

СИСТЕМА ВХОДА И ВЫХОДА

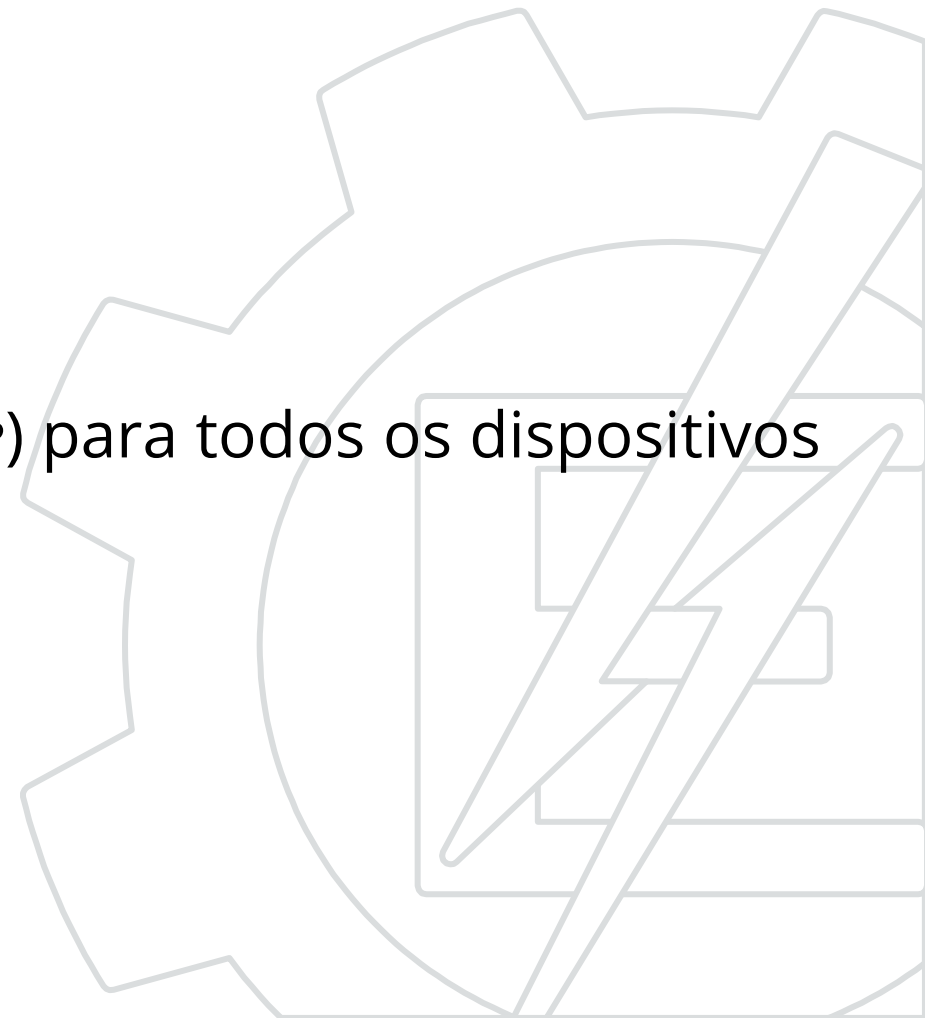
Sistema de Entrada e Saída



Sistema de Entrada e Saída

No Sistema Operacional

- SO controla todos os dispositivos de E/S, possuindo comandos para:
 - Emitir instruções (*read, write, etc.*);
 - Interceptar interrupções;
 - Tratar erros.
- Interface única (API – *Application Program Interface*) para todos os dispositivos devido a(o):
 - Diferenças na construção dos dispositivos;
 - Tempo de processamento dos dispositivos;
 - Tempo de acesso aos dispositivos.

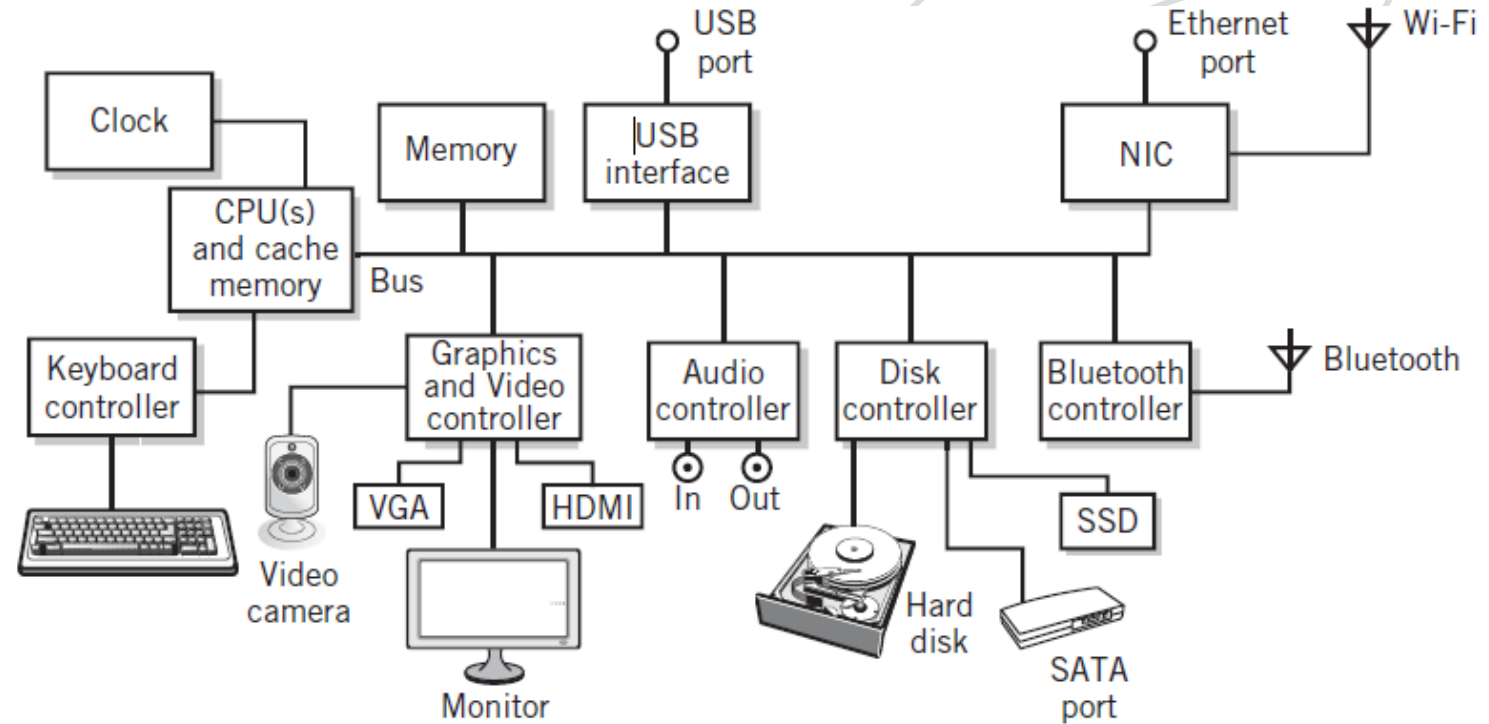


Sistema de Entrada e Saída

Tipos de E/S

Podem ser classificados de acordo com os seguintes tipos:

- De Conexão;
- De Transferência de Dados;
- De Compartilhamento de Conexões.



1) Tipo de Conexão

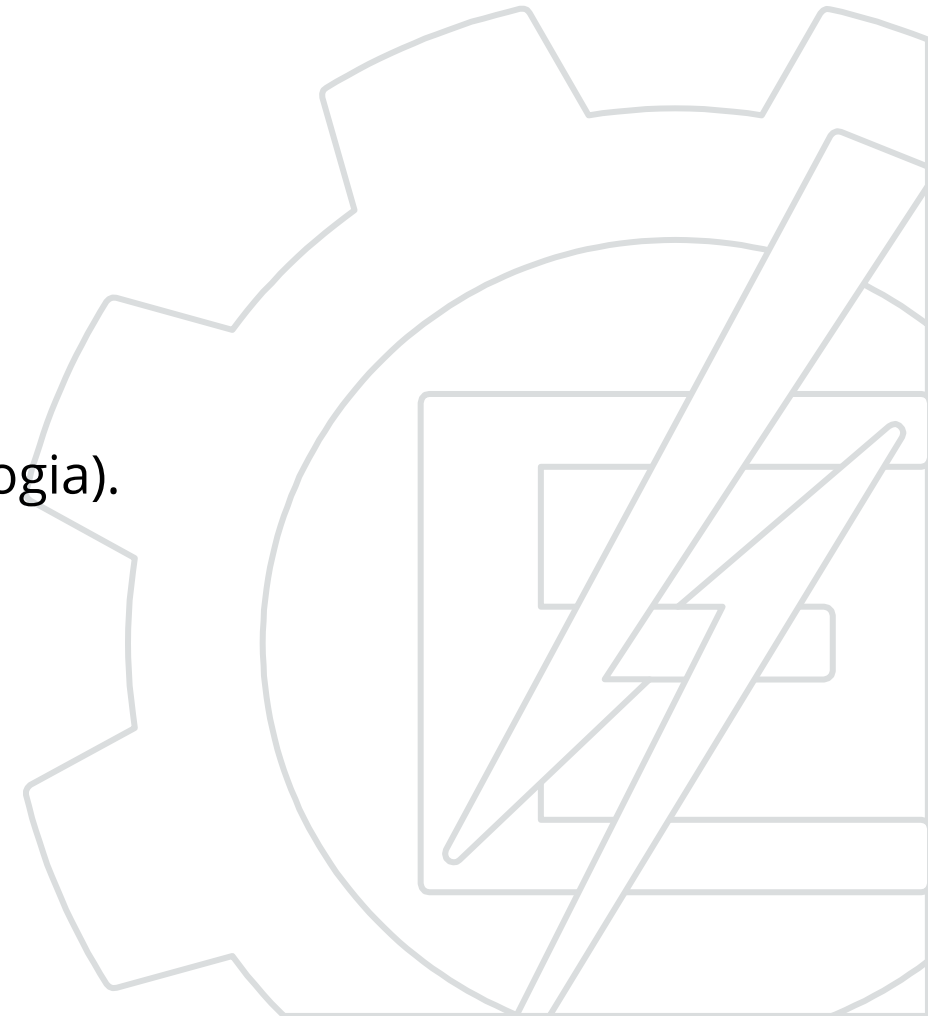
- Natureza da conexão entre módulo de E/S e periférico
- Serial *versus* Paralela

a) **Serial** – única linha de conexão.

- Vantagens: mais barata que a paralela
- Desvantagens: mais lenta que a paralela (contexto / tecnologia).

b) **Paralela** – múltiplas linhas de conexão.

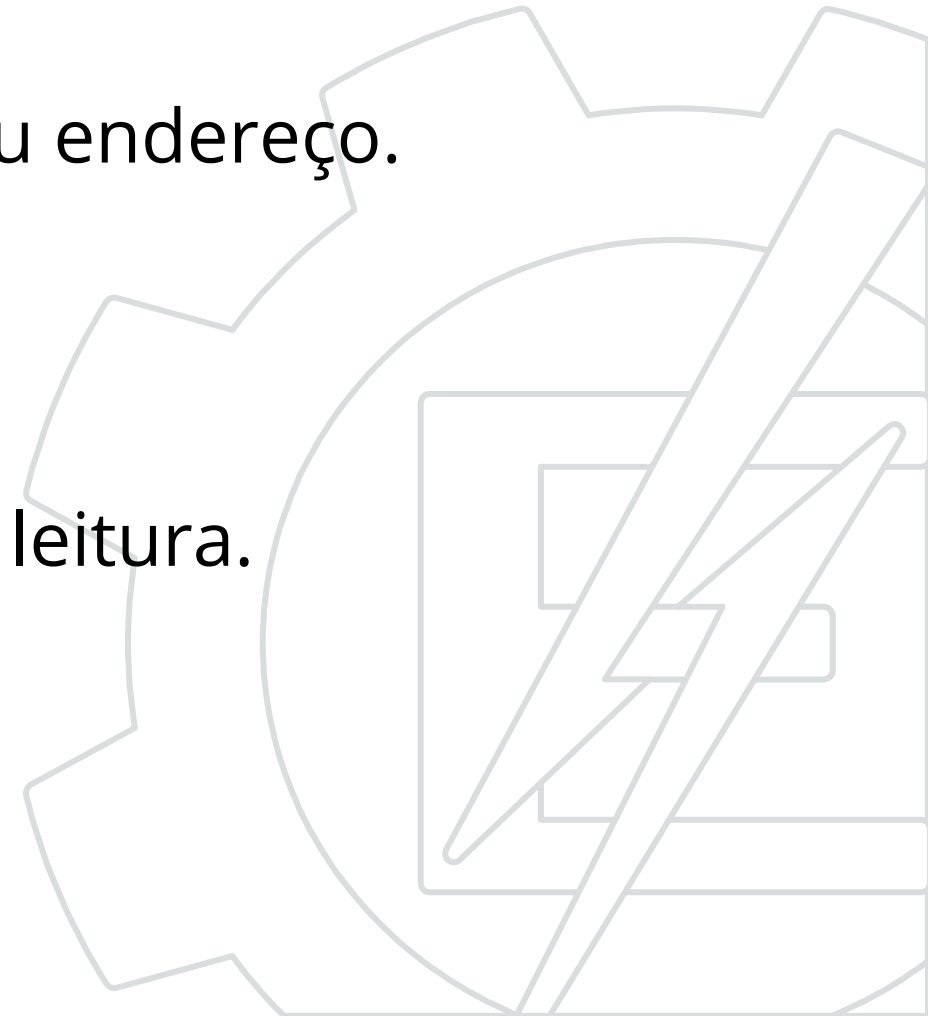
- Vantagens: mais rápida que a serial (contexto / tecnologia).
- Desvantagens: mais cara que a serial (barramentos).



2) Tipo de Transferência de E/S

a) Dispositivos de **bloco** (*block devices*):

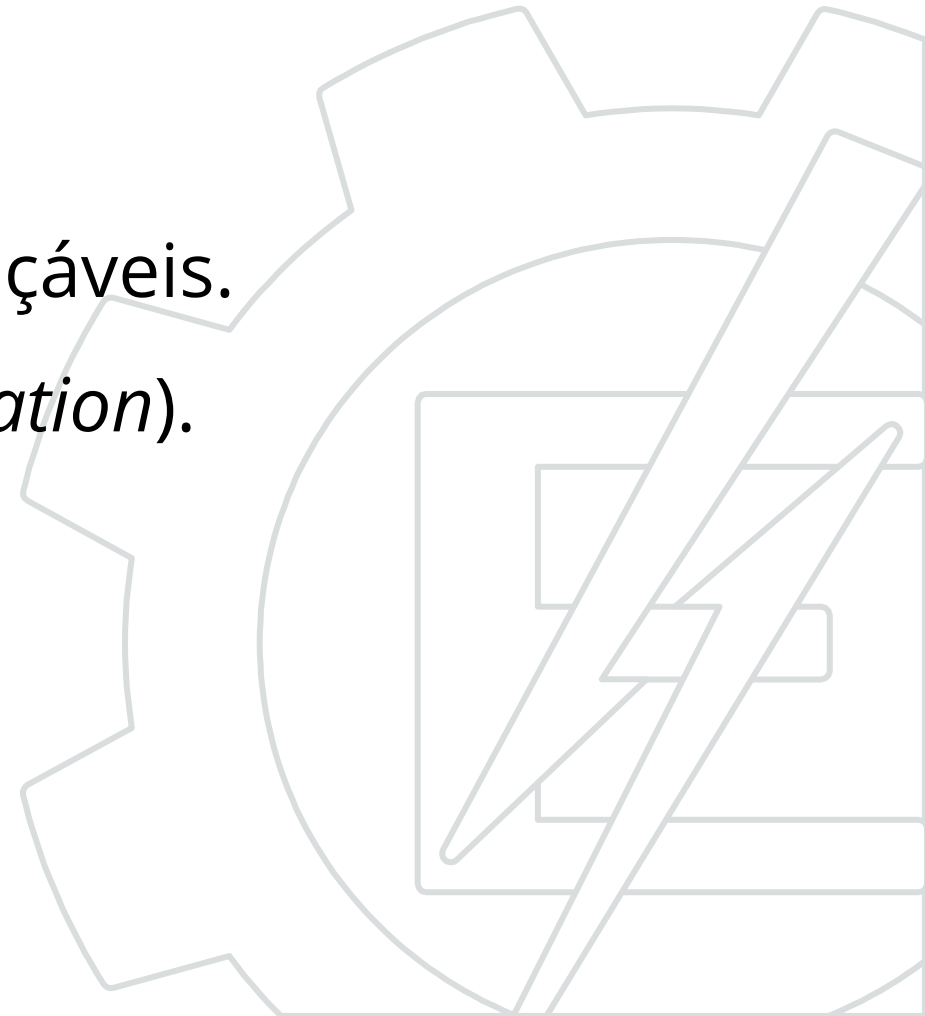
- Blocos de tamanho fixo, cada um com seu endereço.
- Tamanho de 128 a 1024 bytes.
- Transferência com um ou mais blocos.
- Referência de localidade – Otimização da leitura.
- E/S consome tempo.
- Exemplos: HD, CD-ROM, Drive USB, etc.



2) Tipo de Transferência de E/S

b) Dispositivos de **caracter**:

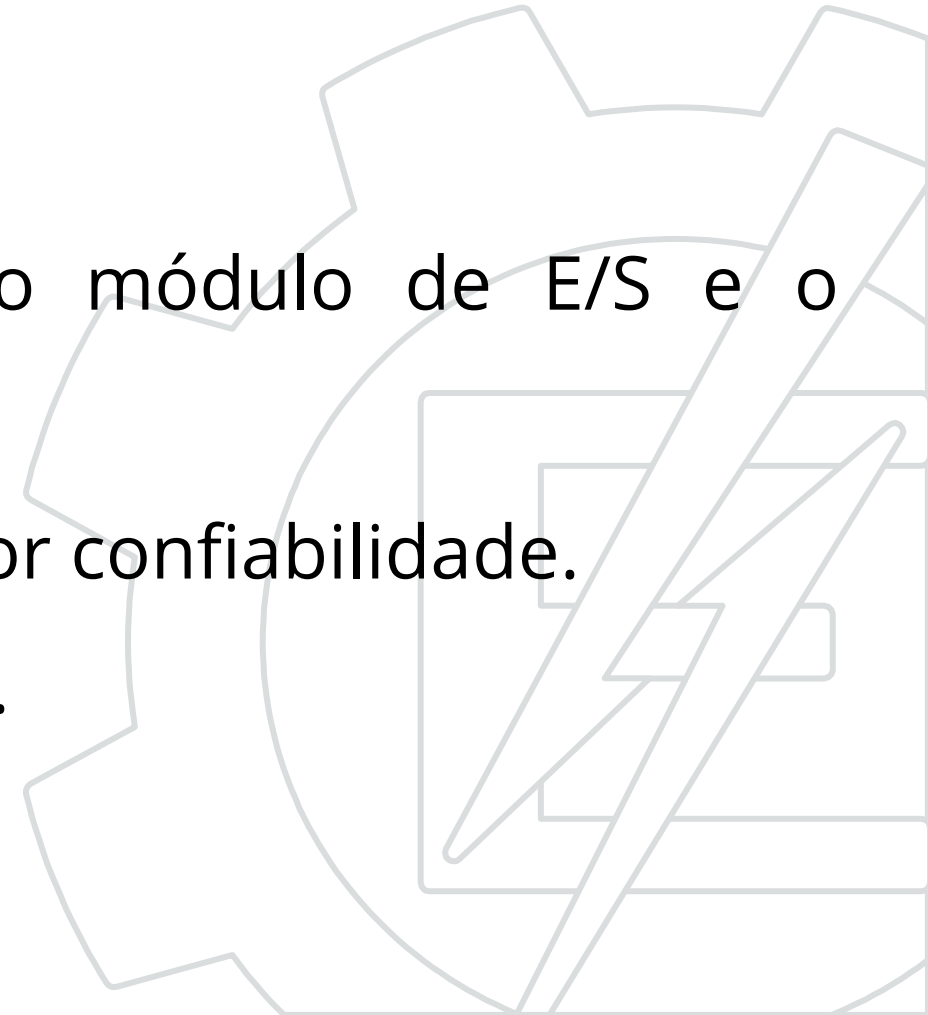
- Acessam um fluxo de caracteres.
- Não consideram blocos e não são endereçáveis.
- Não possuem acesso aleatório (*seek operation*).
- Exemplo: interfaces de rede, mouse, etc.



