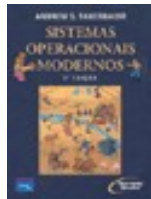


# Capítulo 5

## Entrada/Saída

- 5.1 Princípios do hardware de E/S**
- 5.2 Princípios do software de E/S**
- 5.3 Camadas do software de E/S**
- 5.4 Discos
- 5.5 Relógios
- 5.6 Terminais com base em caracteres
- 5.7 Interfaces gráficas do usuário
- 5.8 Terminais de rede
- 5.9 Gerenciamento de energia

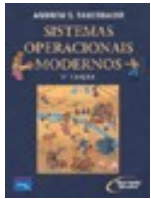
# Princípios do Hardware de E/S



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

Taxas de dados típicas de dispositivos, redes e barramentos

# Controladores de Dispositivos



- Componentes de dispositivos de E/S
  - mecânico
  - eletrônico
- O componente eletrônico é o controlador do dispositivo
  - pode ser capaz de tratar múltiplos dispositivos
- Tarefas do controlador
  - converter fluxo serial de bits em bloco de bytes
  - executar toda detecção e correção de erros necessária
  - tornar o bloco disponível para ser copiado para a memória principal

# Interface vs Controlador



- Interface:
  - Define o padrão de comunicação e interconexão entre o barramento e o dispositivo.
  - É basicamente a documentação
    - os comandos e como usá-los
    - os códigos de erro
  - Exemplos:
    - ATA, SATA, SAS, etc...

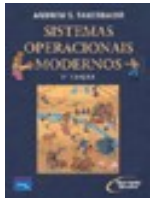
# Interface vs Controlador



- Controlador:
  - É a instanciação da interface.
  - Cada fabricante projeta e constrói a sua porém, para permitir a interconexão com o barramento e sua operação ela deverá adotar um padrão (ou interface)
  - Exemplo:
    - Controladora do disco rígido
    - Placas de rede

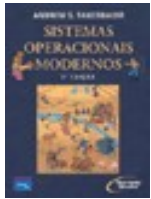
# Princípios do Software de E/S

## Objetivos do Software de E/S (1)



- Independência de dispositivo
  - Programas podem acessar qualquer dispositivo de E/S sem especificar previamente qual (disquete, disco rígido ou CD-ROM)
- Nomeação uniforme
  - Nome de um arquivo ou dispositivo pode ser uma cadeia de caracteres ou um número inteiro que é independente do dispositivo
- Tratamento de erro
  - Trata o mais próximo possível do hardware

# Objetivos do Software de E/S (2)



- Transferências Síncronas vs. Assíncronas
  - transferências bloqueantes vs. orientadas a interrupção
  - utilização de buffer para armazenamento temporário
  - dados provenientes de um dispositivo muitas vezes não podem ser armazenados diretamente em seu destino final
- Dispositivos Compartilháveis vs. Dedicados
  - discos são compartilháveis
  - Impressoras não são

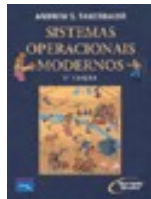
# Comandos de E/S



- CPU envia endereço
  - Identificação do módulo (e dispositivo se mais de 1 dispositivo por módulo)
- CPU envia commando
  - Controle – especifica o que o módulo deve fazer
    - Ex: Rodar disco rígido
  - Teste – checar estado
    - Ex: está ligado? Ocorreu um erro? Esta processando?
  - Leitura/Escrita
    - Transfere dados para o dispositivo através do módulo

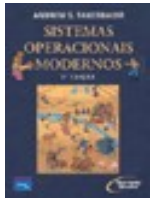


# Endereçamento de Dispositivos de E/S



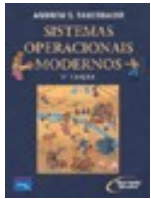
- E/S programada:
  - transferência de dados é parecida com acessos à memória
  - Do ponto de vista da CPU
- Cada dispositivo possui um identificador único associado
- Comandos da CPU contém identificadores (endereço)

# Entrada/Saída Programada



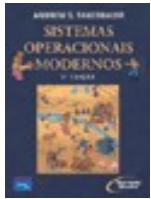
- CPU tem o controle “direto” sobre o dispositivo
  - Verifica estado (ocupado, livre, pronto, etc...)
  - Envia comando de leitura ou escrita
  - Transfere os dados
- A CPU espera até que o módulo de E/S complete a operação
- DESPERDÍCIO DE TEMPO DE PROCESSAMENTO

