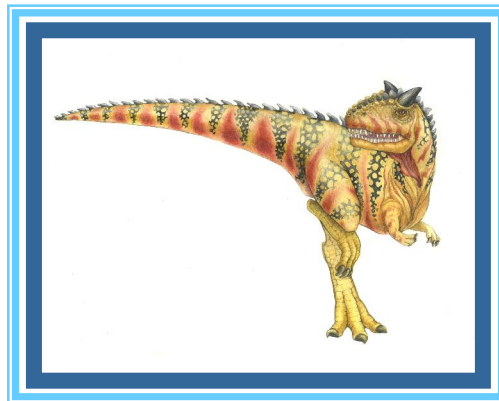


# Capítulo 13: Sistemas de E/S

---



# Sobre a apresentação (About the slides)



Os slides e figuras dessa apresentação foram criados por Silberschatz, Galvin e Gagne em 2009. Essa apresentação foi modificada por Cristiano Costa (cac@unisinós.br). Basicamente, os slides originais foram traduzidos para o Português do Brasil.

É possível acessar os slides originais em <http://www.os-book.com>  
Essa versão pode ser obtida em <http://www.inf.unisinós.br/~cac>



The slides and figures in this presentation are copyright Silberschatz, Galvin and Gagne, 2009. This presentation has been modified by Cristiano Costa (cac@unisinós.br). Basically it was translated to Brazilian Portuguese.

You can access the original slides at <http://www.os-book.com>

This version could be downloaded at <http://www.inf.unisinós.br/~cac>





# Capítulo 13: Sistema de E/S

---

- Hardware de E/S
- Interface de E/S do Aplicativo
- Subsistema de E/S do Kernel
- Transformando Requisições de E/S para Operações de Hardware
- Fluxos (Streams)
- Desempenho





# Objetivos

---

- Explorar a estrutura de um subsistema de E/S de um Sistema Operacional
- Discutir os princípios de E/S de hardware e sua complexidade
- Fornecer detalhes dos aspectos de desempenho do hardware e software de E/S