3) Tipo de Compartilhamento de Conexões

a) Ponto a ponto

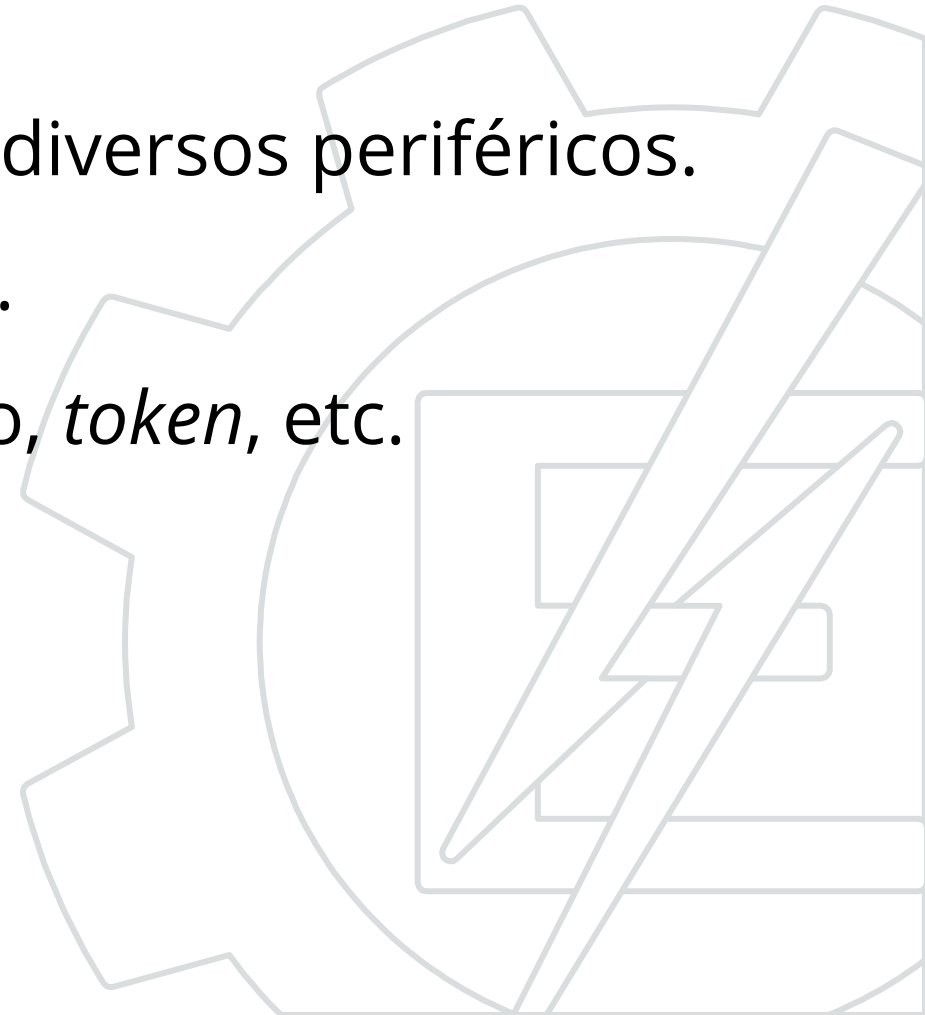
- Conexão mais simples
- Linha dedicada para a ligação entre o módulo de E/S e o periférico.
- Oferece maior grau de paralelismo e maior confiabilidade.
- Protocolo RTS/CTS (*Request/Clear To Send*).



3) Tipo de Compartilhamento de Conexões

b) Multiponto

- Compartilha um conjunto de linhas entre diversos periféricos.
- Maior escalabilidade que a ponto a ponto.
- Não permite paralelismo – Escalonamento, *token*, etc.
- Usada para armazenamento.
- Exemplos: IDE, SCSI, USB, etc.



Sistema de Entrada/Saída

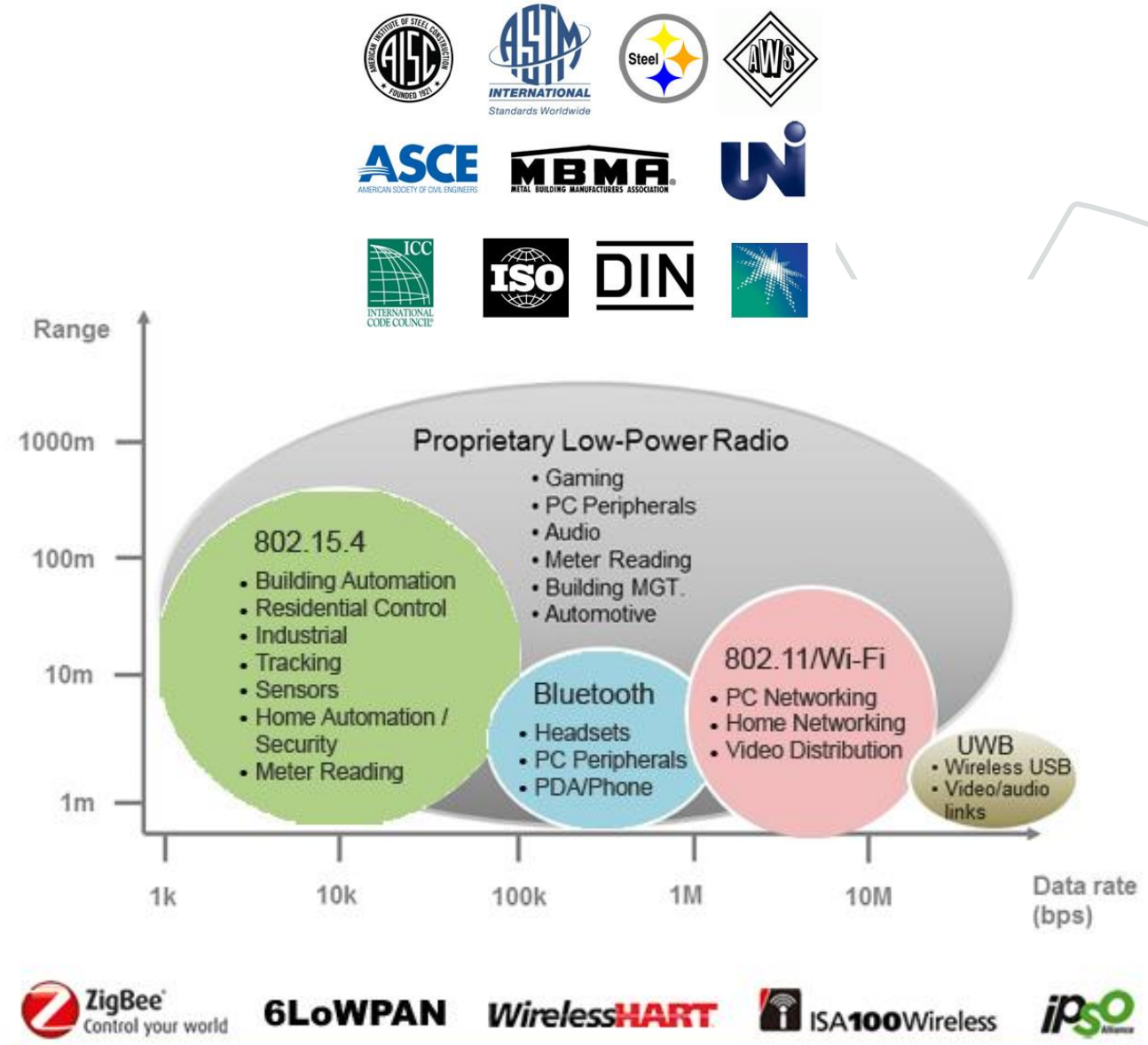
- Princípios de *Hardware*



Sistema de Entrada e Saída

Princípios de *Hardware*

- As unidades de E/S
 - **Componente mecânico** – o dispositivo.
 - **Controladora** de dispositivo:
 - Componente eletrônico – Parte programável.
 - Órgãos de padronização: IEEE, ISO, ANSI, etc.

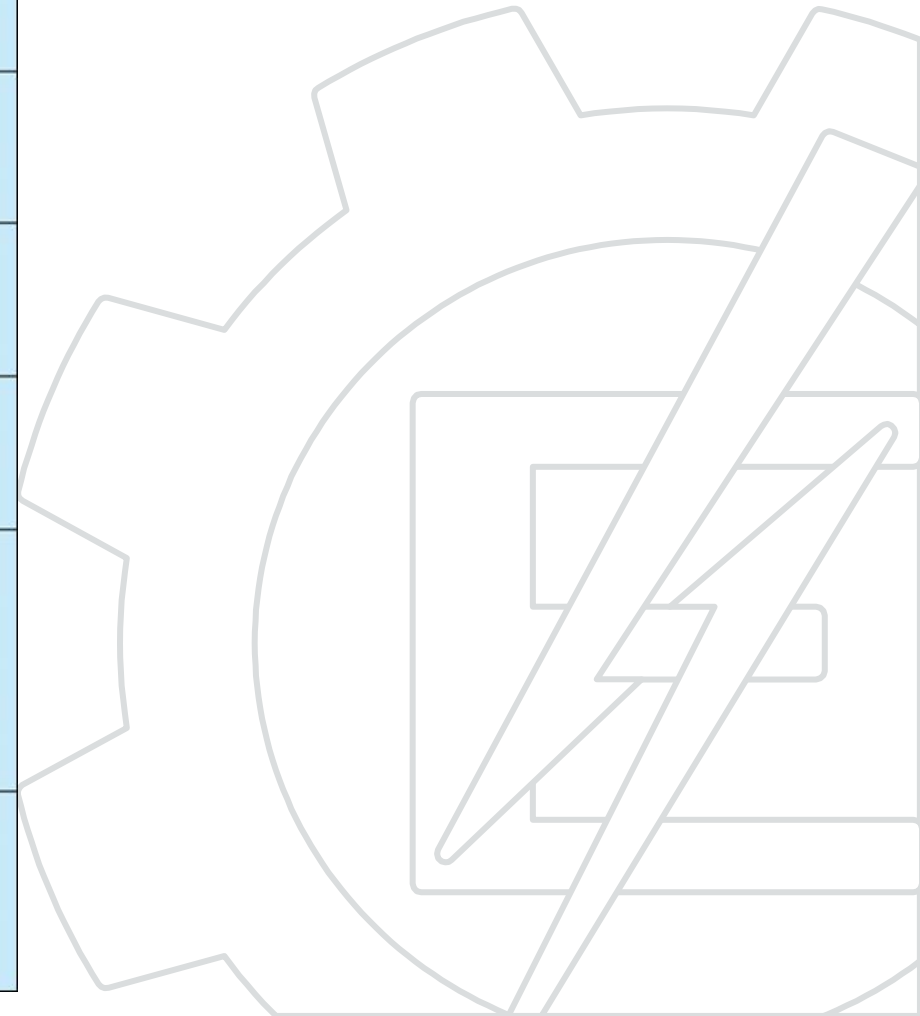


Sistema de Entrada e Saída

Input type	Prime examples	Other examples	Data rate (b/s)	Main uses
Symbol	Keyboard, keypad	Music note, OCR	10s	Ubiquitous
Position	Mouse, touchpad	Stick, wheel, glove	100s	Ubiquitous
Identity	Barcode reader	Badge, fingerprint	100s	Sales, security
Sensory	Touch, motion, light	Scent, brain signal	100s	Control, security
Audio	Microphone	Phone, radio, tape	1000s	Ubiquitous
Image	Scanner, camera	Graphic tablet	1000s-10 ⁶ s	Photos, publishing
Video	Camcorder, DVD	VCR, TV cable	1000s-10 ⁹ s	Entertainment
Output type	Prime examples	Other examples	Data rate (b/s)	Main uses
Symbol	LCD line segments	LED, status light	10s	Ubiquitous
Position	Stepper motor	Robotic motion	100s	Ubiquitous
Warning	Buzzer, bell, siren	Flashing light	A few	Safety, security
Sensory	Braille text	Scent, brain stimulus	100s	Personal assistance
Audio	Speaker, audiotape	Voice synthesizer	1000s	Ubiquitous
Image	Monitor, printer	Plotter, microfilm	1000s	Ubiquitous
Video	Monitor, TV screen	Film/video recorder	1000s-10 ⁹ s	Entertainment
Two-way I/O	Prime examples	Other examples	Data rate (b/s)	Main uses
Mass storage	Hard/floppy disk	CD, tape, archive	10 ⁶ s	Ubiquitous
Network	Modem, fax, LAN	Cable, DSL, ATM	1000s-10 ⁹ s	Ubiquitous

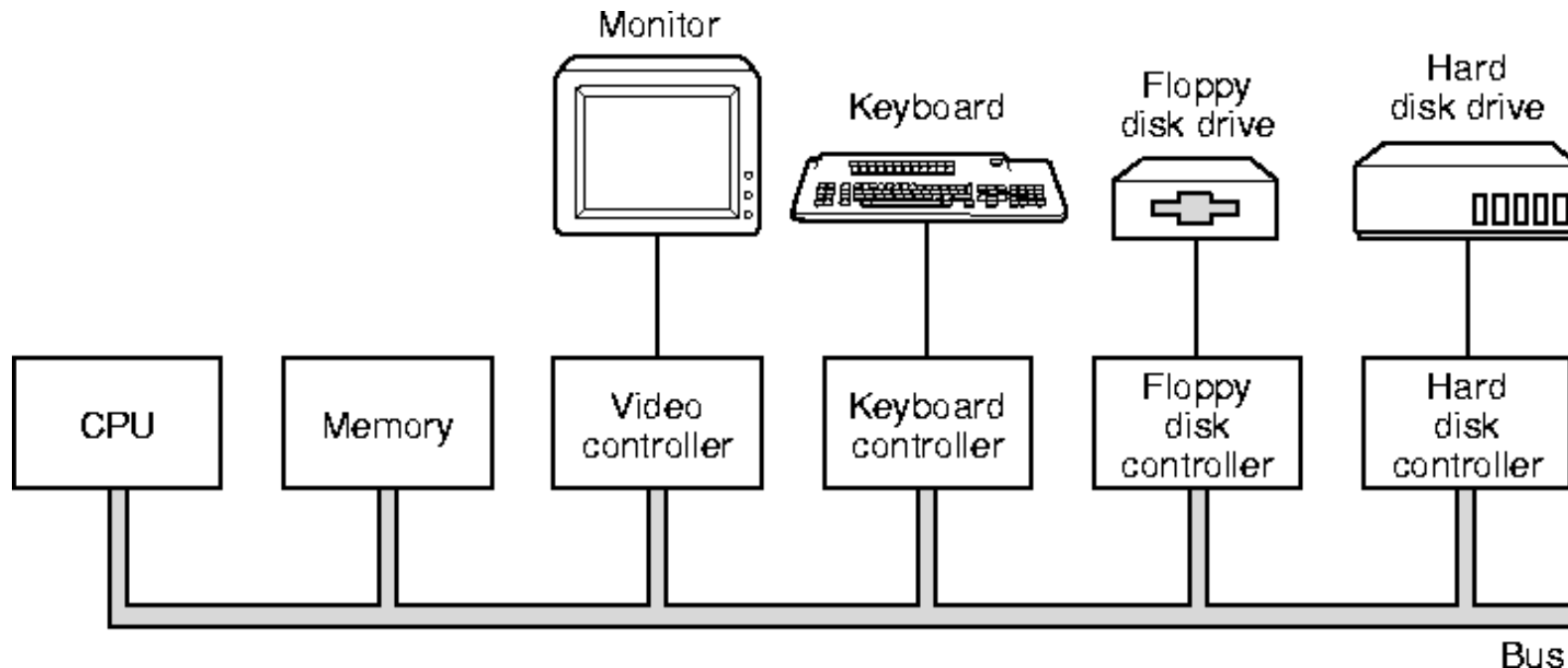
Princípios de *Hardware*

aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read-write	CD-ROM graphics controller disk



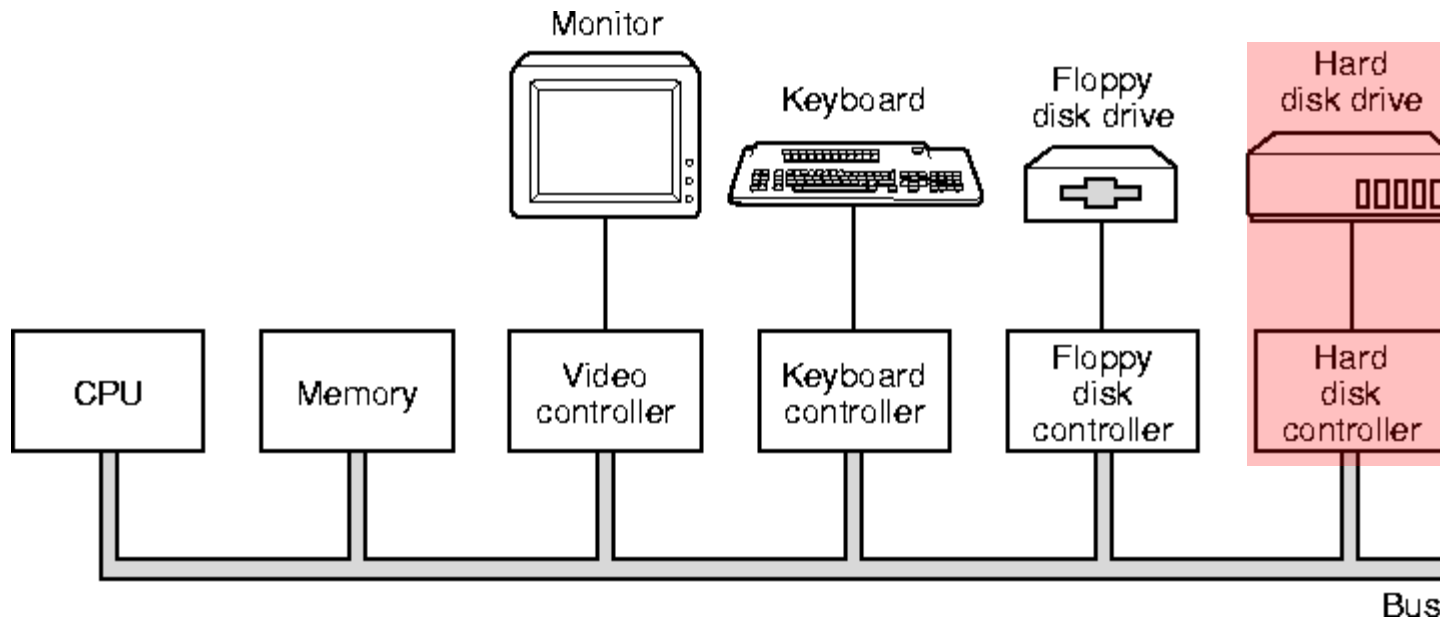
Princípios de *Hardware*

- O SO trata com a controladora – Não lida com os dispositivos.
- Comunicação CPU e Controladoras:
 - Interface de alto nível – utiliza os barramentos comuns.
 - Interface de baixo nível – entre controladora e dispositivo.



