

# Infra-Estrutura de Software

Entrada / Saída

# Diversidade de dispositivos

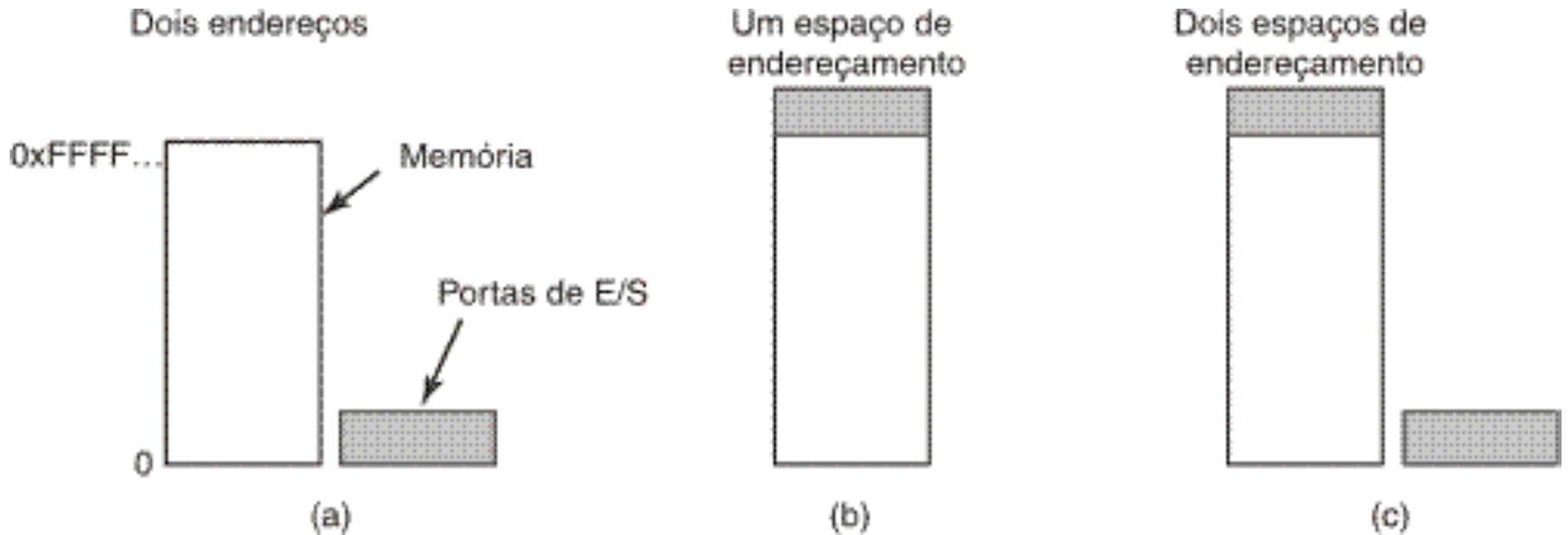
## Hardware de E/S

Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Scanner at 300 dpi	1 MB/sec
Digital camcorder	3.5 MB/sec
4x Blu-ray disc	18 MB/sec
802.11n Wireless	37.5 MB/sec
USB 2.0	60 MB/sec
FireWire 800	100 MB/sec
Gigabit Ethernet	125 MB/sec
SATA 3 disk drive	600 MB/sec
USB 3.0	625 MB/sec
SCSI Ultra 5 bus	640 MB/sec
Single-lane PCIe 3.0 bus	985 MB/sec
Thunderbolt 2 bus	2.5 GB/sec
SONET OC-768 network	5 GB/sec

# E/S: Como a CPU acessa a informação?

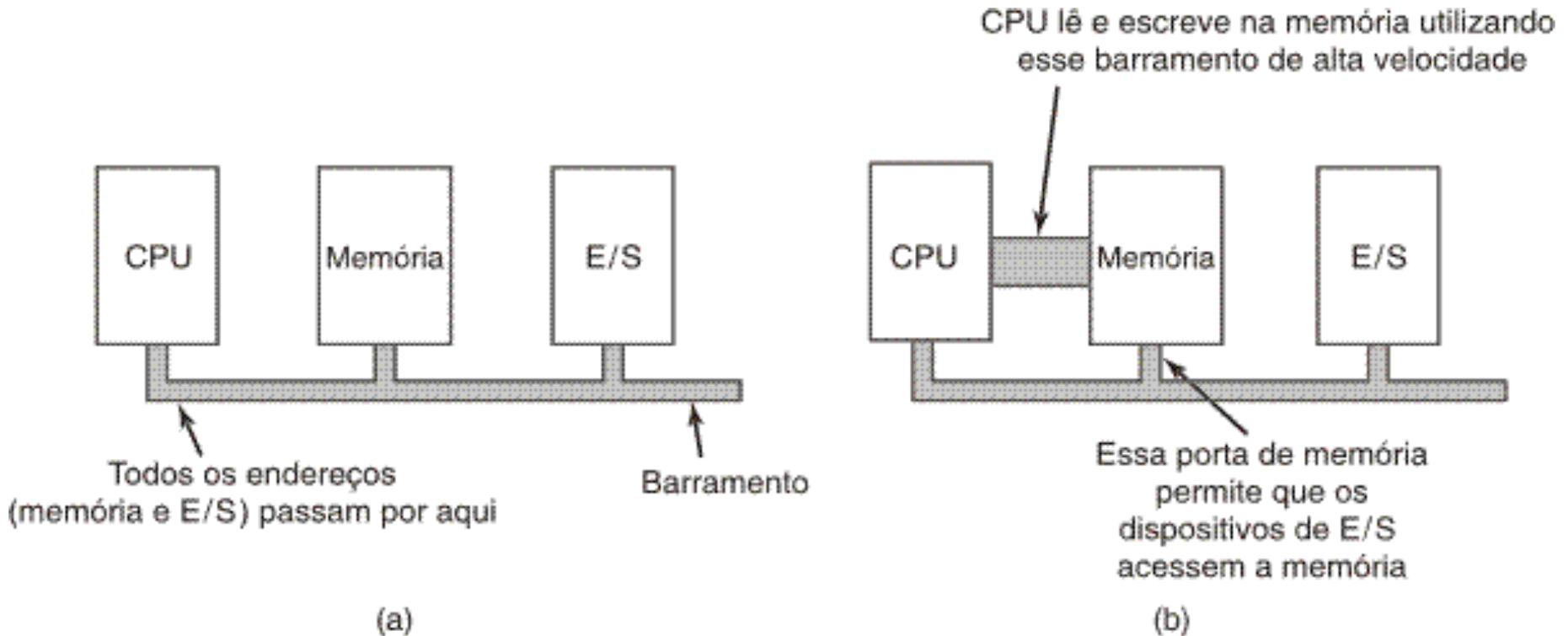
- **Espaço de endereçamento:** conjunto de endereços de memória que o processador consegue acessar diretamente
- A forma de acessar os **registradores (das interfaces) dos periféricos** é definida no projeto do processador:
  - Espaço único
  - Dois espaços, um deles dedicado à E/S (isolada)
- **E/S isolada**
  - Através de **instruções especiais** de E/S
  - Especifica a leitura/escrita de dados numa porta de E/S
- **E/S mapeada em memória**
  - Através de **instruções de leitura/escrita** na memória
- **Híbrido** (ex. IBM-PC):
  - E/S mapeada em memória: memória de vídeo
  - E/S isolada: dispositivos em geral

