

# Sistemas Operacionais

Sistema de entrada e Saída

# Sistema de Entrada e Saída I/O

---

- É uma das principais tarefas de um sistema computacional
- Como máquina abstrata o S.O. deve oferecer uma visão padronizada dos dispositivos de Entrada e Saída
- Controlar acesso concorrente
- Camadas de abstração
- Dispositivos não-confiáveis

# I/O uma das principais tarefas de um computador

---

- Toda computação começa com uma entrada de dados
- Toda computação termina com uma saída de dados
- O tempo de resposta dos dispositivos de I/O são MUITO diferentes (comumente maiores) que o tempo de componentes como memória e processador

# Tempo de resposta de dispositivos

Dispositivo	Taxa de dados
Teclado	10 bytes/s
Mouse	100 bytes/s
Modem 56 K	7 KB/s
Canal telefônico	8 KB/s
Linhas ISDN dual	16 KB/s
Impressora a laser	100 KB/s
Scanner	400 KB/s
Ethernet clássica	1,25 MB/s
USB ( <i>universal serial bus</i> — barramento serial universal)	1,5 MB/s
Câmara de vídeo digital	4 MB/s
Disco IDE	5 MB/s
CD-ROM 40x	6 MB/s
Ethernet rápida	12,5 MB/s
Barramento ISA	16,7 MB/s
Disco EIDE (ATA-2)	16,7 MB/s
FireWire (IEEE 1394)	50 MB/s
Monitor XGA	60 MB/s
Rede SONET OC-12	78 MB/s
Disco SCSI Ultra 2	80 MB/s
Ethernet Gigabit	125 MB/s
Dispositivo de Fita Ultrium	320 MB/s
Barramento PCI	528 MB/s
Barramento da Sun Gigaplane XB	20 GB/s

# Visão padronizada dos dispositivos de Entrada e Saída

---

- Usuário não deve se preocupar se o dispositivo está ligado numa placa ou na placa mãe
- Usuário não deve ter de se preocupar com o tempo de resposta
- Usuário não deve ter de se preocupar com as características mecânicas do dispositivo
- O SO deve tratar da melhor maneira possíveis falhas de hardware

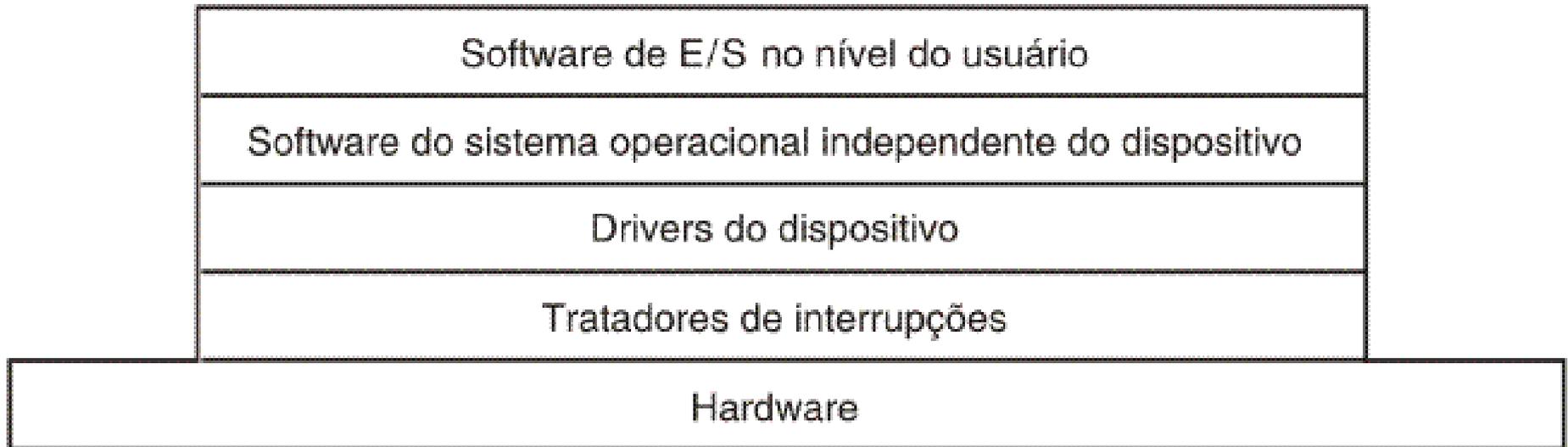
# Controlar acesso concorrente

---

- Alguns dispositivos não são preemptíveis
  - Portas Seriais
- Alguns deles nem são compartilháveis
  - Impressora
  - Gravador de Cds
- O sistema deve evitar situações de deadlock e prover acesso justo a todos os recursos por todos os processos