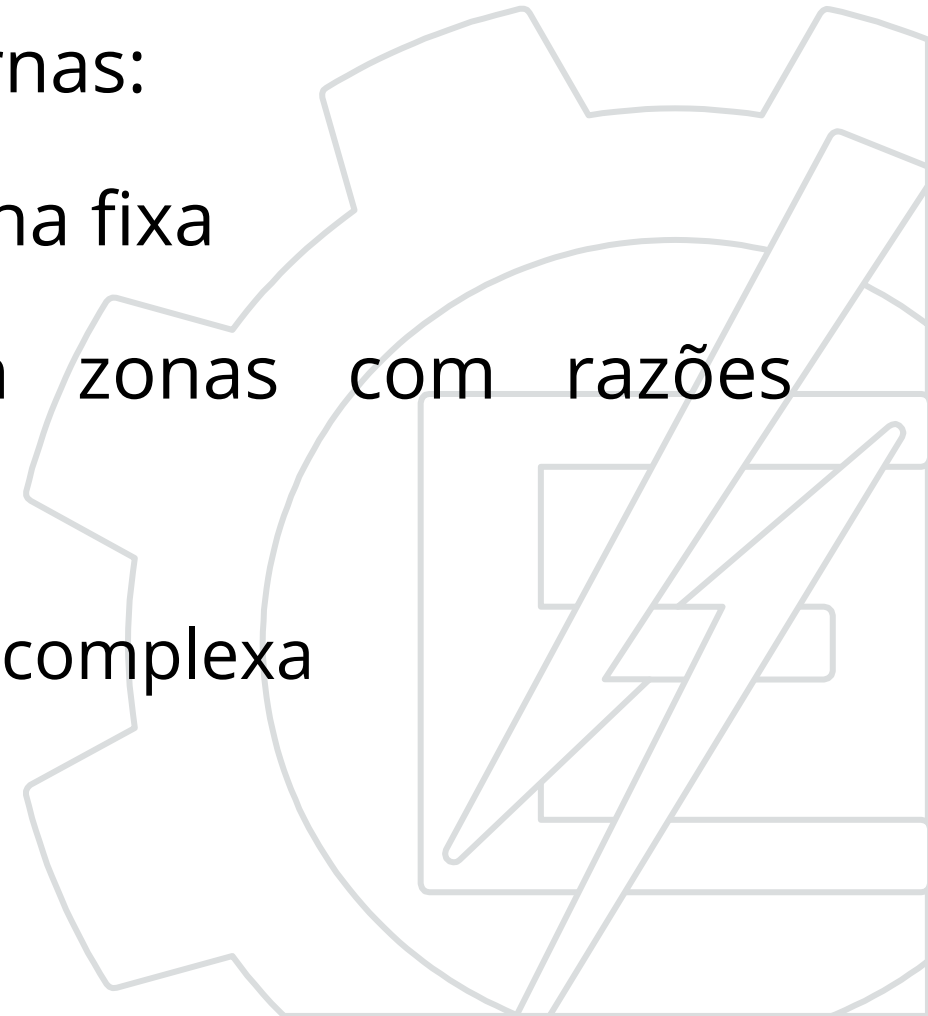
Princípios de *Hardware*

- O SO trata com a controladora – Não lida com os dispositivos.
- Comunicação CPU e Controladoras:
 - Interface de alto nível – utiliza os barramentos comuns.
 - Interface de baixo nível – entre controladora e dispositivo.



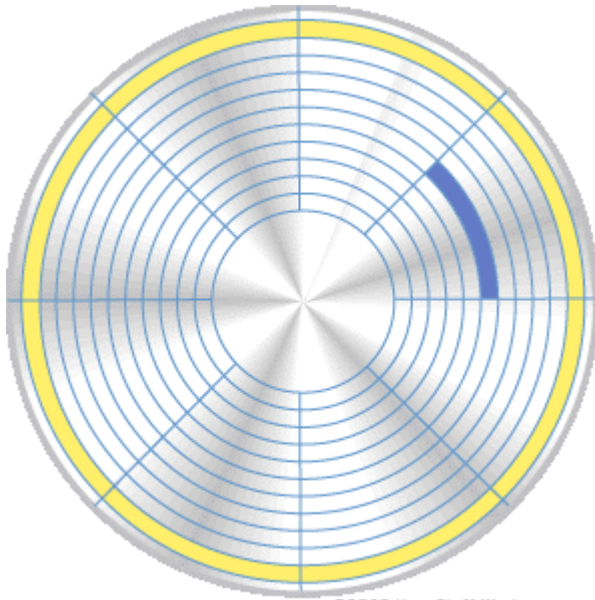
À Æ Ð È Ë Ä Å Æ Ð Ë Æ Z Æ Ð Æ Æ

- Distância linear aumenta nas trilhas externas:
- Discos antigos tinham razão #setores/trilha fixa
- Discos modernos separam disco em zonas com razões diferentes
 - Identificação de localização no disco mais complexa



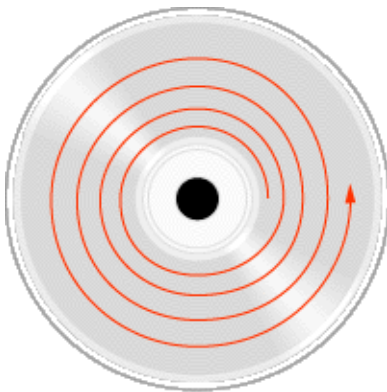
Sistema de Entrada e Saída

Dispositivo de armazenamento



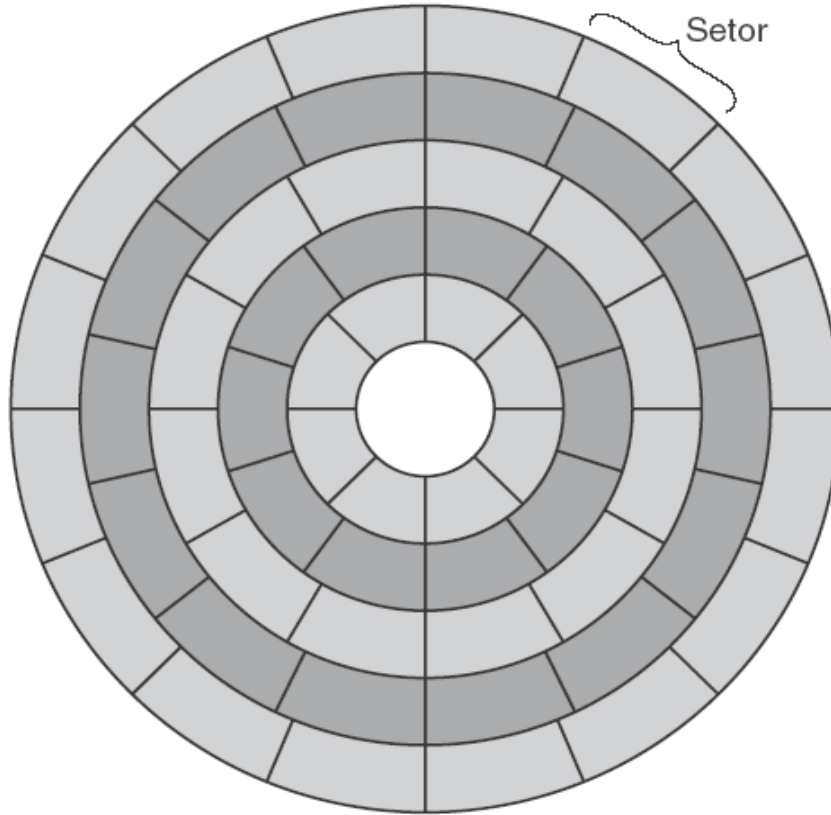
©2000 How Stuff Works

Um disco com uma só zona, cada trilha com oito setores.



©2000 How Stuff Works

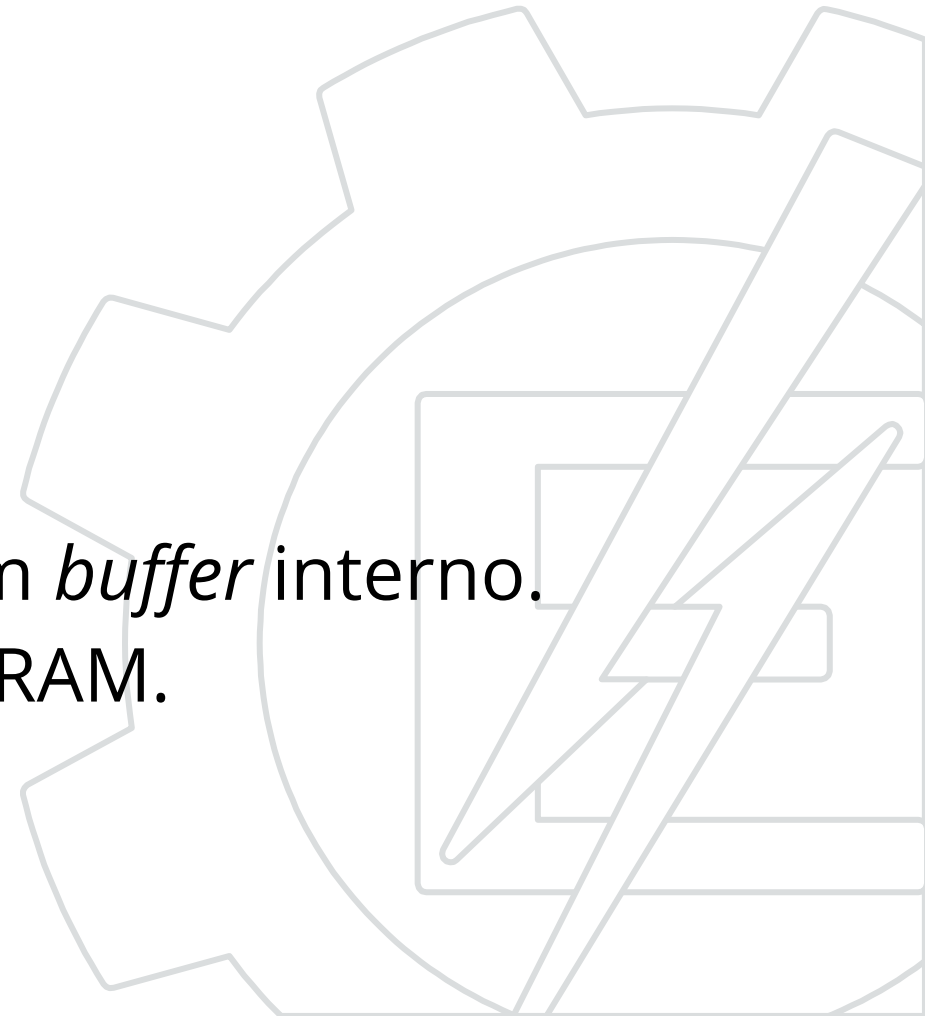
Trilhas em um CD.



Um disco com cinco zonas.
Cada zona tem muitas trilhas.



- Recebe um fluxo de bits com:
 - Preâmbulo
 - Bits do setor
 - *Checksum* (ECC).
- Dispositivo:
 - Monta os bits em bloco e coloca-os em um *buffer* interno.
 - Após verificar *checksum*, copia o bloco na RAM.

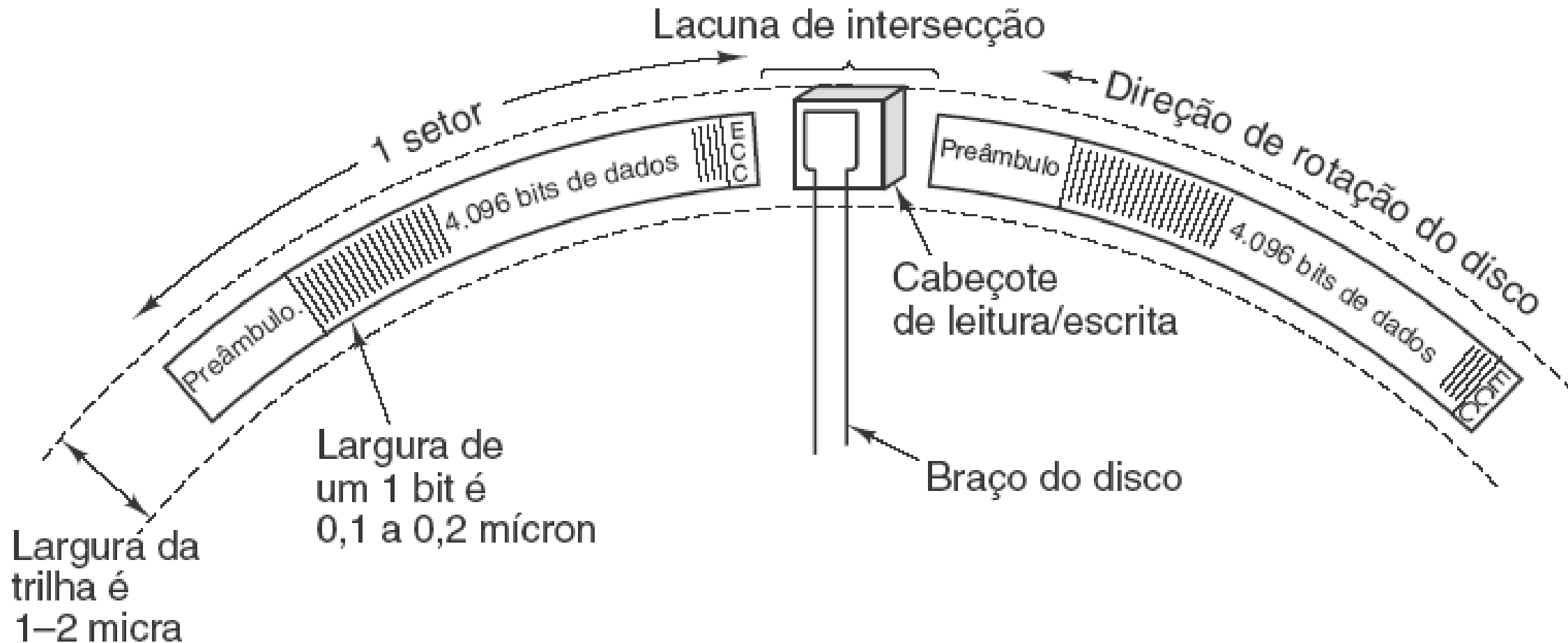


Sistema de Entrada e Saída

Dispositivo de armazenamento

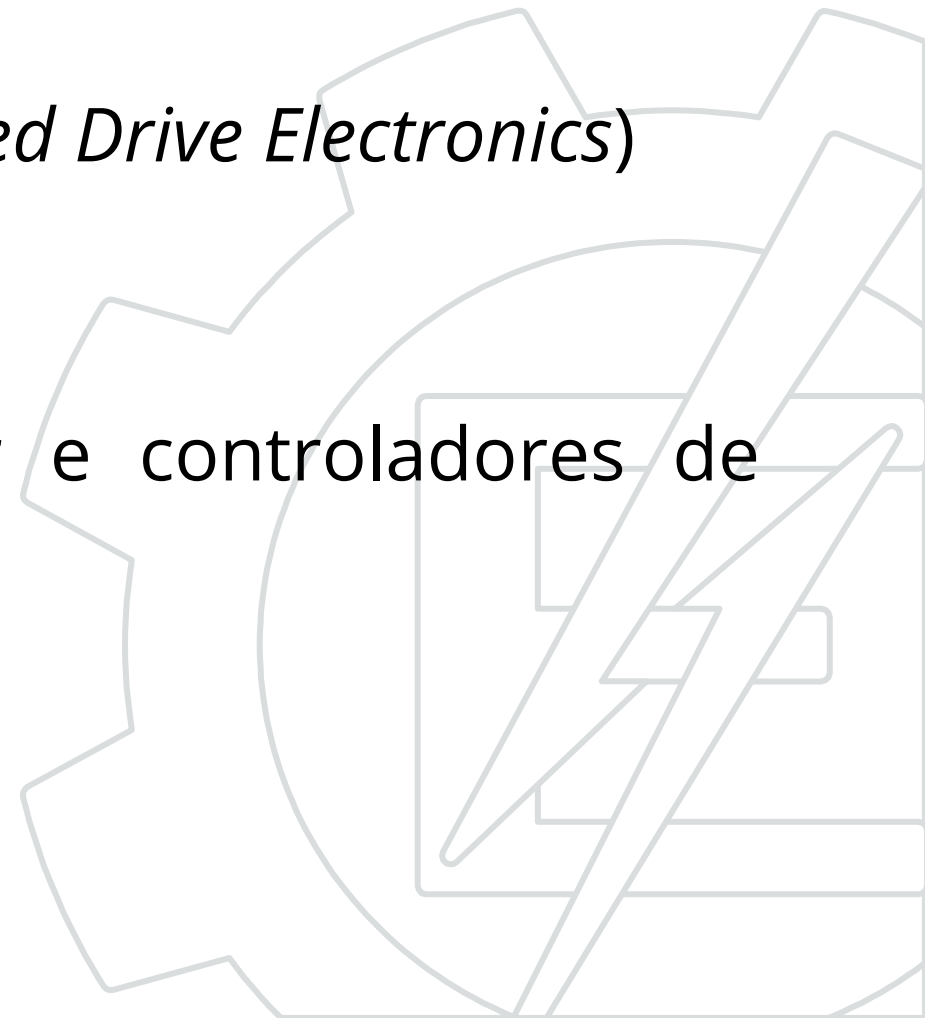
Winchester

- Discos *Winchester*: alusão à numeração dos primeiros discos IBM





- Interface baseada na BIOS IBM PC XT
- Controlador junto ao dispositivo (*Integrated Drive Electronics*)
- Barramento ATA (*PC/AT Attachment*)
- Ainda usa convenções da BIOS (PC XT)
- Barramento simples entre processador e controladores de disco
- Custo e desempenho reduzidos



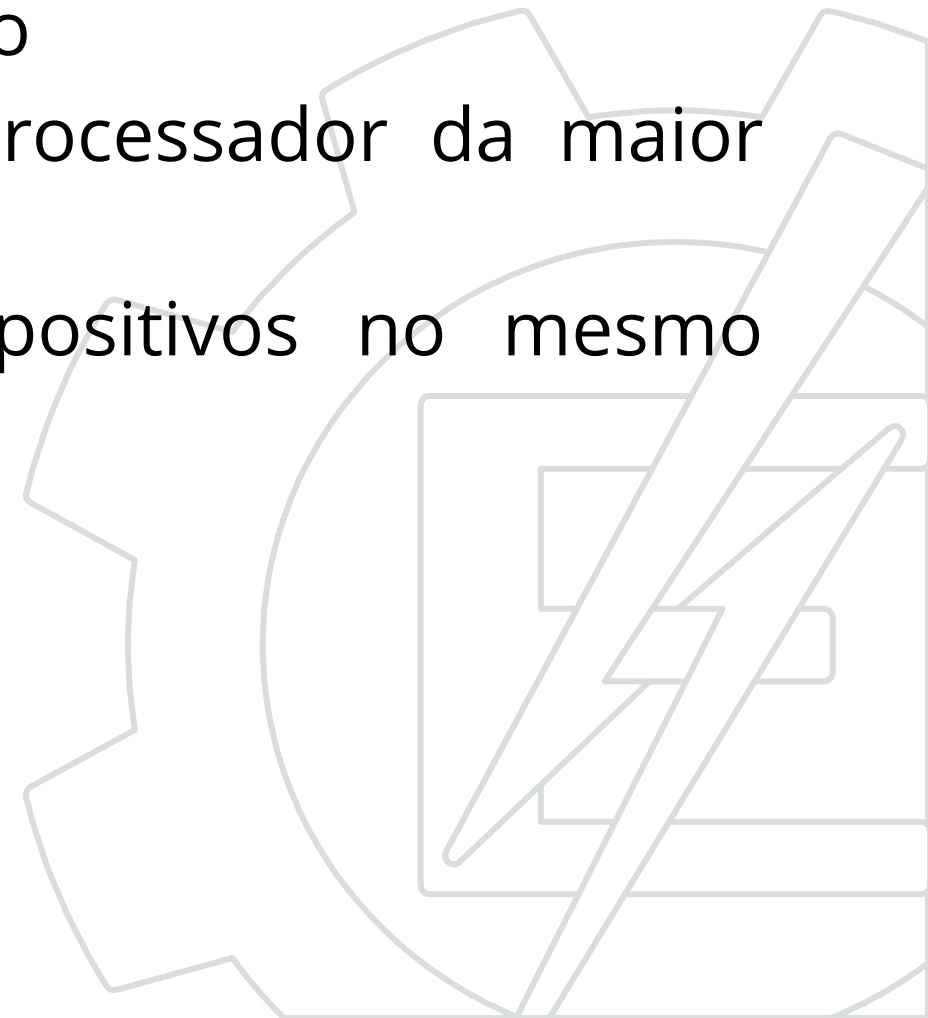
ATA

- Nova geração do barramento ATA (serial)
- Interface mais simples
- Controladores mais poderosos
- Maior velocidade de transferência



ÀĖĐĚĚĂĆÂ

- Barramento padrão de alto desempenho
- Controladores complexos liberam o processador da maior parte do acesso
- Permitem a operação de vários dispositivos no mesmo barramento
- SCSI-1: 8 b, 5 MHz, 5 MB/s
- Wide Ultra2 SCSI: 16 b, 40 MHz, 80 MB/s



Sistema de Entrada e Saída

Controladora de disco

- Padrão denominado **Next Generation Form Factor** ou **NGFF**, posteriormente denominado **M.2**. Comumente o associamos a SSDs de alto desempenho, mas sua aplicação é universal, sendo compatível também com placas Wifi, Bluetooth e NFC.
- Quando falamos sobre M.2 em relação à tecnologia de armazenamento, normalmente falamos sobre um **SSD** em relação ao seu formato.
 - M.2 se refere a um **formato** SSD que se parece com um chiclete. Seu tamanho pequeno e fino é ideal para computadores que são leves e portáteis como laptops, notebooks, NUCs e ultrabooks. Eles ocupam menos espaço do que SSDs de 2,5 pol ou discos rígidos e podem chegar a até 2 TB de capacidade.
 - **M.2 é um formato de SSD**, com base em **NVMe** e **SATA** eles diferem em tecnologia de armazenamento e dependendo de suas necessidades e orçamento, cada um deles tem seus prós e contras.