# Hardware de E/S

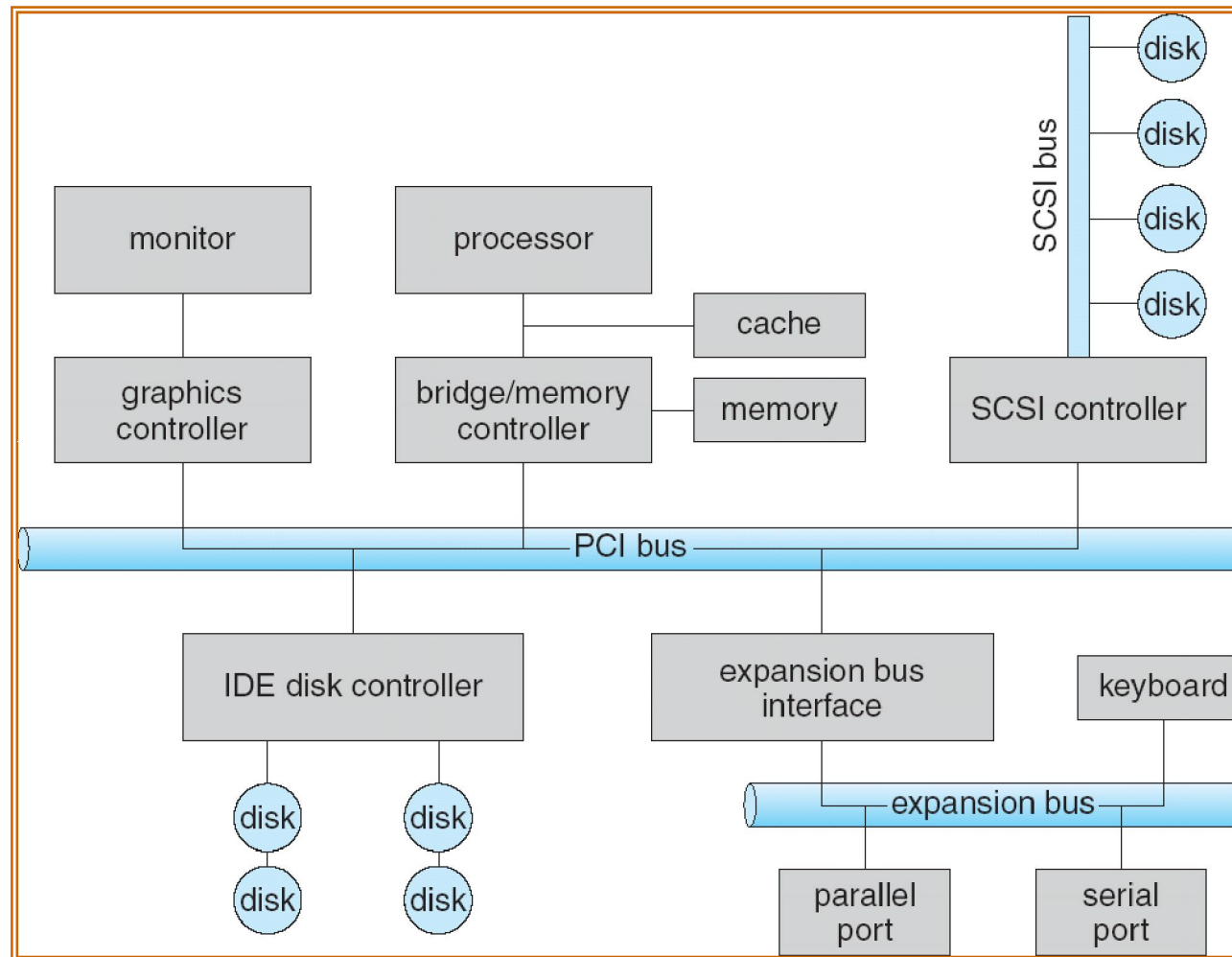
---

- Variedade incrível de dispositivos de E/S
  
- Conceitos Comuns
  - **Porta**
  - **Barramento** (encadeado / *daisy chain* ou acesso direto compartilhado)
  - **Controlador (host adapter)**
  
- Instruções de E/S controlam dispositivos
  
- Dispositivos tem endereços, usados por
  - Instruções diretas de E/S
  - **E/S Mapeado na memória**





# Uma estrutura típica do Barramento do PC





# Localização de Portas de Dispositivos de E/S nos PCs (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 (Sondagem)

---

- Determina o estado de dispositivos
  - Pronto para receber comandos
  - Ocupado
  - Erro
  