# Arquitetura em camadas

---



# Tipos de Dispositivos

---

- Dispositivos de Bloco
- Dispositivos de caractere

# Dispositivos de Bloco

---

- Acesso em conjuntos de  $n$  Bytes (Blocos)
- Cada bloco tem um endereço diferente
- Transferências são em um ou mais blocos
- Ex.:
  - Disco
  - Pen drive
  - CDROM

# Dispositivos de Caractere

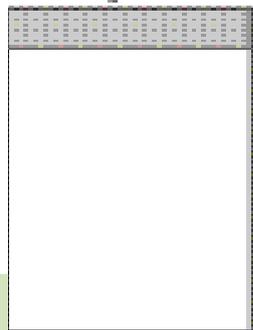
---

- Acesso em Bytes
- Transferências em fluxo de caracteres
- Não endereçável
- Ex.:
  - Impressoras
  - Rede
  - Mouse

# I/O Mapeada em memória

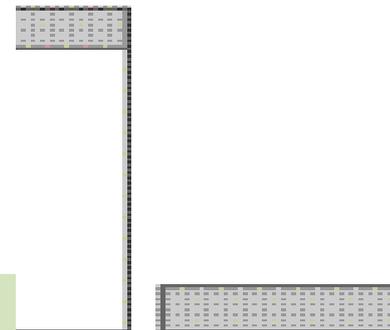
- Os registradores e o buffer de memória ficam em um endereço fixo na memória
- O acesso aos dispositivos é feito do mesmo jeito que se faz com a memória
- Pode ser híbrido com os registradores em um espaço de endereços separado

espaço de  
endereçamento



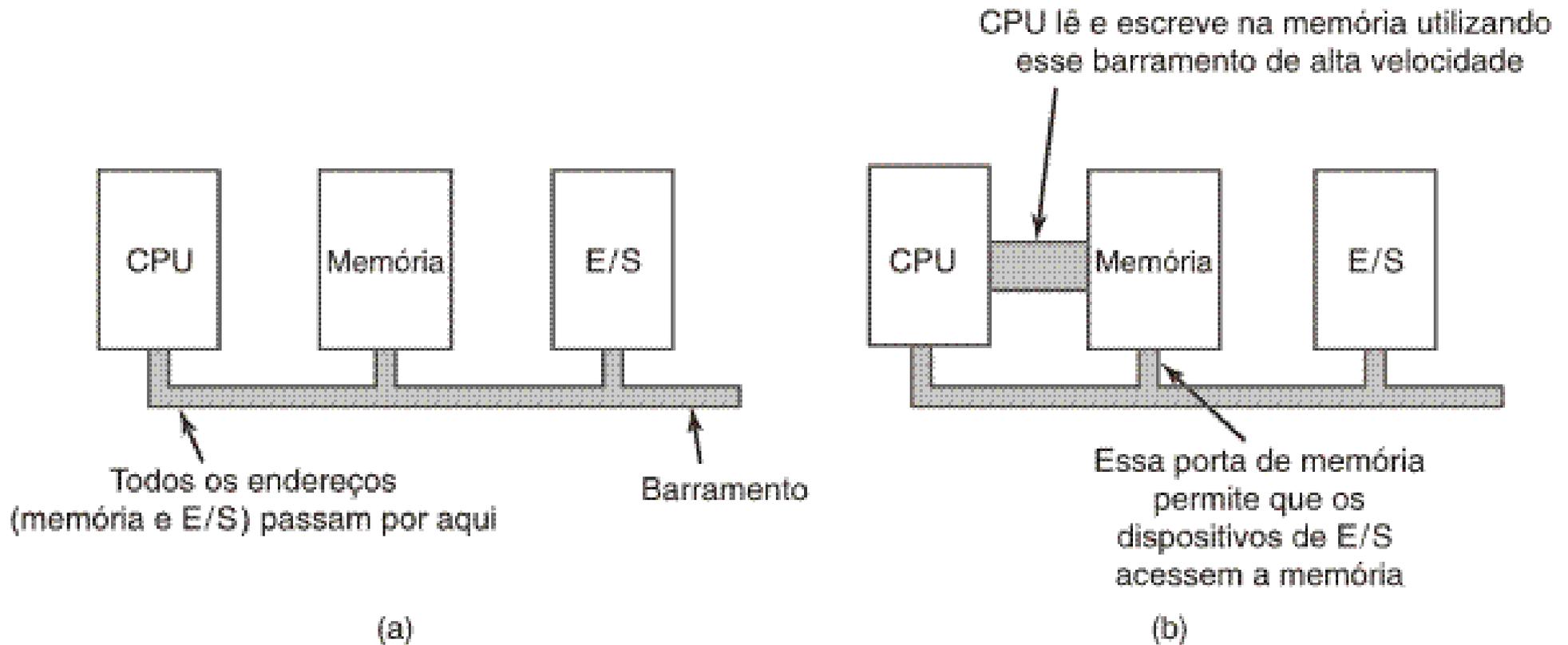
(b)