# Espaços de Memória e E/S



- a) Espaços de memória e E/S separados - E/S isolada
- b) E/S mapeada na memória
- c) Híbrido

# E/S mapeada na memória



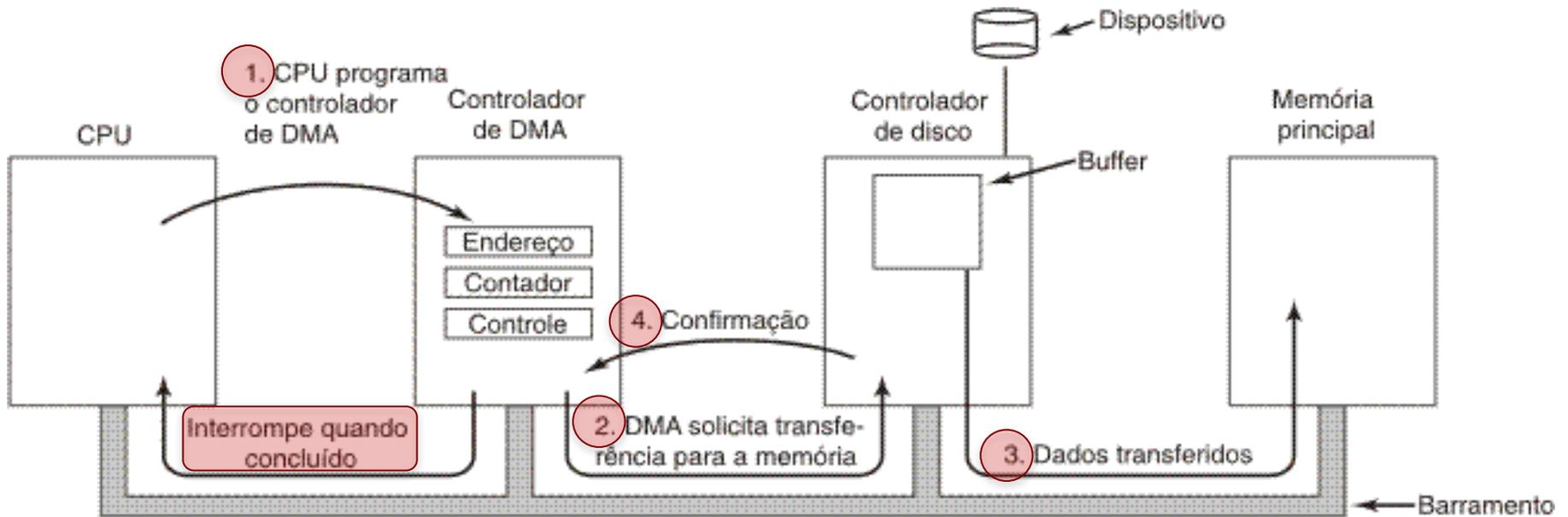
(a) Arquitetura com barramento único

(b) Arquitetura com barramento duplo (dual)

# Como o processador “enxerga” a memória e os demais dispositivos ou como o processador se comunica com o seu exterior

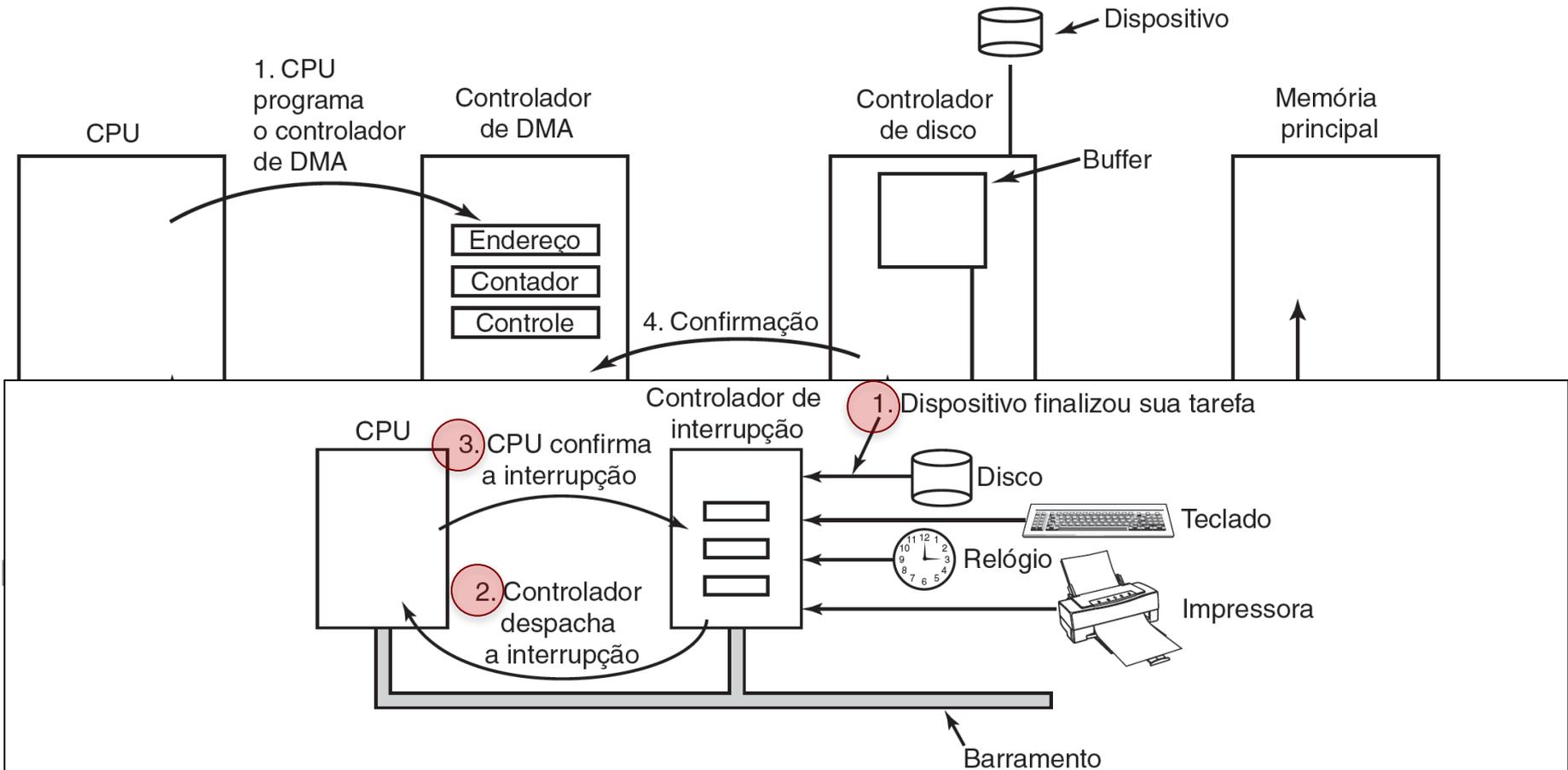
- O processador realiza operações como:
  - Ler um dado da memória
  - Escrever um dado na memória
  - Receber (ler) um dado de dispositivos de E/S
  - Enviar (escrever) dados para dispositivos de E/S
- Nas operações de **acesso à memória**, o processador escreve e lê dados, **praticamente sem intermediários**
- Nos **acessos a dispositivos de E/S**, existem circuitos intermediários, que são as **interfaces**