- Ciclo de **Espera Ocupada** para esperar por E/S do dispositivo



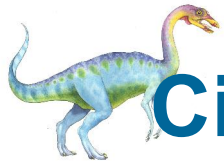


# Interrupções

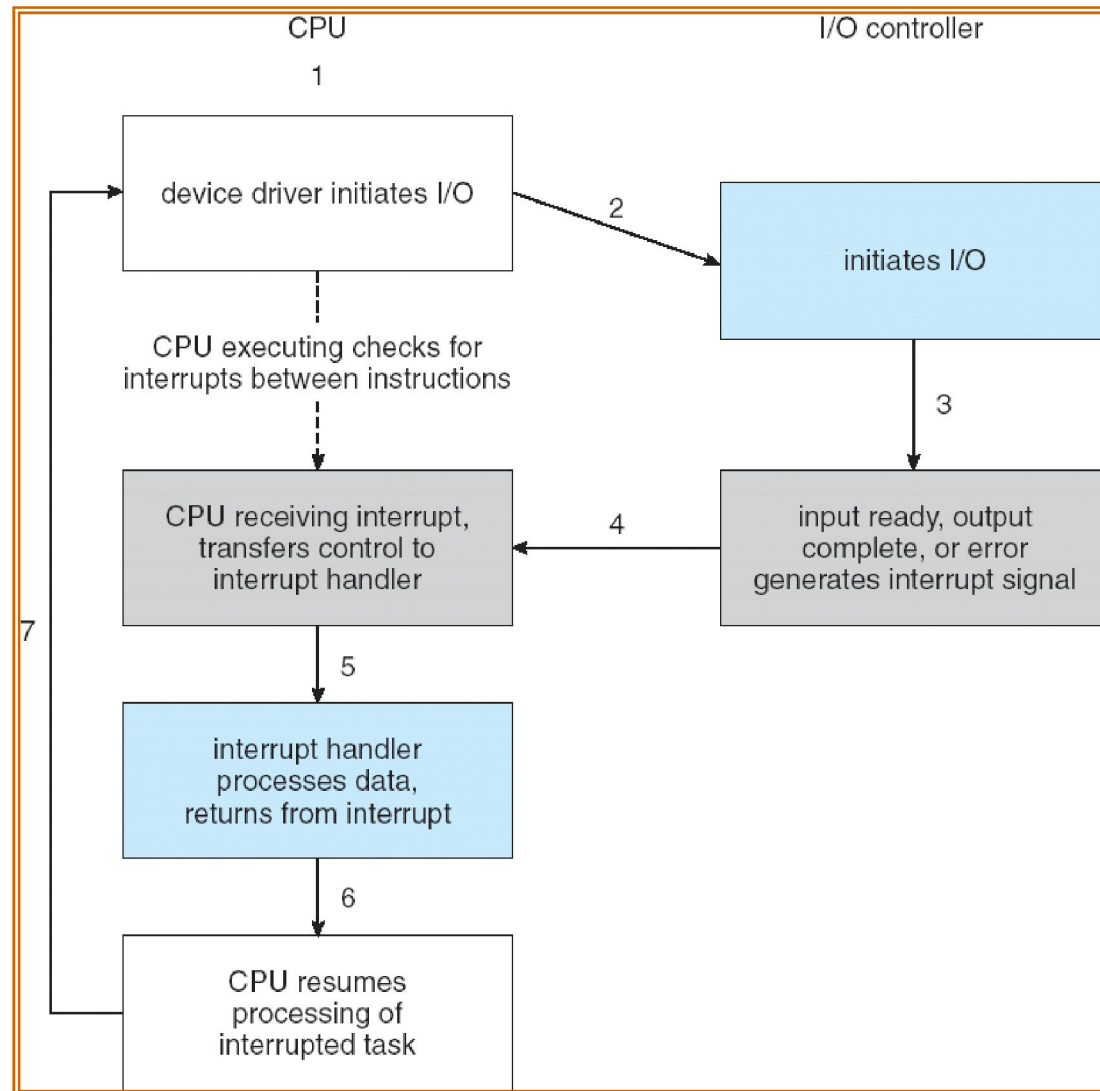
---

- **Linha de Requisição de Interrupção** da CPU ativada por dispositivos de E/S
- **Manipulador de Interrupção** (*Interrupt handler*) recebe interrupções
- **Mascaramento** para ignorar ou atrasar algumas interrupções
- Vetor de Interrupção redireciona interrupção para manipulador correspondente
  - Baseado em prioridade
  - Algumas sem mascaramento
- Mecanismo de interrupção também usado para exceções





# Ciclo de E/S dirigido por Interrupção





# Tabela de Eventos do Processador Intel 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
19–31	(Intel reserved, do not use)
32–255	maskable interrupts