- ***Non-Volatile Memory Express (NVMe)*** é um protocolo de transferência projetado para memória de estado sólido.
- Embora o SATA (*Serial Advanced Technology Attachment*) continue sendo o padrão do setor para protocolos de armazenamento, ele não foi criado especificamente para armazenamento Flash como SSDs e não pode oferecer as mesmas vantagens que o NVMe.
- Eventualmente, SSDs com NVMe irão substituir os SSDs SATA como o novo padrão do setor.

Sistema de Entrada e Saída

Controladora de disco

- Mais velocidade na transferência (55 a 180 IOPS para HDDs contra 3.000 a 40.000 IOPS com SSDs).
- Taxa de transferência de dados mais alta.
- Conexões sem bloqueio que fornecem a cada núcleo de CPU acesso dedicado à fila para cada SSD.
- Paralelismo em massa com mais de 64 mil filas para operações de E/S.

	3,5"	2,5"
5900 RPMs	50 IOPS	-
7200 RPMs	75 IOPS	95 IOPS
10000 RPMs	110 IOPS	140 IOPS
15000 RPMs	150 IOPS	180 IOPS
SSD	1500 IOPS	1500 IOPS

IOPS = Input/Output Operations Per Second

Fontes: <https://www.kingston.com/br/community/articledetail/articleid/57715>

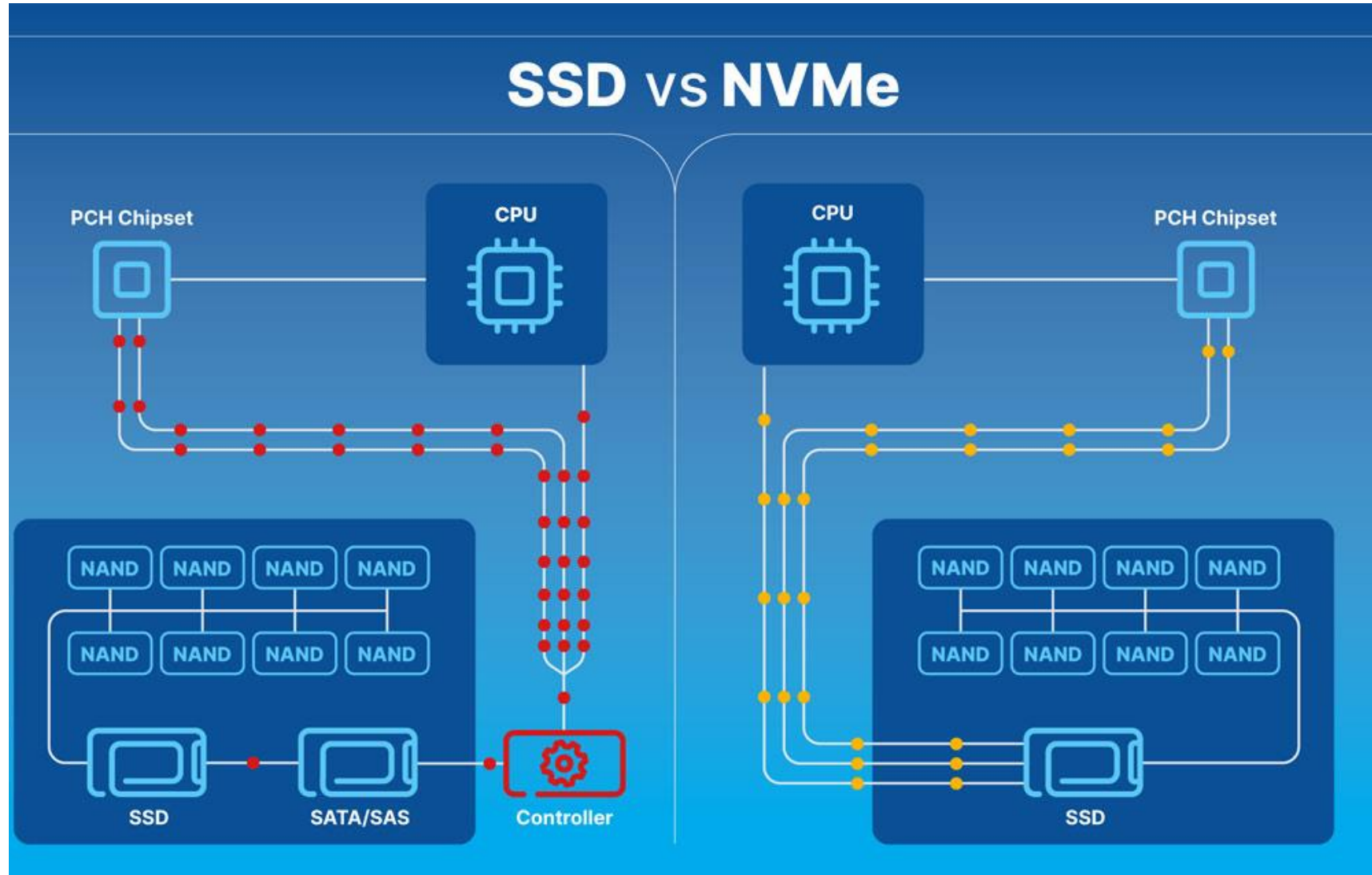


Sistema de Entrada e Saída

Controladora de disco

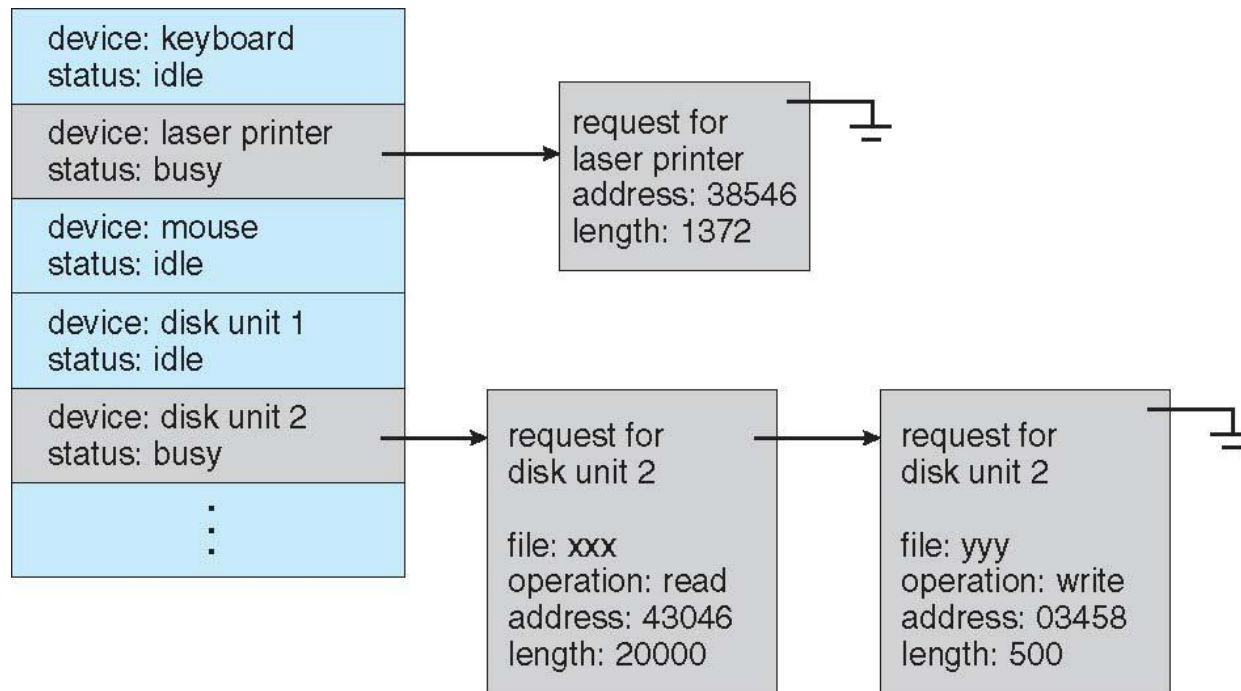
O que é armazenamento SAS?

O SAS (***Serial Attached SCSI***) é uma evolução do padrão SCSI. Em resumo, podemos definir o SAS como uma interface de comunicação ponto a ponto (p2p), que conecta dispositivos de armazenamento em massa a uma placa mãe.



Princípios de *Hardware* - Controladoras

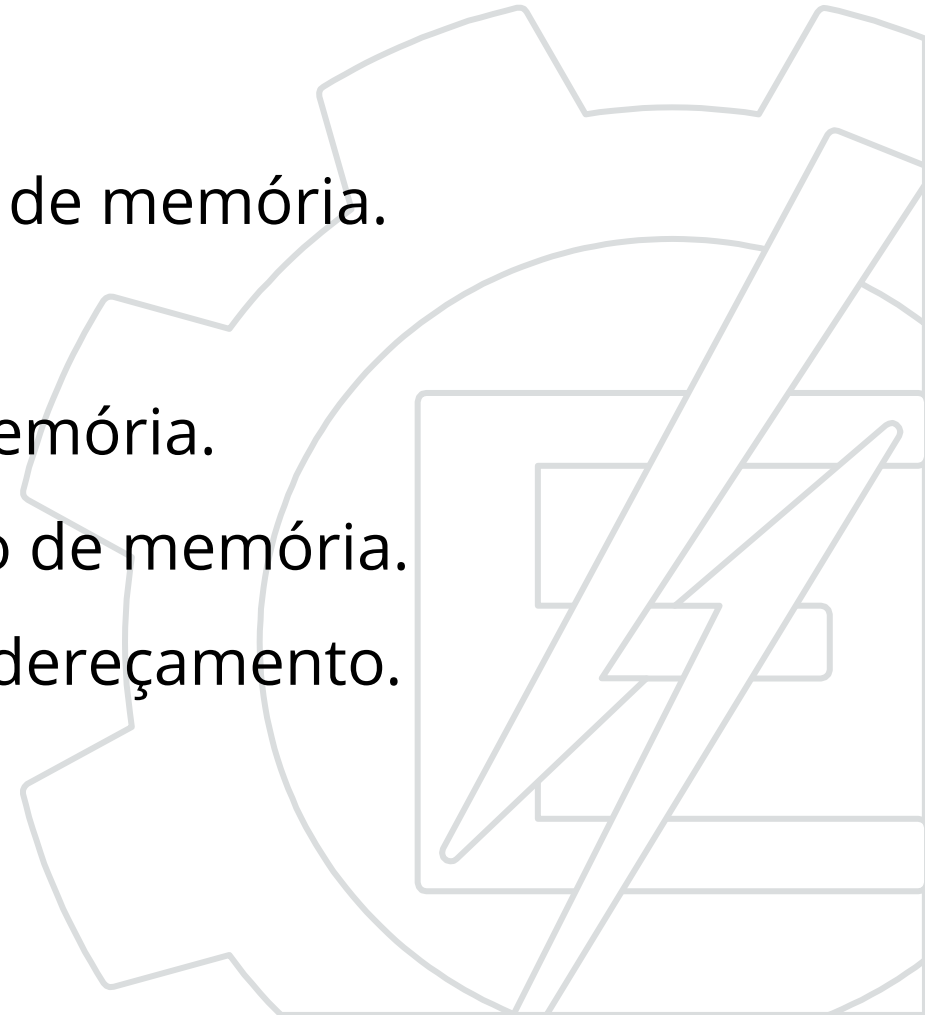
- Cada controladora possui registradores controlados pela CPU para ler/escrever dados no dispositivo.
- O SO pode controlar o dispositivo escrevendo comandos e alterando parâmetros.
- Registradores são utilizados para saber o estado do dispositivo.



Formas de comunicação CPU x Controladora

1) Mapeada na memória

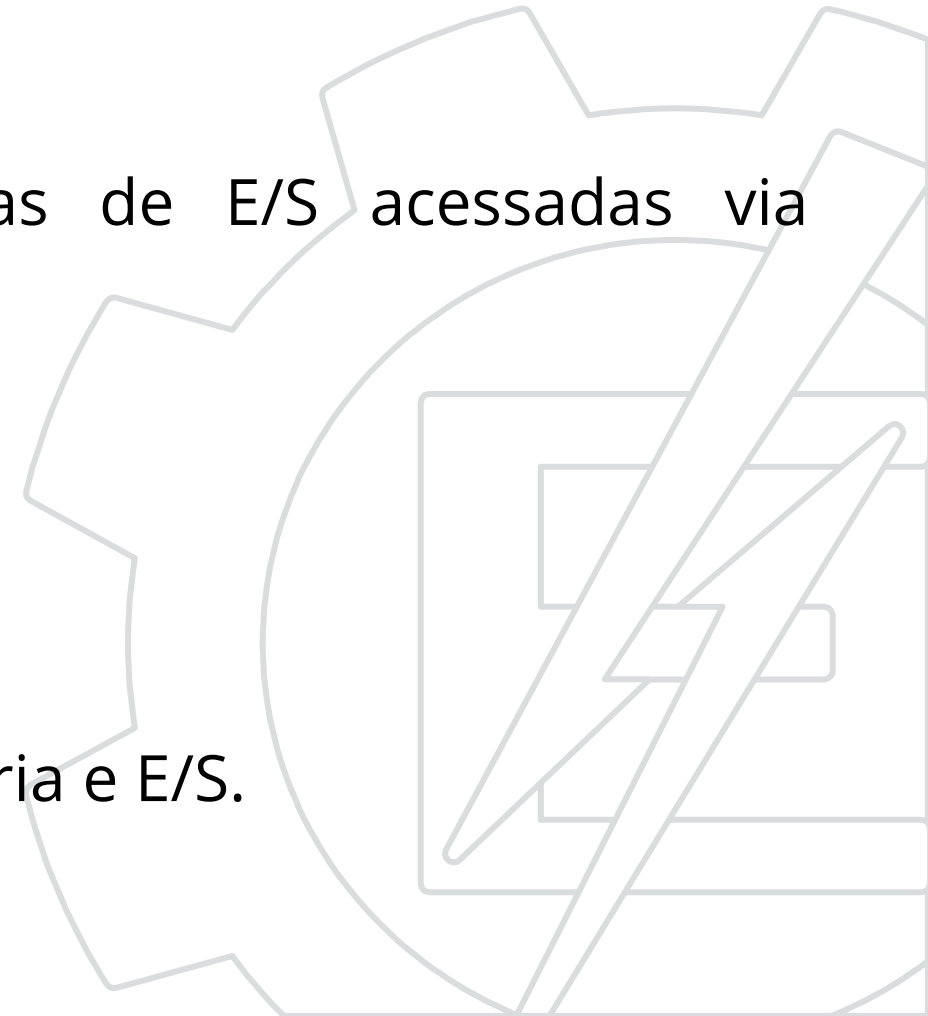
- Localizada dentro do espaço de endereçamento de memória.
- Utiliza um conjunto de endereços reservados.
- Registradores são tratados como posições de memória.
- Todos os registradores são mapeadas no espaço de memória.
- Geralmente se localiza no topo do espaço de endereçamento.



Formas de comunicação CPU x Controladora

2) Mapeada em E/S

- Controladora recebe um número de portas de E/S acessadas via instruções especiais utilizadas apenas pelo SO.
- Por exemplo:
IN REG, PORT
OUT PORT, REG
- Espaços de endereços diferentes para a memória e E/S.



Formas de comunicação CPU x Controladora

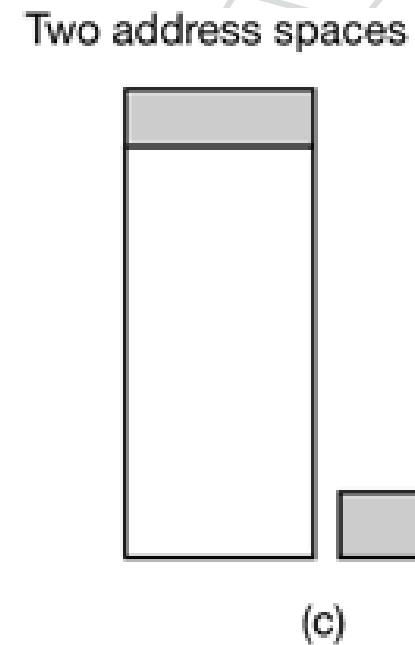
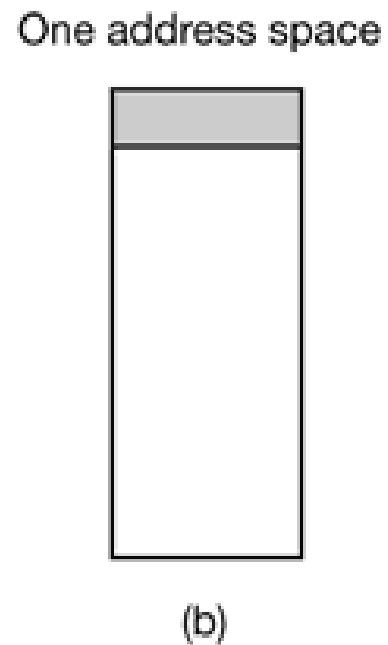
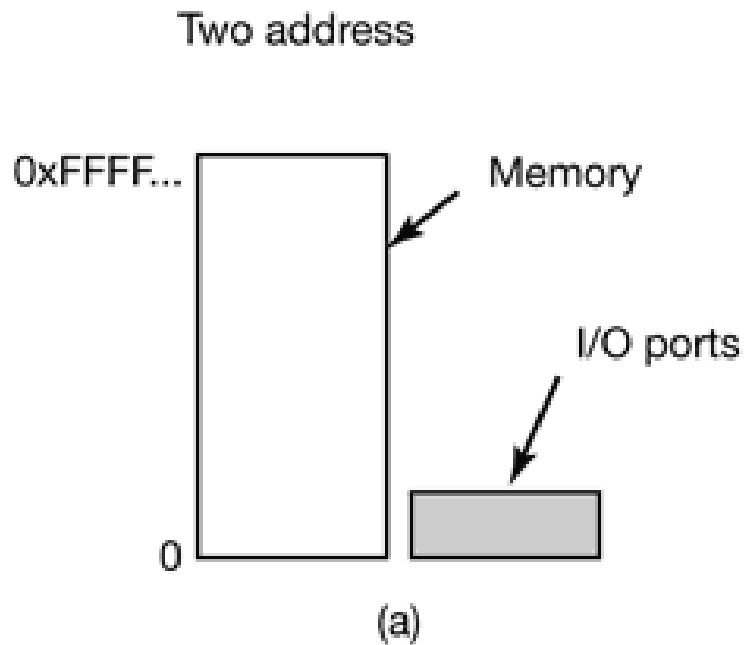
3) Mapeamento híbrido

- Buffers de dados na memória;
- Portas de E/S para controle.
- Exemplo - Pentium
 - 640K a 1M-1 para *buffer* de dados
 - Portas de E/S de 0 a 64K-1 para controle



Formas de comunicação CPU x Controladora

- a) Mapeada em E/S
- b) Mapeada na memória
- c) Mapeamento híbrido



Princípios de *Software*

- **Independência do dispositivo** (CD, HD, *Flash Drive*, etc.) – cabe ao SO cuidar das particularidades.
- Nomenclatura **uniforme** – nomes independentes do dispositivo.
- **Tratamento de erros** - Devem ser realizados o mais próximo possível do HW, sem que o usuário tome conhecimento.
 - Caso ocorra um erro de leitura, a mesma deverá ser repetida de modo que o erro não divulgado às camadas superiores.