# Acesso Direto à Memória (DMA)



Operação de uma transferência com DMA

# Revisitando 'interrupções'



**Figura 5.4** Como ocorre uma interrupção. As conexões entre os dispositivos e o controlador de interrupção atualmente utilizam linhas de interrupção no barramento, em vez de cabos dedicados.

# Como a CPU sabe que o dispositivo já executou o comando?

- E/S Programada
  - CPU lê constantemente o status do controlador e verifica se já acabou (Polling ou Busy-waiting)
  - Desvantagem: Espera até o fim da operação
- E/S por Interrupção
  - CPU é interrompida pelo módulo de E/S e ocorre transferência de dados
  - CPU continua a executar outras operações
  - Desvantagem: toda palavra lida do (ou escrita no) periférico passa pela CPU
- E/S por DMA - Acesso Direto à Memória
  - Quando necessário, o controlador de E/S solicita ao controlador de DMA a transferência de dados de/para a memória
  - Nesta fase de transferência não há envolvimento da CPU
  - Ao fim da transferência, a CPU é interrompida e informada da transação [figura anterior]

# Comunicação S.O.(CPU) – Controlador

## Exemplo de comunicação com dispositivo



# Entrada/Saída

- ✓ Princípios do hardware de E/S
- Princípios do software de E/S
- Camadas do software de E/S
- Gerenciamento de energia

# Objetivos da gerência de E/S

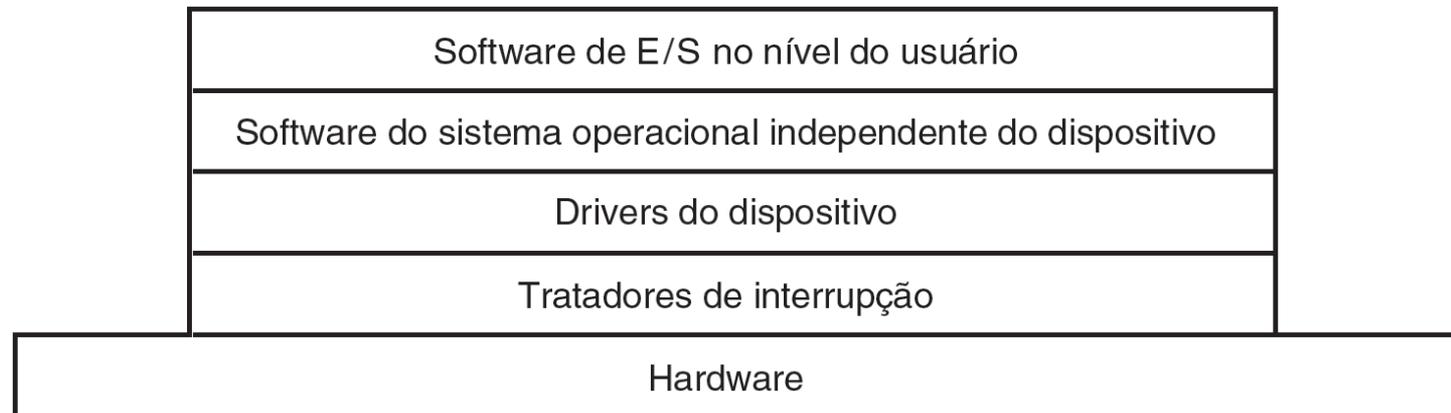
- Eficiência
- Uniformidade (desejável, diante da alta **diversidade**, associada a **heterogeneidade**):
  - Todos dispositivos enxergados da forma mais uniforme possível
- Esconder os detalhes (estes são tratados pelas camadas de mais baixo nível)
- Fornecer abstrações genéricas: read, write, open, close etc.