espaço de  
endereçamento



(c)

# Conexão da CPU com os dispositivos de I/O



# Vantagens do I/O mapeado em memória

---

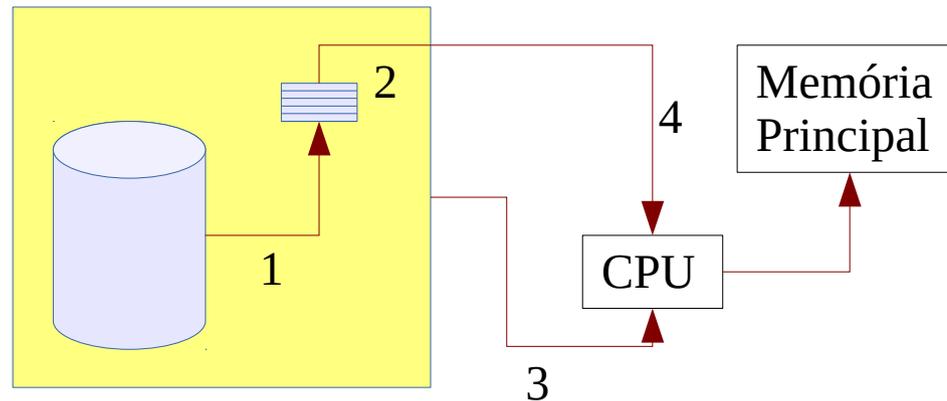
- Programação simplificada: o acesso é feito como um espaço de memória comum
- A proteção a áreas de I/O é mais simples basta nunca oferecer a área de memória de I/O a um processo
- Todas funções de tratamento de memória podem ser usadas para dispositivos de I/O