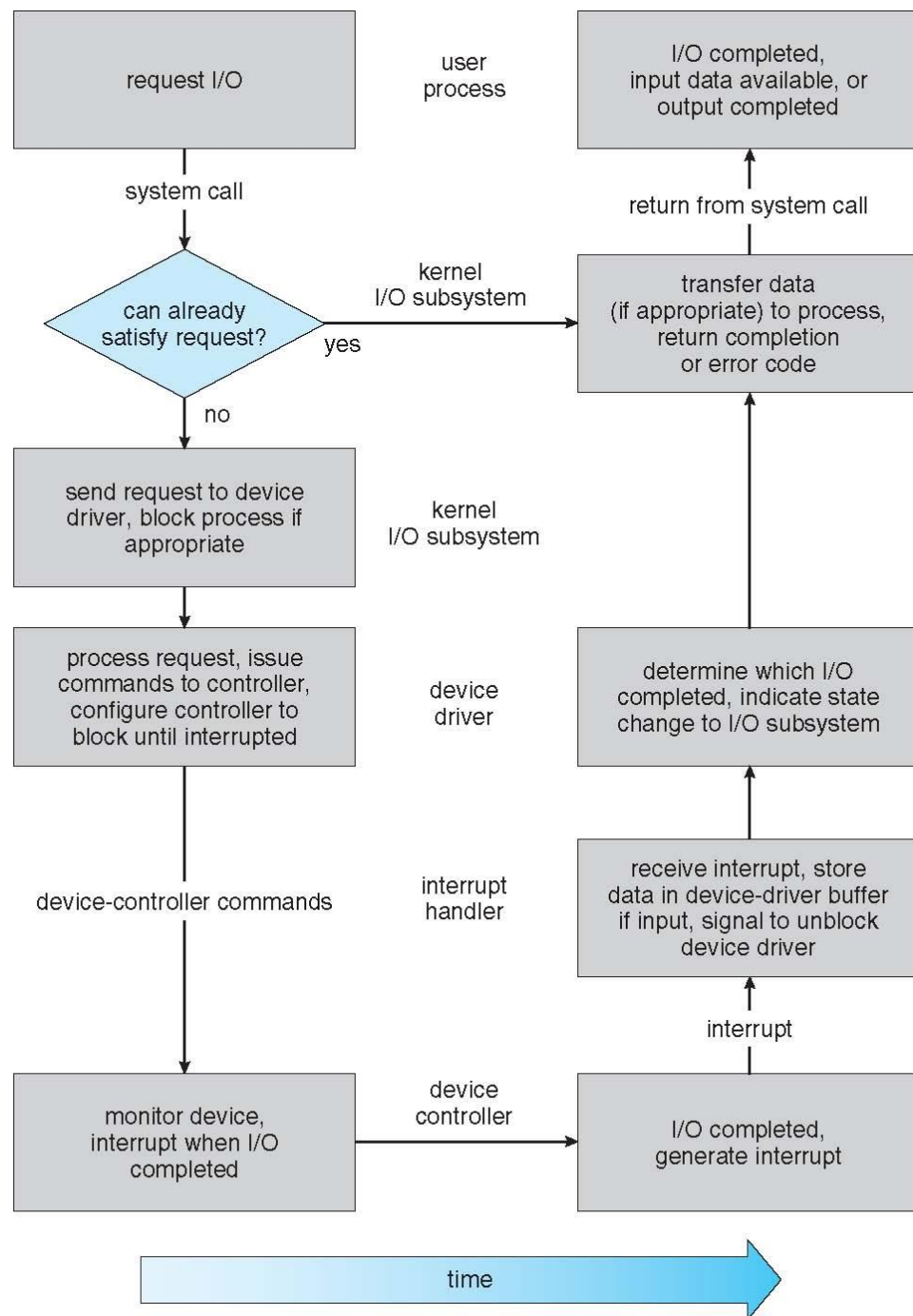
Modos de Operação de E/S

- 1) E/S Programada
 - 2) E/S via interrupções
 - 3) E/S via acesso direto à memória (DMA)
- O que distingue as três formas?
 - Participação da CPU
 - Utilização das interrupções



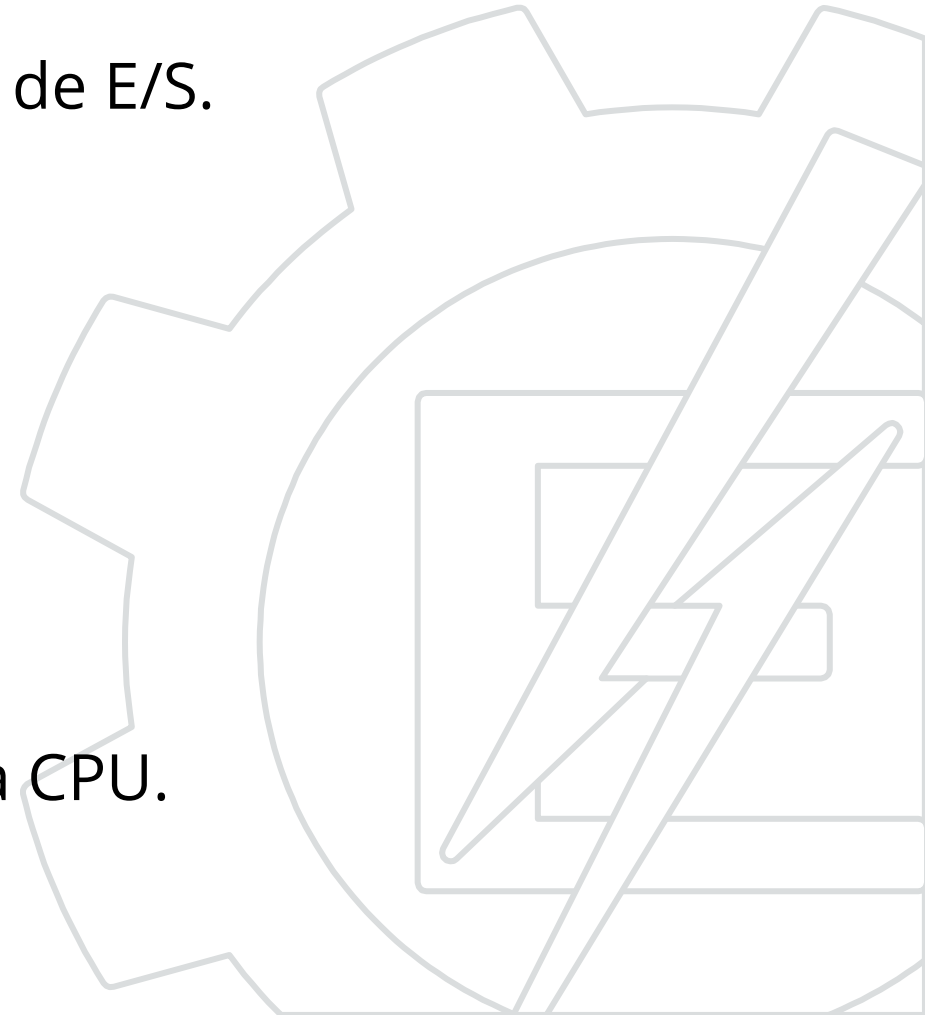
Sistema de Entrada e Saída

Modos de Operação de E/S



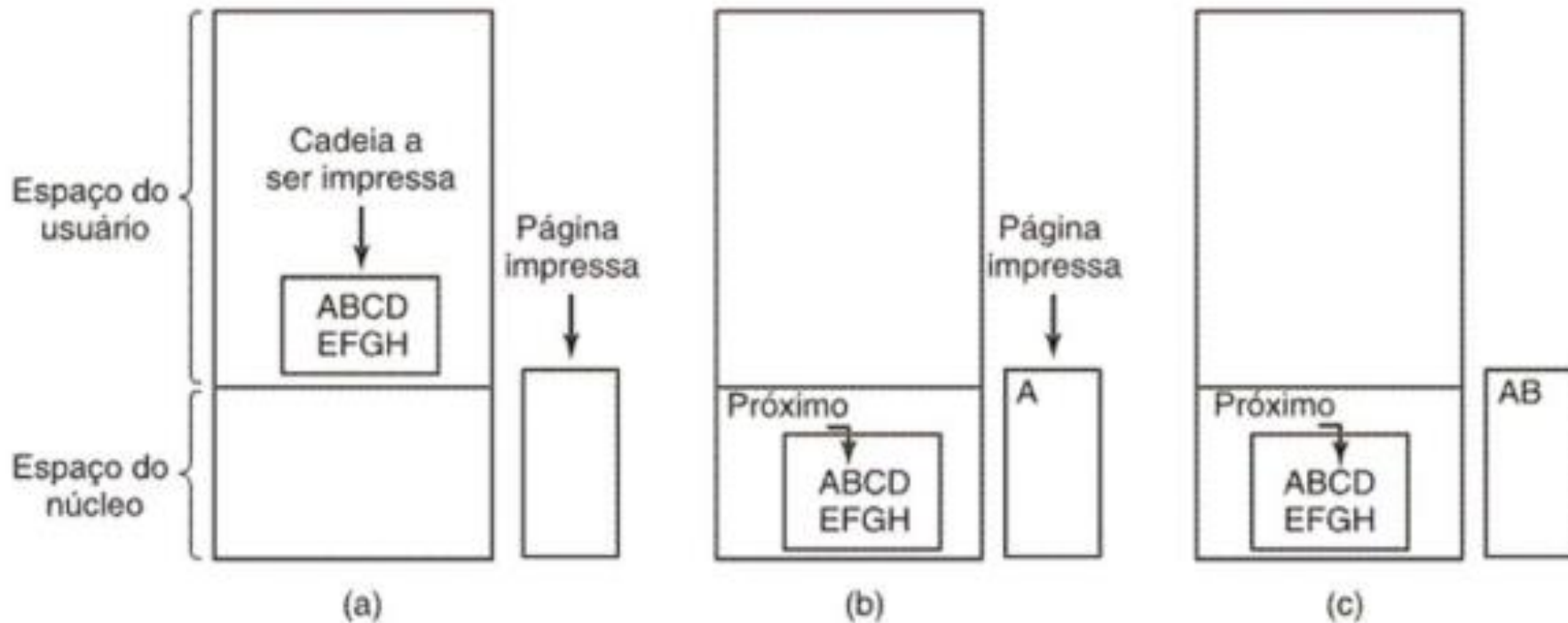
1) E/S Programada

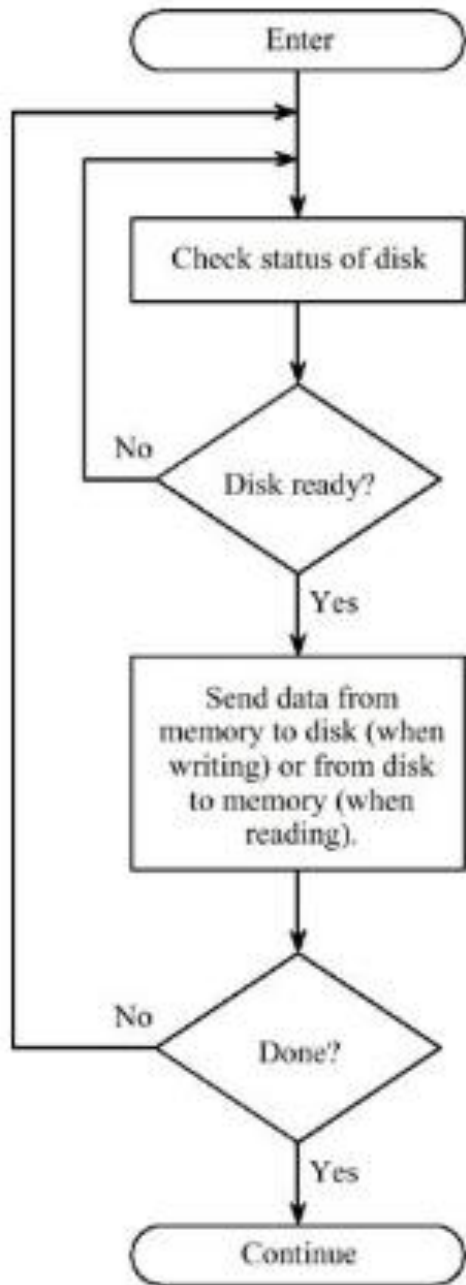
- Forma mais simples de E/S – **tudo é feito pela CPU.**
- Os dados são trocados entre a CPU e o módulo de E/S.
- A CPU executa um programa que:
 - 1) Verifica o estado do módulo de E/S;
 - 2) Envia o comando de operação;
 - 3) **Aguarda o resultado;**
 - 4) Efetua a transferência para o registrador da CPU.



1) E/S Programada

- Forma mais simples de E/S – **tudo é feito pela CPU.**
- Os dados são trocados entre a CPU e o módulo de E/S.



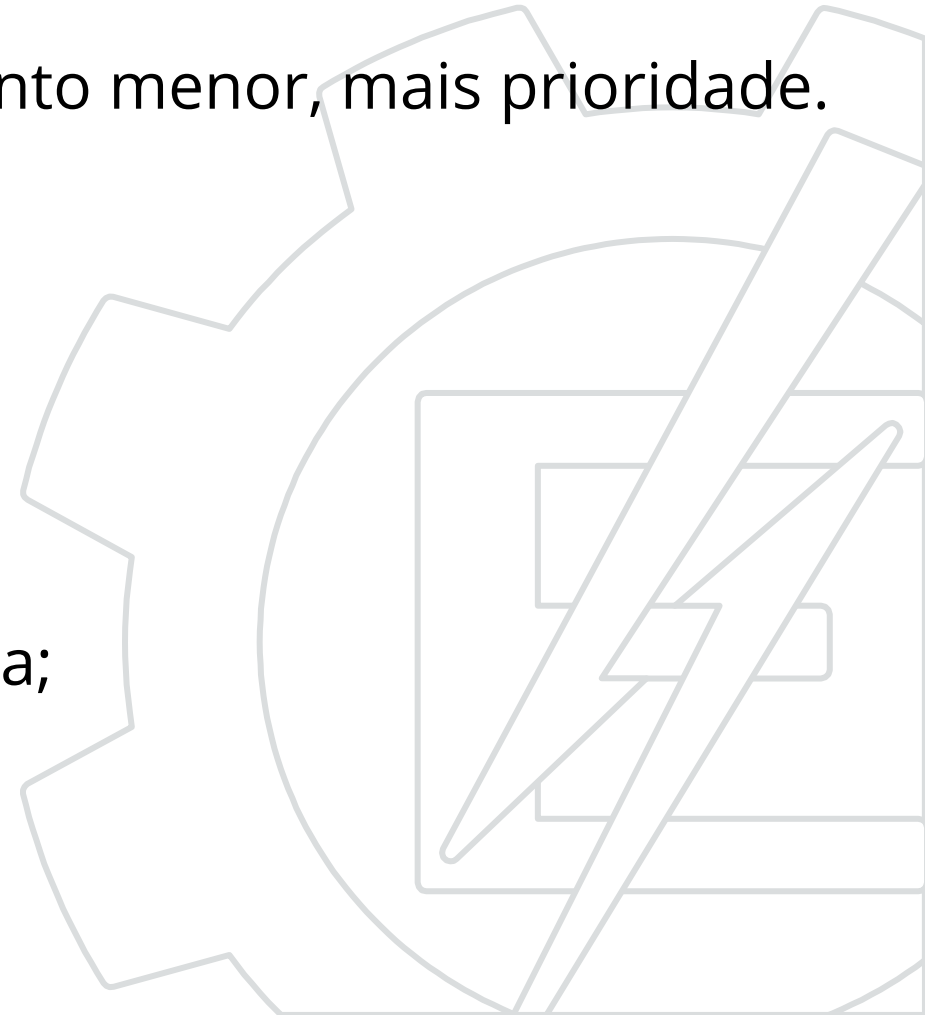


1) E/S Programada

- Desvantagens
 - CPU é ocupada o tempo todo e realiza toda a E/S.
 - Realiza uma **espera ocupada** para a realização da tarefa, também chamada de ***polling***.

2) E/S via Interrupção

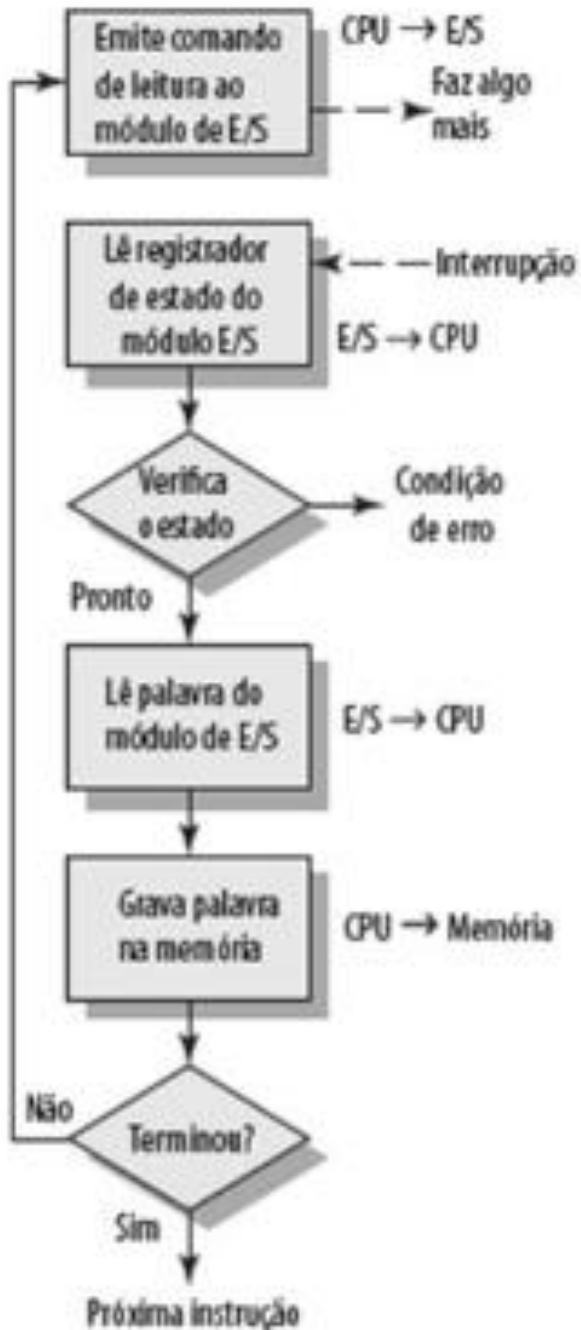
- Supera o problema da espera ocupada.
- Interrupções são identificadas por números. Quanto menor, mais prioridade.
- A CPU executa um programa que:
 - 1) Envia um comando para E/S;
 - 2) CPU executa outra operação;
 - 3) Controladora envia sinal quando E/S finalizada;
 - 4) CPU lê os dados da controladora.



2) E/S via Interrupção

- Exemplo de interrupção:

- 1) CPU requisita uma leitura no disco;
- 2) Controladora lê os dados enquanto a CPU realiza outras tarefas;
- 3) Controladora envia uma **interrupção** à CPU;
- 4) CPU solicita os dados;
- 5) Controladora envia os dados.



