

Gerenciamento E/S

Hardware E/S

Interface de E/S

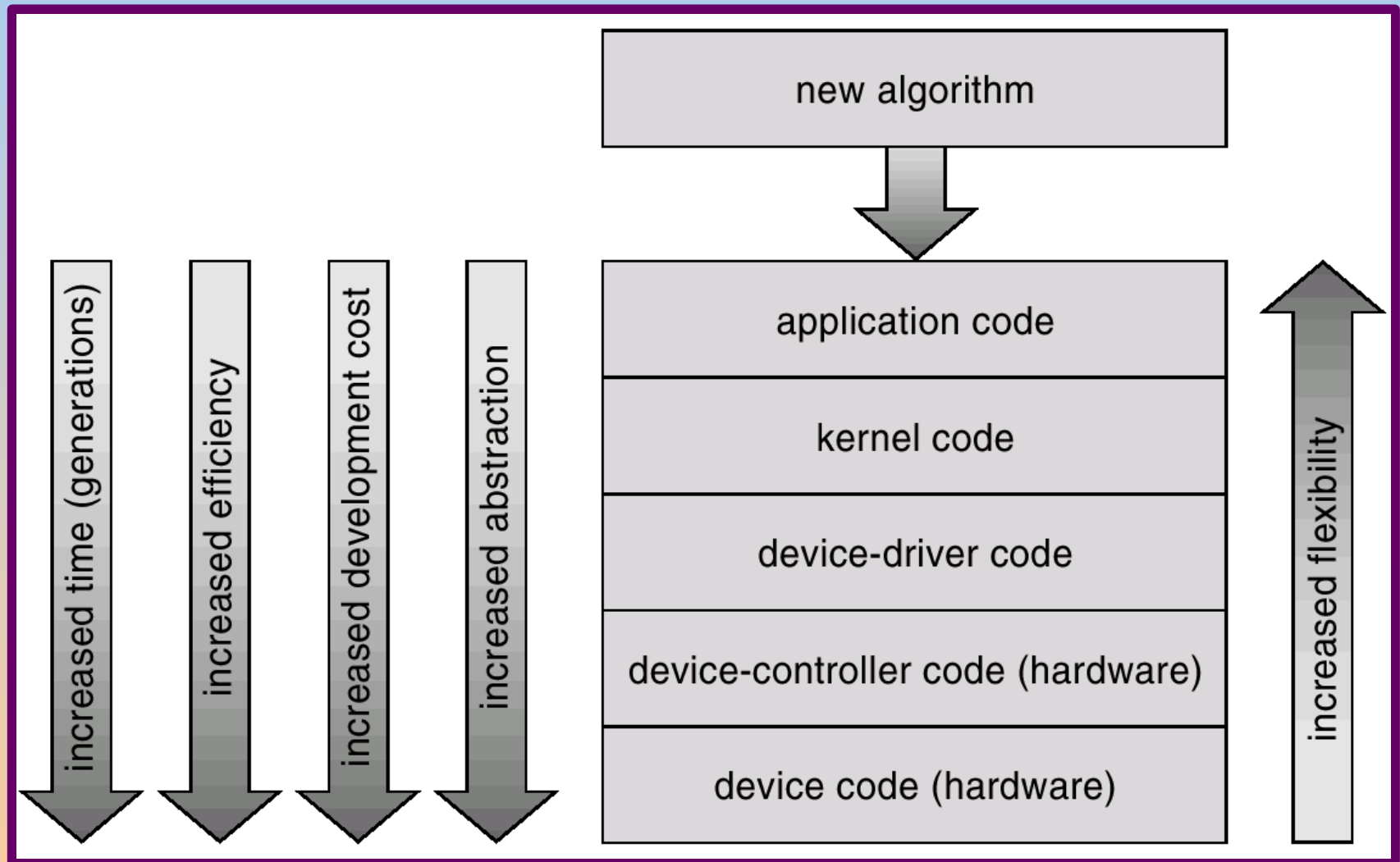
Sub-sistema de E/S do kernel

Transformando requisição de E/S em operações de hardware

Streams

Performance

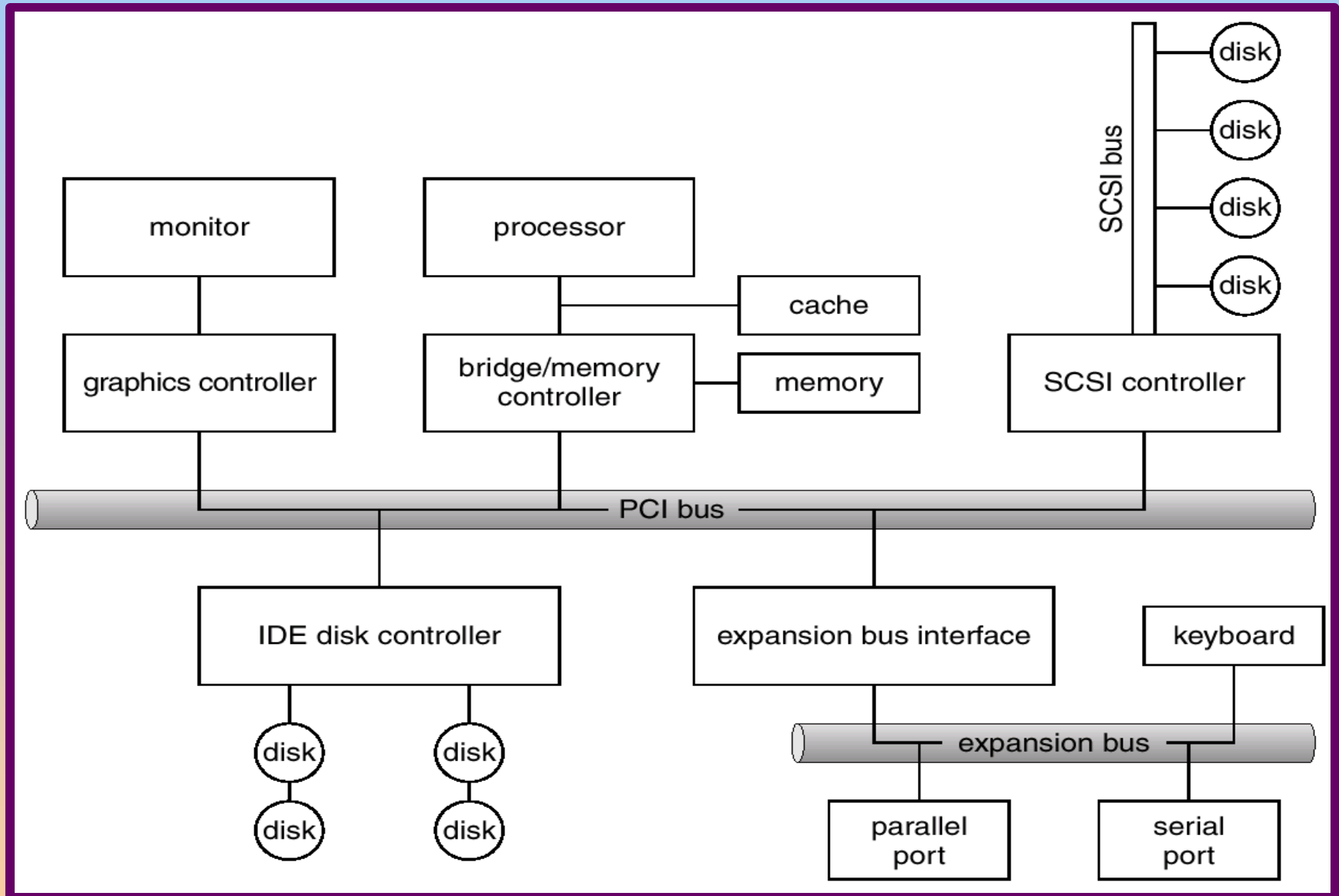
Estrutura básica de E/S



Hardware de E/S

- Problema: grande variedade de dispositivos de E/S
- Conceitos comuns
 - ◆ Porta
 - ◆ Barramento
 - ◆ Controladora
- Dispositivos têm endereços usados para:
 - ◆ Instruções de acesso direto a E/S
 - ◆ E/S mapeada em memória

Estrutura típica de um barramento de PC



Localização de portas de dispositivos de E/S num PC (lista parcial)

I/O address range (hexadecimal)	device
000-00F	DMA controller
020-021	interrupt controller
040-043	timer
200-20F	game controller
2F8-2FF	serial port (secondary)
320-32F	hard-disk controller
378-37F	parallel port
3D0-3DF	graphics controller
3F0-3F7	diskette-drive controller
3F8-3FF	serial port (primary)

Polling

- Determina o estado de um dispositivo
 - ◆ `command-ready`
 - ◆ `busy`
 - ◆ `Error`
- Necessita de ciclos busy-wait para esperar pela resposta do dispositivo.