# Desvantagens do I/O mapeado em memória

---

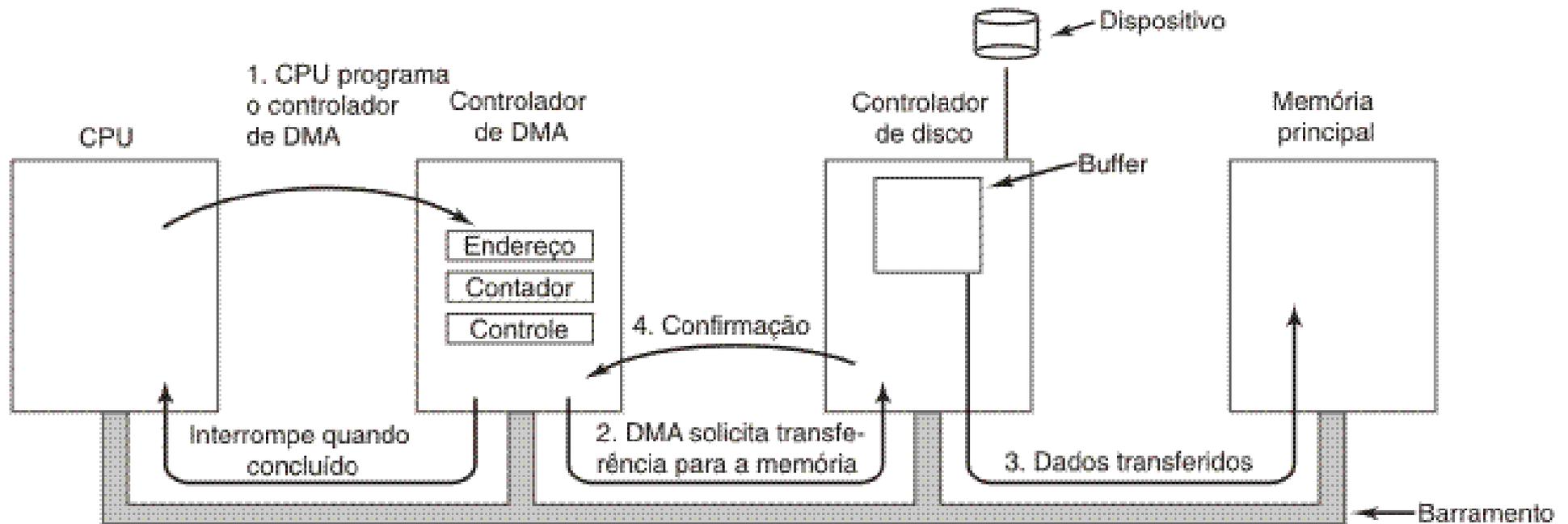
- Deve desabilitar cache de memória
- Quando um pedido de memória é feito todos os dispositivos de I/O e a memória precisam receber o sinal
  - Fácil com um só barramento
  - Complexo para multi-barramentos (Pentium)
  - Possível solução: pedir sempre para a memória se falhar pedir aos dispositivos

# Transferência entre o dispositivo e a memória Sem DMA



- 1 – Controlador Lê do disco
- 2 – Controlador Checa Integridade
- 3 – Controlador Interrompe a CPU
- 4 – SO em loop copia os dados do controlador para memória principal