# E/S Programada - detalhes



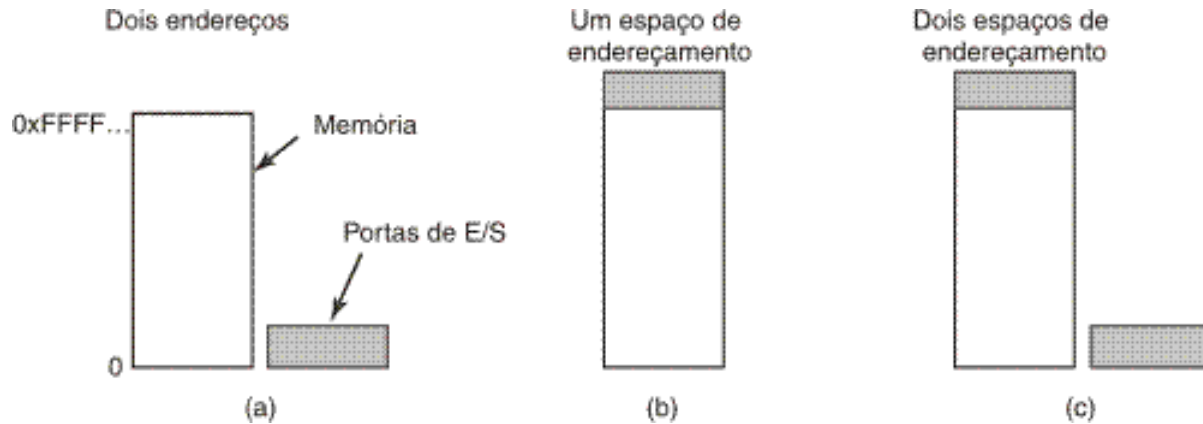
1. CPU requisita operação de E/S
2. Módulo de E/S inicia a operação
3. Módulo de E/S liga bits de estado (ocupado)
4. CPU checa bits de estado periodicamente
5. Módulo de E/S termina operação
6. Módulo de E/S liga bits de estado (pronto)
7. CPU checa bits de estado e “pega” o resultado

# E/S Programada – detalhes (2)



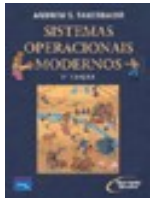
- Observações:
  - Módulo de E/S não informa a CPU diretamente
  - Módulo de E/S não interrompe a CPU
  - CPU pode esperar ou “voltar depois”

# E/S mapeada na memória (1)



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

# E/S Dirigida por Interrupções



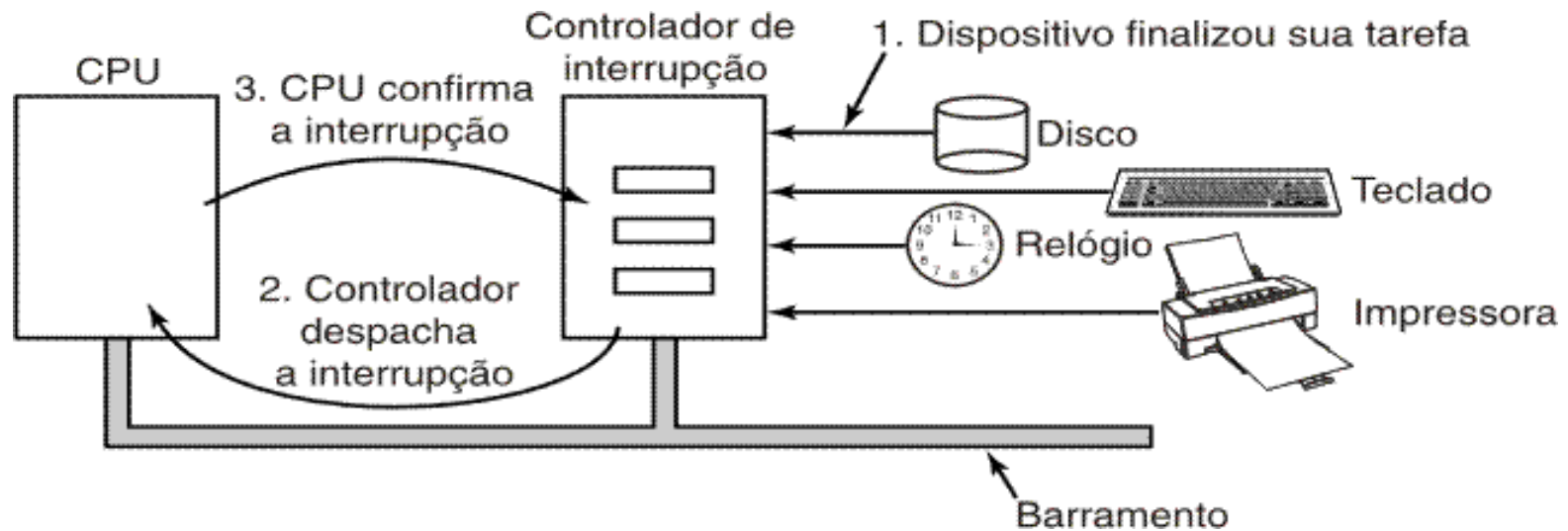
- Elimina o problema de “espera” da CPU
- A CPU não precisa checar o dispositivo repetidamente
- O módulo de E/S faz a interrupção quando terminar