Interrupções

- CPU faz/recebe requisição de interrupção
- Manipulador de interrupções recebe interrupções
- Pode haver mascaramento para ignorar ou atrasar tratamento de interrupções
- Um vetor de interrupções é usado para selecionar o manipulador correto da interrupção:
 - ◆ Baseado em prioridade
 - ◆ Algumas interrupções são não-mascaráveis
- Mecanismo de interrupção também é usado para tratar exceções

Ciclo de interrupção para E/S

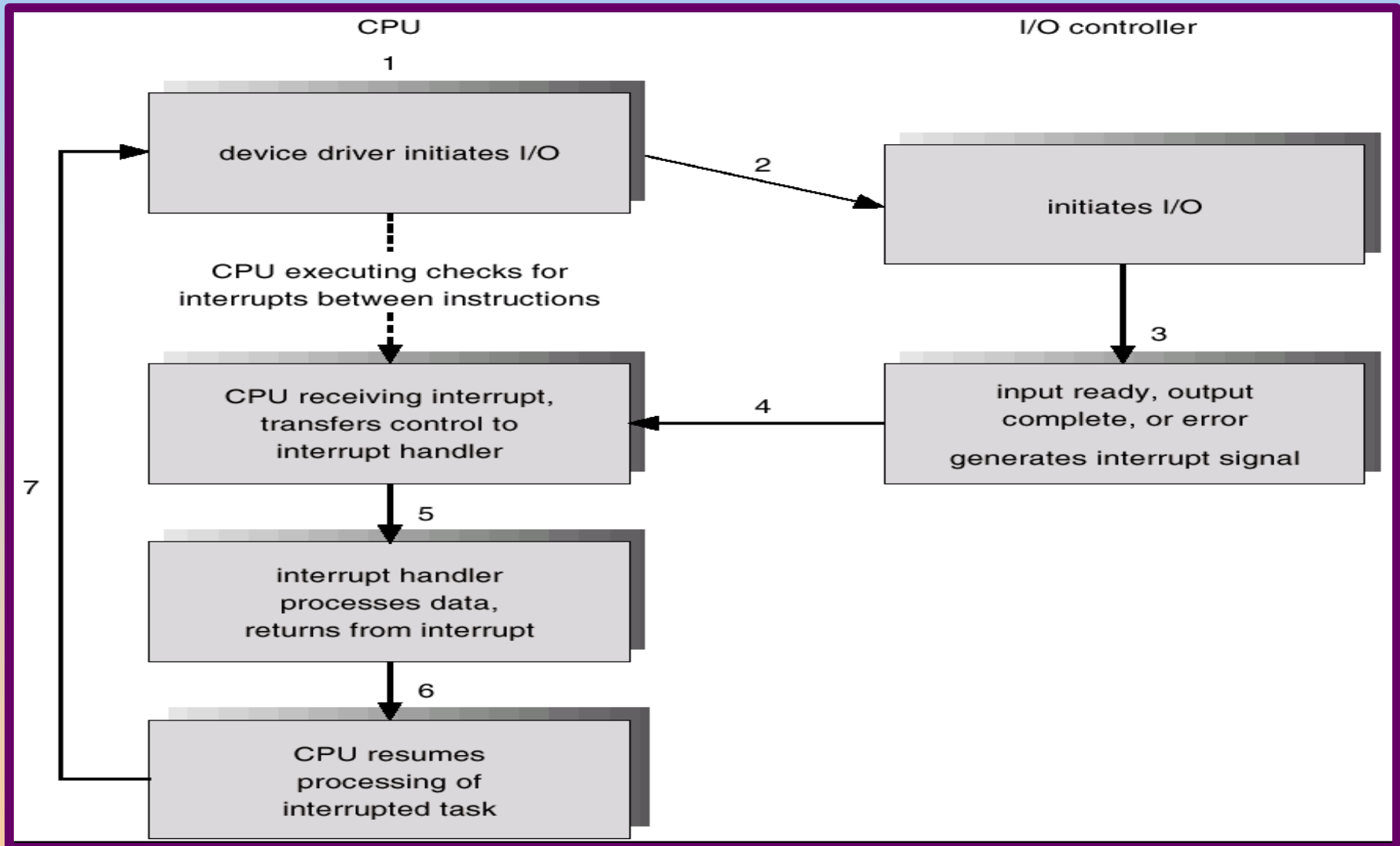


