o SO implemente uma camada chamada de subsistema de E/S, com a função de isolar a complexibilidade dos dispositivos chamada de sistemas de arquivos e da aplicação, possibilitando ao sistema manter sua flexibilidade.

# Estrutura da gerência de Entrada/Saída (E/S)

A gerência de E/S está dividida em duas camadas:

- 1ª camada: visualiza diversos tipos de dispositivos do sistema de um modo único.
- 2ª camada: é específica para cada dispositivo.



Fonte: adaptado de: Machado;  
Maia (2013, p. 208).

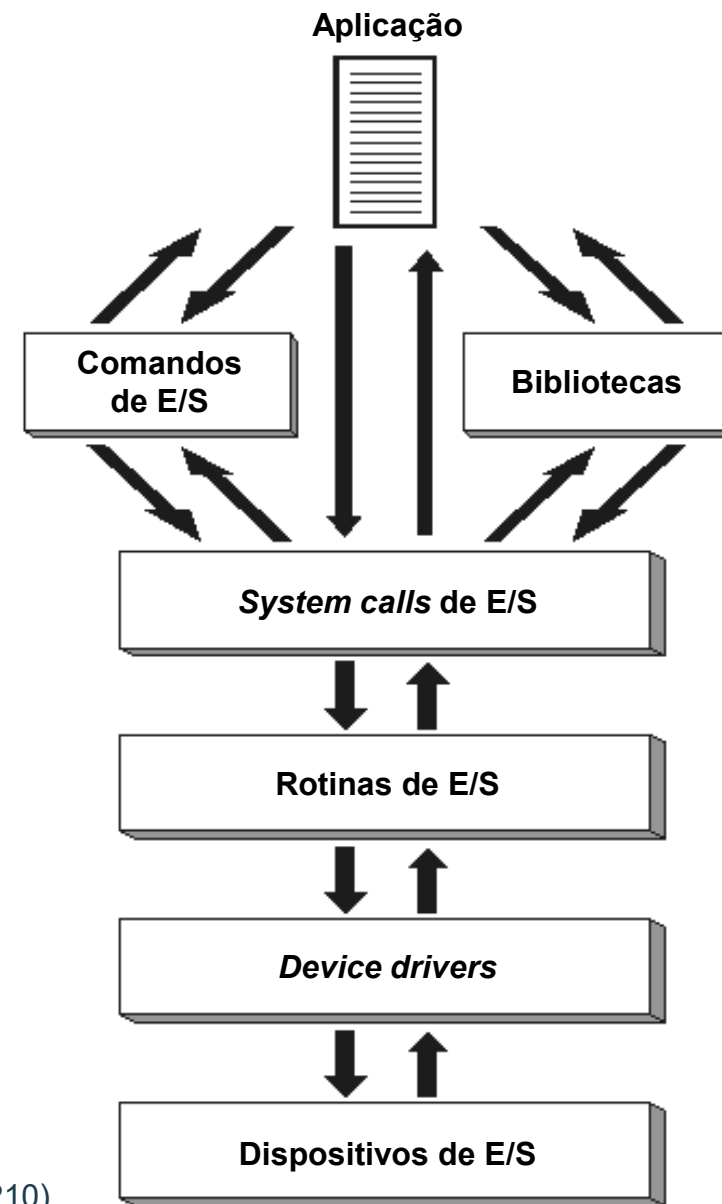
# Acesso ao subsistema de E/S

- SO possui um conjunto de rotinas que torna possível a comunicação com qualquer dispositivo que possa ser conectado ao computador. São denominadas de rotinas de Entrada/Saída e fazem parte do subsistema de E/S.
- As operações de E/S são realizadas pelo System Calls que chama a rotina de E/S do núcleo do SO.
  - Dessa forma, é possível escrever um programa que manipule arquivos, estejam eles em disquetes, discos rígidos, CDs, fitas magnéticas, sem ter que alterar o código para cada tipo de dispositivo.
  - Esses *System Calls* de Entrada/Saída têm como objetivo a simplificação da interface entre as aplicações e os dispositivos.

# Operações de E/S

As operações de E/S podem ser classificadas em:

- Operação síncrona: quando o processo que realizou a operação fica aguardando em estado de espera por seu término.
- Operação assíncrona: 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 que a operação foi terminada.