# Transferência entre o dispositivo e a memória Com DMA



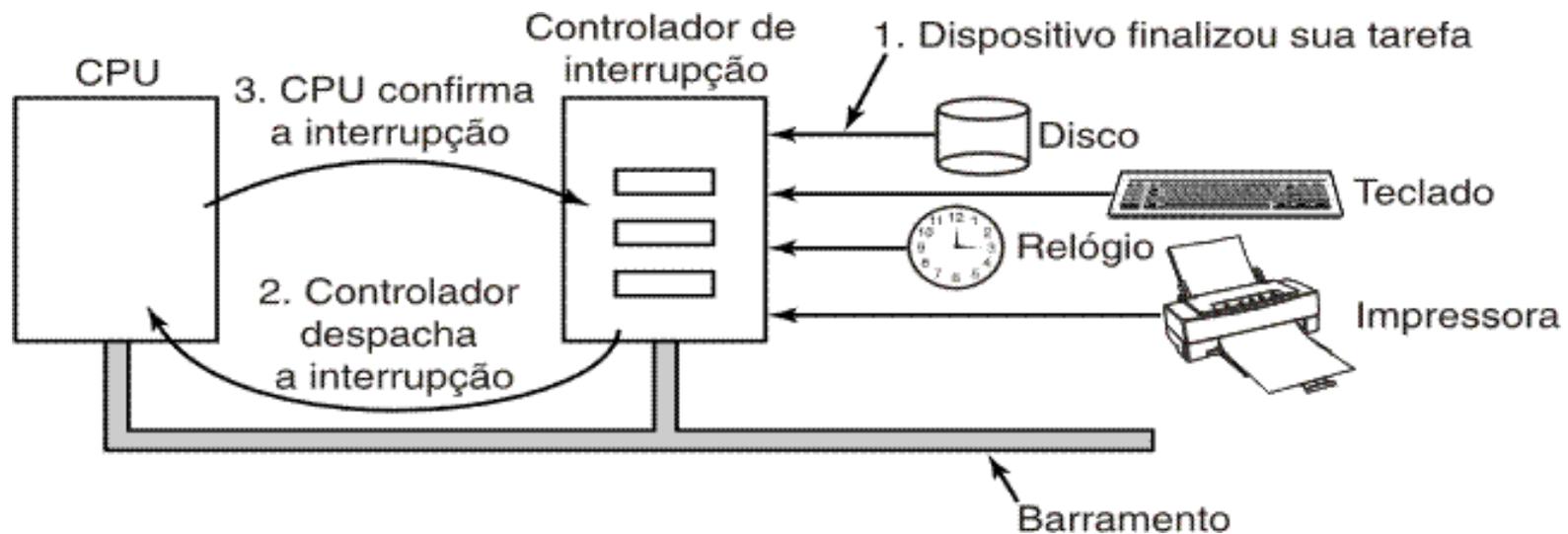
# Desvantagens de usar DMA

---

- A CPU é muito mais rápida que o controlador de DMA
- Se os dispositivos de I/O não são tão lentos é melhor que a CPU espere pelo I/O
- Usando DMA a CPU espera pelo controlador DMA que espera pelo dispositivo
- Em sistemas embarcados isso implica em um custo por vezes muito alto

# Interrupções

- Provêem comunicação assíncrona com a CPU
- Funcionamento:



# Gerenciadores de Interrupções

---

- Nível mais baixo antes do físico
- São pequenos trechos de código executados quando ocorre uma interrupção
- É a responsável por comunicar aos níveis superiores quando uma operação foi completada
- Devem ser escondidos de outros blocos do S.O.

# Gerenciador de Interrupções

---

1. salva registradores que ainda não foram salvos pelo hardware de interrupção
2. estabelece contexto para rotina de tratamento de interrupção
3. estabelece uma pilha para a rotina de tratamento de interrupção
4. sinaliza o controlador de interrupção, reabilita as interrupções
5. copia os registradores de onde eles foram salvos
6. executa rotina de tratamento de interrupção
7. escolhe o próximo processo a executar
8. estabelece o contexto da MMU para o próximo processo a executar
9. carrega os registradores do novo processo
10. começa a executar o novo processo