Sistema de Entrada e Saída

E/S via Interrupção

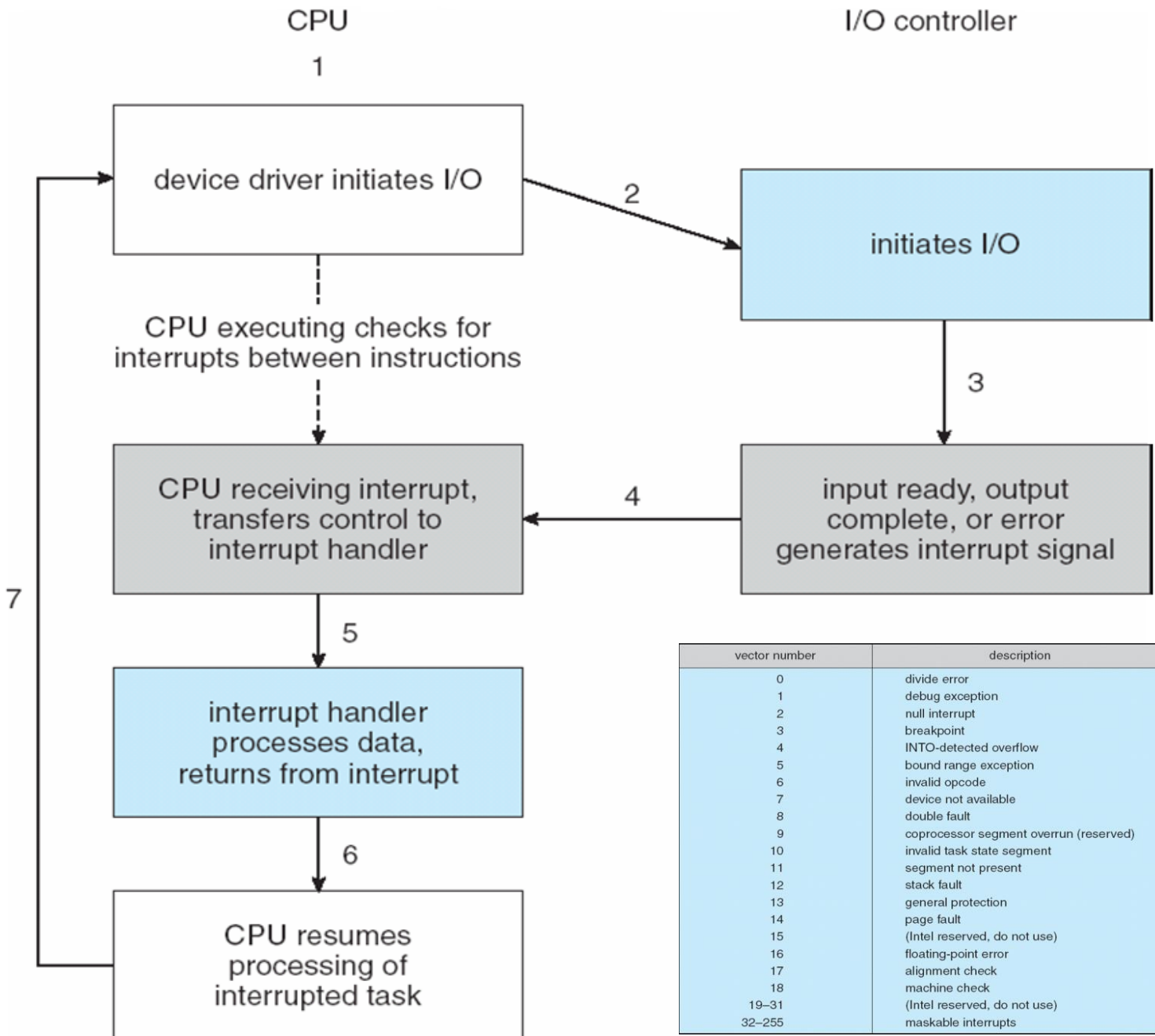
Below is a table showing the Interrupt Numbers and Names.

INT #	Normal Use	INT #	Normal Use	INT #	Normal Use
0	Divide by Zero (Internal)	12	BIOS Get Memory Size	24	Critical Error Handler *
1	Single Step Debug	13	BIOS Diskette Service	25	DOS Absolute Disk Read *
2	NMI *	14	BIOS Comm. Services *	26	DOS Absolute Disk Write *
3	Breakpoint *	15	BIOS Misc. System Services *	27	Terminate and Stay Resident (TSR)
4	Arithmetic Overflow	16	BIOS Keyboard Services	28	DOS safe *
5	Print Screen *	17	BIOS Printer Services	29	DOS TTY
6	Invalid Opcode	18	Execute	2A	MS Net
7	CPU Reserved	19	System Warm Reboot *	2F	"Multiplex" *
8	System Timer *	1A	BIOS Clock Services	33	Microsoft Mouse Services
9	Hardware Keyboard	1B	Ctrl-Break Handler *	67	EMS Services
A	Cascade to IRQ 9	1C	User Timer Tick Interrupt	70	Real Time Clock
B	COM 2*	1D	Video Init. Parameters	71	Redirect to IRQ 2
C	COM 1*	1E	Disk Init. Parameters *	72	USER HARDWARE
D	LPT 2	1F	Graph Display Char Table*	73	USER HARDWARE
E	Floppy Diskette Controller	20	DOS Terminate Program	74	IBM Mouse (Hardware) *
F	LPT1	21	DOS Services	75	Math Coprocessor
10	BIOS Video Services	22	DOS Termination Address	76	Hard Disk Controller
11	BIOS Get Equipment Status	23	CtrlC Handler *	77	USER HARDWARE

IRQ	Usage
0	system timer (cannot be changed)
1	keyboard controller (cannot be changed)
2	cascaded signals from IRQs 8-15
3	second RS-232 serial port (COM2: in Windows)
4	first RS-232 serial port (COM1: in Windows)
5	parallel port 2 and 3 or sound card
6	floppy disk controller
7	first parallel port
8	real-time clock
9	open interrupt
10	open interrupt
11	open interrupt
12	PS/2 mouse
13	math coprocessor
14	primary ATA channel
15	secondary ATA channel

Sistema de Entrada e Saída

E/S via Interrupção



3) E/S via acesso direto à memória (DMA)

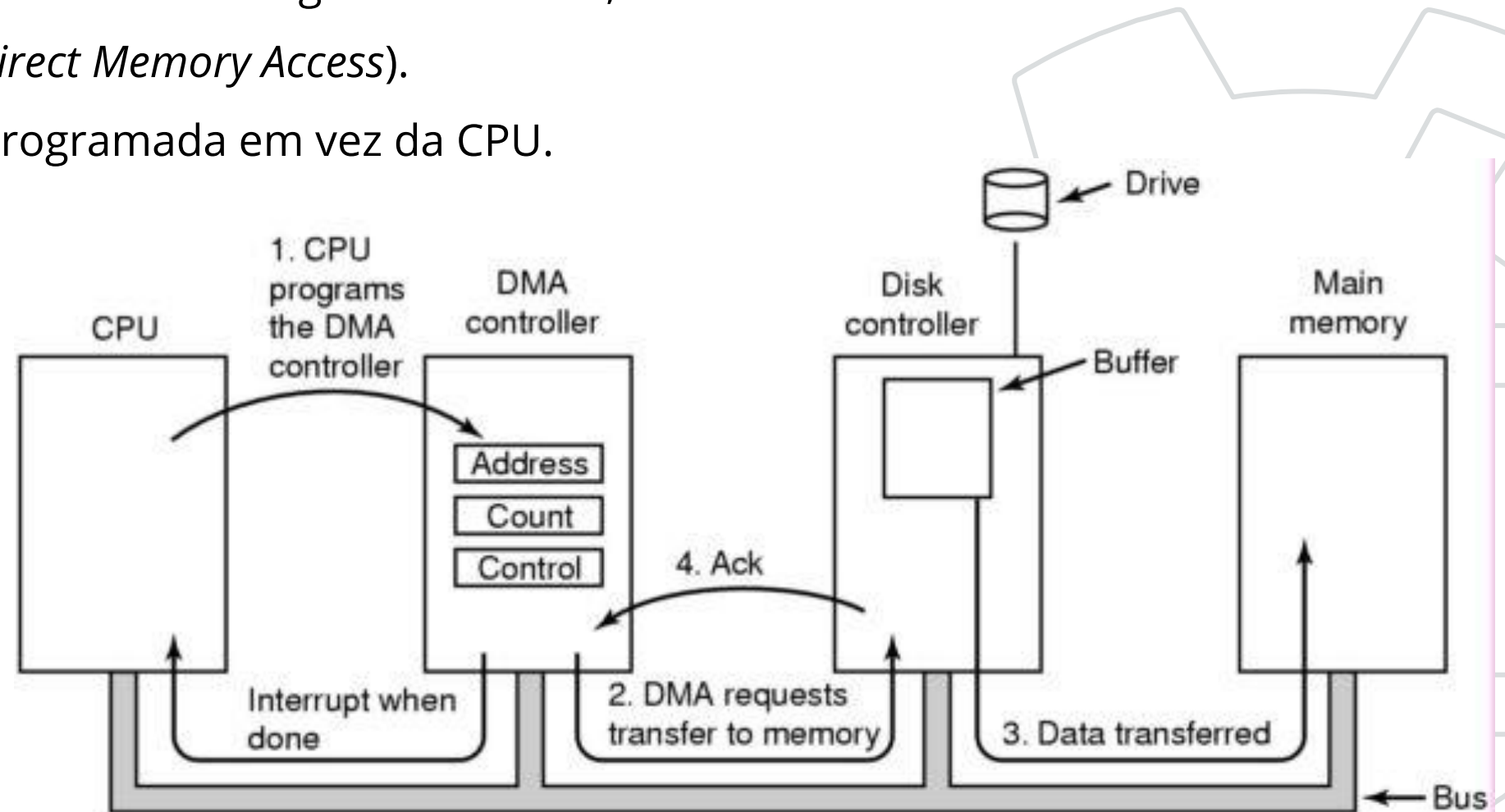
- Requer *software* e *hardware*.
- Esta solução retira a CPU do gerenciamento, mas necessita de uma controladora DMA (*Direct Memory Access*).
- A DMA realiza a E/S Programada em vez da CPU.
- Inconvenientes das técnicas anteriores:
 - Limitam a capacidade de transferência da CPU;
 - CPU fica ocupada no gerenciamento;
 - Desempenho cai para grande quantidade de dados.



Sistema de Entrada e Saída

3) E/S via acesso direto à memória (DMA)

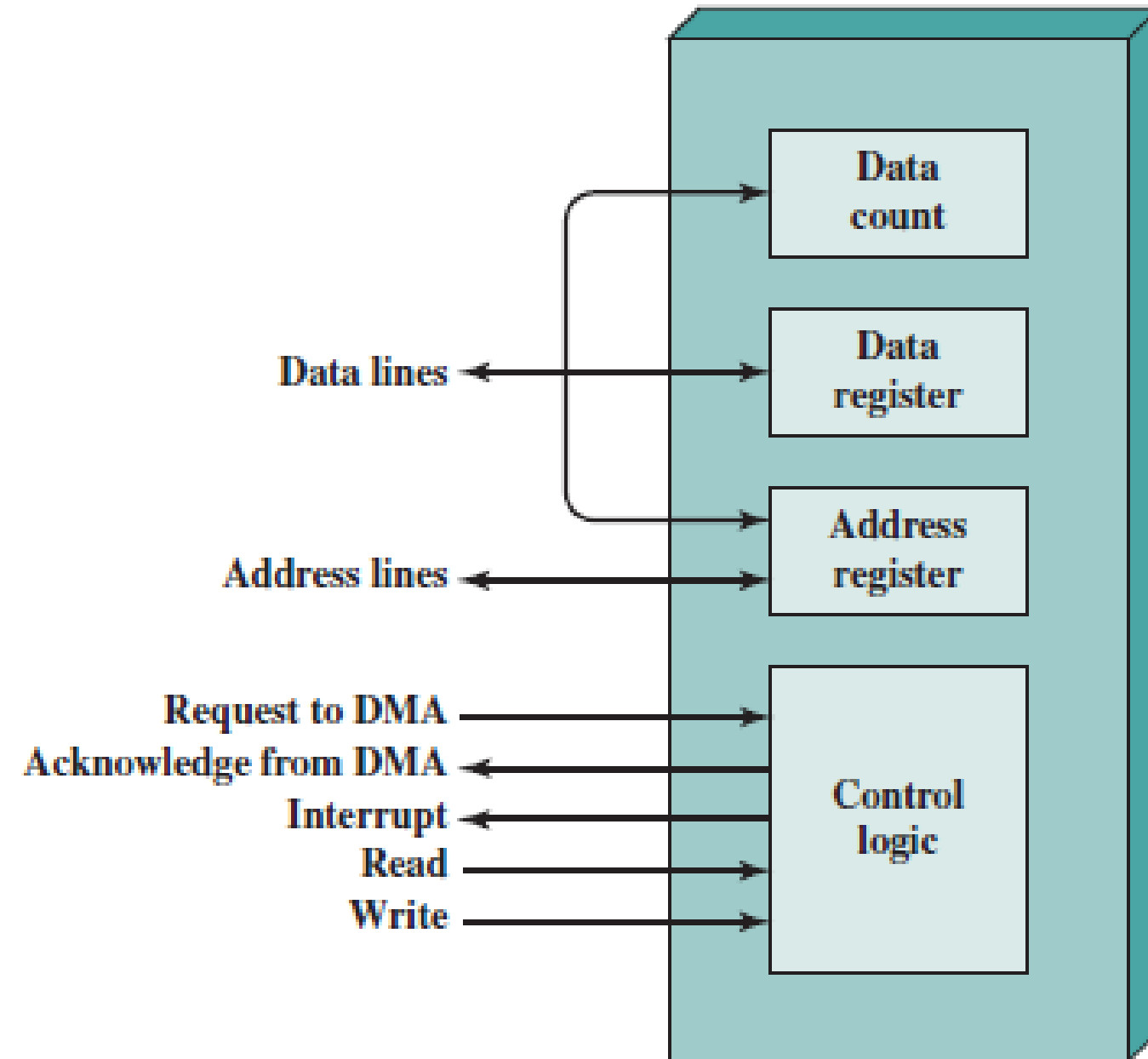
- Requer *software* e *hardware*.
- Esta solução retira a CPU do gerenciamento, mas necessita de uma controladora DMA (*Direct Memory Access*).
- A DMA realiza a E/S Programada em vez da CPU.



Sistema de Entrada e Saída

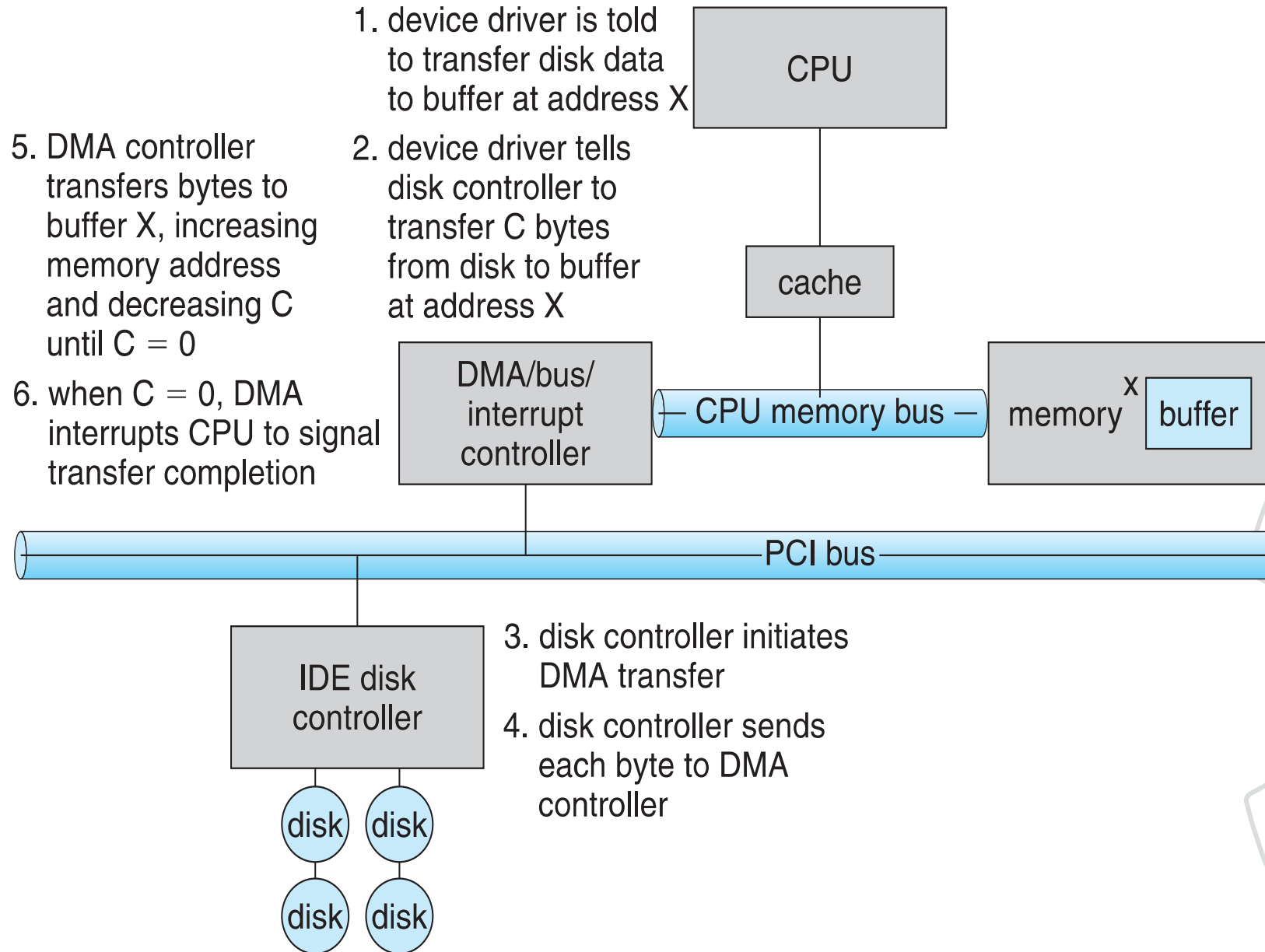
3) E/S via acesso direto à memória (DMA)

- Informações necessárias:
 - a) Endereço de memória;
 - b) Quantidade de bytes;
 - c) Porta de E/S a ser utilizada;
 - d) Direção de transferência (do ou para o dispositivo)
 - e) Unidade de transferência (um byte ou palavra por vez)