# E/S Dirigida por Interrupção

## Operação Básica

1. CPU emite um comando de leitura
2. Módulo de E/S vai buscar os dados
  - Enquanto a CPU executa outras instruções
3. Módulo de E/S interrompe a CPU
  - Ao terminar de buscar os dados
4. CPU faz requisita os dados
5. Módulo de E/S transfere dados para CPU

# Interrupções Revisitadas



Como ocorre uma interrupção. Conexões entre dispositivos e controlador de interrupção usam linhas de interrupção no barramento em vez de fios dedicados

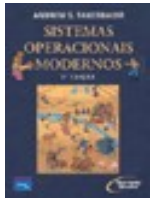


# Do ponto de vista da CPU...



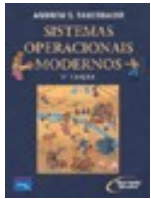
- Emite comando de leitura
- Executa outras instruções
- Verifica se houve interrupção ao final de cada ciclo de execução
- Se a CPU foi interrompida:
  - Salva contexto (Registradores: CP, MAR, MBR, ...)
  - Processa interrupção (pode levar algum tempo)
    - Busca dados e armazena
  - Restaura contexto
- CPU continua a execução de onde parou

# Acesso Direto à Memória



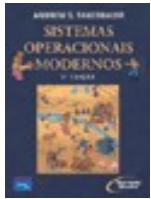
- DMA (Direct Memory Access)
- E/S dirigida por interrupção e E/S programada precisam que a CPU “trabalhe”
  - Como a taxa de transferência dos dispositivos é limitada
  - CPU fica ocupada
- DMA é a resposta

# Funcionamento do DMA



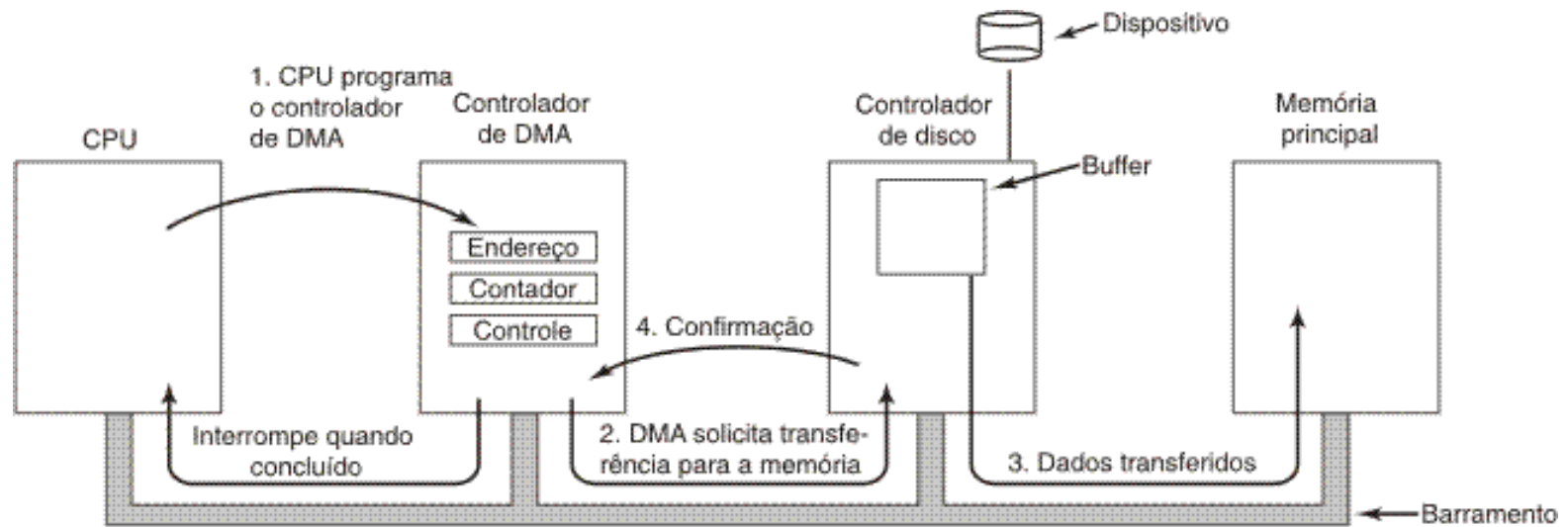
- O Controlador de DMA é um módulo adicional do barramento
- O controlador de DMA “toma” E/S da CPU