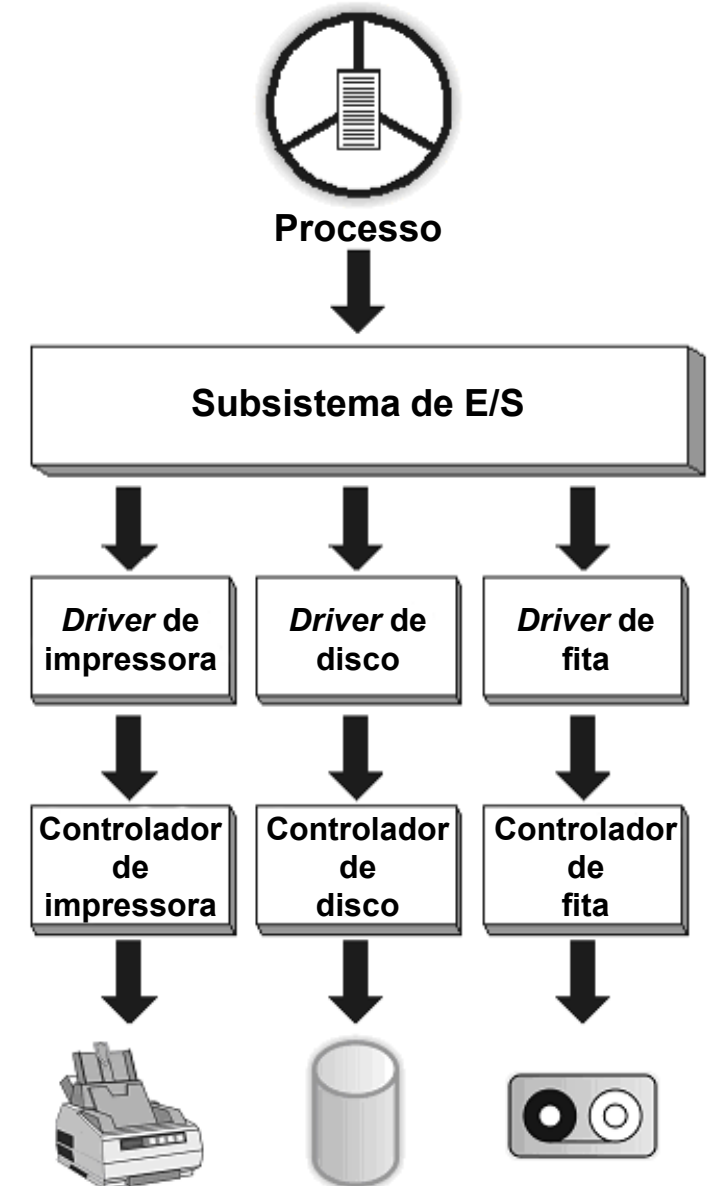


## *Device drivers*

- *Device driver*, ou somente *driver*, tem como função implementar a comunicação do subsistema de E/S com os dispositivos por controladores.
- Os *drivers* recebem os comandos gerais sobre os acessos aos dispositivos e traduzem para comandos específicos que poderão ser acessados pelas funções dos controladores e com isso eles possam entendê-las e executá-las.
- Cada *driver* manipula um tipo de dispositivo ou grupo de dispositivos semelhantes e normalmente o SO possui diferentes *drivers* para cada recurso computacional.

# Device drivers

- Devido ao grau de dependência entre os *drivers* e o restante do núcleo do SO, os fabricantes desenvolvem para um mesmo dispositivo diferentes *drivers*, cada um para um SO.



# Controladores de dispositivo

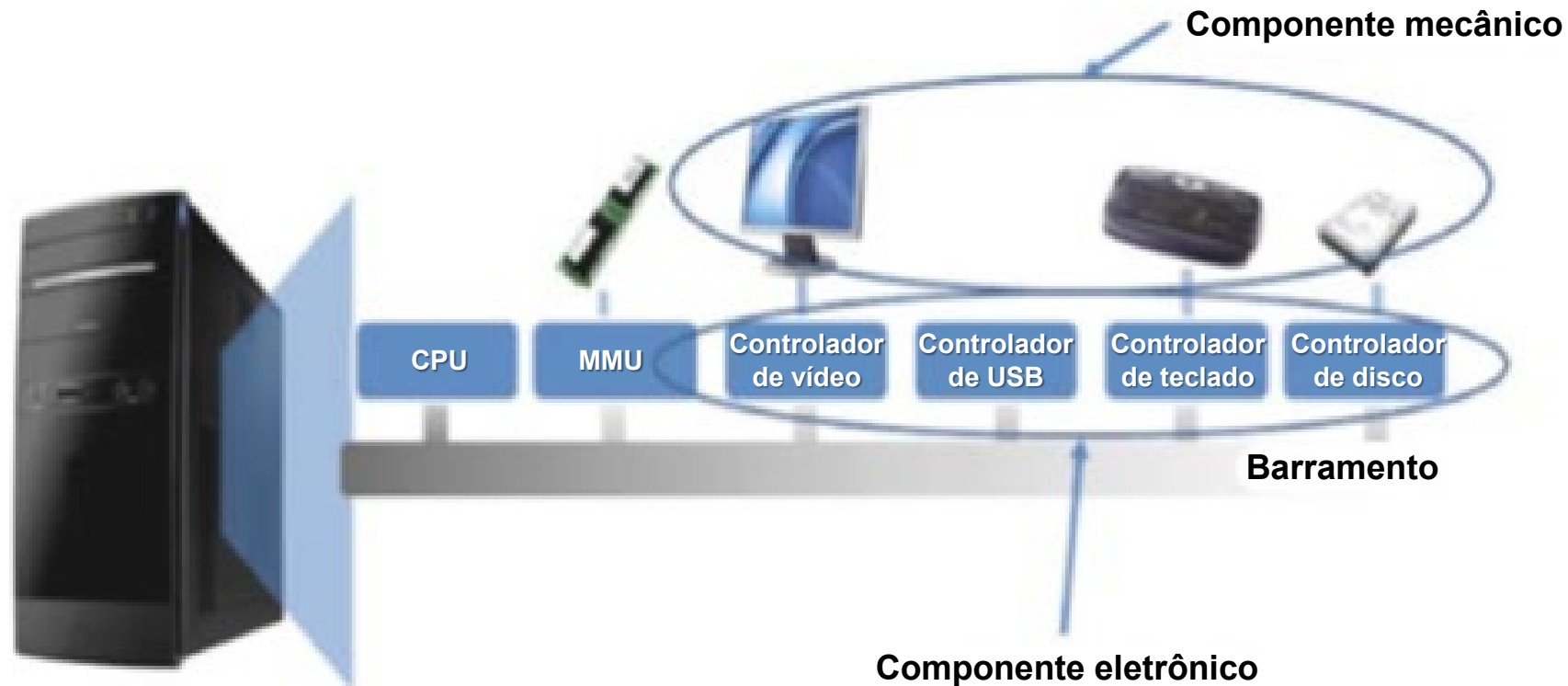
- São componentes de *hardware* responsáveis por manipular diretamente os dispositivos de E/S.
- O *driver* se comunica com os dispositivos pelos controladores.
- Ele pode ser uma placa independente conectada a um *slot* do computador ou implementada na mesma placa do processador.
- O controlador possui memória e registradores próprios.



# Controladores de dispositivo

As unidades de entrada e saída típicas são constituídas por dois componentes:

- Componente mecânico: é o dispositivo mais aparente para o usuário final, ou seja, impressora, teclado, *mouse* e outros.
- Componente eletrônico: controlador de dispositivo ou adaptador. Esses adaptadores são inseridos em um conector de expansão localizado na placa-mãe do computador.



# Tipos de dispositivos de E/S

- Os dispositivos de entrada e saída típicos e os que daremos ênfase são divididos em duas categorias: dispositivos de blocos e dispositivos de caractere.
- Dispositivos de blocos: entre outras características, armazenam informações em blocos de tamanho fixo e endereço próprio. Todas as transferências estão em unidades consecutivas de um ou mais blocos. Para essa categoria, cada bloco pode ser lido ou escrito independentemente de todos os outros.



# Tipos de dispositivos de E/S

- Dispositivos de caractere: nesse caso há o envio e o recebimento de caracteres.
- Diferentemente dos dispositivos de bloco, os dispositivos de caractere não são endereçáveis e não possuem funcionalidades de posicionamento.



## Software de E/S

- O *software* de E/S deve estar suportado pelo conceito de independência do *hardware*. Isso parte do pressuposto que deveria ser possível que os programas pudessem acessar os dispositivos de E/S sem a necessidade específica de conhecer o dispositivo. Exemplo: leitura de um arquivo de entrada tanto em um disco rígido, em CD, DVD, USB ou na nuvem.
- Programas de E/S deveriam estar alheios ao tratamento de erros, ficando a cargo dos níveis mais próximos ao *hardware* esse tratamento, ou seja, o controlador deveria resolver o problema e, se não conseguisse, então o *driver* do dispositivo deveria tratar.

## Software de E/S

- Definição do tipo de transferência síncrona ou assíncrona do dispositivo de E/S é fundamental.
- Na transferência síncrona, o modo é de bloqueio e na assíncrona é orientada à interrupção.
- A utilização de *buffers* para armazenamento temporário envolve frequentes e elevadas operações de cópia, gerando um impacto considerável no desempenho de entrada e saída.

Utilização de dispositivos dedicados *versus* dispositivos compartilhados:

- Dispositivos não compartilhados (dedicados) podem apresentar grandes problemas, bem como impasses.

# Software de E/S

- Tipicamente, os *softwares* de E/S possuem quatro camadas e estão logo acima do *hardware*.
- Cada camada do *software* de entrada e saída tem função específica e interface com as camadas vizinhas.

Os quatro níveis de uma estrutura de E/S são:

1. Rotinas dos serviços de interrupção.
2. *Drivers* dos dispositivos.
3. *Software* de E/S independente de dispositivo.
4. *Software* de E/S do espaço do usuário.



# Interatividade

O gerenciamento de entrada e saída tem como princípio básico a abstração, tornando a interação do programador com a máquina algo muito mais fácil e permitindo que os programas e os *hardwares* evoluam de forma independente, porém estruturada. Os dispositivos de entrada e saída típicos são classificados em:

- a) Dispositivos voláteis e dispositivos não voláteis.
- b) Dispositivos locais e dispositivos de armazenamento.
- c) Dispositivos mecânicos e dispositivos eletrônicos.
- d) Dispositivos dedicados e dispositivos compartilhados.
- e) Dispositivos de blocos e dispositivos de caractere.

# Resposta

O gerenciamento de entrada e saída tem como princípio básico a abstração, tornando a interação do programador com a máquina algo muito mais fácil e permitindo que os programas e os *hardwares* evoluam de forma independente, porém estruturada. Os dispositivos de entrada e saída típicos são classificados em:

- a) Dispositivos voláteis e dispositivos não voláteis.
- b) Dispositivos locais e dispositivos de armazenamento.
- c) Dispositivos mecânicos e dispositivos eletrônicos.
- d) Dispositivos dedicados e dispositivos compartilhados.
- e) Dispositivos de blocos e dispositivos de caractere.



# Dispositivos de E/S

- Permitem a comunicação entre o SO e o mundo externo.

São classificados em:

- Entrada de dados: CD-ROM, teclado, *mouse*.
- Saída de dados: impressoras, projetor.
- Entrada/Saída de dados: modem, disco rígido, tela *touch-screen*.

# Dispositivos de E/S

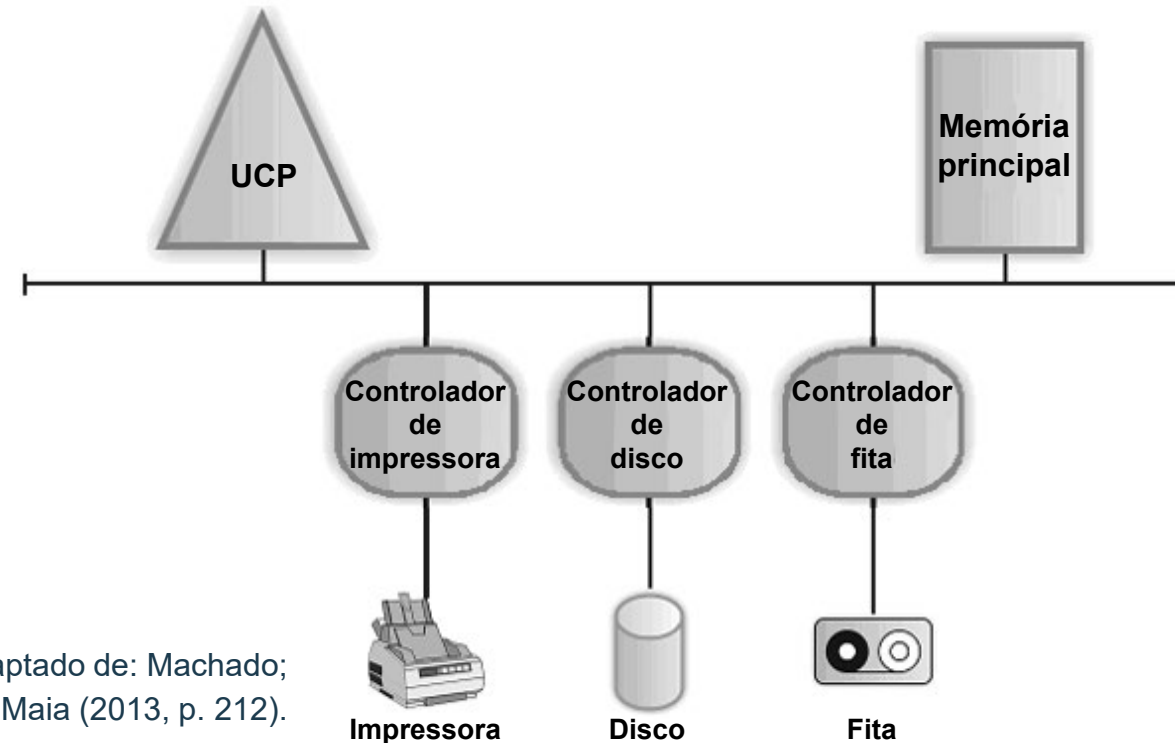
Padronização de equipamentos de E/S:

- ISO.
- IEEE.
- ANSI.
  
- Velocidades típicas de E/S.

Dispositivos	Velocidades
Teclado	10 B/s
Mouse ótico	100 B/s
Interface infravermelha (IrDA-SIR)	15 KB/s
Interface paralela padrão	125 KB/s
Interface de áudio digital S/PDIF	384 KB/s
Interface de rede <i>Fast Ethernet</i>	11.6 MB/s
<i>Pendrive</i> ou disco USB 2.0	60 MB/s
Interface de rede <i>Gigabit Ethernet</i>	116 MB/s
Disco rígido SATA 2	300 MB/s
Interface gráfica <i>high-end</i>	4.2 GB/s

# Controladores

- São utilizados na execução de instruções enviadas pelo *driver*. Em operações de leitura, armazenam em seu *buffer* interno uma sequência de bits provenientes do *driver* até formar um bloco. Após verificar se no bloco existem erros, são transferidos para um *buffer* de E/S na memória principal.
- A transferência do bloco para o *buffer* pode ser realizada por um controlador de DMA.



Fonte: adaptado de: Machado;  
Maia (2013, p. 212).

# Dispositivos de E/S

Em função da forma com que os dados são armazenados, são classificados em:

- Dispositivos estruturados: armazenam as informações em blocos de tamanho fixo, possuindo cada qual um endereço que pode ser lido/gravado. Ex.: disco rígido.

Podem ser divididos em:

- Acesso direto: quando um bloco pode ser recuperado pelo seu endereço.
- Acesso sequencial: quando, para se acessar um bloco, o dispositivo percorre sequencialmente os demais blocos.
  - Dispositivos não estruturados: enviam ou recebem uma sequência de caracteres sem estar estruturada no formato de um bloco de mesmo tamanho. Ex.: terminais, impressoras, interfaces de rede.

# Relógio

- É necessário manter o funcionamento de segundos, minutos, horas, data e ano; mesmo que o computador esteja desligado.
- Com isso, quando o equipamento estiver ligado e em funcionamento, é o relógio que irá fornecer o tempo real e atual para o ambiente.
- Cálculo do tempo do processo na CPU e tempo de alternar entre os demais processos.

Componentes do relógio:

- Oscilador de cristal.
- Controlador.
- Registrador de apoio.



# Dispositivos específicos

Controladoras Raid (*Redundant Array of Independent Disks*):

- Raid é uma estrutura que se propõe a solucionar problemas associados ao armazenamento de grandes quantidades de dados. Ela é associada sempre à cópia de segurança.
- Princípio fundamental de uma estrutura Raid: combinar vários discos rígidos físicos em uma estrutura lógica de discos de forma a aumentar a confiabilidade e o desempenho do ambiente.

# Dispositivos específicos

- Conjunto de discos (*array*) independentes (*independent*) em Raid armazena informações de forma redundante (*redundant*), viabilizando a recuperação de dados em caso de falha física de um dos discos (*disks*).

A estrutura de Raid é dividida em níveis:

- RAID 0
- RAID 1
- RAID 5
- RAID 0+1

# Dispositivos específicos

## Nível 0: Striping

- Dados escritos são divididos entre os diferentes discos físicos que compõem o disco Raid.

## Nível 1: Mirroring

- Espelhamento de disco(s), sendo um determinado dado escrito simultaneamente em um disco primário e em um disco secundário de cópia.
- Nível 5: os dados são divididos entre os diferentes discos e, para cada *strip*, é calculada a paridade. Essa informação de paridade fica distribuída entre os diferentes discos.
  - Nível 0+1: combinação dos níveis 0 (*striping*) e 1 (*mirroring*), na qual os dados são divididos entre os discos para melhorar o rendimento e também utilizar outros discos para duplicação de dados.



# Interatividade

No que consiste a técnica de *striping*, utilizada em níveis de Raid?

- a) Consiste no espelhamento de dados de um disco em outro.
- b) Utilização de um disco somente para controle de paridade.
- c) É um *driver* utilizado em discos SCSI.
- d) Processo de gravação de dados distribuídos em dois ou mais discos.
- e) Forma de distribuição de blocos, trilhas e setores em um disco.

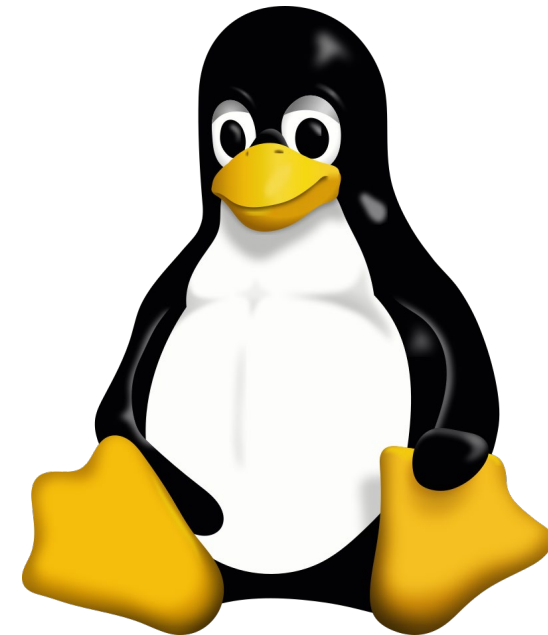
## Resposta

No que consiste a técnica de *striping*, utilizada em níveis de Raid?

- a) Consiste no espelhamento de dados de um disco em outro.
- b) Utilização de um disco somente para controle de paridade.
- c) É um *driver* utilizado em discos SCSI.
- d) Processo de gravação de dados distribuídos em dois ou mais discos.
- e) Forma de distribuição de blocos, trilhas e setores em um disco.

# Sistema operacional GNU/Linux

- Em 1991, o finlandês Linus Torvalds desenvolveu um novo kernel chamado Linux.
- Ele se uniu com as ferramentas desenvolvidas pelo projeto GNU, surgindo o que é conhecido atualmente como GNU/Linux.
- O desenvolvimento do Linux foi baseado no sistema Minix, criado em 1987 por Andrew Tanenbaum.
- O Minix se baseava nos padrões do Unix e, na época, era usado apenas na área acadêmica para estudos e desenvolvimentos específicos.



**Tux, mascote oficial do Linux.**

Fonte: <https://commons.wikimedia.org/wiki/Linux#/media/File:Tux.svg>.

# Sistema operacional GNU/Linux

- Linux é um sistema operacional de código aberto, utilizado por muitos desenvolvedores e times de TI para gerenciar seus ambientes e desenvolver novos códigos para o mercado.
- Desde o princípio, o Linux foi desenvolvido para ser um sistema operacional multitarefa e multiusuário.
- Em 1983, Richard Stallman iniciou o projeto que ficou conhecido como GNU, em que seu 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.
  - Em 1985, Richard Stallman fundou a Free Software Foundation (FSF), uma organização sem fins lucrativos que se dedica à eliminação de restrições sobre cópia, estudo e modificação de programas de computador (bandeiras do movimento do *software* livre).

# Software livre

*Software* livre significa que os usuários possuem as quatro liberdades essenciais:

- (Liberdade 0) para executar o programa.
- (Liberdade 1) para estudar e mudar o código-fonte do programa.
- (Liberdade 2) para redistribuir cópias exatas.
- (Liberdade 3) para distribuir versões modificadas.

# Software livre

- O termo software livre se refere à liberdade que o usuário tem de executar, distribuir, modificar e repassar as alterações sem que, para isso, tenha que pedir permissão ao autor do programa.
- Além de disponibilizar o código-fonte, o *software* livre agrega pessoas em comunidades de apoio que criam documentação e oferecem apoio na internet.
- É um sistema de suporte de ajuda horizontal e solidária, gerando redes de conhecimento e compartilhamento de informação.
  - Ser livre para fazer essas coisas significa que você não tem que pedir ou pagar pela permissão, uma vez que esteja de posse do programa.

# Utilização de Linux em dispositivos móveis

- Linux vem ganhando espaço não só em computadores, mas agora também em dispositivos móveis.
- O sistema operacional móvel Android tem sua base de apoio no núcleo do sistema operacional Linux e é o sistema operacional presente na maioria dos *smartphones* no mundo.

# Copyleft

- Copyleft é uma extensão das quatro liberdades básicas e ocorre na forma de uma obrigação.
- O copyleft diz que qualquer um que distribui *software*, com ou sem modificações, tem que passar adiante a liberdade de copiar e modificar novamente o programa (FSF).
- O copyleft garante que todos os usuários têm liberdade.
- Você verá o símbolo do copyleft, palavra que é um trocadilho com copyright, e cuja tradução aproximada seria “deixamos copiar” ou “cópia permitida”.



Fonte:  
<https://w7.pngwing.com/pngs/1020/307/png-transparent-copyleft-free-art-license-x-mark-words-phrases-copyright-contract.png>



# Distribuições do Linux

- Com base 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.
- Uma **distribuição** ou **distro** 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.
- Cada distribuição possui suas características particulares e muitas vezes exclusivas.
  - Observação: nem todas as distribuições Linux são gratuitas. Para evitar esse tipo de confusão, elas são organizadas em categorias.

# Distribuições do Linux

## ▪ Distribuições livres

Mantidas por comunidades de colaboradores que não visam a lucros sobre suas distribuições. As distribuições livres estão nas seguintes versões:

- Debian v 9.1.
- CentOSv 7.0.
- Slackwarev 14.1.



Fonte: [debian.org](http://debian.org)



# CentOS

Fonte: [centos.org](http://centos.org)

# Distribuições do Linux

- **Distribuições corporativas**

Mantidas por empresas que cobram pelo suporte prestado, possuem as seguintes características:

- Disponibilidade de suporte técnico especializado, conforme a distribuição utilizada.
- Homologação para a instalação de alguns programas corporativos.

Versões atuais são as seguintes:

- Oracle Linux v.7.
- Red Hat Enterprise Linux v.7.
- SuseLinux Enterprise v.12.

# Distribuições do Linux

- **Distribuições para iniciantes (estação de trabalho)**

São aquelas que permitem ao usuário fazer tudo aquilo que faria no sistema operacional que já domina, com interface gráfica amigável, opções claras, usabilidade e ferramentas gráficas. As versões atuais são as seguintes:

- Linux Mint 18.2.
- Ubuntu 16.10.
- Kubuntu 16.10.

# Distribuição do Linux – Distro

- Também chamadas de *releases*, essas versões são empacotadas de maneiras diferentes por vários coletivos comerciais e comunitários, que chamamos de distribuições ou simplesmente “distros”.
- As principais distros comerciais são Mandrake, RedHat e Suse Conectiva.
- As distros comunitárias mais conhecidas são Slackware e Debian.
- A distribuição comercial vende serviços de suporte para empresas que querem utilizar a sua versão de Linux.



Fonte:  
<https://www.dic.asdanet.com.br/2019/01/melhores-distribuicoes-linux-para.html>

# Interatividade

Em relação a *software* livre, assinale a alternativa correta.

- a) Qualquer *software* que não cobra pela licença é um *software* livre.
- b) Após alterar o código-fonte de um *software* livre, é possível vender a licença desse *software* modificado.
- c) Após o pagamento da licença, qualquer *software* vira *software* livre.
- d) Um dos requisitos fundamentais do *software* livre é acesso ao código-fonte.
- e) Como o *software* é livre, nenhuma empresa desenvolvedora terá quaisquer receitas advindas de serviços desse *software*.

# Resposta

Em relação a *software* livre, assinale a alternativa correta.

- a) Qualquer *software* que não cobra pela licença é um *software* livre.
- b) Após alterar o código-fonte de um *software* livre, é possível vender a licença desse *software* modificado.
- c) Após o pagamento da licença, qualquer *software* vira *software* livre.
- d) Um dos requisitos fundamentais do *software* livre é acesso ao código-fonte.
- e) Como o *software* é livre, nenhuma empresa desenvolvedora terá quaisquer receitas advindas de serviços desse *software*.

# Características do Linux

- Permite carregamento e descarregamento dinâmicos de *drivers* em tempo de execução.
- É um sistema multiusuário, com proteção entre processos.
- Usa um *scheduler* para partilhar o tempo do processador entre os processos ativos.
- Novos processos compartilham seletivamente partes do seu ambiente ancestral.
  - Isso permite trabalhar em *multi-threading*.



# Características do Linux

Um processo Linux possui um modelo de memória organizado em quatro partes:

- Texto – código do programa.
- Dados não inicializados.
- Dados inicializados.
- Pilha – espaço de memória para variáveis locais e passagem de parâmetros.

# Kernel

- Com o apoio de uma comunidade da internet, o Linux cresceu a partir de um kernel inicial, até realizar boa parte dos serviços do Unix.
- Inicialmente, o trabalho girou somente em torno do kernel, um módulo original do sistema.
- O Linux, como o conhecemos hoje, agregou desde então muitos módulos adicionais, alguns originais, outros adaptados.

# Buscando documentação

- Uma parte importante de qualquer sistema operacional é a documentação, os manuais técnicos que descrevem o uso e o funcionamento dos programas.
- O sistema GNU/Linux possui uma ampla documentação técnica oficial.
- A documentação dos comandos pode ser acessada pelo programa **MAN** (manual), seguido pelo comando desejado.

```
LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
```

Manual page ls(1) line 1 (press h for help or q to quit)

# Aspecto de segurança do GNU/Linux

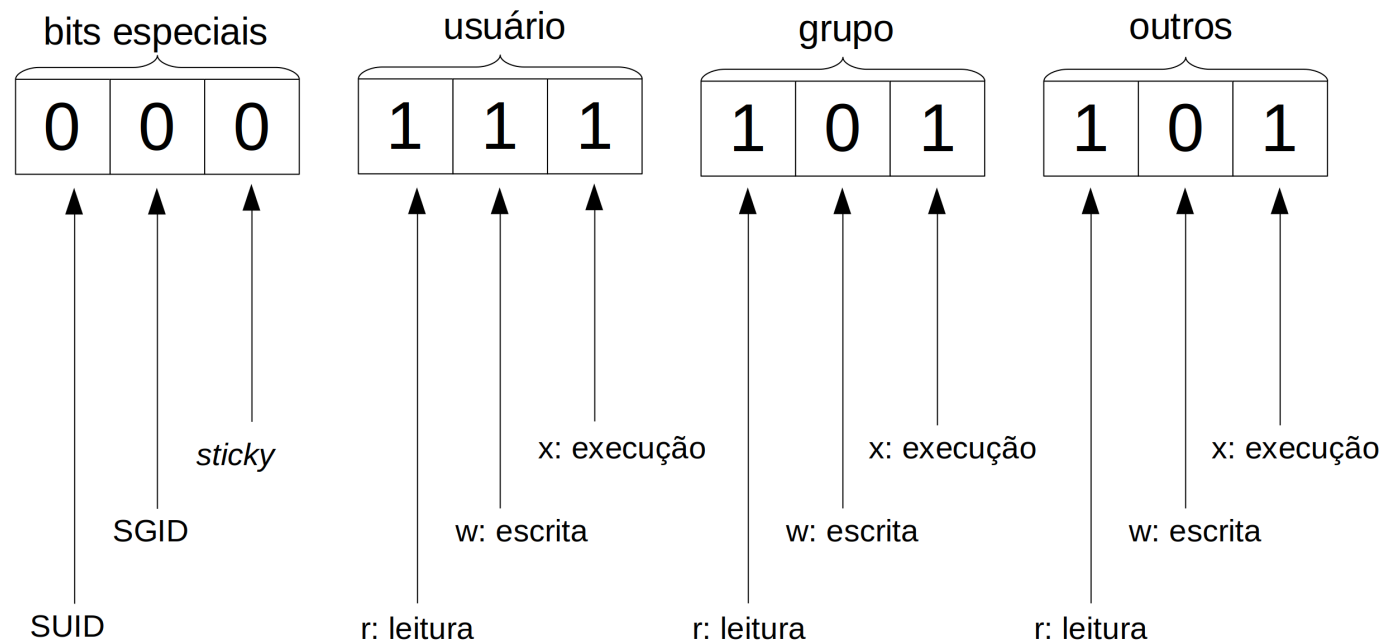
- Similaridade com o sistema UNIX.