# Princípios básicos do software de E/S

- Subsistema de E/S é complexo, dada a **diversidade** de periféricos
- Padronizar ao máximo para reduzir número de rotinas
  - Novos dispositivos não alteram a visão do usuário em relação ao SO
- Organizado em camadas

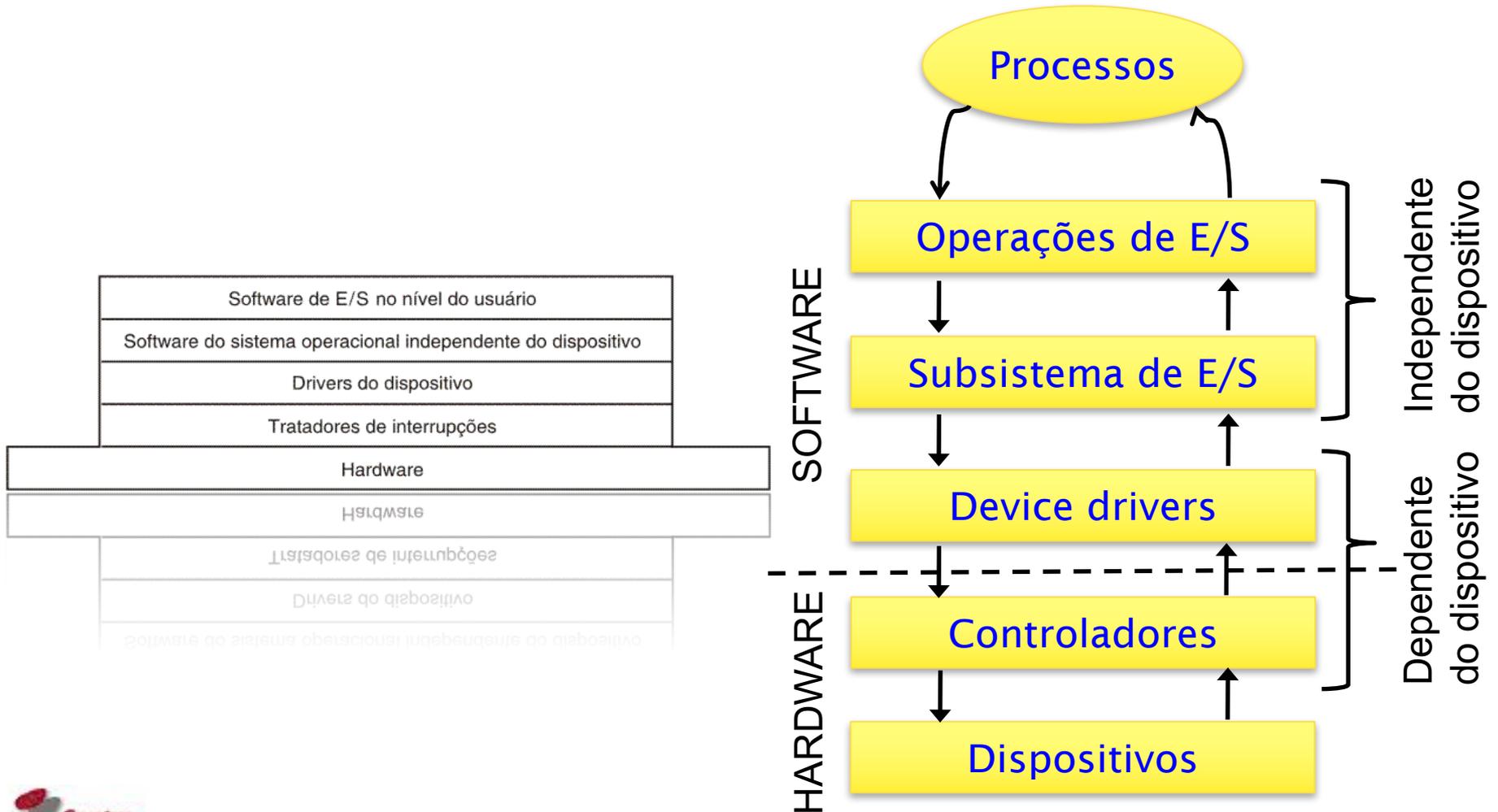
# Visão Geral do software de E/S

- Tratador de interrupção
  - É acionado ao final da operação de transferência
  - Aciona driver
- Driver de dispositivo
  - Recebe requisições
  - Configura (aciona) o controlador
- E/S independente de dispositivo
  - Nomes e proteção
  - bufferização
- E/S em nível de usuário
  - Chamadas de E/S (*System Calls*)