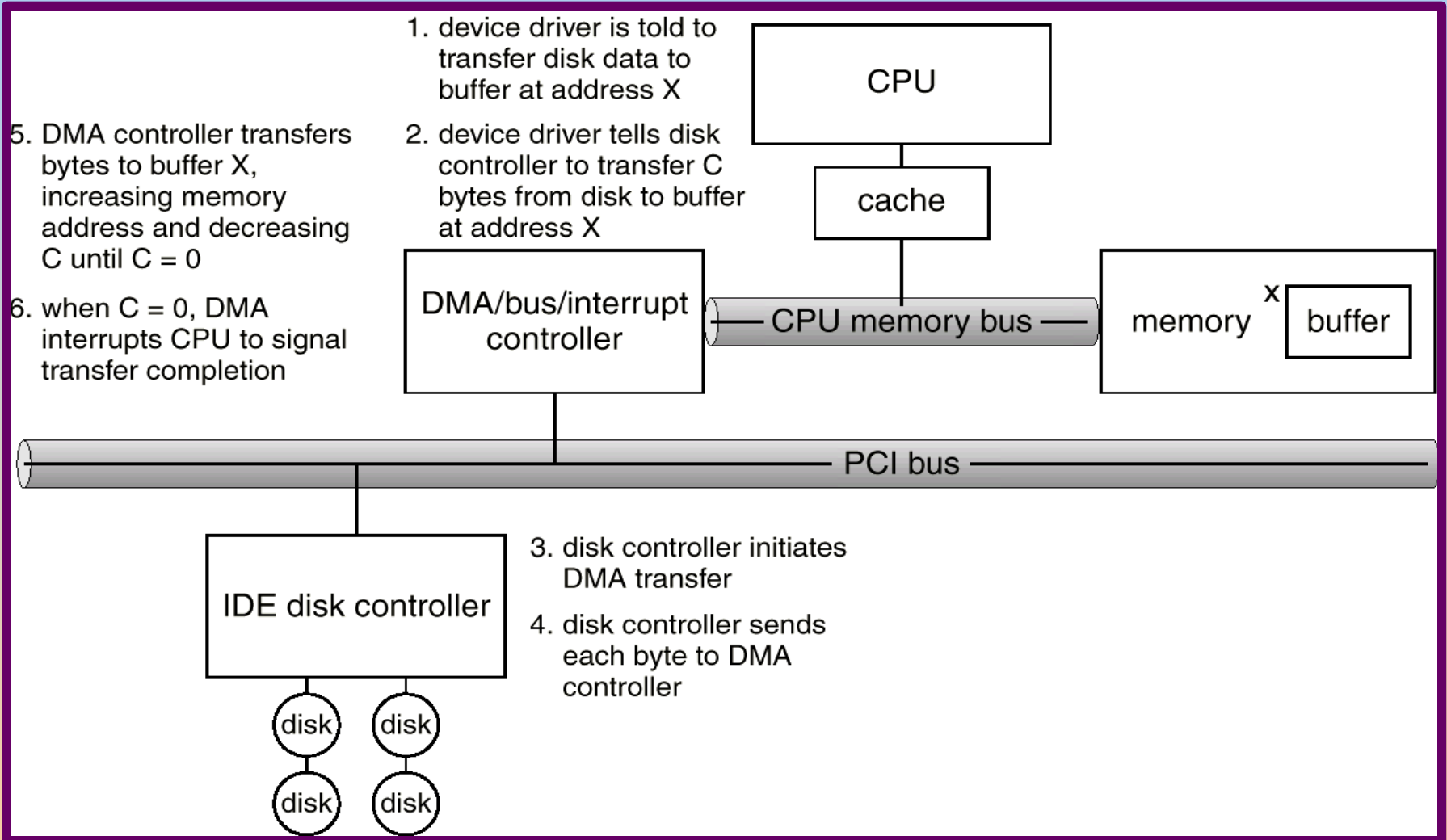
Tabela de Interrupções da família Pentium

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19Ð31	(Intel reserved, do not use)
32Ð255	maskable interrupts

Acesso direto à memória (DMA)

- Usado para evitar E/S programada para movimentação de grandes blocos de dados
- Requer uma controladora de DMA
- Libera a CPU para outras tarefas, uma vez que transfere dados diretamente entre dispositivo E/S e memória.