# Operação do DMA



- CPU passa ao controlador de DMA:
  - Tipo de operação: Leitura/Escrita
  - Endereço do dispositivo
  - Endereço inicial
  - Quantidade de dados a serem transferidos
- CPU vai fazer outro trabalho
- Controlador de DMA faz a transferência
- Controlador de DMA envia interrupção quando terminar

# Acesso Direto à Memória (DMA)



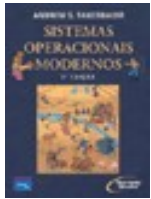
## Operação de uma transferência com DMA

# Transferência DMA

## Roubo de Ciclos

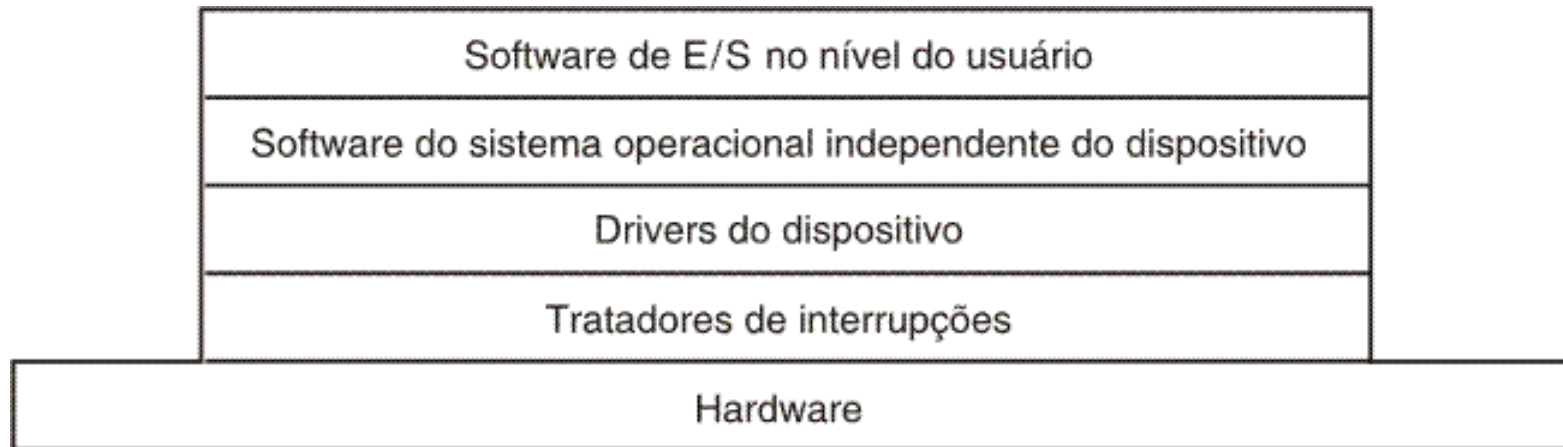
- Controlador DMA usa o barramento por um ciclo
- Transfere uma palavra de dados
- CPU é suspendida antes de acessar barramento
  - i.e. antes de buscar/escrever dados em algum dispositivo ou memória
- Deixa CPU mais lenta
  - Melhor do que a CPU fazer o trabalho diretamente

## Por outro lado...



- Qual o efeito do cache durante uma operação DMA?
  - O cache visa diminuir o acesso à memória pela CPU
  - Economiza acessos ao barramento também
  - DMA usa muito o barramento
- Por outro lado...
  - E se o endereço de memória que está sendo atualizado estiver no cache?
  - INCONSISTÊNCIA !!!
  - “Pendurar” o DMA no cache?