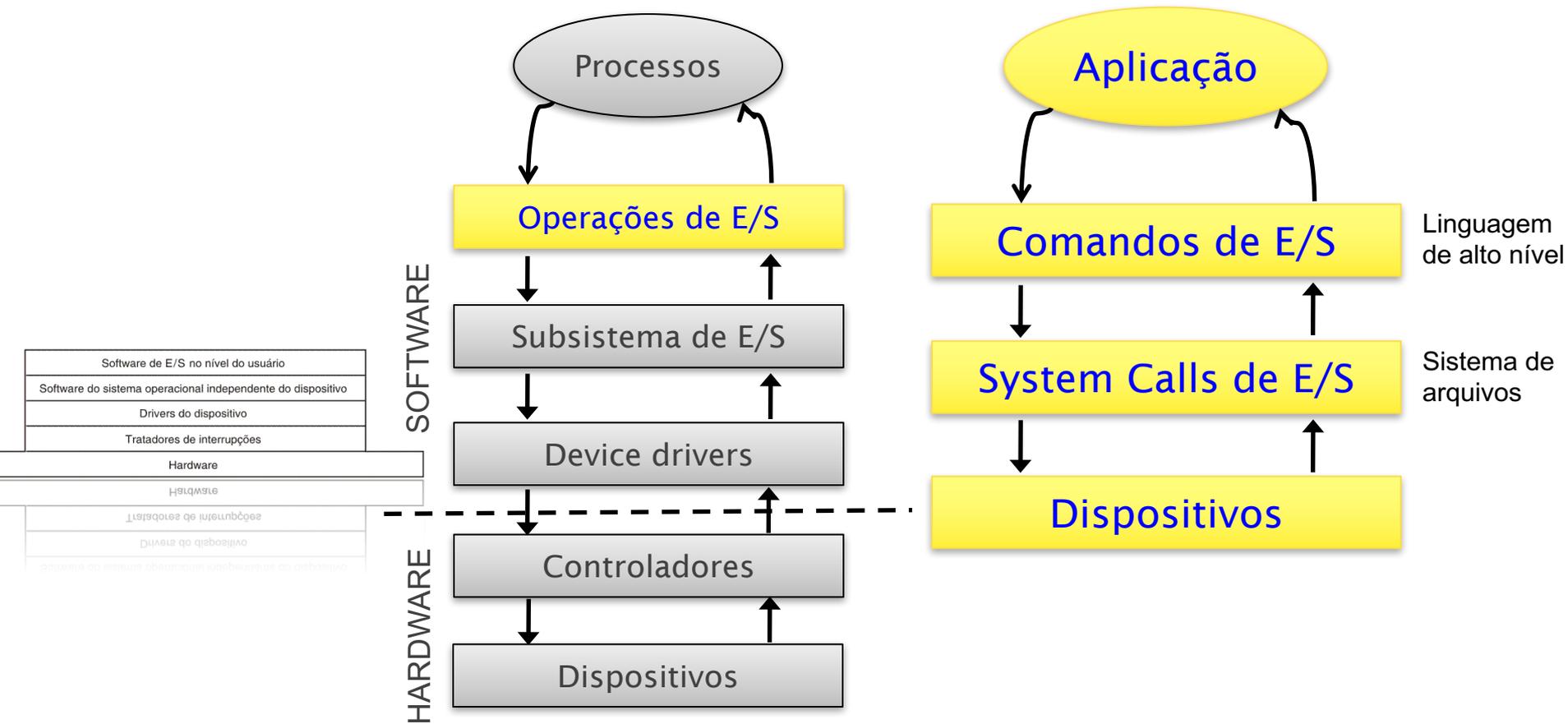


**Figura 5.10** Camadas do software de E/S.

# Camadas do Software de E/S



# Camadas do Software de E/S

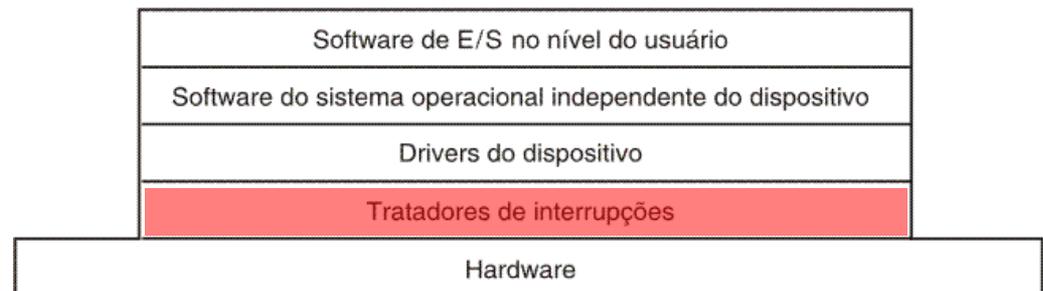


# Tratador(es) de Interrupção

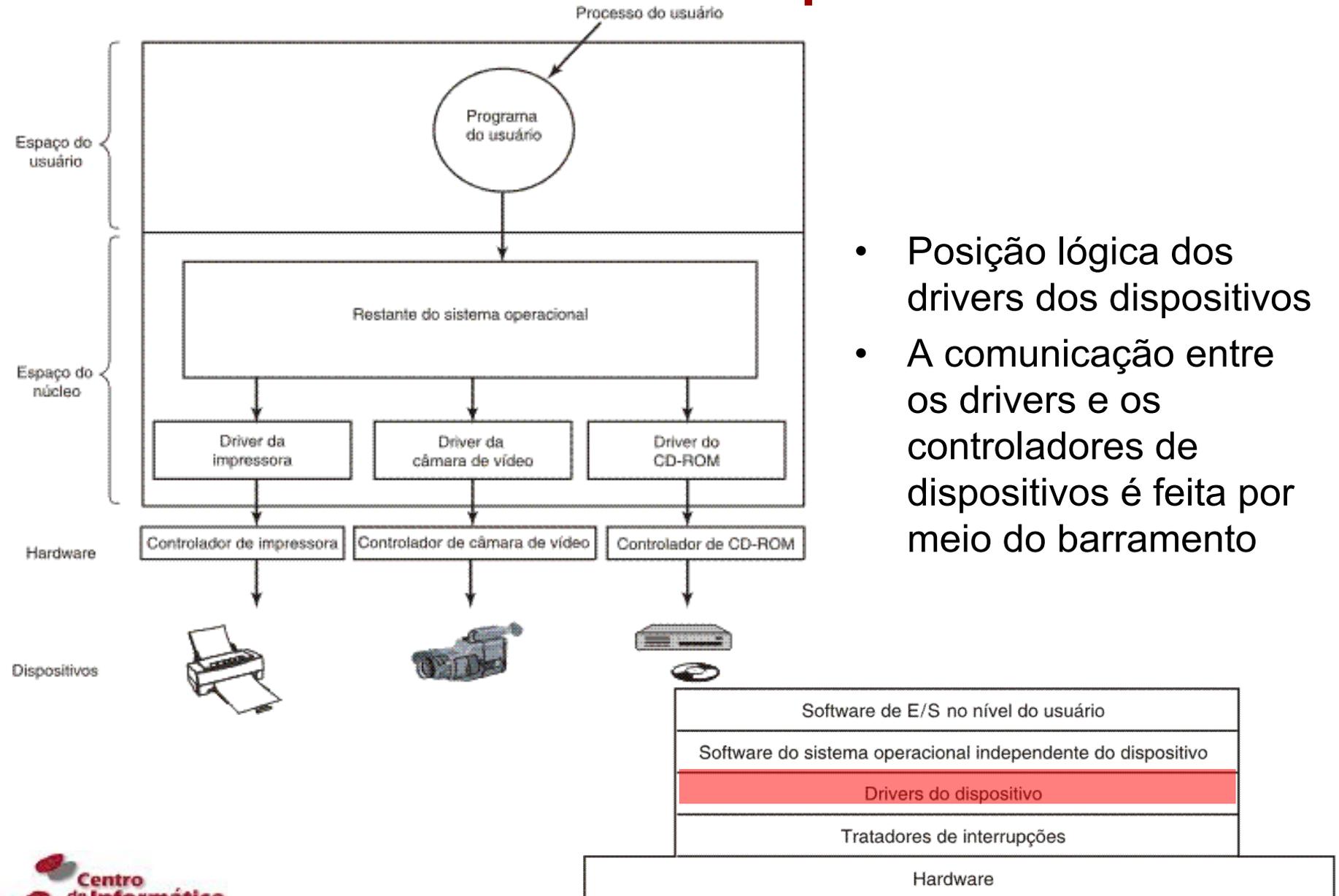
- As interrupções devem ser escondidas (transparentes) o máximo possível

Para tanto:

- bloqueia o *driver* que iniciou uma operação de E/S
- rotina de tratamento de interrupção cumpre sua tarefa
- notifica que a E/S foi completada
- e então desbloqueia o *driver* que a chamou



# Drivers dos Dispositivos



- Posição lógica dos drivers dos dispositivos
- A comunicação entre os drivers e os controladores de dispositivos é feita por meio do barramento

# Software de E/S Independente de Dispositivo

Funções do software de E/S independente de dispositivo

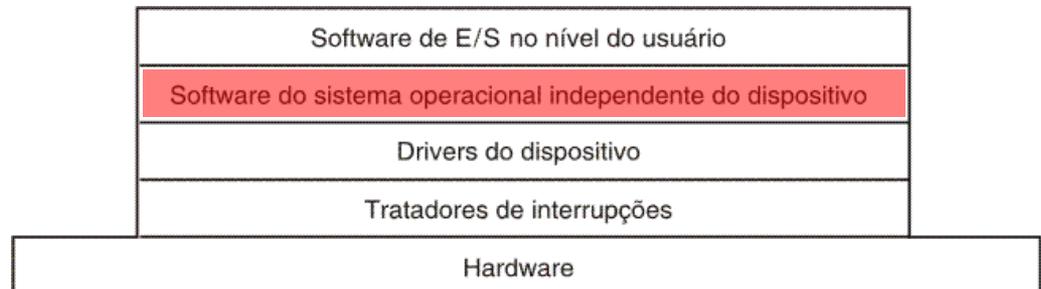
Interface uniforme para os drivers dos dispositivos

Armazenamento em buffer

Relatório de erros

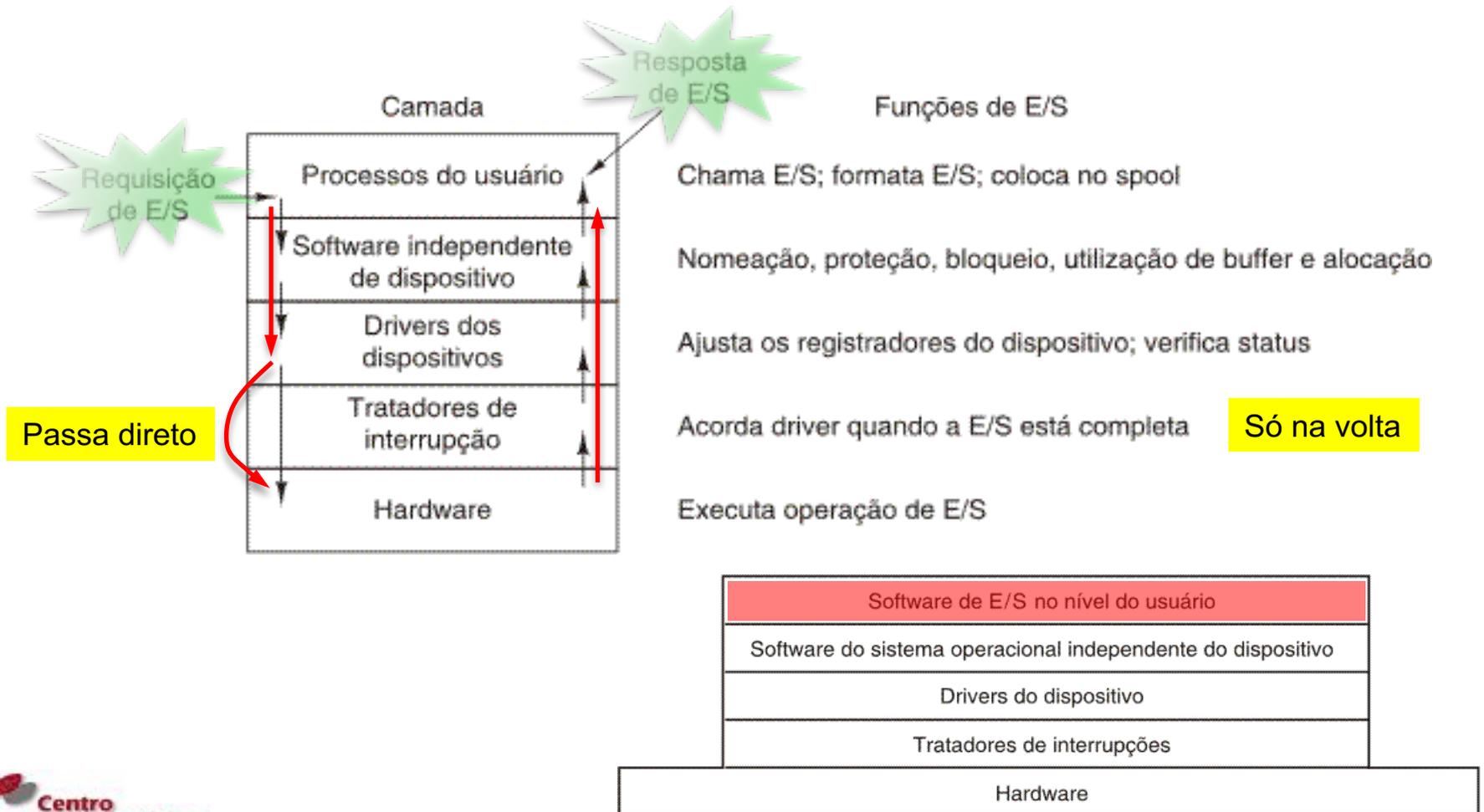
Alocação e liberação de dispositivos dedicados

Fornecimento de tamanho de bloco independente de dispositivo



# Software de E/S do Espaço do Usuário

## Camadas do sistema de E/S e as principais funções de cada camada



# **Entrada/Saída**

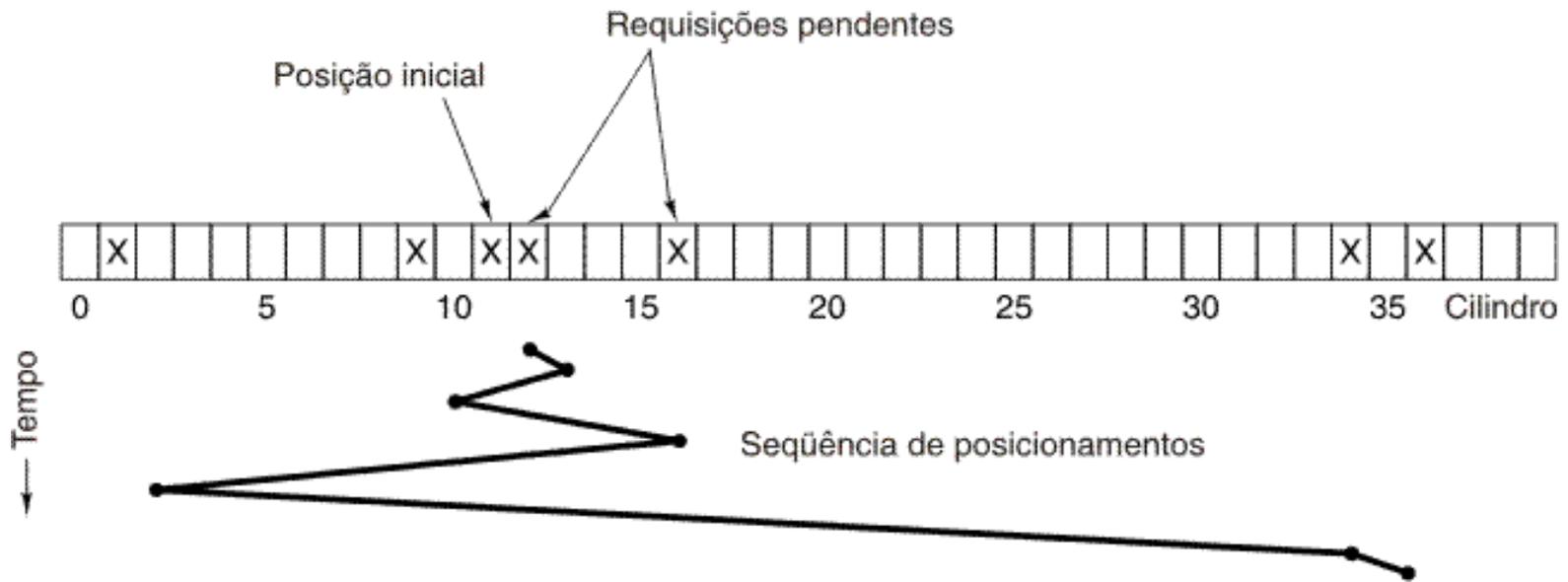
## **(Curiosidades)**

- **E/S em disco**
- **Gerenciamento de energia**

# Algoritmos de Escalonamento de Braço de Disco (1)

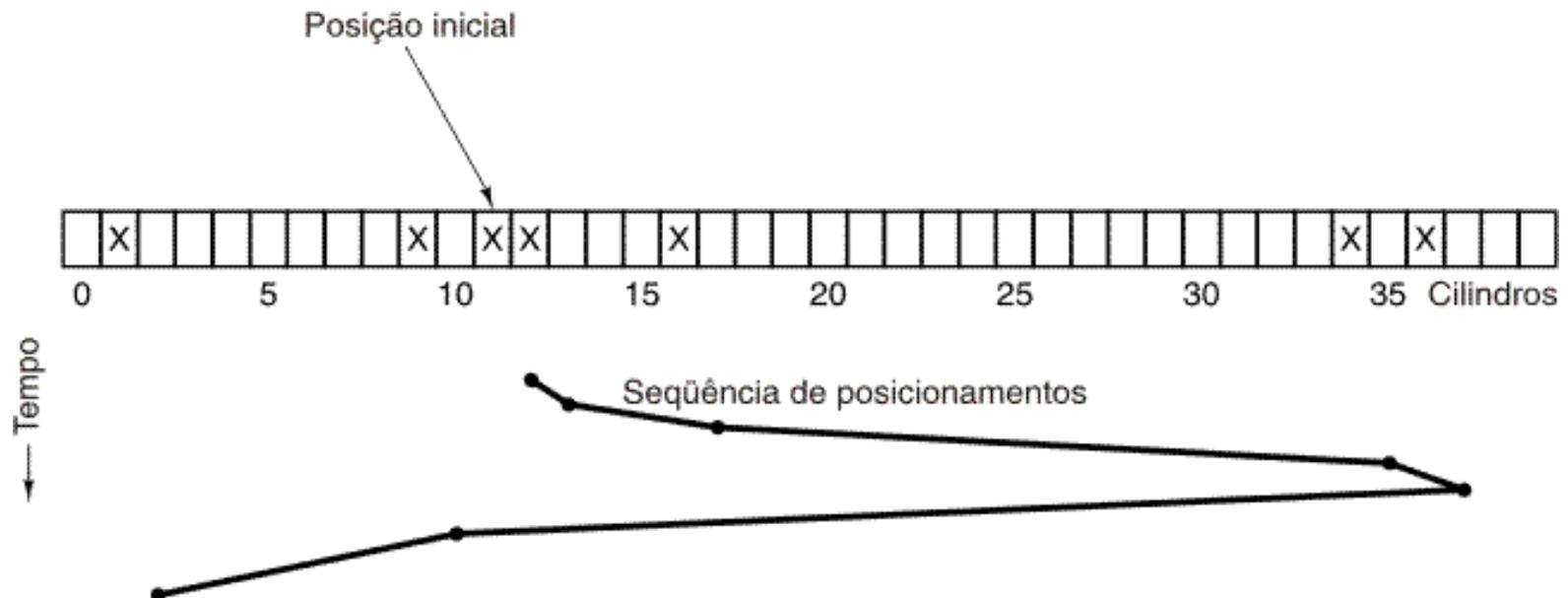
- Tempo necessário para ler ou escrever um bloco de disco é determinado por 3 fatores
  1. tempo de posicionamento
  2. atraso de rotação
  3. tempo de transferência do dado
- Tempo de posicionamento domina
- Checagem de erro é feita por controladores

# Algoritmos de Escalonamento de Braço de Disco (2)



Algoritmo de escalonamento de disco *Posicionamento Mais Curto Primeiro (SSF)*

# Algoritmos de Escalonamento de Braço de Disco (3)



O algoritmo do elevador para o escalonamento das requisições do disco

# Entrada/Saída

## (Curiosidades)

- ✓ E/S em disco
- Gerenciamento de energia

# Gerenciamento de Energia (1)

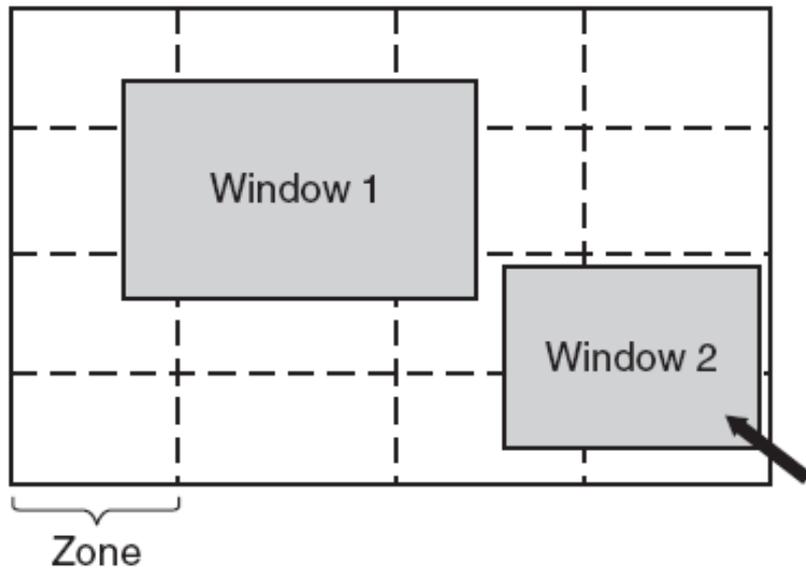
Dispositivo	Li et al. (1994)	Lorch e Smith (1998)
Monitor de vídeo	68%	39%
CPU	12%	18%
Disco rígido	20%	12%
Modem		6%
Som		2%
Memória	0,5%	1%
Outros		22%

*Tecnologia mais avançada* →

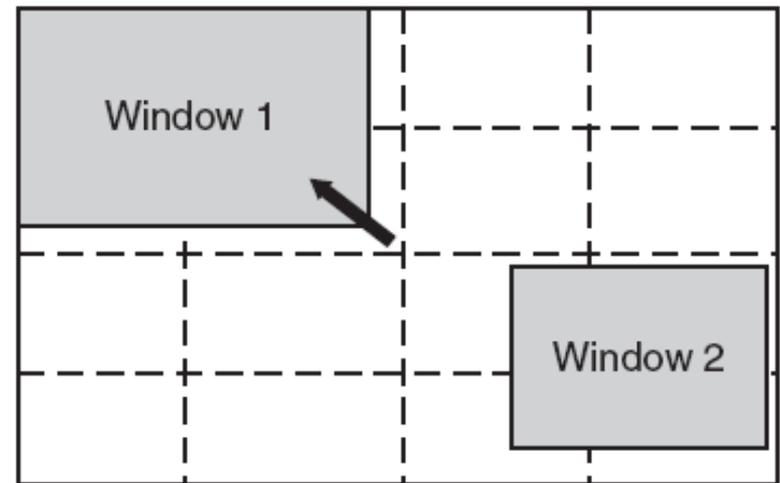
*Aumento do desempenho* →

Consumo de energia de várias partes de um laptop

# Gerenciamento de Energia (2): O uso de zonas para iluminação do monitor de vídeo



(a)

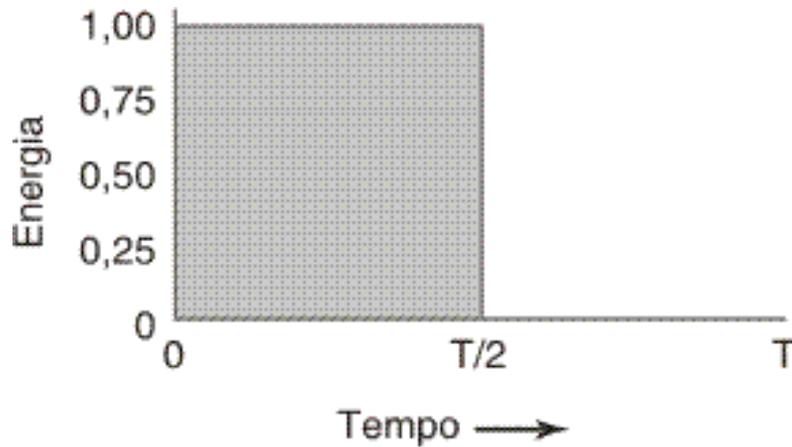


(b)

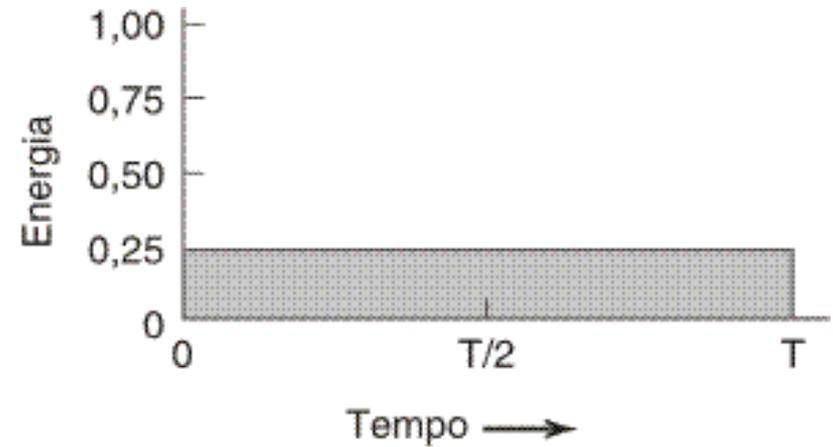
(a) Janela 2 selecionada e não se move.

(b) Janela 1 selecionada e se move para reduzir o número de zonas iluminadas.

# Gerenciamento de Energia (3)



(a)



(b)

- a. Execução em velocidade máxima do relógio
- b. Cortando a voltagem pela metade
  - corta a velocidade do relógio também pela metade,
  - consumo de energia cai para 4 vezes menos

# Gerenciamento de Energia (4): Impactos na Entrada/Saída

- Dizer aos programas para usar menos energia
  - pode significar experiências mais pobres para o usuário
- Exemplos
  - muda de saída colorida para preto e branco
  - reconhecimento de fala com vocabulário reduzido
  - menos resolução ou detalhe em uma imagem

# Conclusões: princípios básicos do software de E/S

- Subsistema de E/S é complexo dada a diversidade de periféricos
- Padronizar ao máximo para reduzir número de rotinas
  - Novos dispositivos não alteram a visão do usuário em relação ao SO
- Organizado em camadas