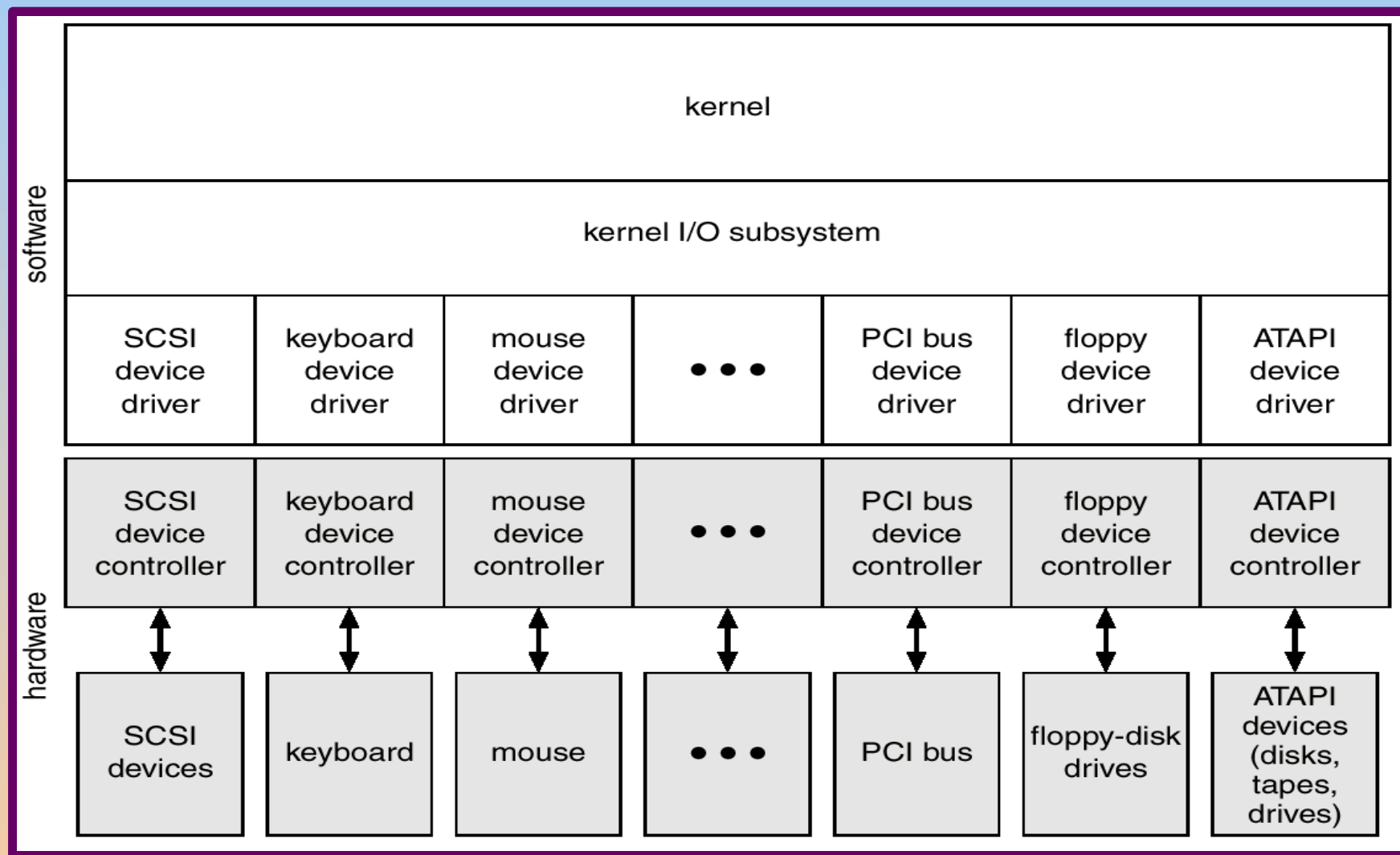
Seis passos básicos de DMA



Interface de E/S

- Systems calls de E/S encapsulam comportamentos de dispositivos em classes genéricas
- Camada de driver de dispositivo (device driver) esconde diferenças entre controladoras de E/S do kernel
- Dispositivos variam em muitas dimensões
 - ◆ Stream de caracteres ou blocos
 - ◆ Acesso seqüencial ou direto
 - ◆ Compartilhado ou dedicado
 - ◆ Velocidade de operação
 - ◆ Leitura-escrita, só leitura ou só escrita

Uma estrutura de E/S típica



Características de dispositivos de E/S

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

Dispositivos de bloco e caractere

■ Dispositivos de bloco incluem drives de disco

- ❖ Comandos incluem leitura(read), escrita(write) e posicionamento(seek)
- ❖ Permitem leitura direta (Raw I/O) ou acesso por sistema de arquivos
- ❖ Possibilidade de uso de arquivos mapeados em memória

■ Dispositivos de caractere incluem teclados, mouse e portas seriais

- ❖ Comandos incluem `get`, `put`

Dispositivos de Rede

- Variam entre bloco e caractere para ter sua própria interface
- Unix e Windows NT/9i/2000 incluem interface de socket
 - ◆ Separa protocolo de rede da operação de rede
- Variadas estruturas de comunicação(pipes, FIFOs, streams, filas , mailboxes)

Clocks e Timers

- Disponibilizam temporização para o SO
- Timer – variável interna
- Clock – dispositivo de hardware
- Timer normalmente é alterado com base em interrupções de clock.