# Acesso Direto a Memória

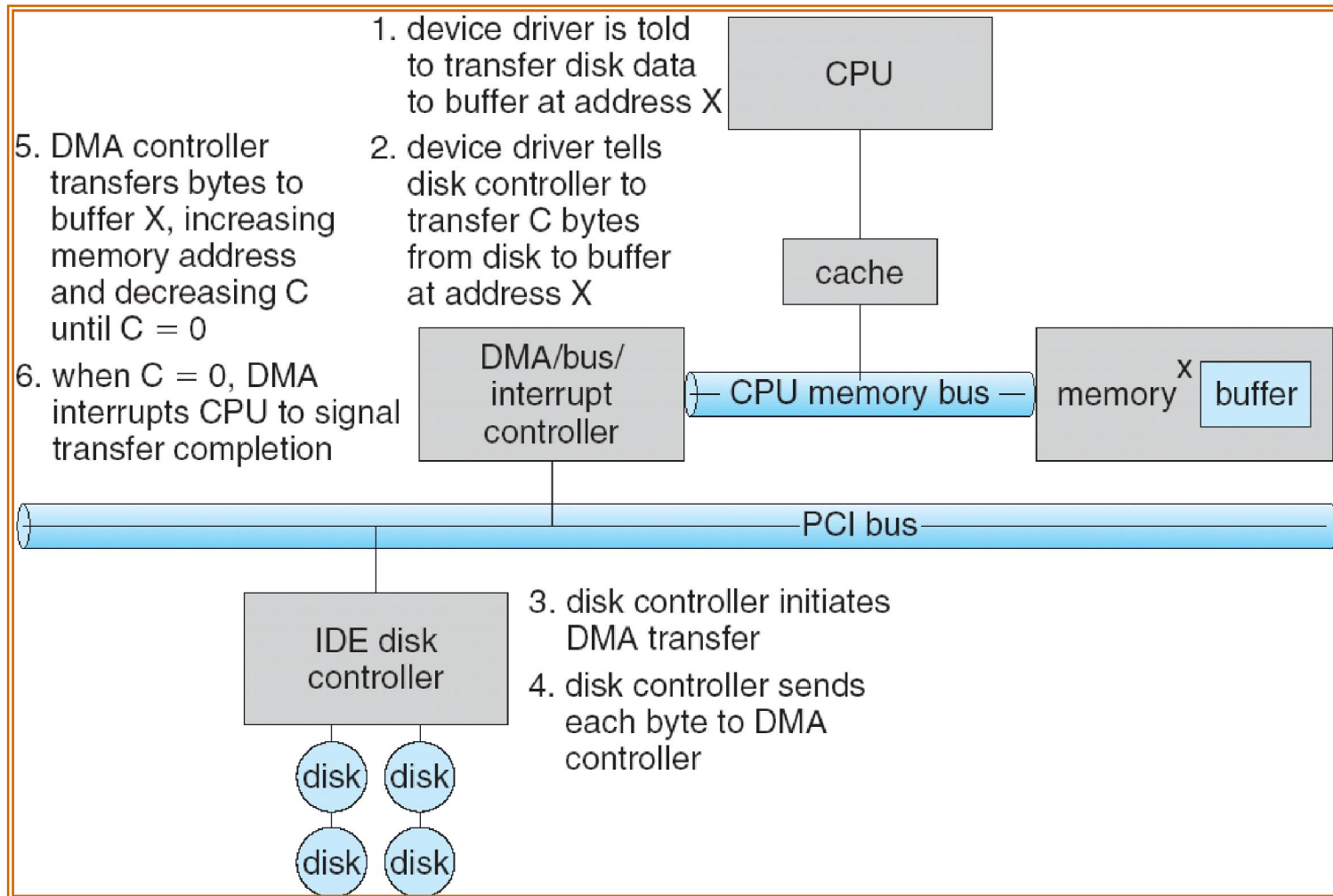
---

- Usado para evitar E/S programável para um grande movimento de dados
- Necessita de controladora de **DMA**
- Bi-passa CPU para transferir dados diretamente entre o dispositivo de E/S e a memória