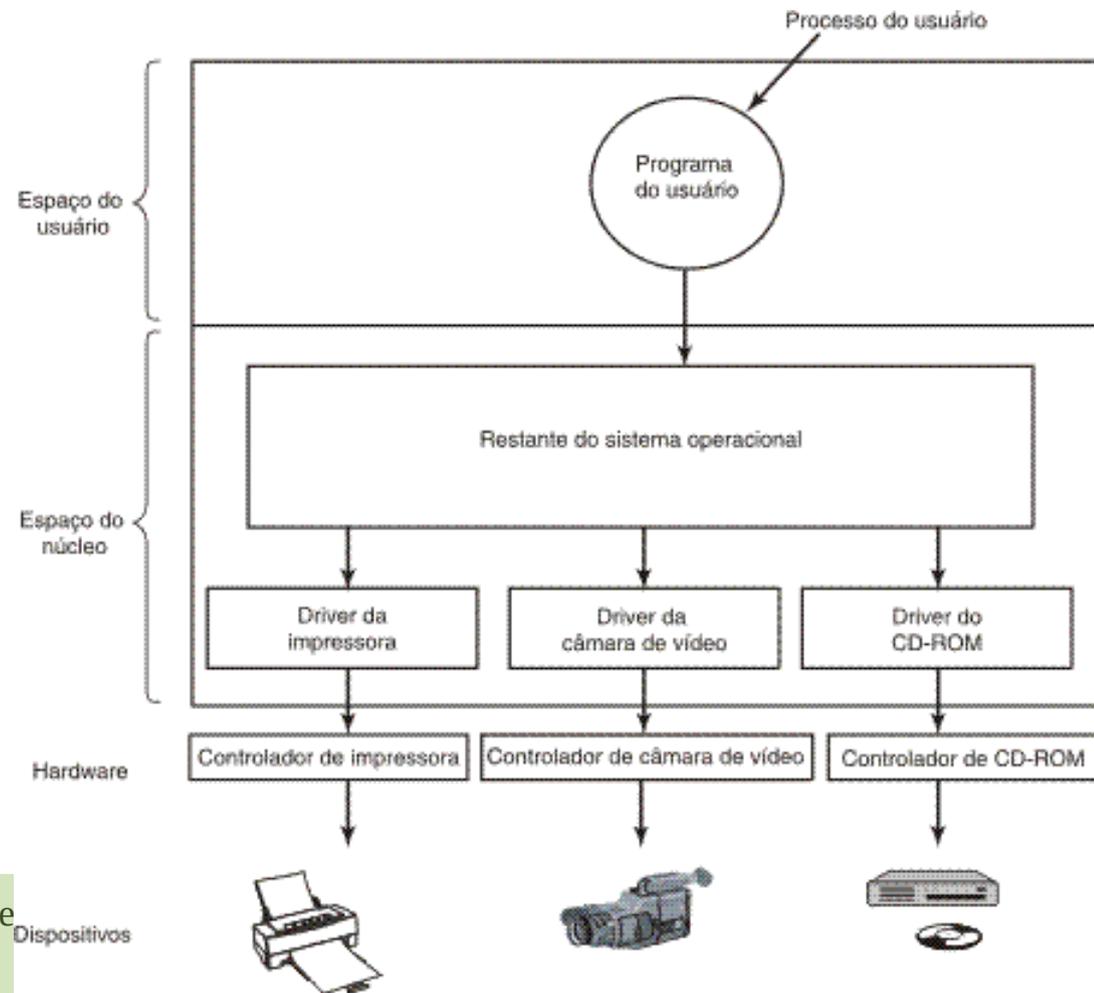
# Drivers de dispositivos

---

- Contem todo o código dependente do dispositivo
- Comunica-se com as controladoras enviando os comandos e preenchendo registradores
- Tradução de instruções abstratas para comandos concretos do dispositivo
- Tratamento de possíveis erros do dispositivo
- O criador do hardware desenvolve para o S.O.
- Espaço do Kernel X Espaço do usuário

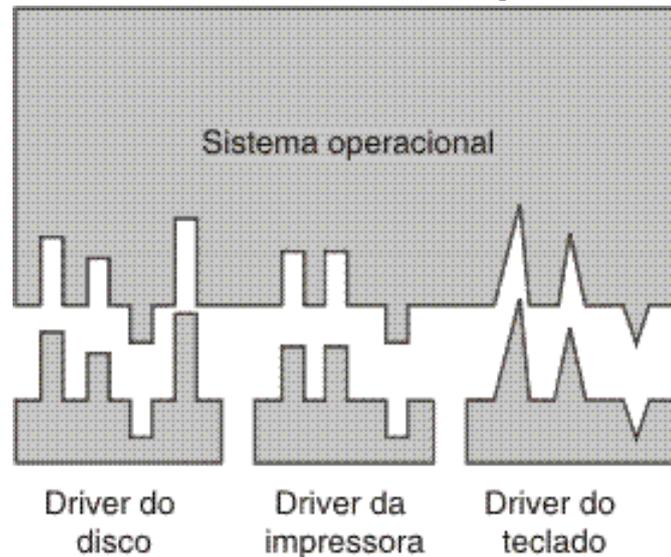
# Drivers de Dispositivos

- O SO oferece uma interface padrão para o desenvolvimento de drivers
  - De bloco
  - De caractere