E/S por bloqueamento(blocking) e sem bloqueamento (nonblocking)

- Por bloqueamento – processo é suspenso até E/S ser realizada completamente
 - ◆ Fácil de usar e entender
 - ◆ Insuficiente para algumas necessidades

- Sem bloqueamento – realizações intermediárias de E/S
 - ◆ Utiliza bufferização
 - ◆ Implementação via multi-threading
 - ◆ Retorna rapidamente com contador de blocos escritos ou lidos

- Assíncrono – Processo roda enquanto E/S é realizada
 - ◆ Difícil de usar
 - ◆ Sub-sistema de E/S informa o processo quando a E/S for realizada.

Sub-sistema de E/S do Kernel

■ Escalonamento

- ◆ Algumas requisições de E/S são feitas via filas de dispositivos

■ Buffering – armazenar dados em memória enquanto ocorre transferência entre dispositivos

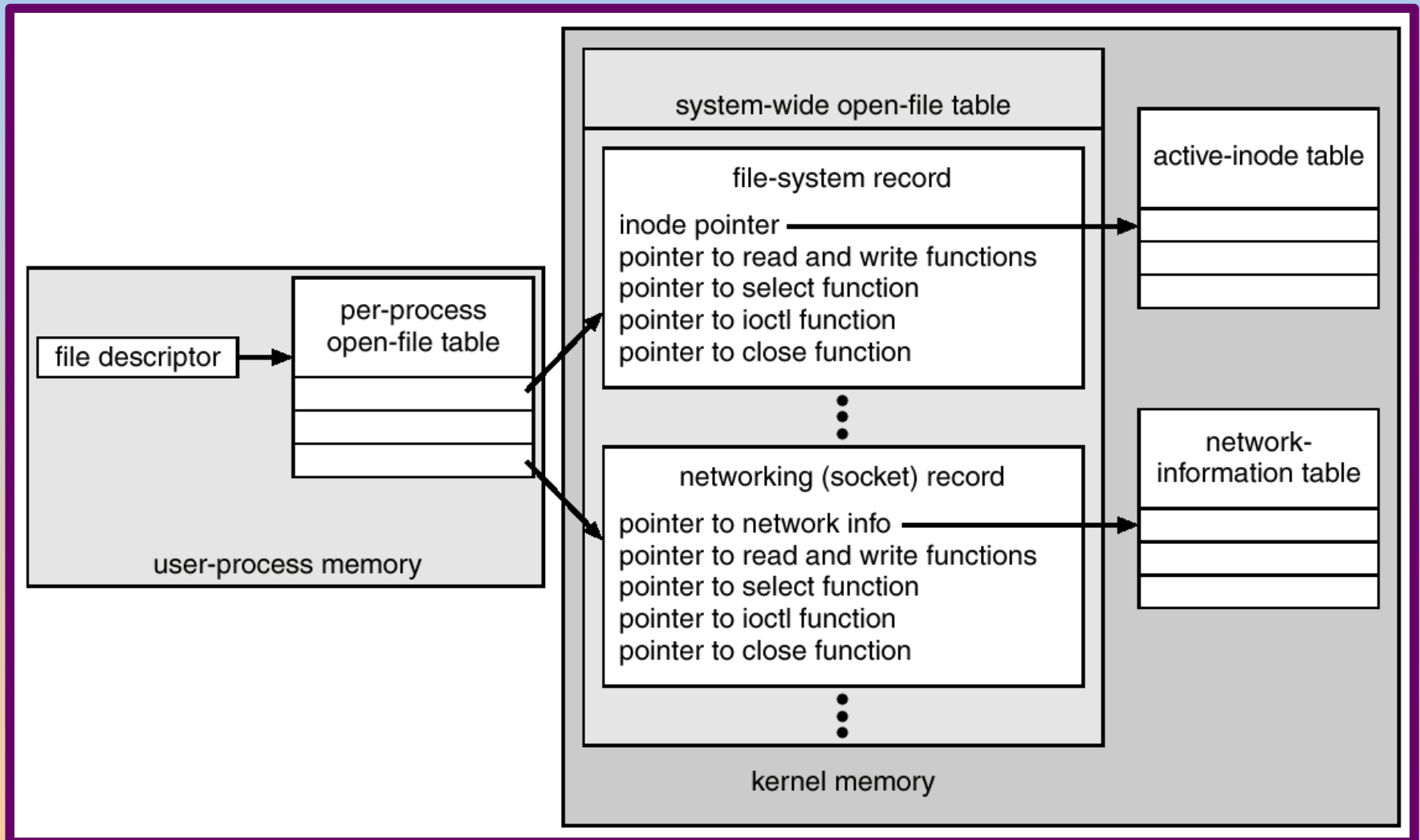
Sub-sistema de E/S do Kernel

- Caching – memória de alta velocidade que armazena dados acessados
 - ◆ Mantém sempre uma cópia
 - ◆ Estrutura básica para aumentar performance
- Spooling – manter saída para um dispositivo
 - ◆ Se o dispositivo consegue atender uma requisição de cada vez
 - ◆ Exemplo típico: impressora

Estruturas de Dados para E/S no Kernel

- Kernel mantém informação de estado para componentes de E/S, incluindo tabelas de arquivos abertos, conexões de rede, estado de dispositivos de caractere, etc.
- Estruturas mais complexas podem ser necessárias para gerenciar buffers, alocação de memória, sincronização de cópias (dirty buffers).
- É comum o uso de métodos de orientação a objetos e passagem de mensagens para implementar E/S.