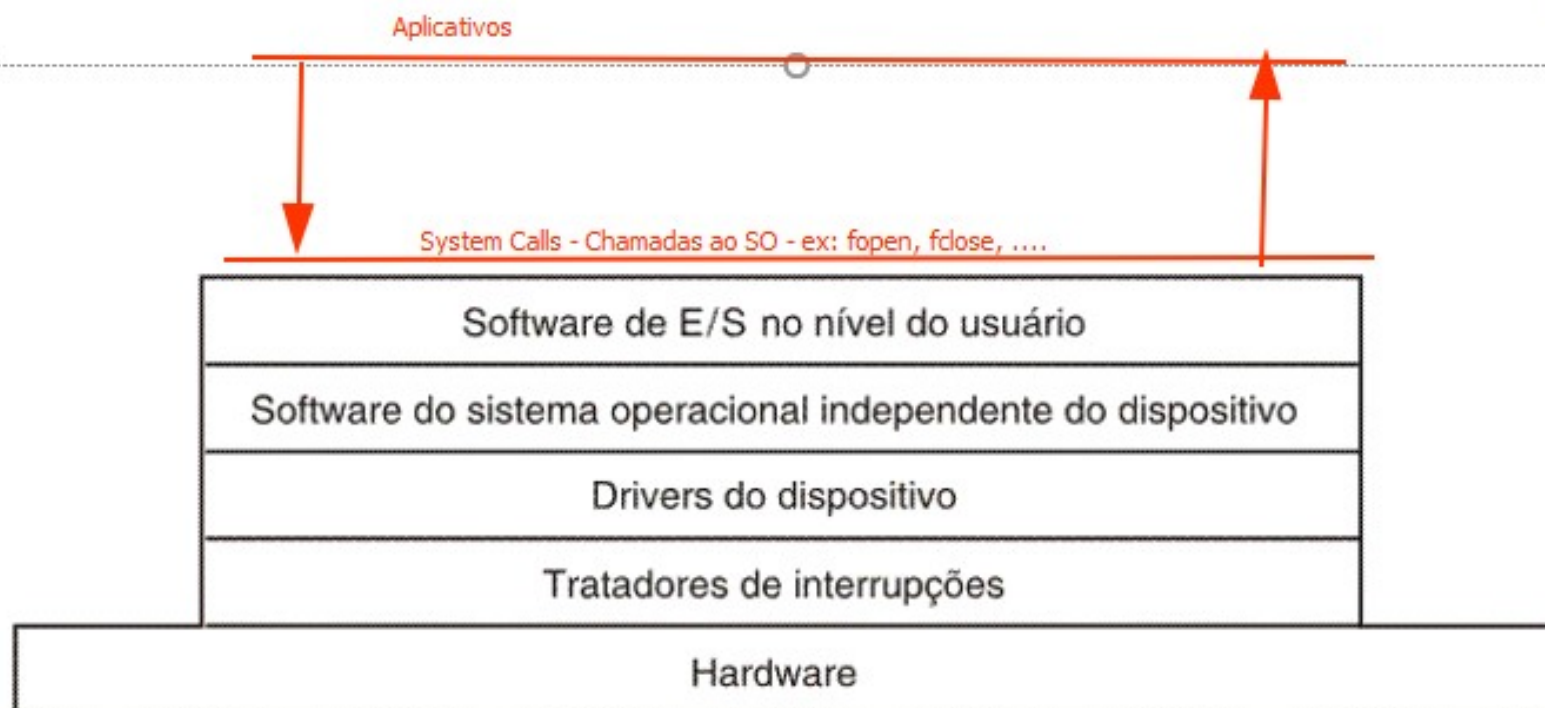
# Camadas do Software de E/S



Camadas do sistema de software de E/S

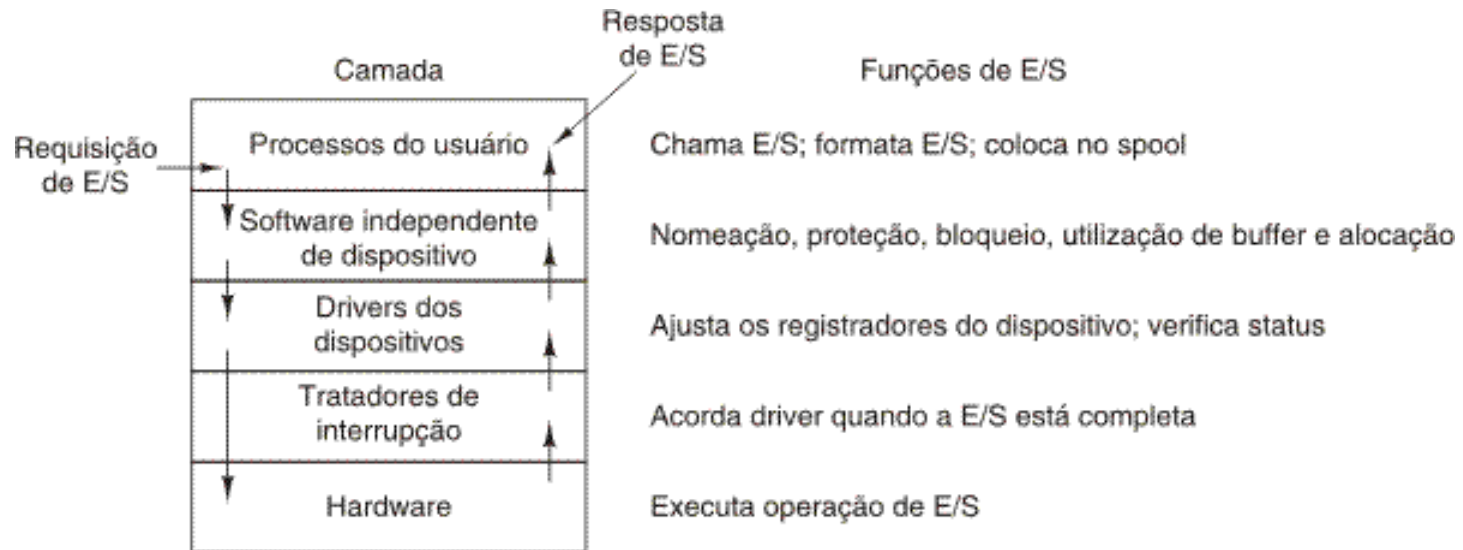


# Camadas do Software de E/S



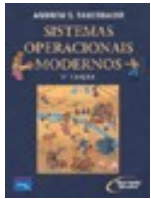
## Camadas do sistema de software de E/S

# Software de E/S no Espaço do Usuário



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