# Seis passos para realizar transferência DMA





# Interface de E/S da Aplicação

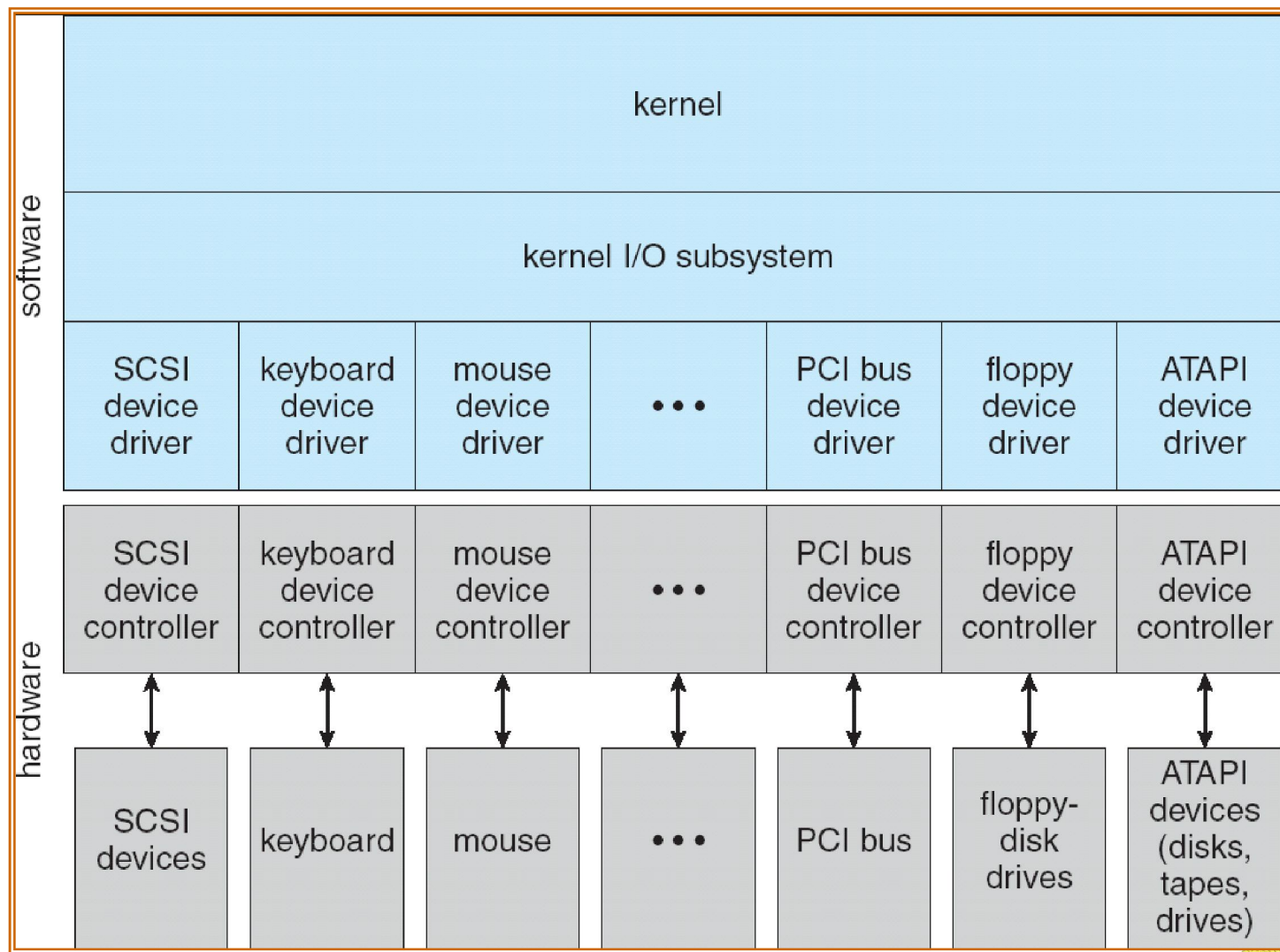
---

- Chamadas de sistema de E/S encapsulam comportamentos de dispositivos em classes genéricas
- