



UNIDADE III

Sistemas Operacionais Abertos e Mobile

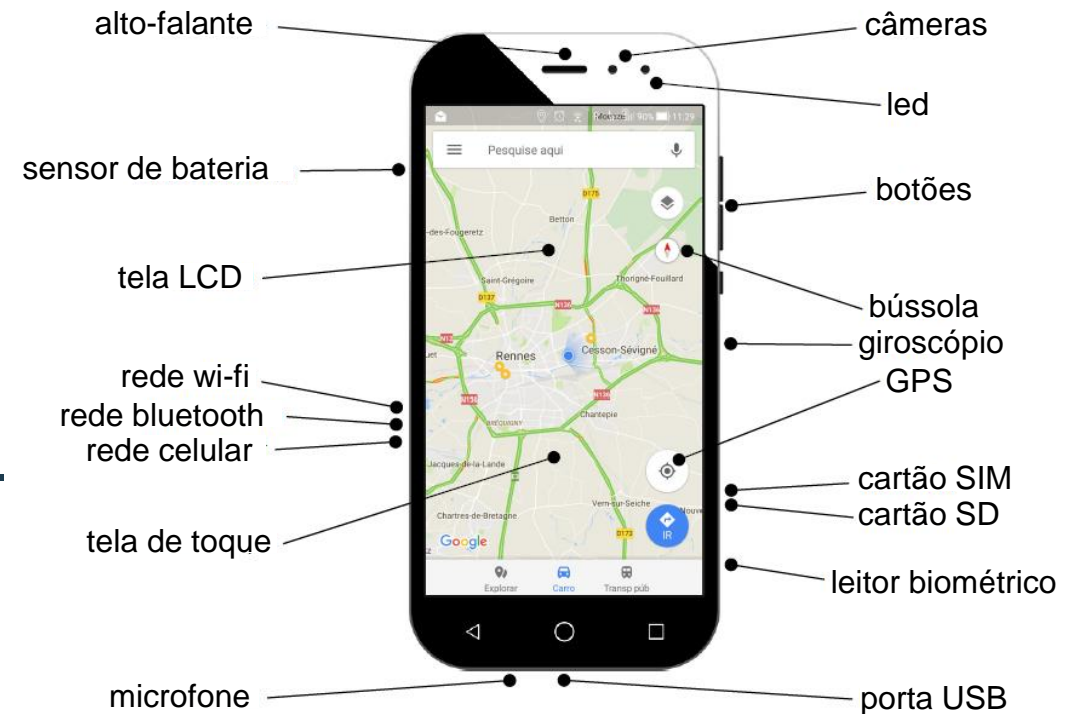
Prof. Me. Michel Fernandes

Gerência de Entrada/Saída

- Interação da CPU com o mundo exterior.
- Função: informações físicas \Leftrightarrow informações digitais.

Grande variedade de dispositivos:

- Interação com usuário: *mouse*, teclado, tela, fones.
- Armazenamento de dados: discos, SSDs, CD-ROMs.
- Impressão e captura de dados: impressora, escâner.



- Comunicação com outros sistemas: ethernet, Bluetooth.
- Gerência do sistema: tempo, energia, temperatura.
- Localização: GPS, bússola, giroscópio.

Fonte: adaptado de: Maziero (2019, p. 235).

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*.

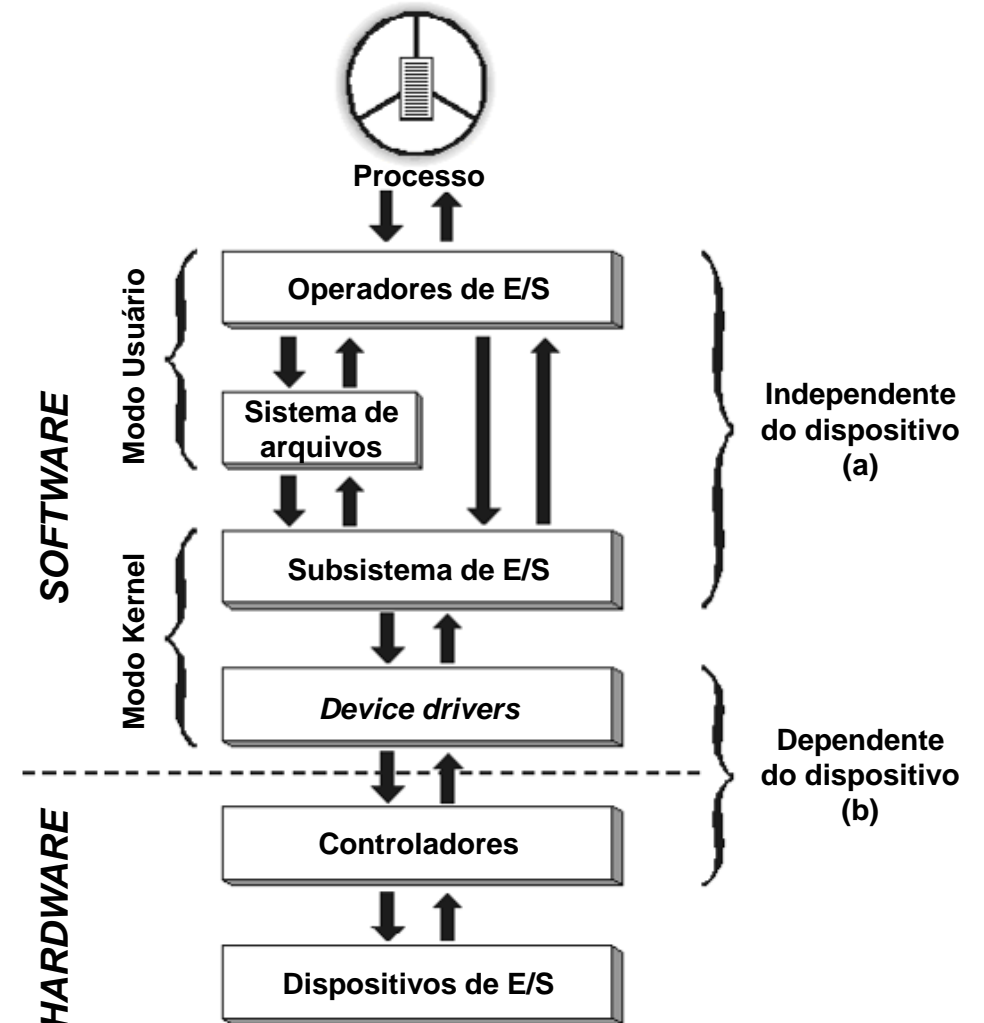
Gerência de Entrada/Saída (E/S)

- 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 chamados 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).

Componentes de sistema de E/S

Entrada de dados:

- Sensor: transforma grandeza física em sinal elétrico.
- Amplificador: aumenta e limpa o sinal elétrico.
- Conversor AD: converte o sinal em informação digital.

Saída de dados:

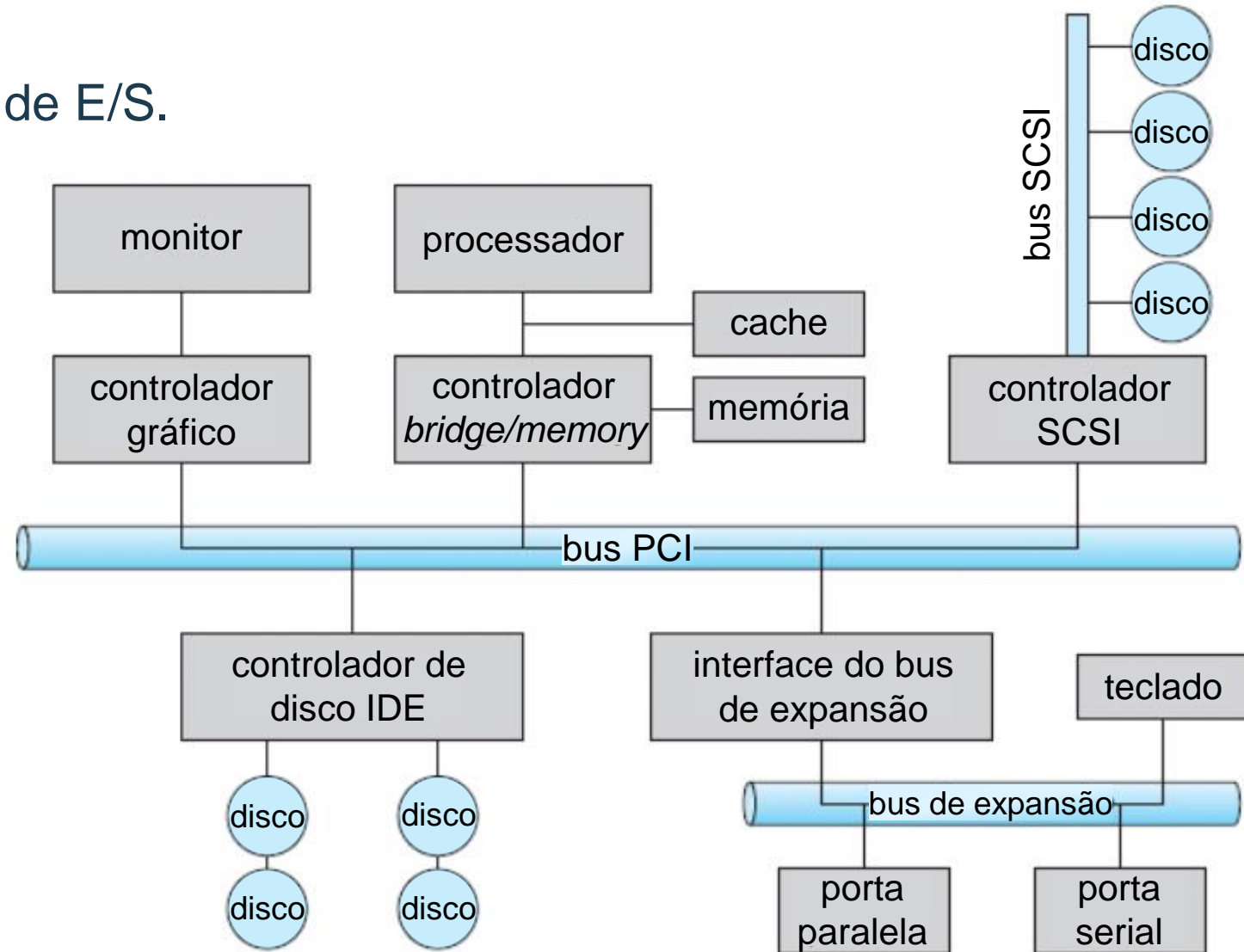
- Conversor DA: converte dados em sinais elétricos.
- Transdutor: transforma sinais elétricos em ações físicas.

Elementos comuns:

- *Buffer*: armazena dados coletados e/ou enviados.
- Controlador de barramento: permite acesso da CPU.
- Microcontrolador: gerencia o *hardware* do dispositivo.
- *Firmware*: código executado pelo controlador.

Barramentos

- Barramento: via de comunicação.
- Interliga CPU, memória e dispositivos de E/S.
- Parte do *chipset* da placa-mãe.
- Componentes rápidos: CPU, RAM, portas AGP e PCI-express (vídeo).
- Componentes lentos: PCI, USB, SATA, BIOS, controladores legados.



Barramentos

- Velocidades típicas de dispositivos de entrada e saída

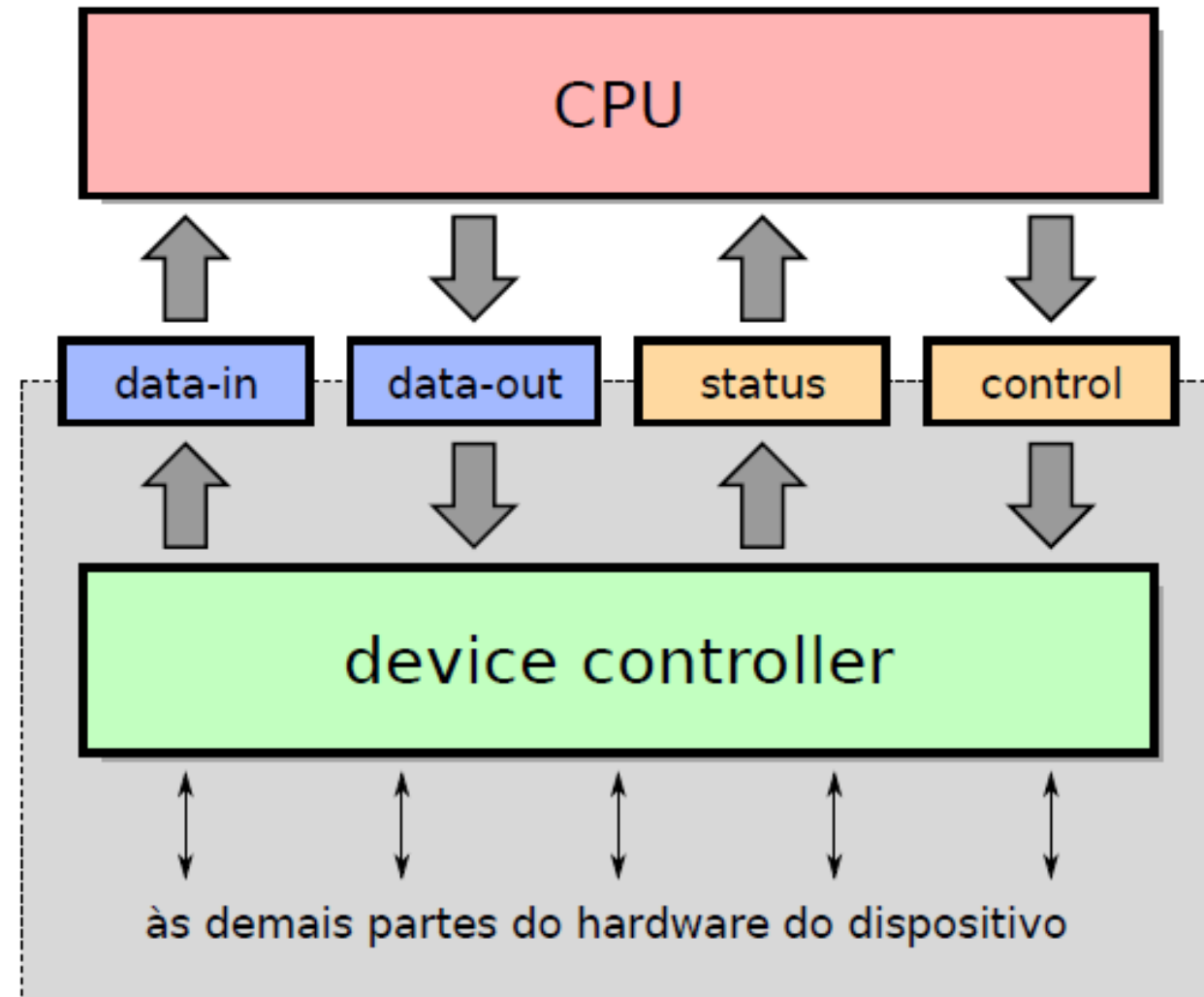
Dispositivo	Velocidade
Teclado	10 B/s
<i>Mouse</i> ótico	100 B/s
Interface infravermelha (IrDA-SIR)	14 KB/s
Interface paralela padrão	125 KB/s
Interface de áudio digital S/PDIF	384 KB/s
Interface de rede <i>Fast Ethernet</i>	11.6 MB/s
<i>Pendrive</i> ou disco USB 2.0	60 MB/s
Interface de rede <i>Gigabit Ethernet</i>	116 MB/s
Disco rígido SATA 2	300 MB/s
Interface gráfica <i>high-end</i>	4.2 GB/s

Interfaces de acesso

- Interação CPU *device* feita por portas de E/S.

Existem 4 tipos básicos de portas:

- Entrada.
- Saída.
- *Status*: consultar o estado do dispositivo.
- Controle: enviar comandos ao dispositivo e alterar a configuração do dispositivo.



Padronização de interfaces

- Interfaces seguindo um padrão oficial.

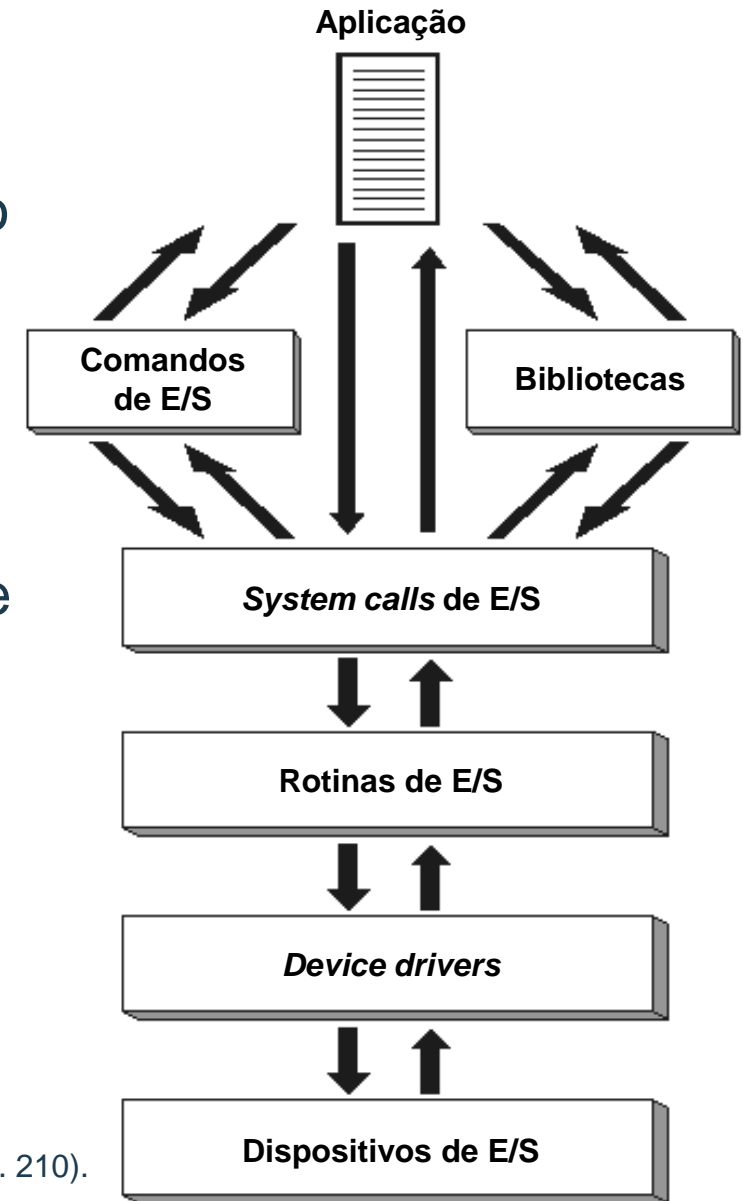
Fornecido por organizações reconhecidas internacionalmente como:

- IEEE.
 - ISO.
 - ANSI e outras.
-
- Os fabricantes seguirão os padrões definidos como os tipos de conectores, as tensões elétricas e propriedades mecânicas.

Operações de E/S

As operações de E/S podem ser classificadas em:

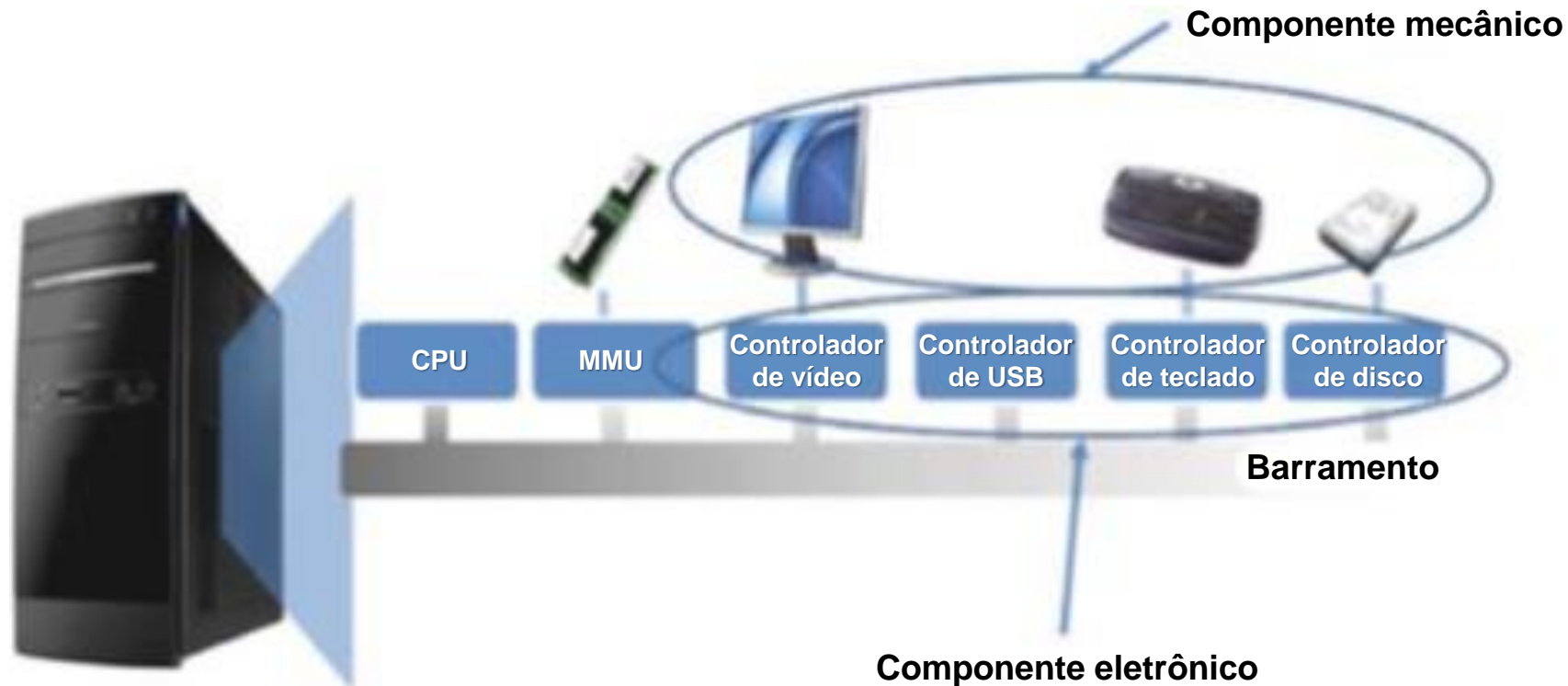
- Operação síncrona: quando o processo que realizou a operação fica aguardando em estado de espera por seu término.
- Operação assíncrona: quando o processo que realizou a operação não aguarda seu término e continua pronto para ser executado. Nesse caso deve existir uma sinalização que indique que a operação foi terminada.



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.



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.

Interatividade

Considere as seguintes afirmações sobre a gerência de dispositivos de Entrada/Saída:

- I. Em função da quantidade e variedade de dispositivos, é uma função muito complexa.
- II. A gerência de E/S visualiza diversos tipos de dispositivos do sistema de um modo único sem tratar suas especificidades.
- III. A velocidade dos barramentos existentes é para todos os dispositivos.

É correto o que se afirma em:

- a) I, apenas.
- b) II, apenas.
- c) I e II, apenas.
- d) I e III, apenas.
- e) II e III, apenas.

Resposta

Considere as seguintes afirmações sobre a gerência de dispositivos de Entrada/Saída:

- I. Em função da quantidade e variedade de dispositivos, é uma função muito complexa.
- II. A gerência de E/S visualiza diversos tipos de dispositivos do sistema de um modo único sem tratar suas especificidades.
- III. A velocidade dos barramentos existentes é para todos os dispositivos.

É correto o que se afirma em:

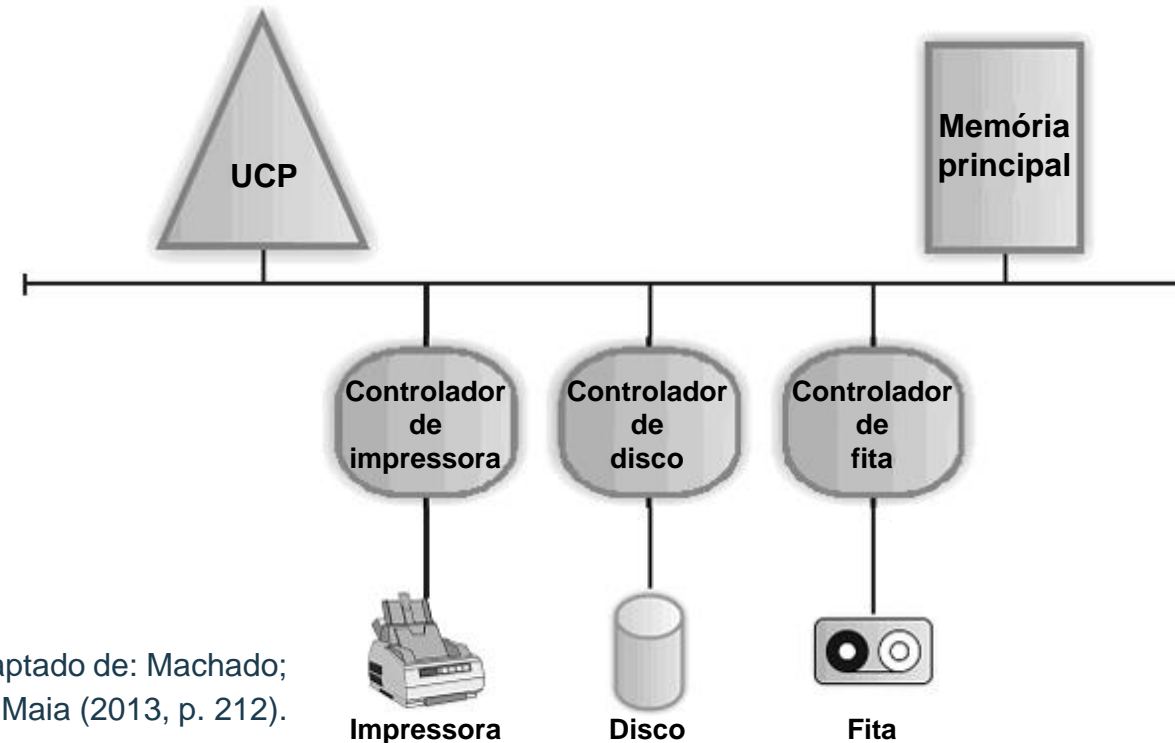
- a) I, apenas.
- b) II, apenas.
- c) I e II, apenas.
- d) I e III, apenas.
- e) II e III, apenas.

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.

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.

Disco rígido

- Dispositivo de armazenamento magnético.

Características:

- Criado em 1954.
- Um ou mais discos metálicos.
- Velocidade de rotação entre 4.200 e 15.000 RPM.
- Capacidade entre 100 GB e 10 TB.
- Taxa de transferência entre 0,5 e 2,0 Gbps.
- Latência entre 2 e 10 ms.

Estrutura lógica de um disco rígido

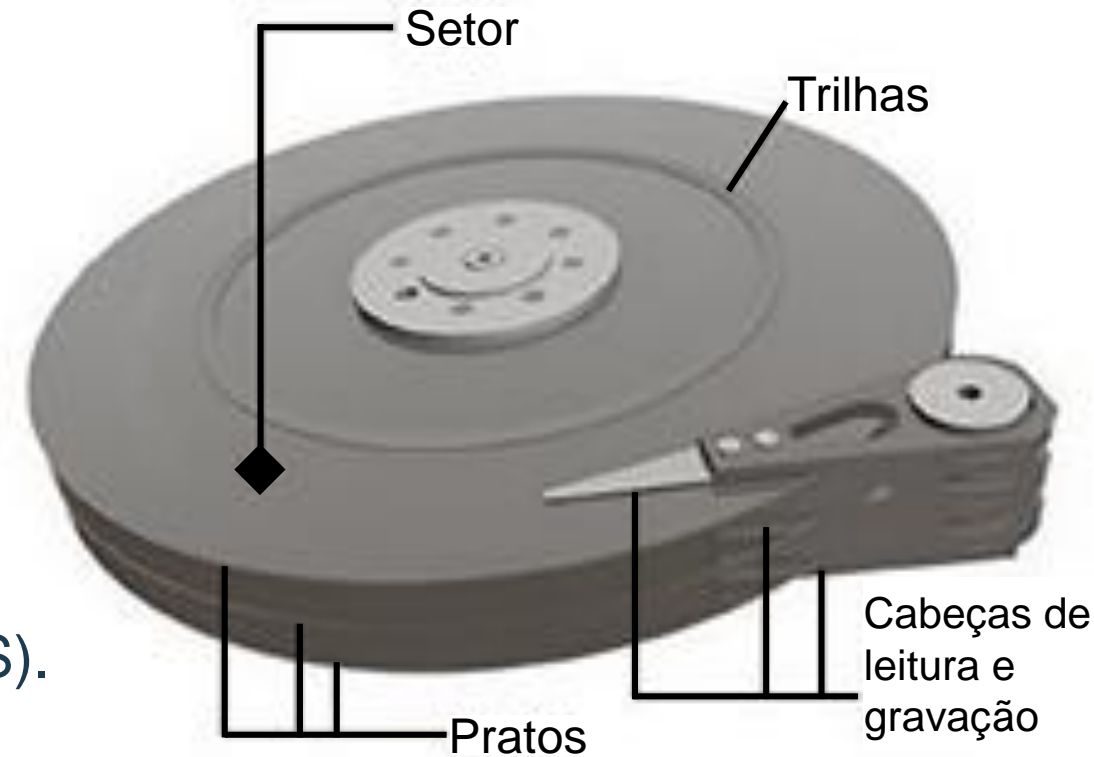
- Faces (ou cabeças): duas por disco metálico.
- Trilhas (ou cilindros): faixas concêntricas.
- Setores: “fatias” angulares.

Blocos físicos:

- Interseção entre cabeça, trilha e setor.
- Tamanho fixo de 512 ou 4.096 bytes.

Endereçamento dos blocos:

- Esquema CHS: *Cylinder, Head, Sector* (interno).
- Esquema LBA: *Large Block Array* (firmware ou BIOS).



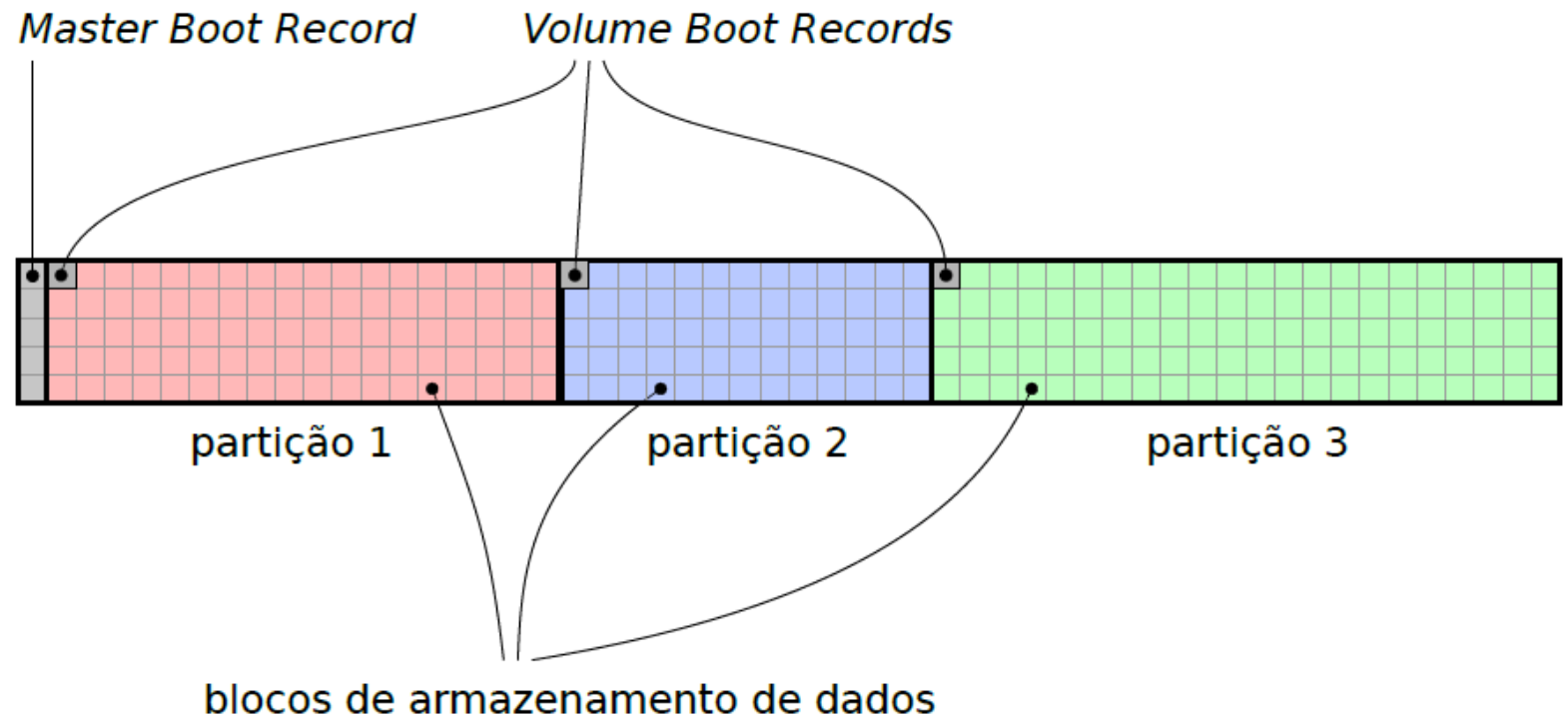
Fonte: adaptado de: Córdova Jr. (2018, p. 204).

Acessos ao disco rígido

- O disco é um dispositivo lento!
- Latência rotacional $t_r \approx 5$ ms.
- Tempo de busca $t_s \approx 10$ ms (*seek time*).
- O disco é um dispositivo sequencial: trata um pedido por vez!
- Tratamento dos pedidos de acesso ao disco.
- Pedidos dos processos são mantidos em uma fila.
- Busca-se desempenho e justiça.

Partições do disco rígido

- Disco rígido é um dispositivo de blocos.
- Há uma pequena área de configuração reservada no início do disco e pode existir uma ou mais partições, que representam espaços independentes.
- *Boot sector* ou *Master Boot Record* (MBR).
- Área reservada na bloco de inicialização da partição: *Volume Boot Record* (VBR).



Estratégia RAID

Problemas dos discos rígidos:

- Discos são lentos.
- Discos podem falhar, levando à perda de dados.

Solução: estratégia de conjunto redundante de discos econômicos (RAID):

- *Redundant Array of Independent Disks.*

RAID

- Criar um disco lógico a partir de discos físicos.
- Operações em paralelo permitem maior desempenho.
- Redundância (cópias) permitem tolerar falhas.
- Implementado em *hardware* dedicado ou *software*.
- Opera com blocos (abaixo dos arquivos).
- Ela é associada sempre à cópia de segurança.
- Princípio fundamental: combinar vários discos rígidos físicos em uma estrutura lógica de discos de forma a aumentar a confiabilidade e o desempenho do ambiente.

Dispositivos específicos

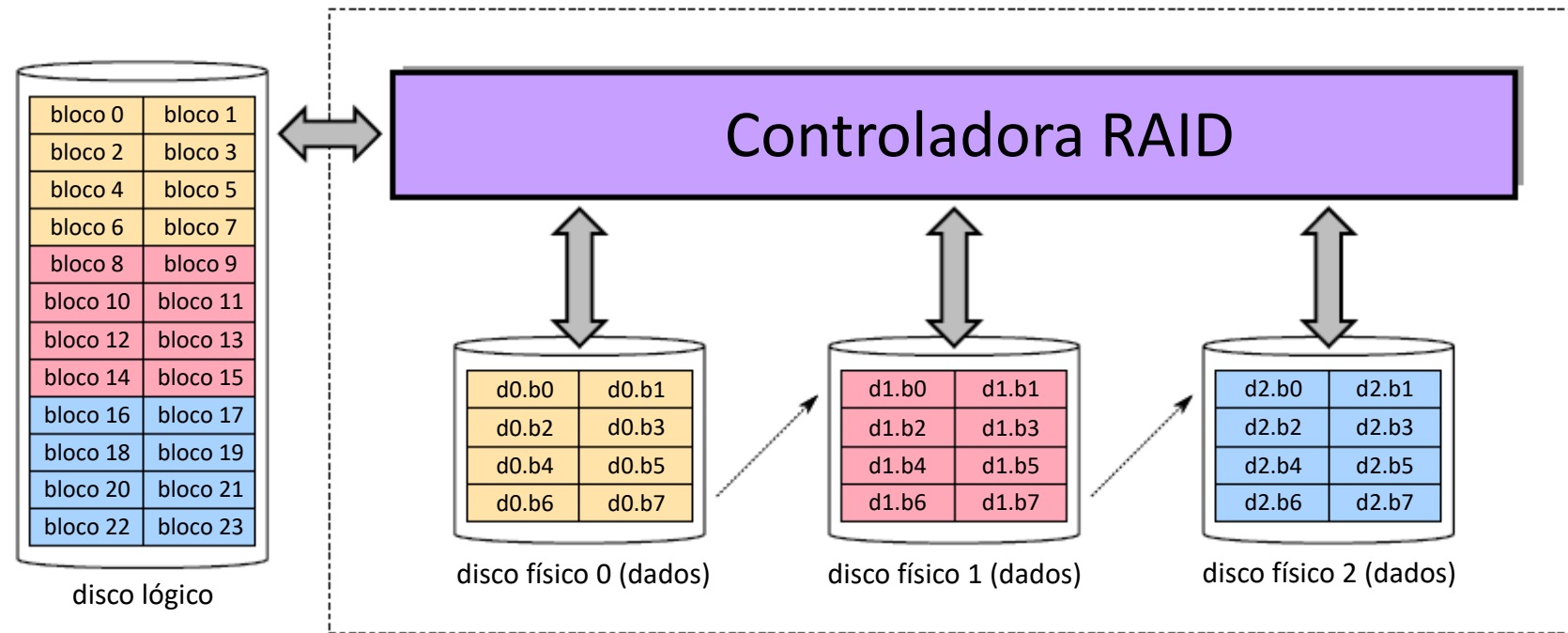
- Conjunto de discos (*array*) independentes (*independent*) em Raid armazena informações de forma redundante (*redundant*), viabilizando a recuperação de dados em caso de falha física de um dos discos (*disks*).

A estrutura de Raid é dividida em níveis:

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

RAID Nível 0 – Linear

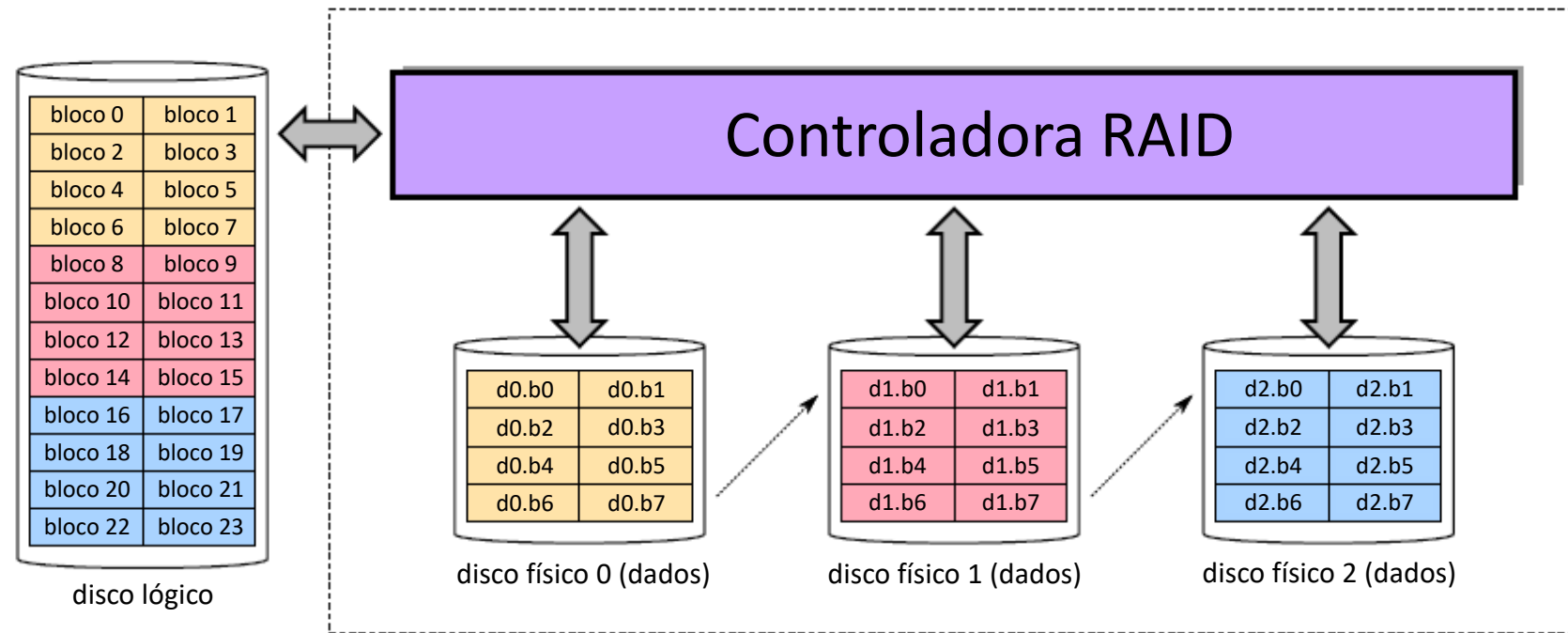
- Estratégia: concatena discos em sequência.
- Mais espaço e velocidade, sem redundância.



Fonte: adaptado de: Maziero (2019, p. 271).

RAID Nível 0 – *Striping*

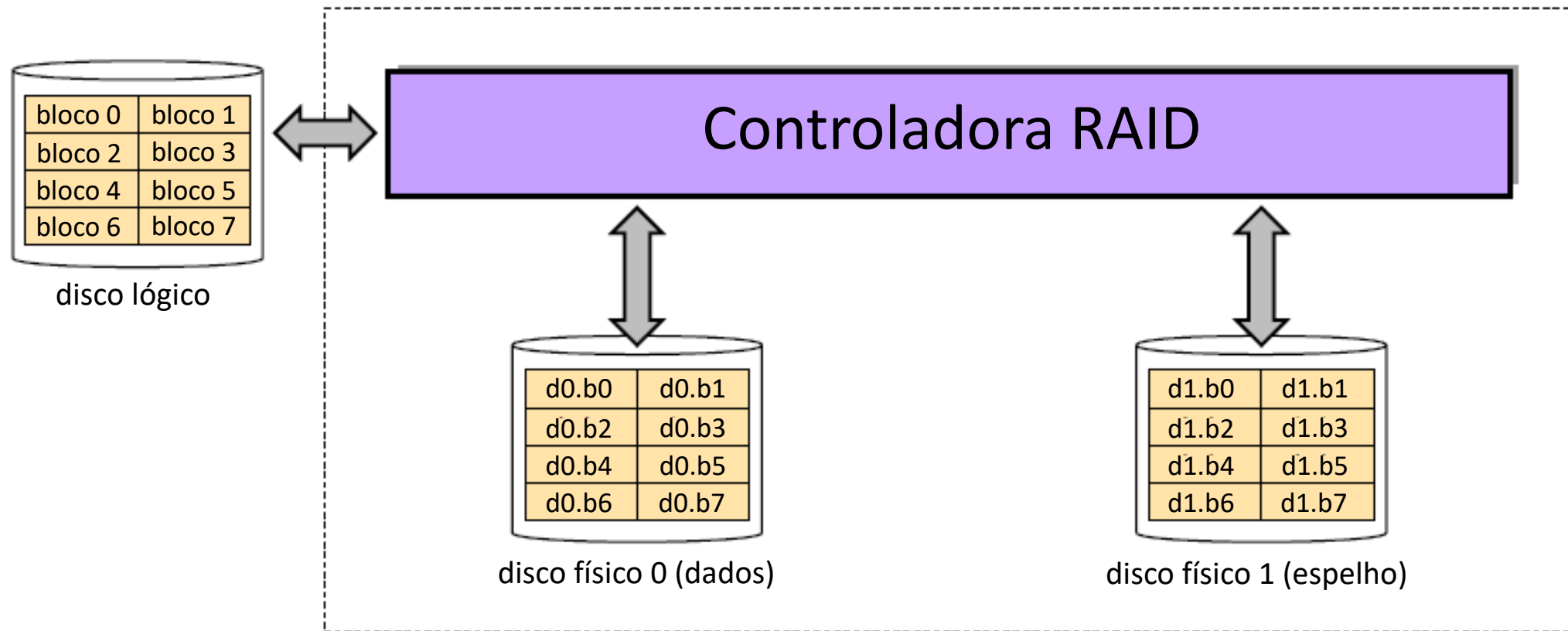
- Estratégia: concatena discos em faixas de blocos.
- Desempenho mais equilibrado entre os discos.



Fonte: adaptado de: Maziero (2019, p. 271).

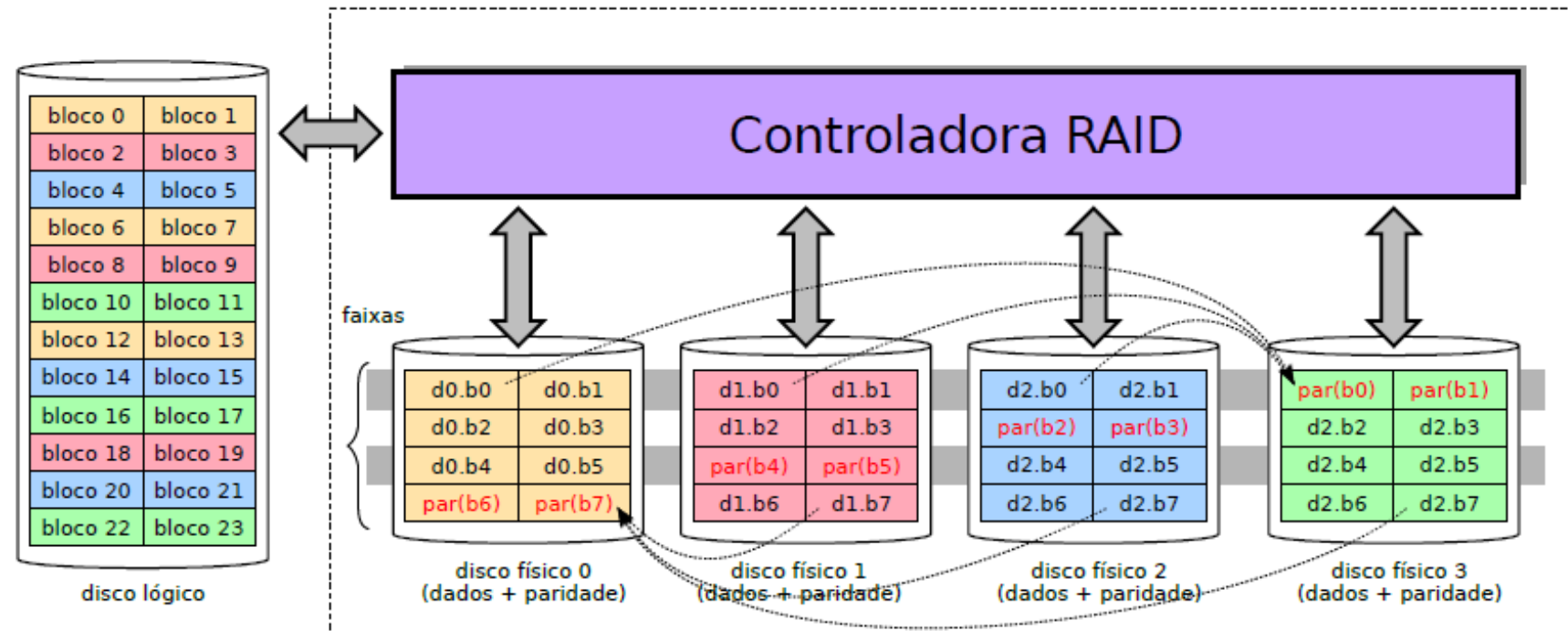
RAID Nível 1 – Espelhamento

- Estratégia: espelhamento (cópias do disco).
- Boa velocidade, tolera falhas de disco, mas tem alto custo.



RAID Nível 5 – Paridade distribuída

- Estratégia: blocos de paridade espalhados nos discos.
- Mais velocidade com tolerância a falhas e baixo custo.



Fonte: Maziero (2019, p. 273).

Combinação de RAID

- É impossível combinar diferentes níveis de RAID.
- Nível 0+1: combinação dos níveis 0 (*striping*) e 1 (*mirroring*), em que 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.

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* desse 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.



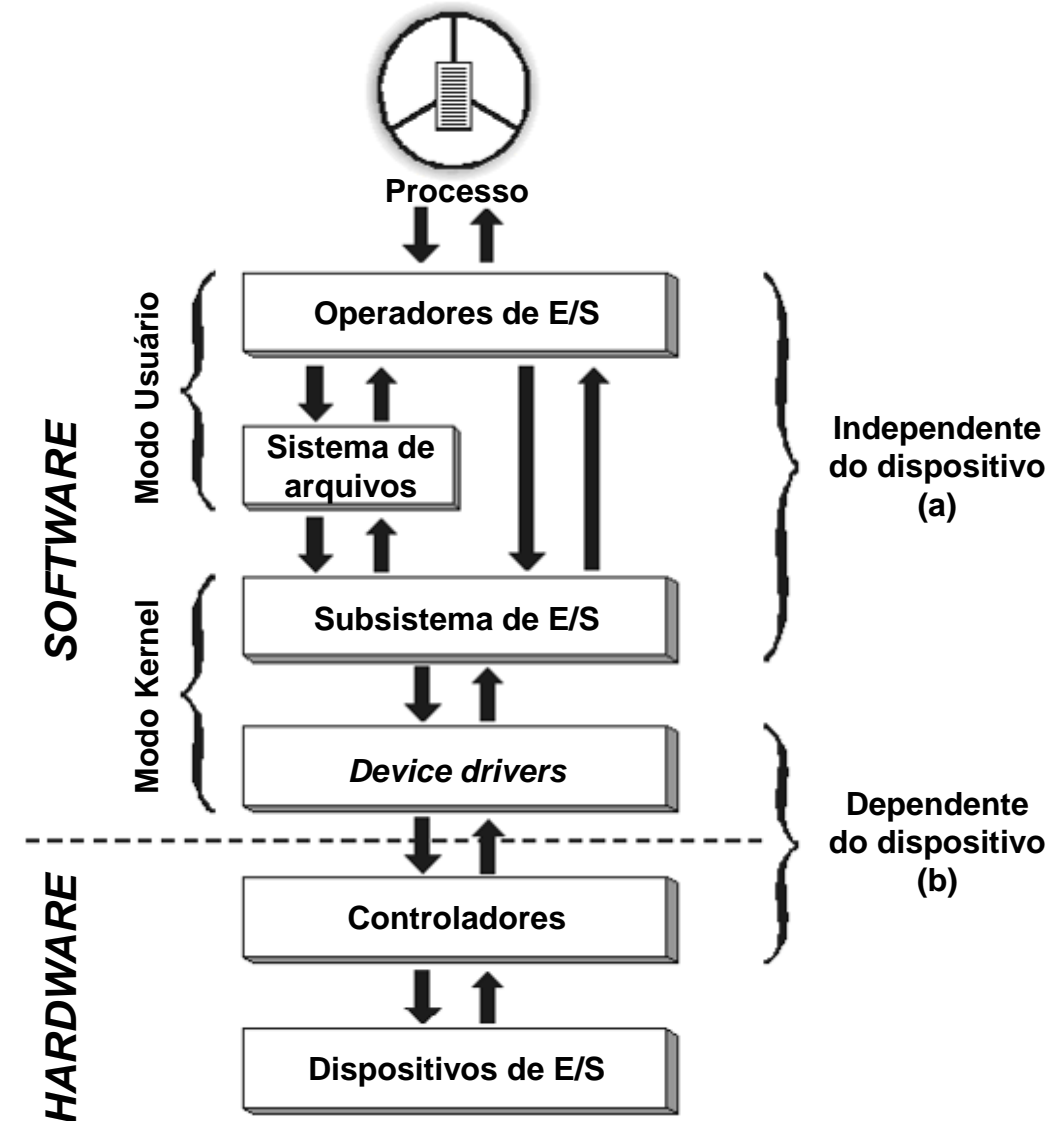
Software de E/S

Software que interage com os dispositivos:

- *Drivers* (ou pilotos).
- Abstrações de baixo nível (*sockets*, blocos etc.).

Grande diversidade de dispositivos:

- Muitos dispositivos distintos = muitos *drivers*.
- 60% do código-fonte do núcleo Linux são *drivers*.

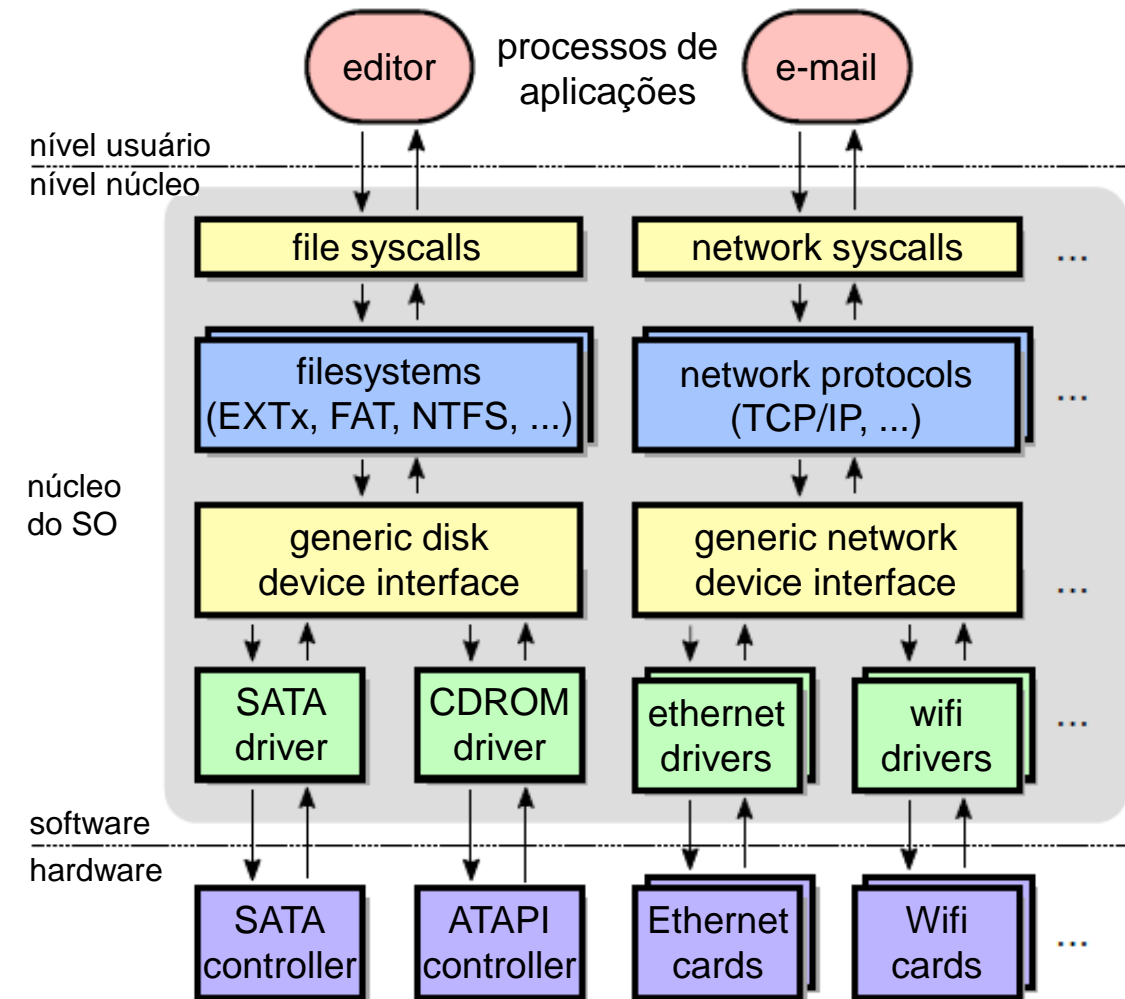


Estrutura em camada do *software* de E/S

- Código do sistema operacional é estruturado em camadas, que levam da interação direta com o *hardware*.

1ª camada de *software* no núcleo do SO:
drivers de dispositivos.

- Código que interage diretamente com cada controlador.
- Acessa portas de E/S e trata interrupções.
- Gerencia o dispositivo.



Estrutura em camada do *software* de E/S

Dispositivos genéricos (*Generic Device Interface*):

- Visão genérica de dispositivos similares (discos).
- Discos: vetores de blocos de dados.
- Interfaces de rede.
- Abstrações: implementar abstrações mais complexas como sistemas de arquivos e protocolos de rede.
 - Chamadas de sistema: interface oferecida aos processos fornecidos às aplicações para acessar as abstrações construídas pelas camadas inferiores, como arquivos, diretórios, *sockets* de rede etc.

Estrutura geral do núcleo

Drivers:

- Interagem com os dispositivos.
- Acessam portas de E/S e tratam interrupções.

Dispositivos genéricos:

- Visão genérica de dispositivos similares (discos).
- Discos: vetores de blocos de dados.
- Interfaces de rede.
- Abstrações: sistemas de arquivos e protocolos de rede.
- Chamadas de sistema: interface oferecida aos processos.

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.

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.



Tipos de dispositivos de E/S

Dispositivos de rede

- Blocos de dados de tamanho variável (mensagens).
- Envios de blocos de forma sequencial.
- NDIS – *Network Device Interface Specification*.
- Ex.: *Interfaces Ethernet, Bluetooth*.

Dispositivos gráficos

- Renderização de texto e gráficos em uma tela.
- Usam uma área de RAM compartilhada (*frame buffer*).
- Acesso por bibliotecas específicas (DirectX, DRI).

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.

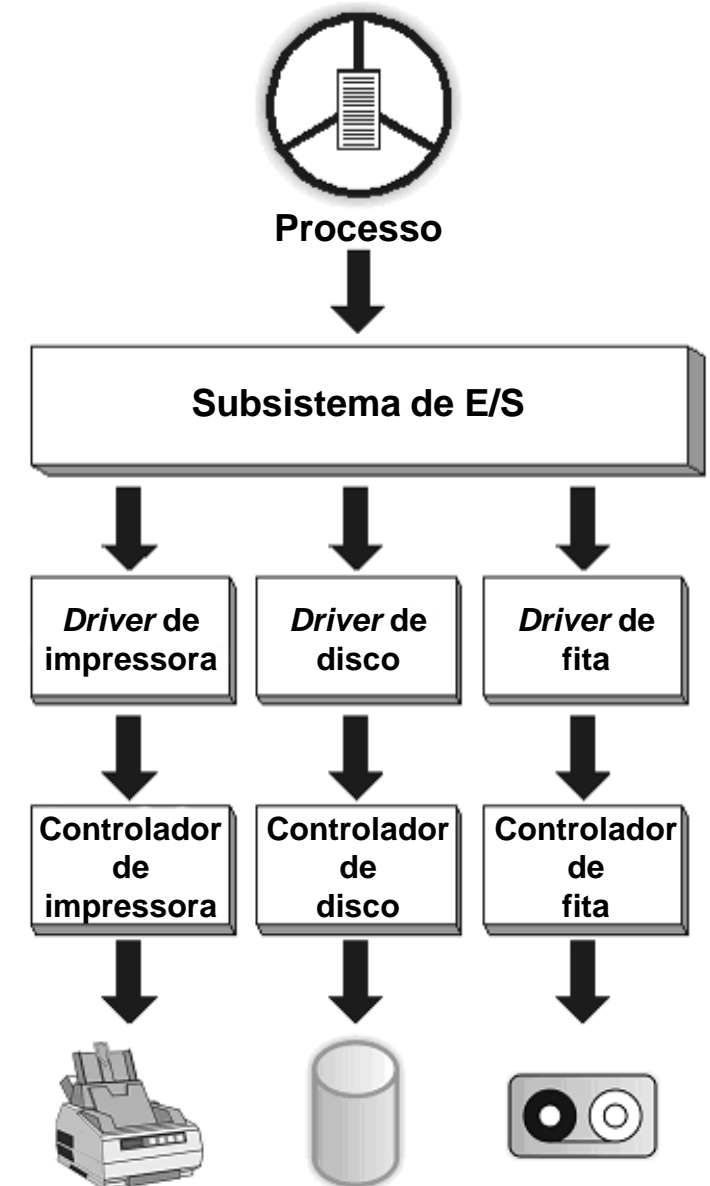
Resposta

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.

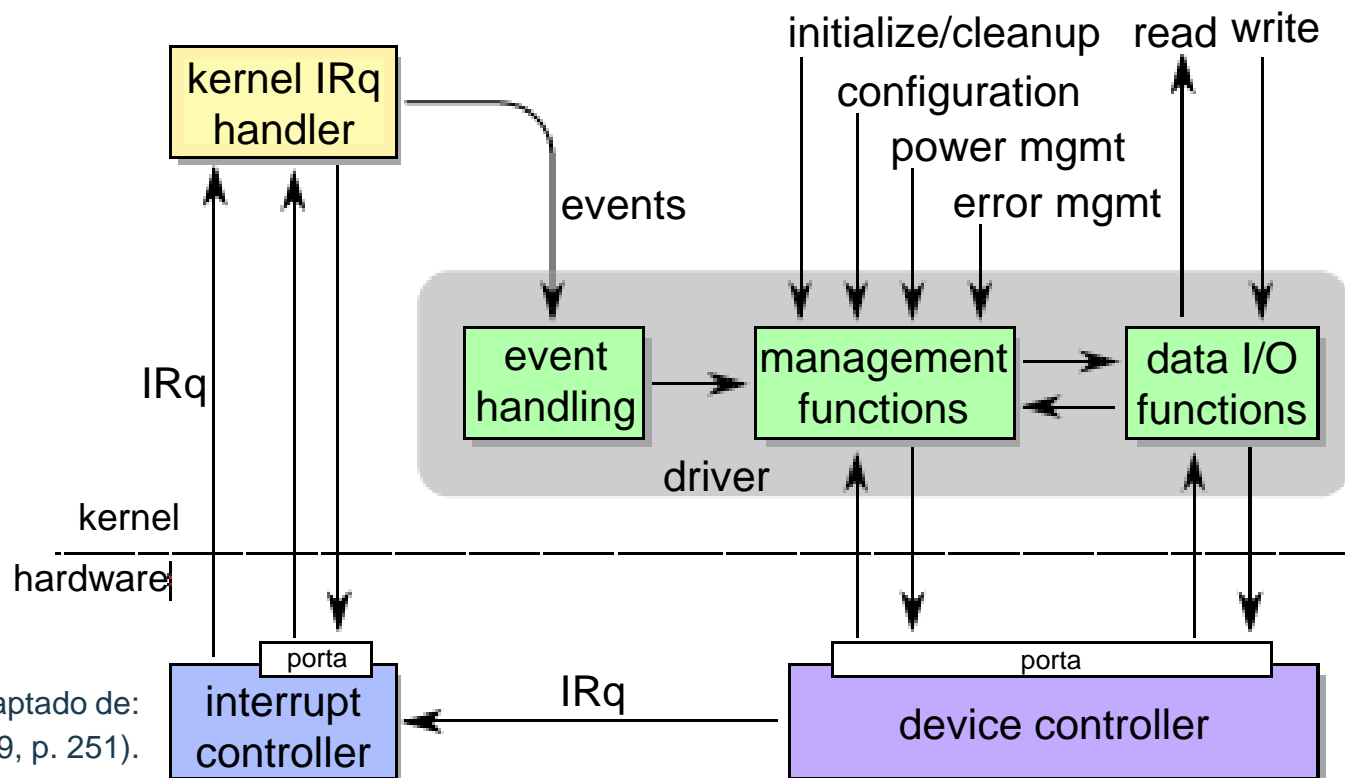
Device drivers

- 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.
- Componente de baixo nível do SO:
 - Executa geralmente em modo núcleo.
 - Interage com o *hardware* do dispositivo.
 - Um *driver* para cada tipo de dispositivo.
- Funcionalidades: entrada/saída, gerência e tratamento de eventos.



Estrutura de um *driver*

- Funções de entrada/saída:
 - Transferência de dados de dispositivo núcleo.
 - Caracteres, blocos, mensagens.
- Funções de gerência:
 - Inicialização e configuração do dispositivo.
 - Inicialização e configuração do *driver*.
 - Syscalls: `ioctl()`, `DeviceIoControl()`.
- Funções de tratamento de eventos:
 - Interrupções geradas pelo dispositivo.



Fonte: adaptado de:
Mazeiro (2019, p. 251).

Estratégia de interação

- Como o núcleo interage com os dispositivos?
- Por meio dos *drivers*!

Os *drivers* implementam estratégias de interação:

- Por programa (ou por *polling*).
- Por eventos (ou por interrupções).
- Por acesso direto à memória (DMA).

Polling ou por *program*

- Estratégia de entrada/saída mais simples.
- Também chamada varredura ou *polling*.

O *driver* pede a operação e aguarda sua conclusão:

Etapas

1. Espera o dispositivo estar pronto (*status*).
2. Escreve dado na porta de saída (*data out*).
3. Escreve comando (*control*).
4. Espera dispositivo concluir a operação (*status*).

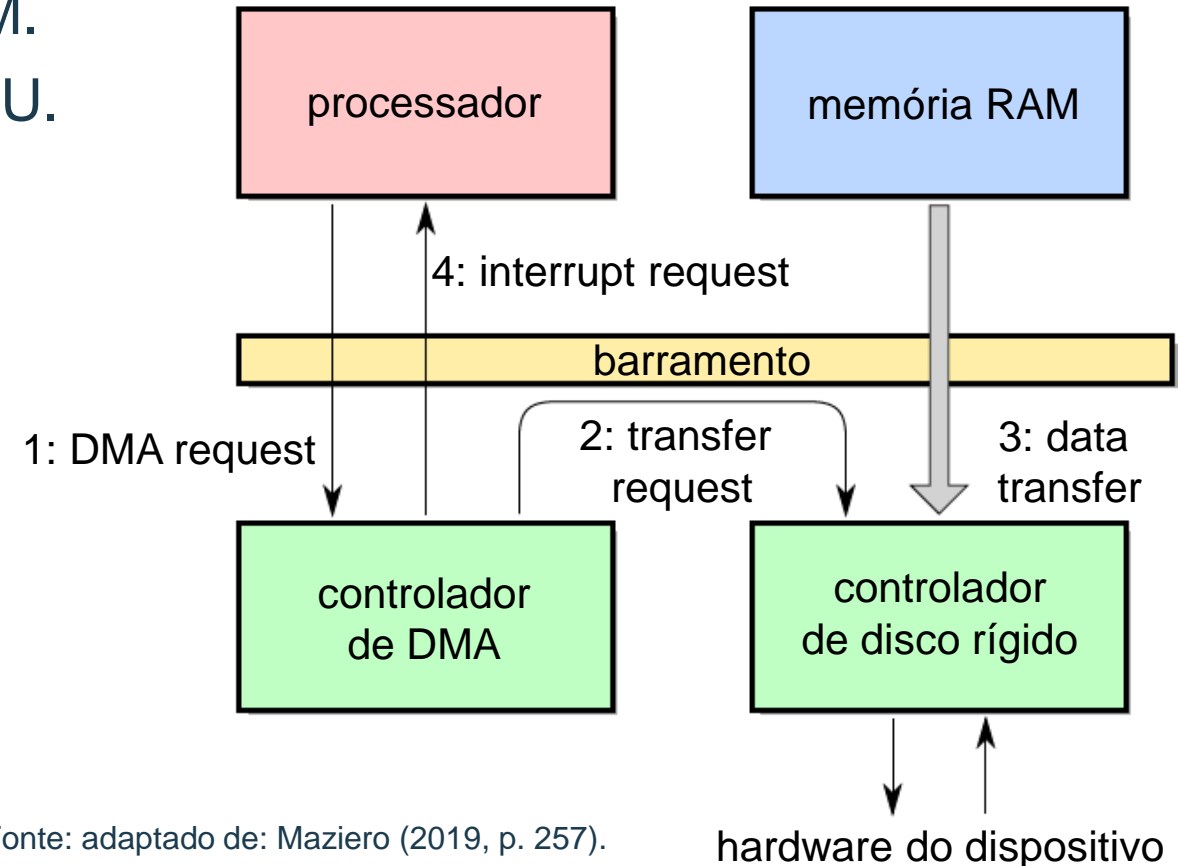
Acesso direto à memória (DMA)

- Permite transferência direta entre a memória principal e os controladores de E/S.
- Os dados não precisam passar pela CPU (desempenho).
- Muito usado para transferir grandes volumes de dados.
- Processador acessa as portas do controlador de DMA associado ao dispositivo.
- Controlador de DMA solicita ao controlador do disco a transferência de dados da RAM para o disco e aguarda a conclusão da operação.
- Controlador do disco rígido recebe os dados da memória (pode ser feita repetidas vezes).
- Controlador de DMA notifica o processador sobre a conclusão da operação, usando uma requisição de interrupção (IRq).

Acesso direto à memória

Passos:

- A CPU programa o controlador DMA com os parâmetros (endereços e tamanho da transferência).
- O controlador de DMA interage com o controlador do disco para transferir os dados da RAM.
- O controlador do disco recebe os dados da RAM.
- No final, o controlador de DMA interrompe a CPU.

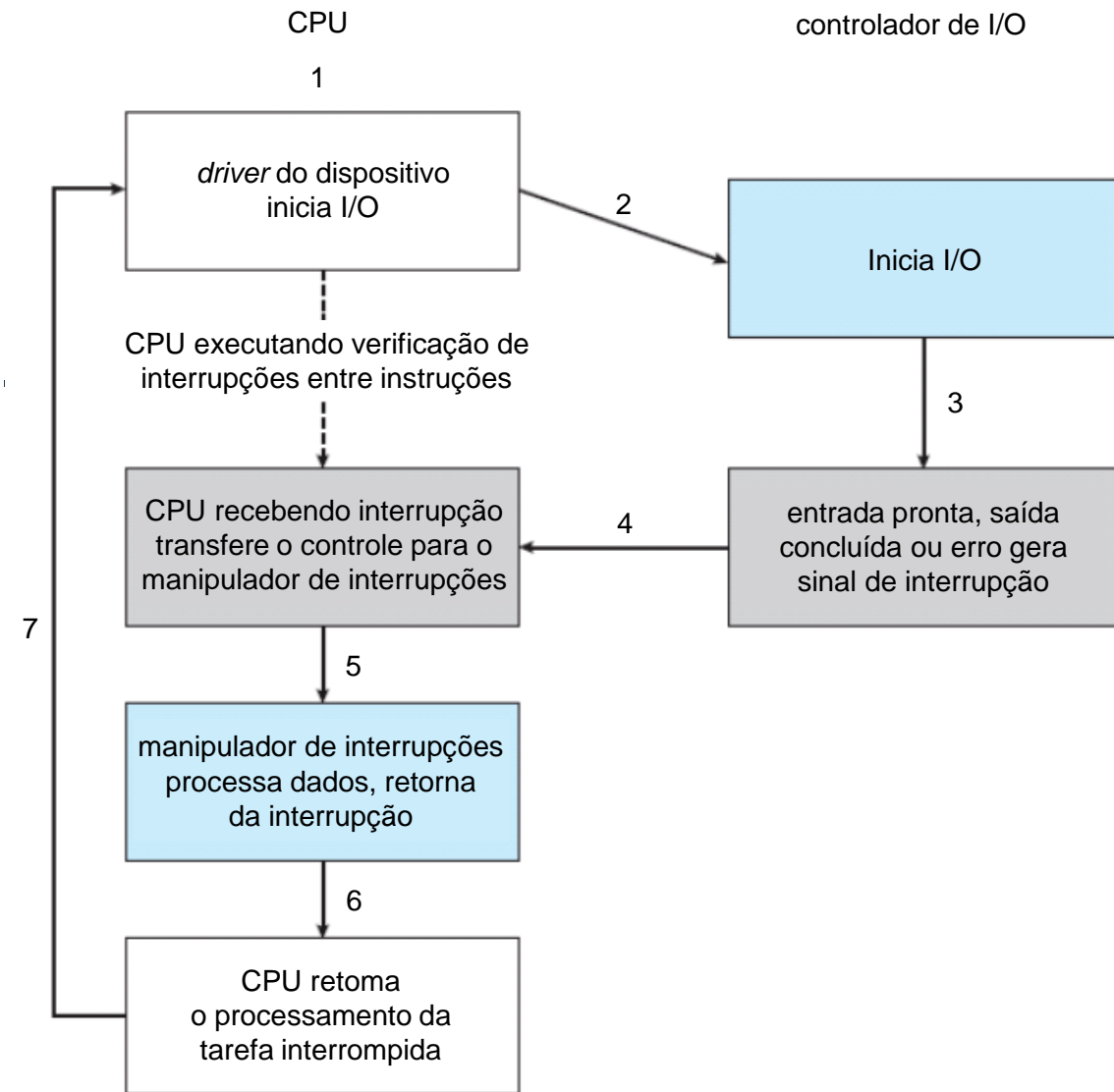


Fonte: adaptado de: Maziero (2019, p. 257).

Por eventos ou por interrupções

Interrupções são tratadas por *handlers* (tratadores):

- Compõem a estrutura dos *drivers*.
- Operam usualmente com interrupções inibidas.
- Devem ser muito rápidas, portanto, simples.
- Ciclo de interrupção de E/S apresentado na figura.



Por eventos ou por interrupções

Estratégia básica:

1. Requisitar a operação desejada.
2. Suspender o fluxo de execução (tarefa atual).
3. O dispositivo gera uma IRq ao concluir.
4. O *driver* retoma a operação de E/S.

A operação de E/S pelo *driver* é dividida em duas etapas:

1. Uma função de E/S inicia a operação.
2. Uma função de tratamento de evento a continua.

Interatividade

Qual o nome da interação entre os dispositivos de E/S que se baseia no tratamento de eventos previamente programados?

- a) Varredura.
- b) Interrupção.
- c) Acesso direto à memória.
- d) Chamadas de sistema.
- e) *Driver*.

Resposta

Qual o nome da interação entre os dispositivos de E/S que se baseia no tratamento de eventos previamente programados?

- a) Varredura.
- b) **Interrupção.**
- c) Acesso direto à memória.
- d) Chamadas de sistema.
- e) *Driver*.

Referências

- CÓRDOVA JÚNIOR, R. S.; LEDUR, C. L.; MORAES, I. S. *Sistemas operacionais*. Porto Alegre: Grupo A, 2019.
- FURUKAWA, F.; NUNES, R. *Fundamentos de Sistemas Operacionais*. São Paulo: Editora Sol, 2011.
- 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.

Referências

- OLIVEIRA, R. S.; CARISSIMI, A. S.; TOSCANI, S. S. *Sistemas operacionais* - V.11 (Livros didáticos informática UFRGS). 4. ed. Porto Alegre: Bookman, 2010.
- TANENBAUM, S. A.; WOODHULL, S. *Sistemas operacionais: projetos e implementação – o livro do Minix*. 3. ed. Porto Alegre: Bookman, 2008.
- TANENBAUM, A. S.; BOS, H. *Sistemas Operacionais Modernos*. 4. ed. São Paulo: Pearson, 2016.

ATÉ A PRÓXIMA!