## Mecanismos de autenticação

- *Login*: mecanismos de checagem de senhas.

## Controle de acesso em três categorias:

- ao usuário ou dono do arquivo;
- ao grupo desse usuário;
- aos usuários restantes.



# Aspecto de segurança do GNU/Linux

As permissões para um arquivo podem ser:

- de leitura, representada pela letra “r” (do inglês *read*);
  - de escrita, representada pela letra “w” (do inglês *write*);
  - de execução, representada pela letra “x” (do inglês *execute*, segunda letra).
  - A ausência de permissão é representada por um traço: “-”.
- 
- Alterações das permissões pelo comando “chmod”.

# Aspecto de segurança do GNU/Linux

## *Access Control Lists (ACLs)*

- Permitem ajustar ainda mais as permissões e também funcionam como uma camada adicional de controle de acesso discricionário.
- Complementando o controle de acesso discricionário, temos sistemas como o AppArmor e o SELinux, que operam no controle de acesso obrigatório, ou seja, não podem ser modificados pelo usuário e são configurados pelo administrador da máquina.

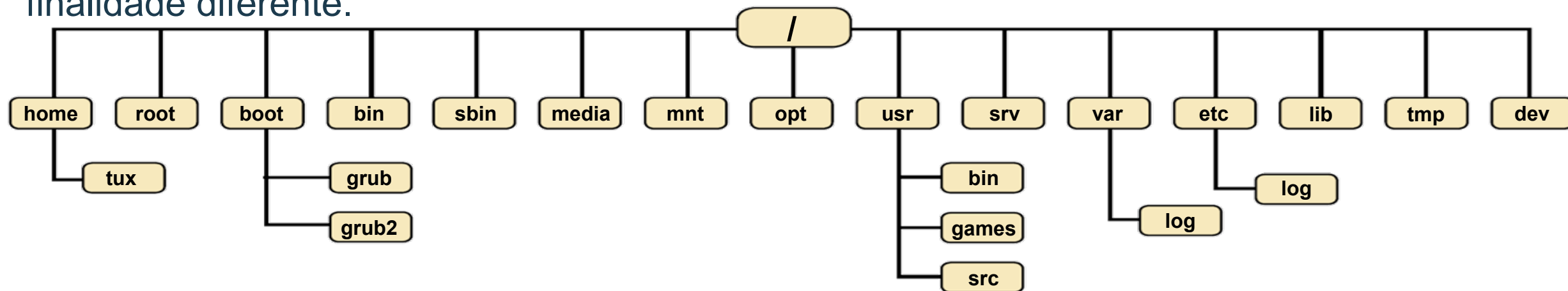
# Sistema de arquivos e diretórios

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: é conhecido como *Second Extended File System* e utiliza blocos do mesmo tamanho para armazenar os arquivos.
- EXT3: permite utilizar o sistema de cotas e pode trabalhar com blocos de 1, 2 e 4 kilobytes.
- EXT4: pode facilmente ser convertido para o formato ext4 e conta com melhorias no desempenho e na capacidade de armazenamento.

# Sistema de arquivos e diretórios

- Devido à existência de diversas distribuições disponíveis, foi preciso realizar uma padronização nos diretórios encontrados.
- Essa padronização é conhecida como *File System Hierarchy Standard* (FHS).
- Essa estrutura está dividida de uma forma hierárquica, em que cada diretório possui uma finalidade diferente.



Estrutura do FHS.  
Fonte: autoria própria.



# Shell

- Características do Linux e do Unix é a ampla utilização da linha de comando ou terminais.

Existem diversos “shells” disponíveis e cada um deles apresenta características próprias.

Exemplos de shell:

- Bourne shell ou sh.
  - GNU Bourne-Again shell ou Bash.
  - Korn shell ou ksh.
  - C shell, entre outros.
- 
- Shell funciona como uma interface (em modo de texto) entre o usuário e o sistema operacional e também funciona como uma linguagem interpretada.

# Interatividade

A quem se deve pedir permissão para utilizar, alterar e distribuir o *software* livre?

- a) MIT.
- b) Free Software Foundation.
- c) Projeto GNU.
- d) Campos.
- e) Ninguém.

# Resposta

A quem se deve pedir permissão para utilizar, alterar e distribuir o *software* livre?

- a) MIT.
- b) Free Software Foundation.
- c) Projeto GNU.
- d) Campos.
- e) Ninguém.

# Referências

- FURUKAWA, F.; NUNES, R. *Fundamentos de sistemas operacionais*. São Paulo: Editora Sol, 2011.
- HENESSY, J. L.; PATTERSON, D. A. *Arquitetura de computadores – uma abordagem quantitativa*. 5. ed. Rio de Janeiro: Campus, 2014.
- MACHADO, F. B., MAIA, L. P. *Arquitetura de sistemas operacionais*. 5. ed. Rio de Janeiro: LTC, 2013.
- MAZIERO, C. A. *Sistemas operacionais: conceitos e mecanismos [recurso eletrônico]*. Curitiba: DINF/UFPR, 2019.

**ATÉ A PRÓXIMA!**