# Software de E/S Independente de Dispositivo (1)



Interface uniforme para os drivers dos dispositivos

Armazenamento em buffer

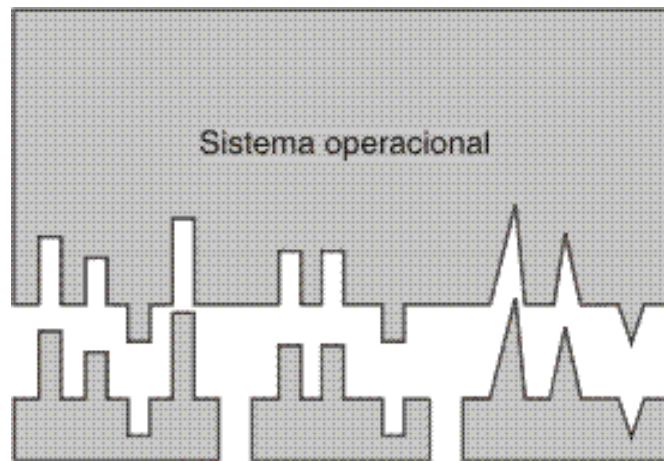
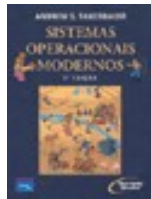
Relatório dos erros

Alocação e liberação de dispositivos dedicados

Fornecimento de tamanho de bloco independente  
de dispositivo

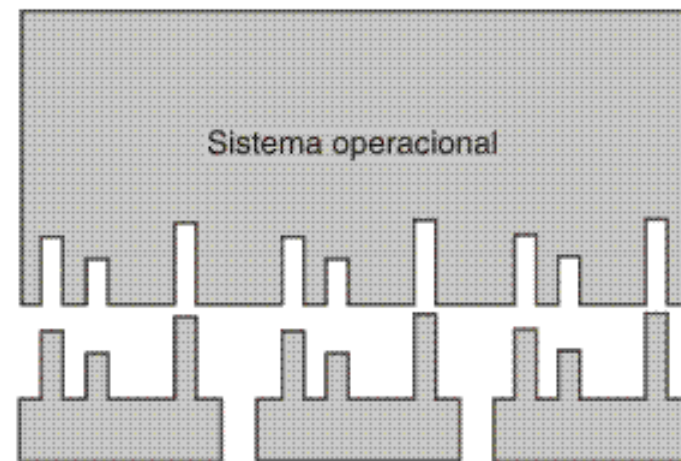
## Funções do software de E/S independente de dispositivo

# Software de E/S Independente de Dispositivo (2)



Driver do disco      Driver da impressora      Driver do teclado

(a)



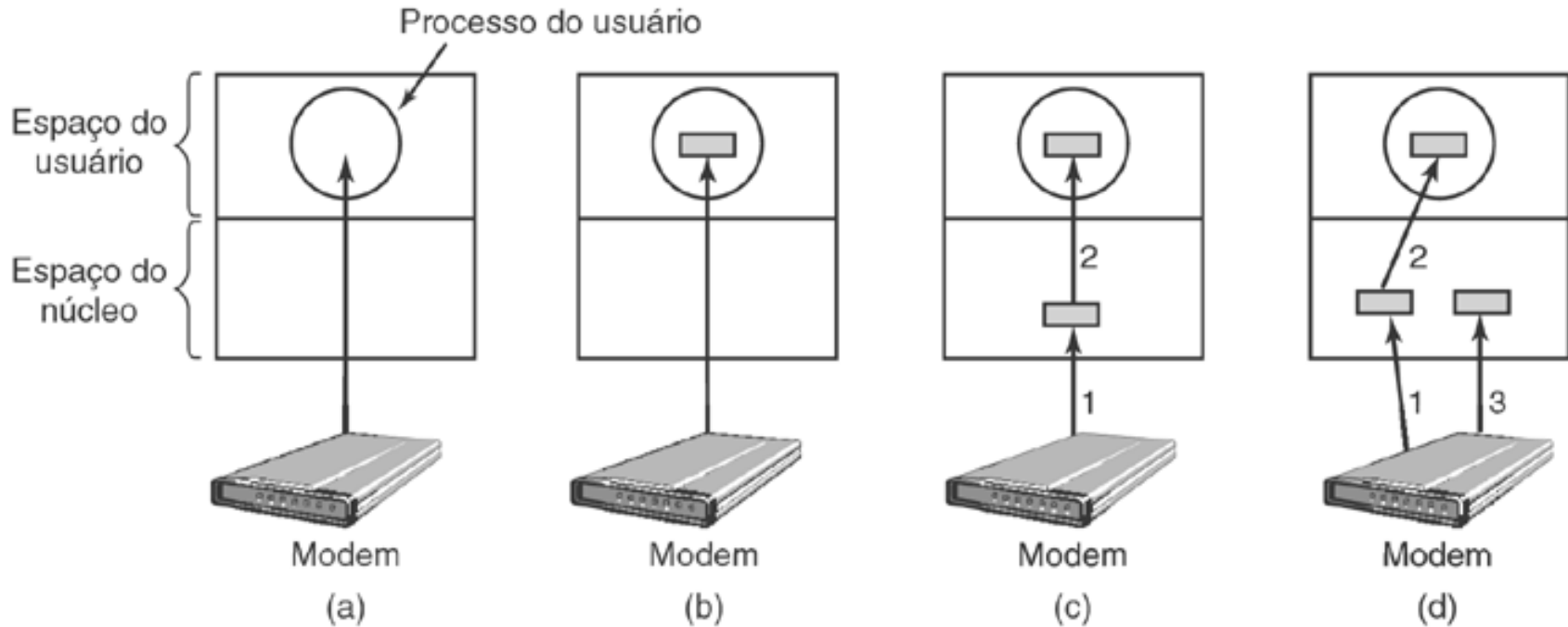
Driver do disco      Driver da impressora      Driver do teclado

(b)

(a) Sem uma interface-padrão do driver

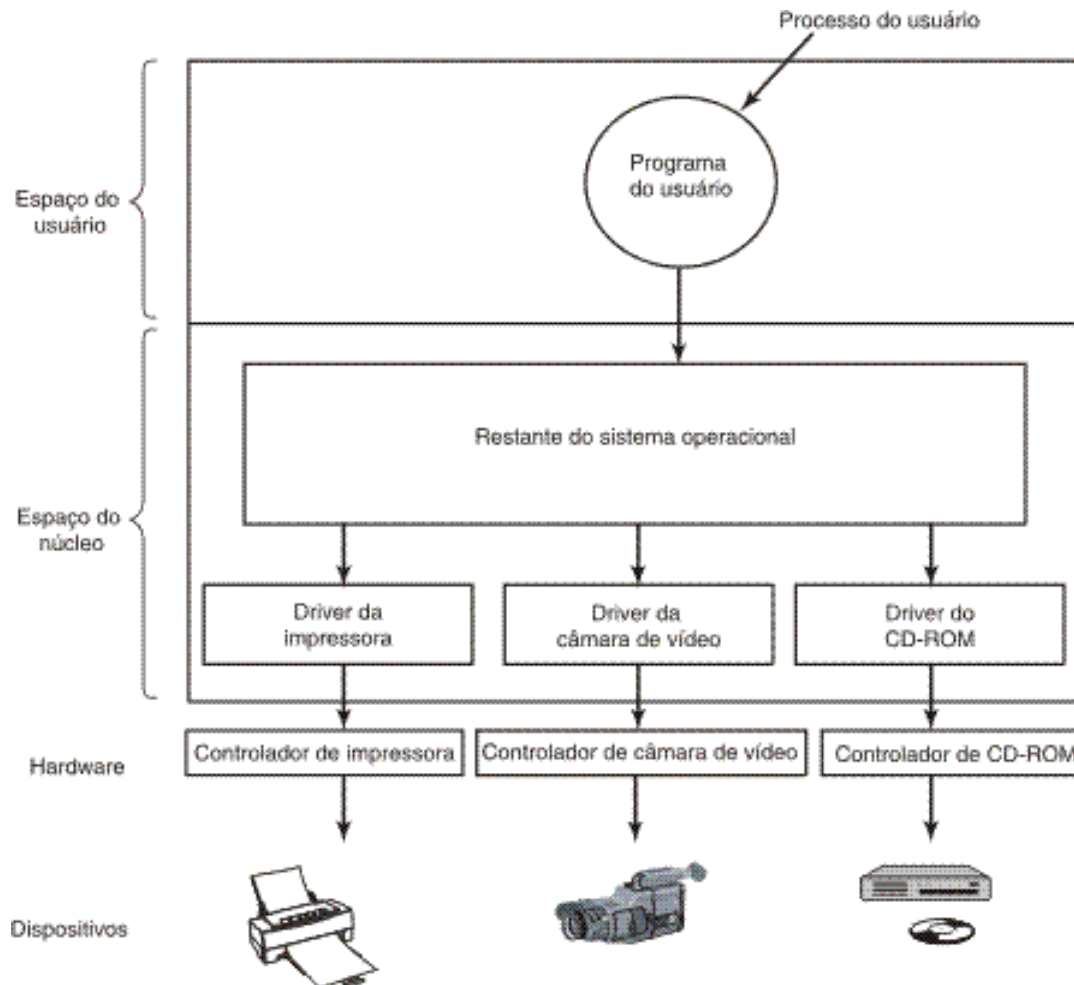
(b) Com uma interface-padrão do driver

# Software de E/S Independente de Dispositivo (3)



- 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

# Drivers dos Dispositivos



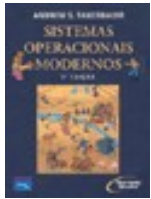
- Posição lógica dos drivers dos dispositivos
- A comunicação entre os drivers e os controladores de dispositivos é feita por meio do barramento

# Tratadores de Interrupção (1)



- As interrupções devem ser escondidas o máximo possível
  - uma forma de fazer isso é bloqueando o driver que iniciou uma operação de E/S até que uma interrupção notifique que a E/S foi completada
- Rotina de tratamento de interrupção cumpre sua tarefa
  - e então desbloqueia o driver que a chamou

# Tratadores de Interrupção (2)



- **Passos que devem ser executados em software depois da interrupção ter sido concluída**
  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

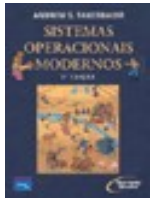


# Buffer vs Cache



- Cache:
  - Mecanismo de armazenamento temporário
  - Composto por uma memória ultra rápida e lógica em eletrônica embarcada
  - Geralmente serve como intermediário no acesso a disco rígido e memória RAM
  - Monitora as requisições:
    - Caso detenha a informação requisitada, ela retorna a informação para o solicitante
    - Caso não tenha, repassa a solicitação para o dispositivo de armazenamento e após obter a resposta salva uma cópia na cache

# Buffer vs Cache



- Buffer:
  - Região de memória, geralmente localizada em controladores.
  - 1) Agrupa blocos de dados para envio em lote para o dispositivo de E/S
    - Diminuir a quantidade de acessos aumentando a quantidade de dados enviados de cada vez
  - 2) Conciliar transferência de dados entre dispositivos com velocidades diferentes.
  - Exemplo:
    - Implantação em software de gravadoras de DVDs
    - Buffer de envio e recepção de dados em placas de rede

# Exercícios