Estrutura de E/S UNIX

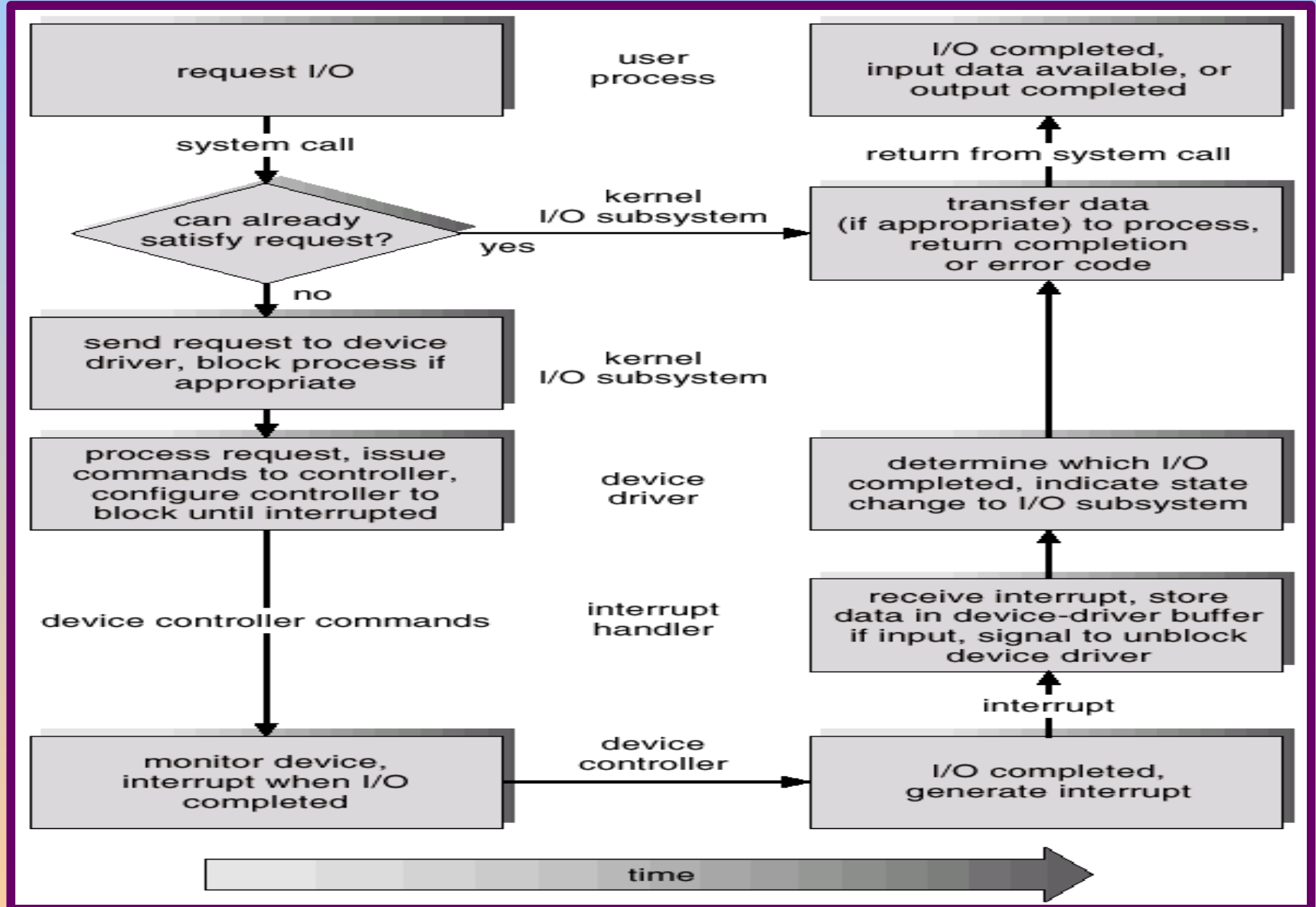


Transformação de requisições de E/S para operações em Hardware

■ Considere-se um processo que precise ler um arquivo de disco:

- ◆ Determinar qual dispositivo armazena o arquivo
- ◆ Traduzir o nome do arquivo para a representação de disco
- ◆ Ler dados do disco para um buffer
- ◆ Tornar os dados disponíveis para o processo
- ◆ Retornar o controle ao processo

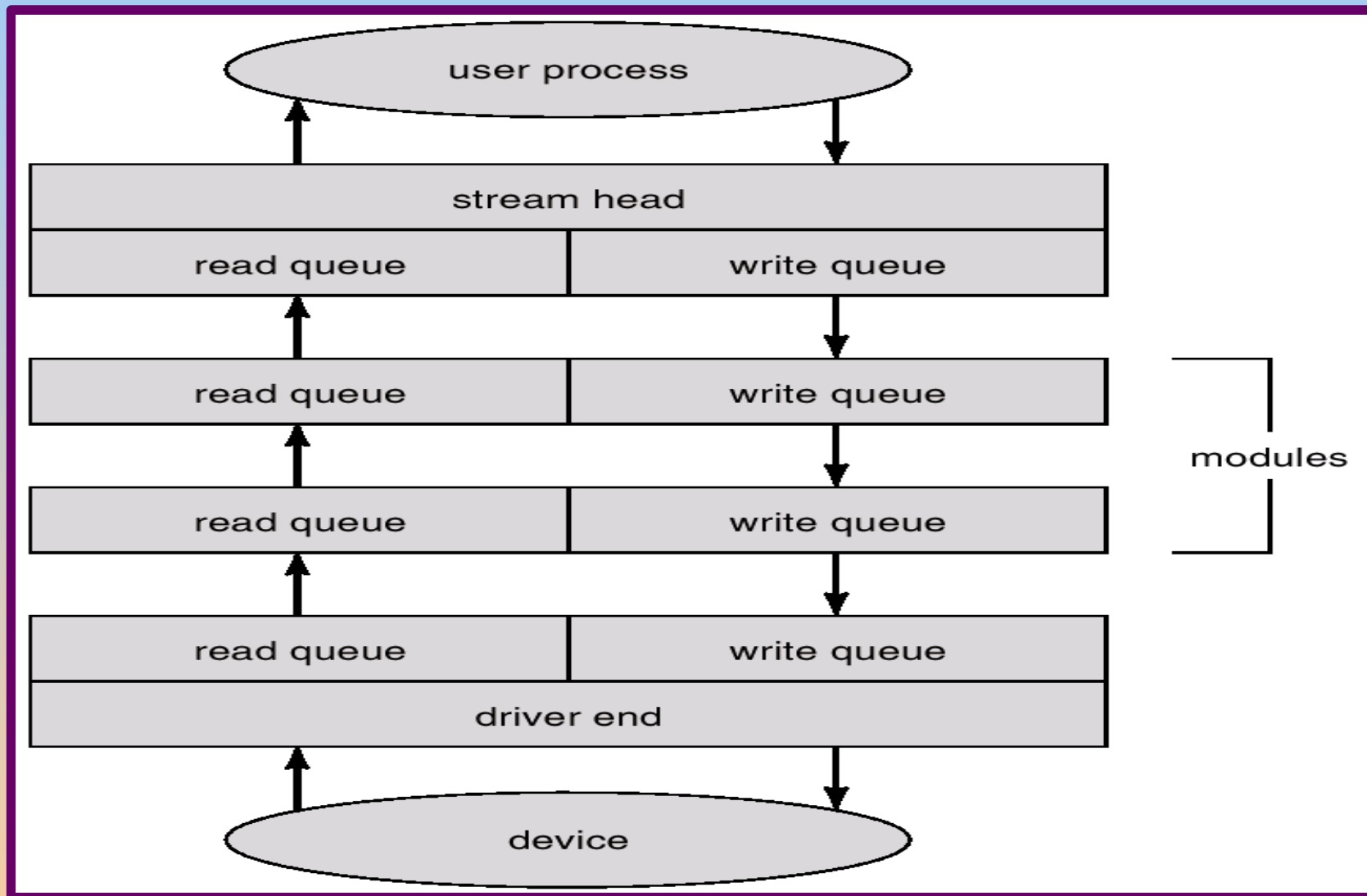
Ciclo de vida de uma requisição de E/S



STREAMS

- **STREAM** – Um canal de comunicação full-duplex entre um processo de nível de usuário e um dispositivo.
- Um STREAM consiste de:
 - interfaces de cabeçalho **STREAM** com o processo do usuário
 - interfaces **driver end** com o dispositivo
 - zero ou mais módulos STREAM entre eles.
- Cada módulo contém uma fila de leitura e uma de escrita.
- Passagem de mensagem é um mecanismo usado para implementar comunicação entre as filas.

A estrutura de STREAMS



Comunicações entre máquinas