Sistema de Entrada e Saída

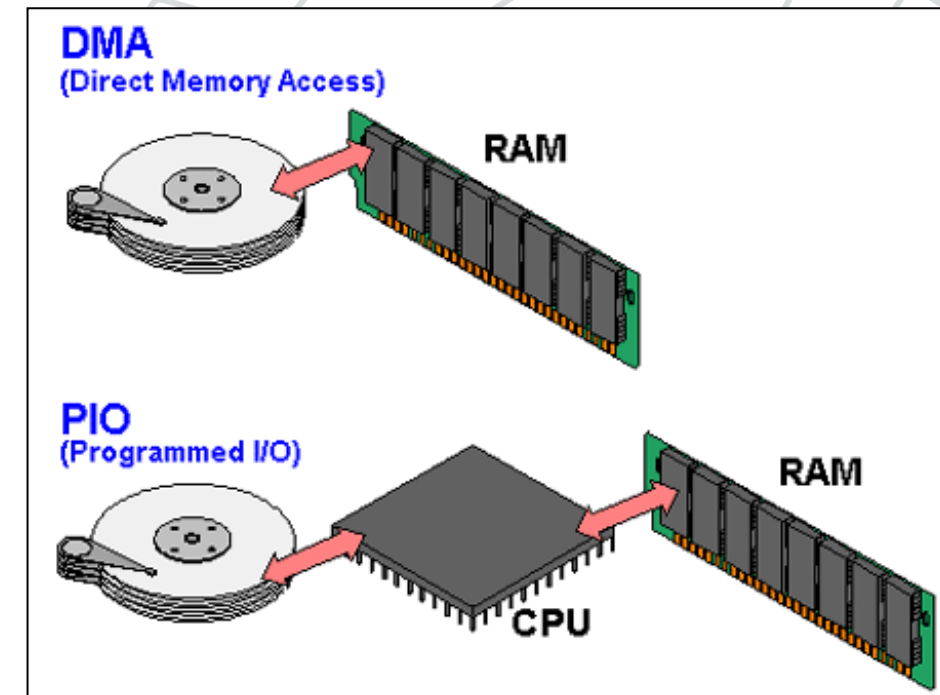
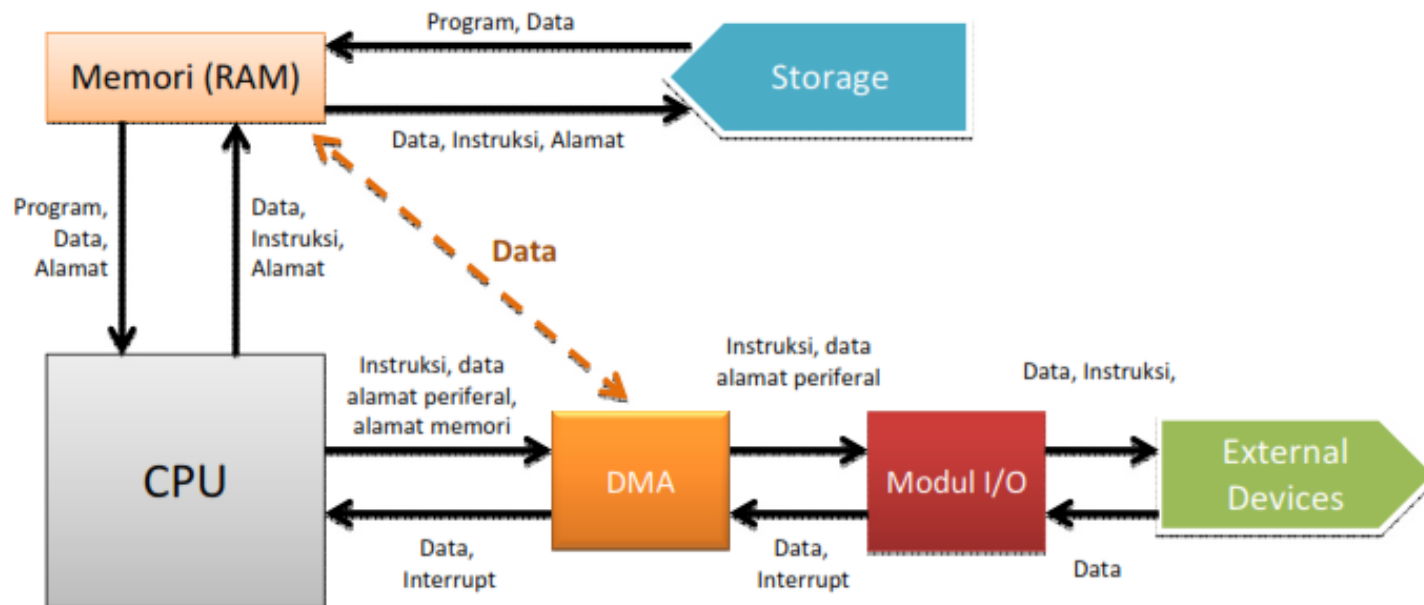
E/S via acesso direto à memória (DMA)



Sistema de Entrada e Saída

3) E/S via acesso direto à memória (DMA)

- Desvantagens:
 - A CPU pode ser mais rápida que a controladora DMA.
 - Arquitetura mais cara com DMA.
- Vantagens:
 - DMA executa E/S programada.
 - Controladora DMA realiza todo o trabalho e libera a CPU.



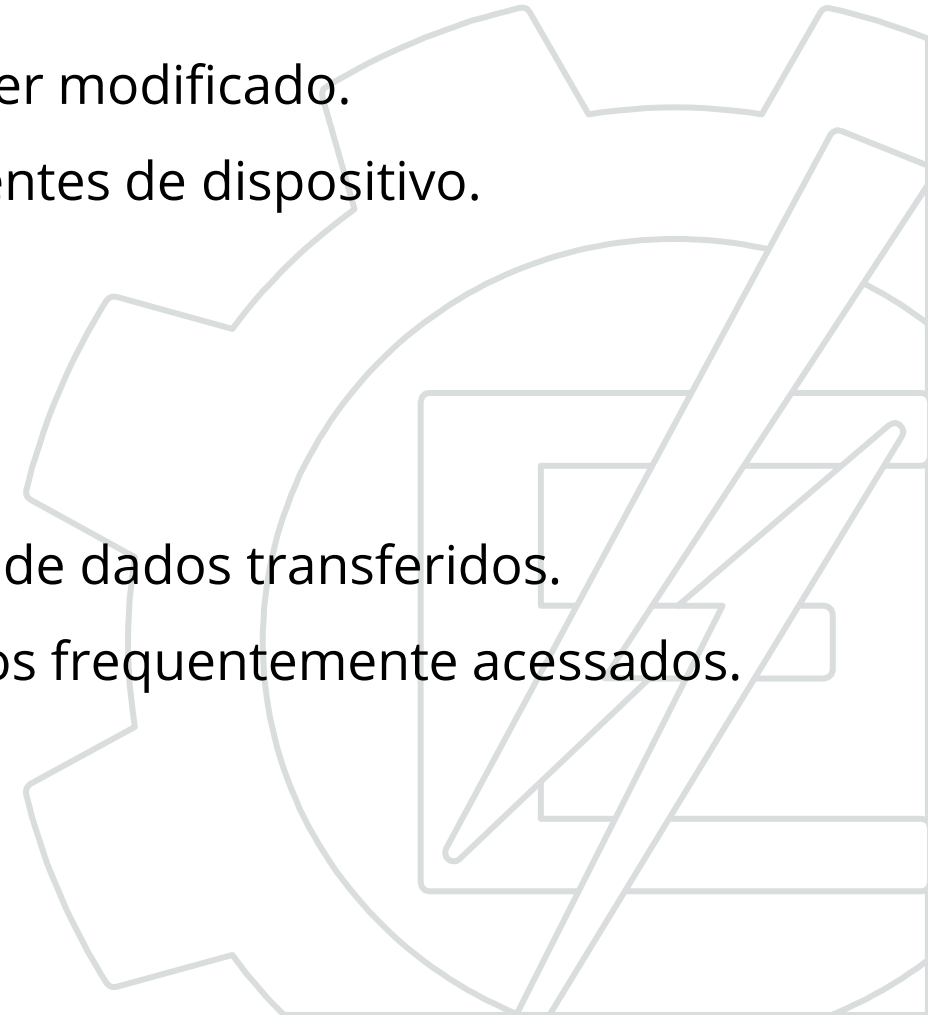
Princípio de camadas

- Facilita a **independência** dos dispositivos, fornecendo **modularidade** e **coesão**.
- Camadas mais baixas:
 - Detalhes do *hardware*;
 - *Drivers* e manipuladores de interrupção.
- Camadas mais altas:
 - Interface para o usuário;
 - Aplicações de usuário;
 - Chamadas de sistema – parte independente de E/S.



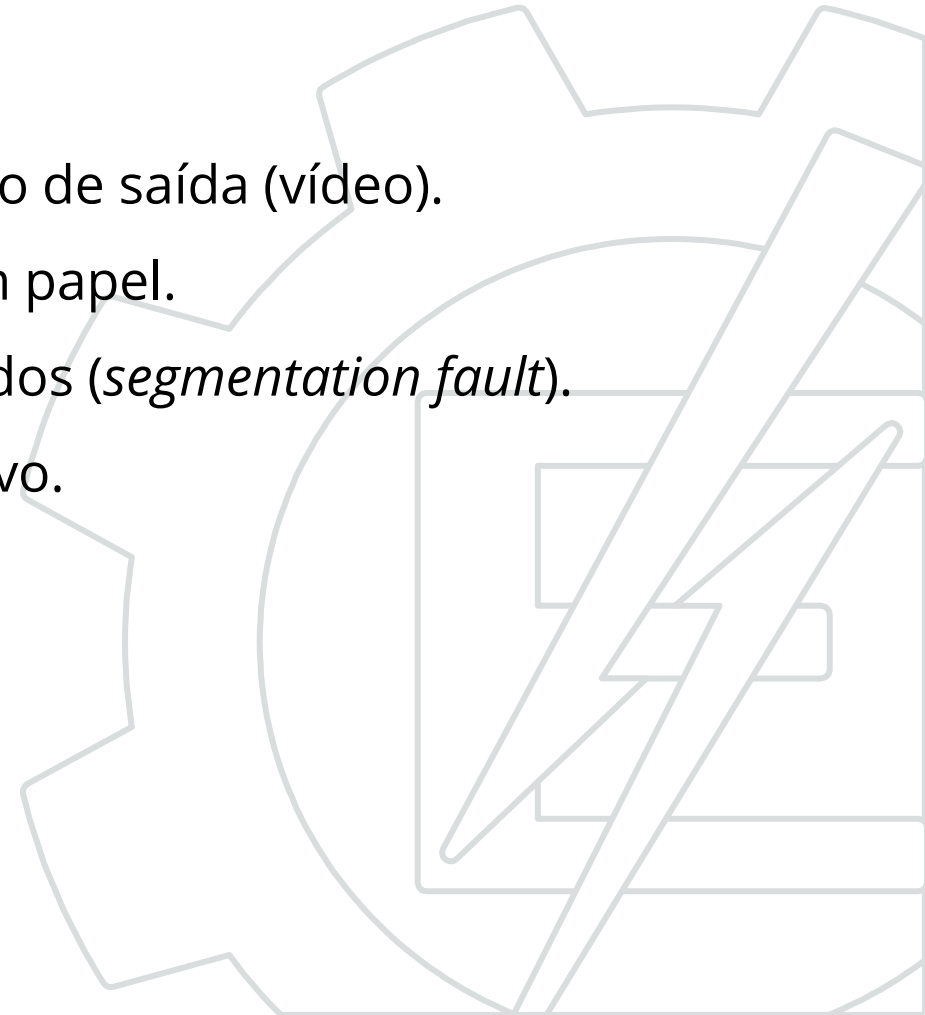
Independência do dispositivo

- Fornecer interface uniforme ao *software* de usuário.
- Evitar que a cada novo dispositivo criado o SO tenha que ser modificado.
- Software de E/S – há partes específicas e outras independentes de dispositivo.
- A parte independente:
 - Executa E/S comuns a todos os dispositivos;
 - Realiza o escalonamento de E/S;
 - Provê *buffering* – ajuste de velocidade e da quantidade de dados transferidos.
 - Provê *cache* de dados– armazena um conjunto de dados frequentemente acessados.



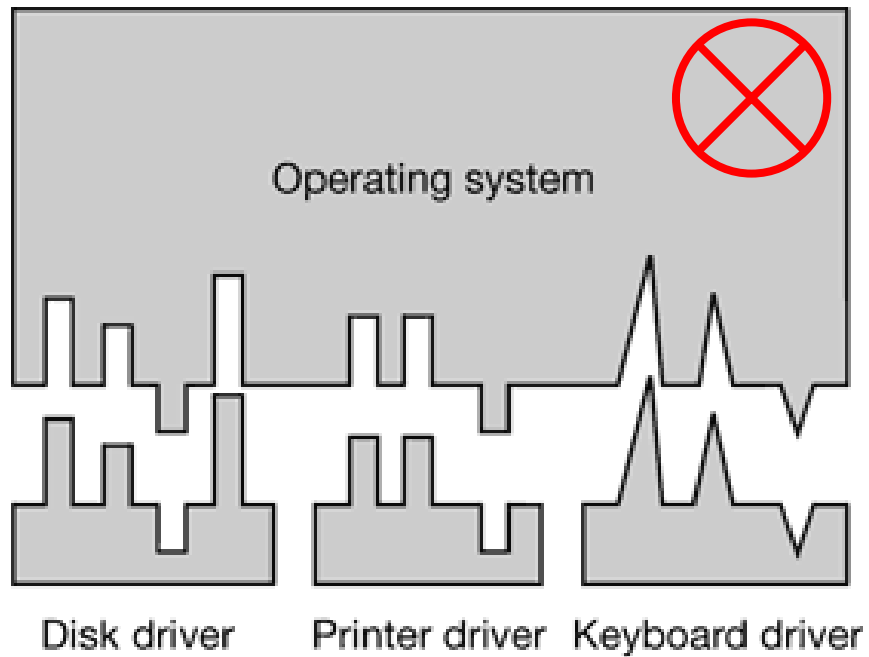
Independência do dispositivo

- Outras funções da parte independente:
 - Reportar erros e proteger contra acessos indevidos:
 - Erros de programação – Ex.: leitura de um dispositivo de saída (vídeo).
 - Erros de E/S – Ex.: imprimir em uma impressora sem papel.
 - Erros de memória – Ex.: escrita em endereços inválidos (*segmentation fault*).
 - Definir tamanhos de blocos independentes do dispositivo.

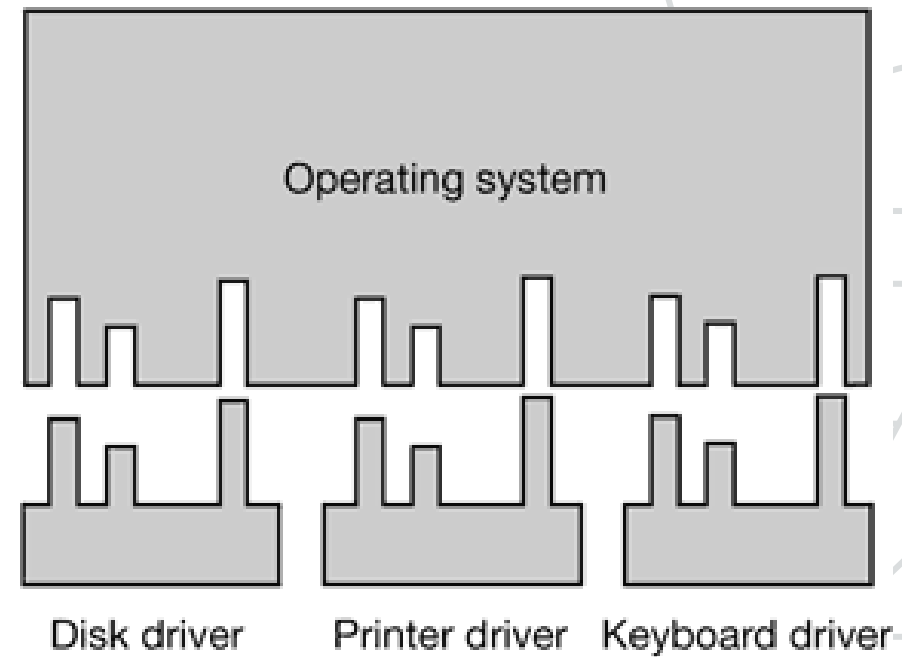


Princípios de *Software*

- Fornecer **interface uniforme** (API) – *read, write, send, receive*, etc.
- Padronizar as funções dos *drivers* – cada fabricante fornece sua função.



(a)



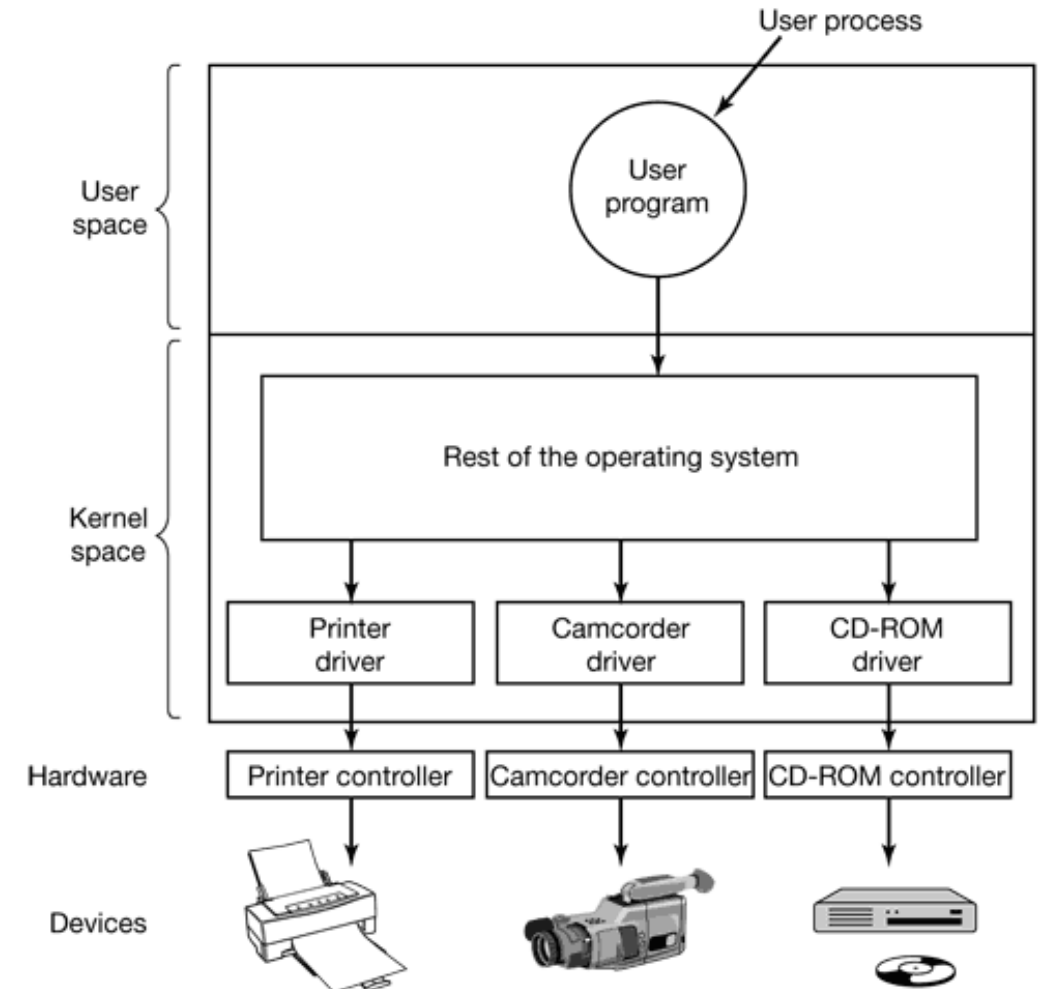
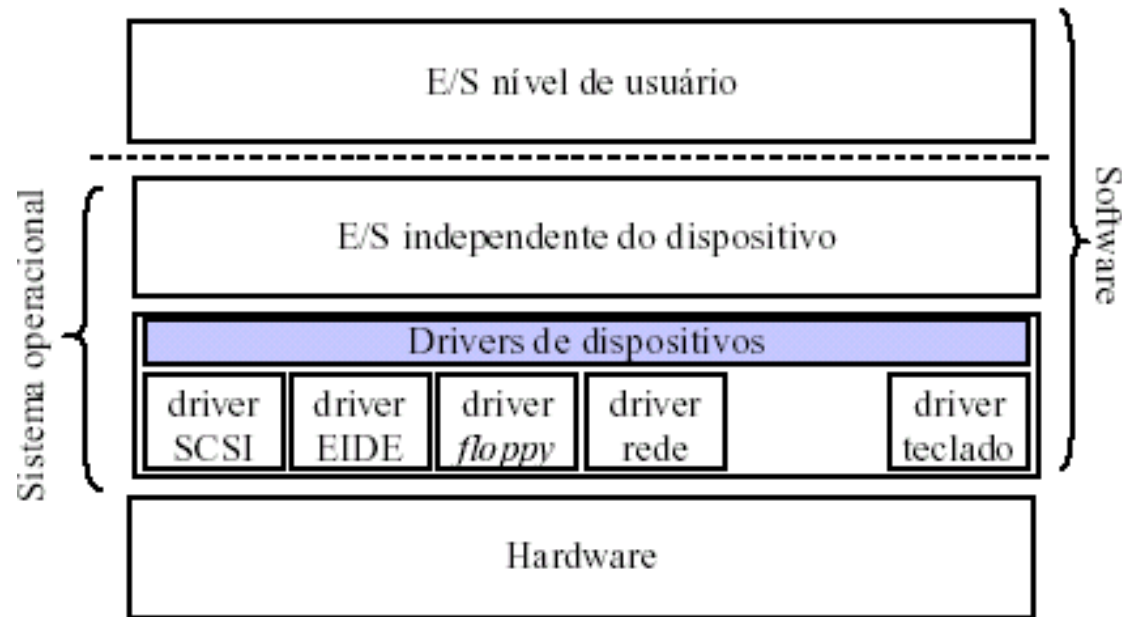
(b)