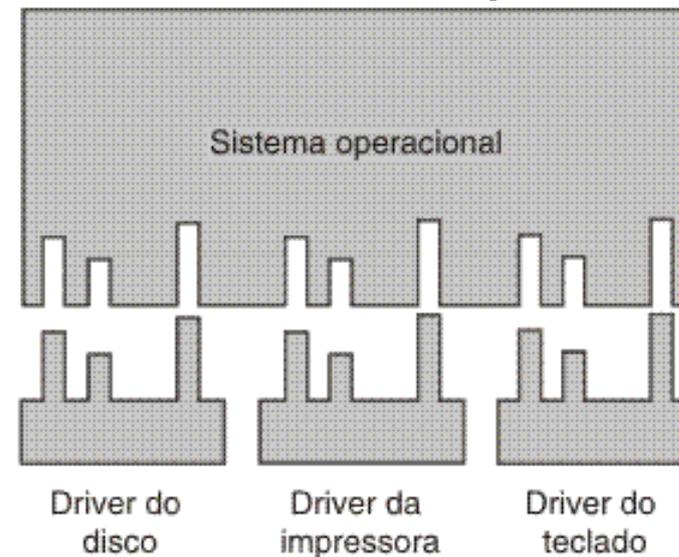


# Software Independente de dispositivo(1)

- Interface uniforme para os drivers dos dispositivos



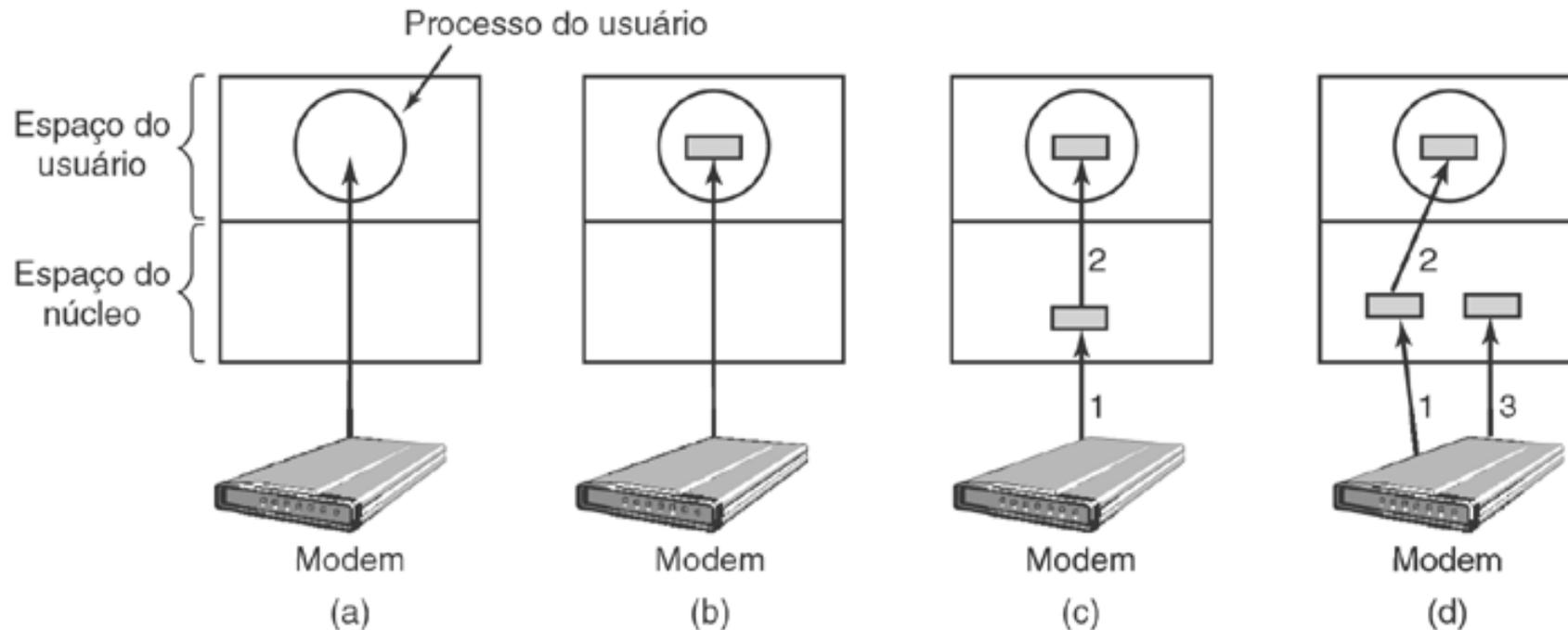
(a)



(b)

# Software Independente de dispositivo(2)

- Armazenamento em buffer



- a) Entrada sem utilização de buffer
- b) Utilização de buffer no espaço do usuário
- c) Utilização de buffer no núcleo seguido de cópia para o espaço do usuário
- d) Utilização de buffer duplo no núcleo

# Visão geral do Software de I/O