Sistema de Entrada e Saída

Drivers

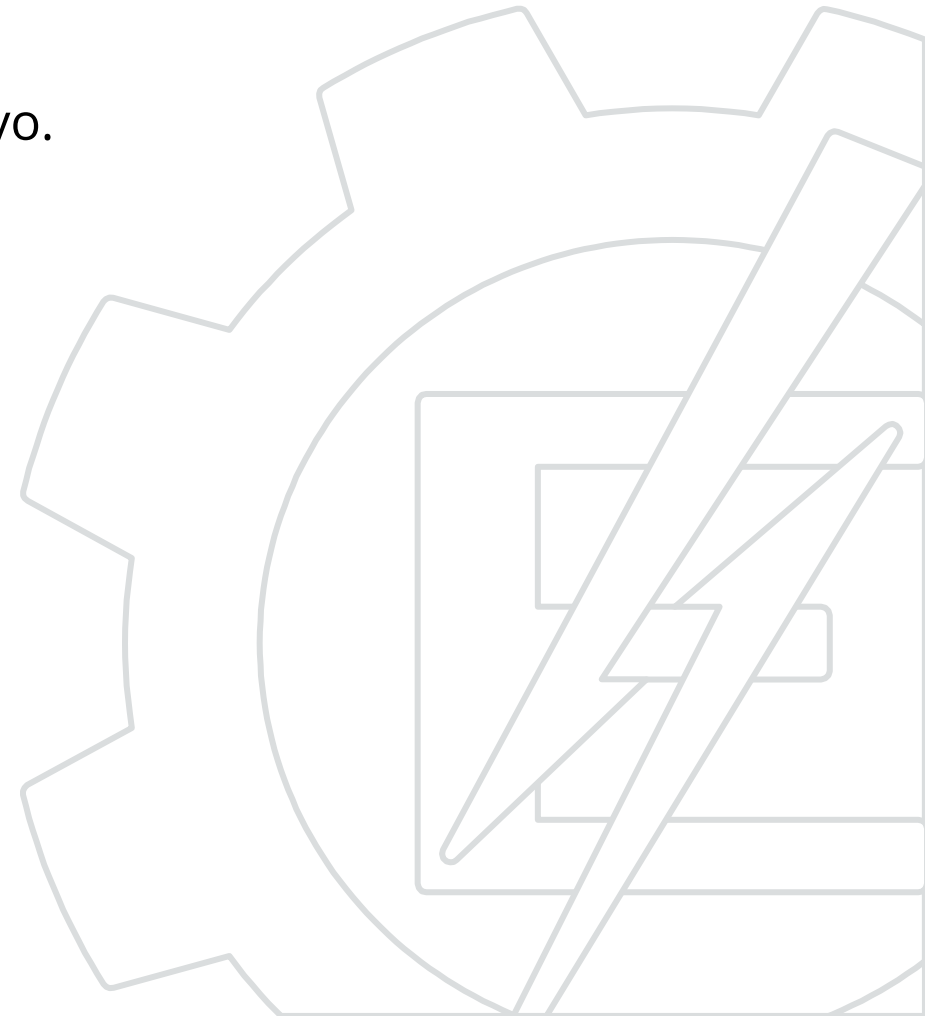
Drivers – Parte dependente do dispositivo

- Escritos pelo fabricante do dispositivo, de acordo com a interface definida.



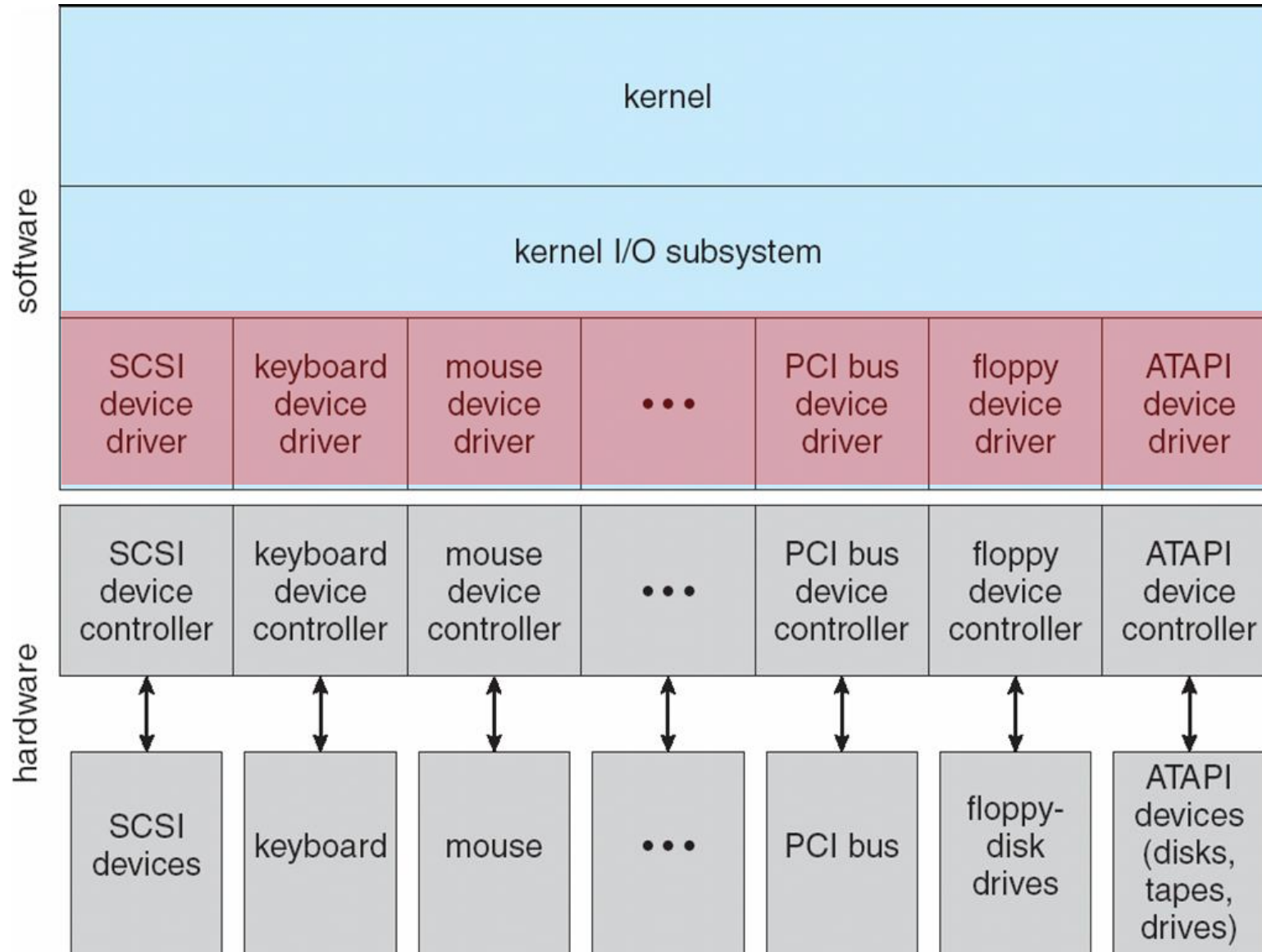
Drivers – Parte dependente do dispositivo

- Escritos pelo fabricante do dispositivo, de acordo com a interface definida.
- S.O.s diferentes requerem diferentes *drivers*:
 - Fazem parte do *kernel* e possuem acesso total ao dispositivo.
 - Podem causar problemas no SO.
- Código específico para E/S e para controle:
 - Disco Winchester (HD) com pratos e braço;
 - Disco de estado sólido (SSD).
- Processo de utilização:
 - Compila o código do *driver*
 - ***insmod*** instala o objeto do *driver*.
 - ***rmmod*** remove o *driver*.



Sistema de Entrada e Saída

Drivers

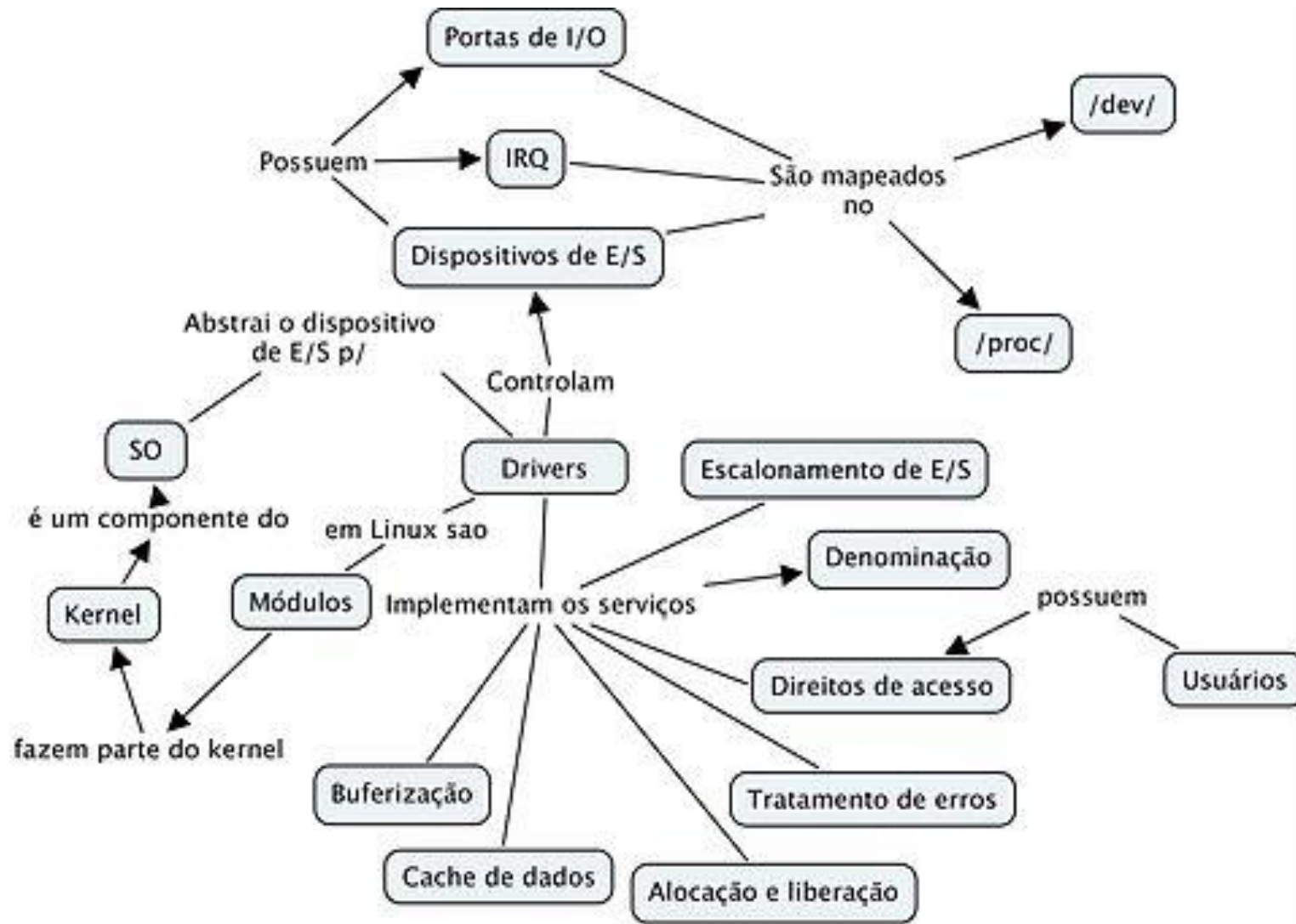


Drivers – Parte dependente do dispositivo

- Podem ser carregados dinamicamente (por exemplo, **DLLs**).
- São utilizados para as solicitações de leitura/escrita realizadas pelo *software*:
 - Verificam o pedido realizado - *check-up*;
 - Inicializam o dispositivo, se necessário;
 - Gerenciam as necessidades de energia do dispositivo;
 - Criam um *log* de eventos.

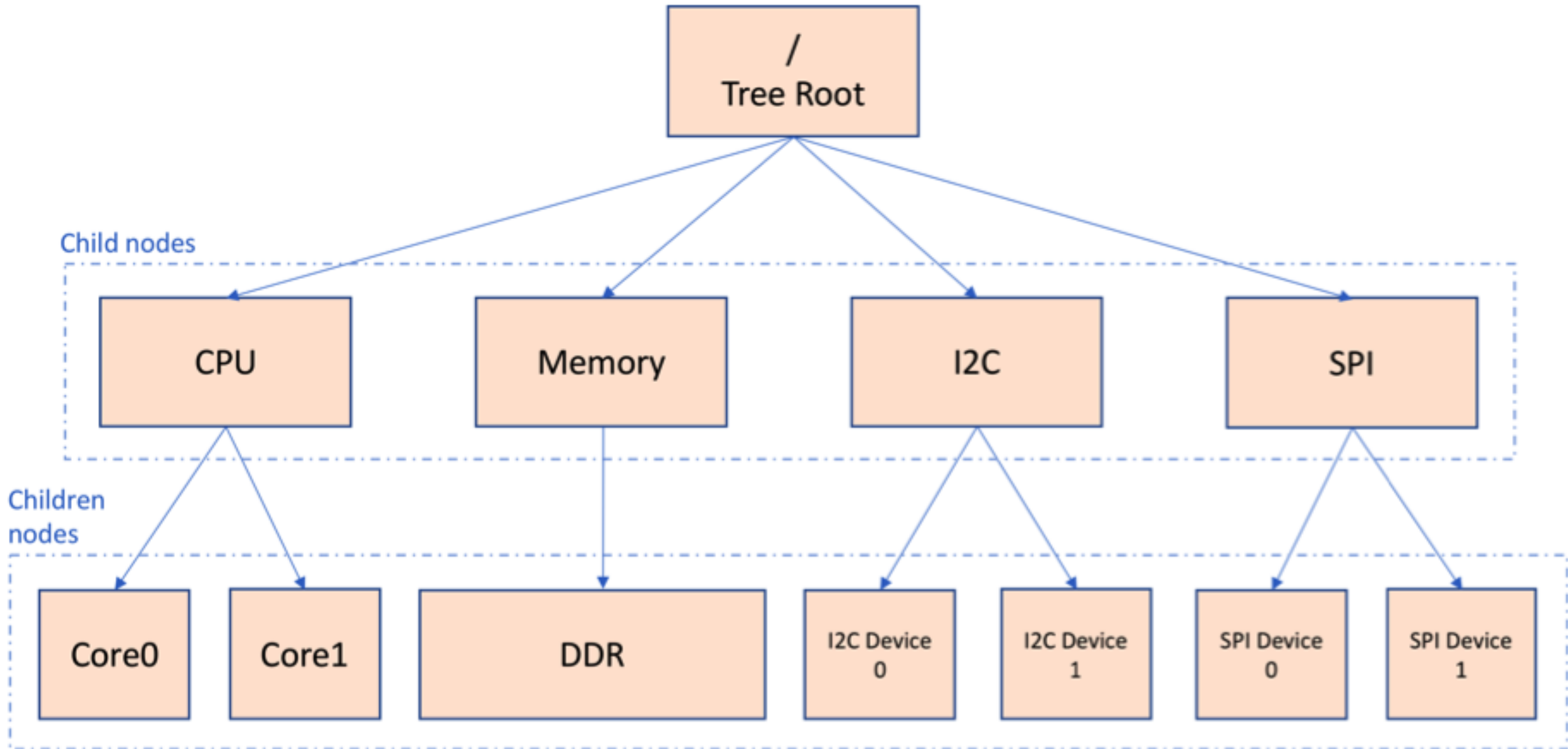


Sistema de Entrada e Saída



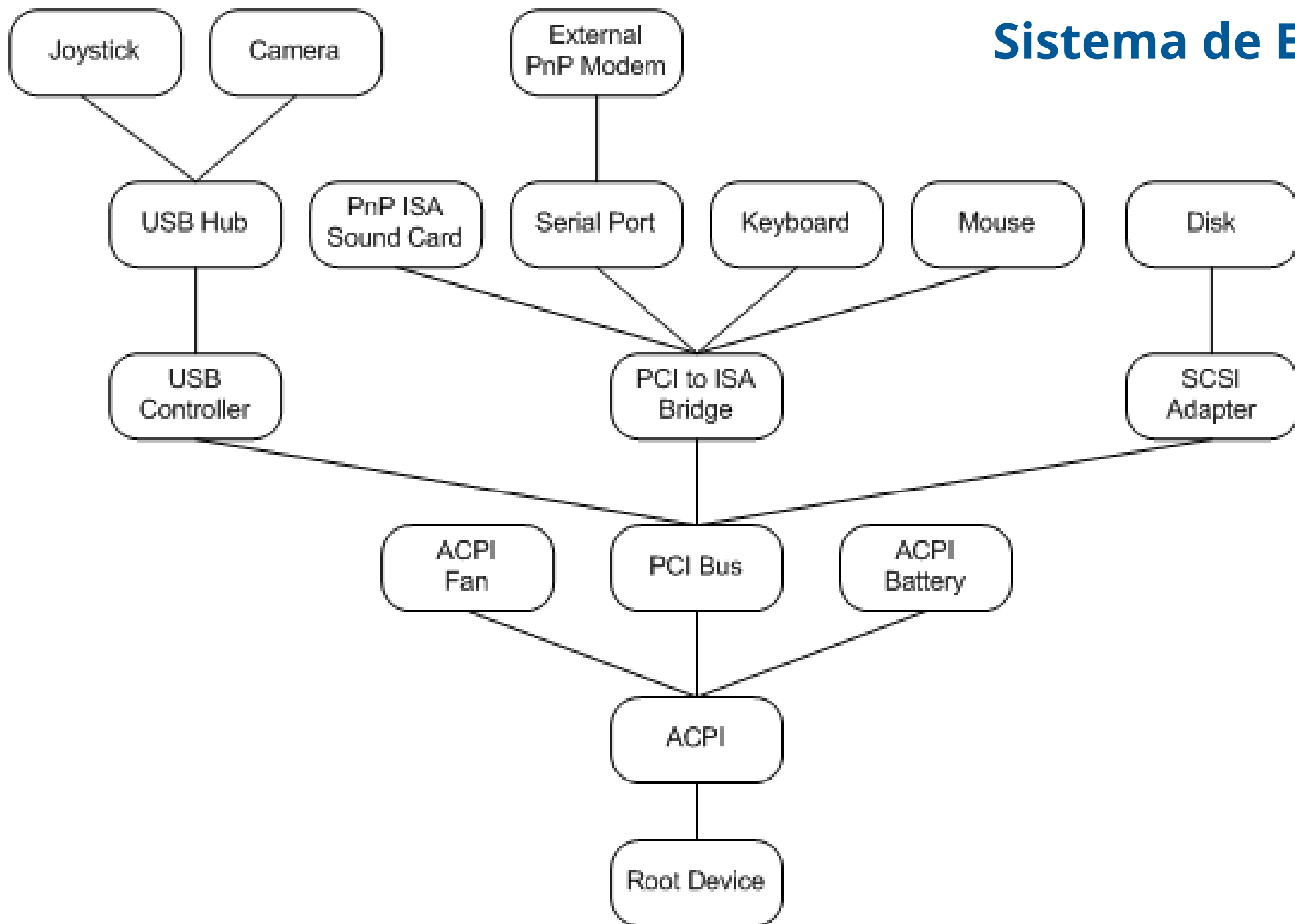
Sistema de Entrada e Saída

Device Tree



Sistema de Entrada e Saída

Device Tree



СИСТЕМА ВХОДА И ВЫХОДА

Sistema de Entrada e Saída

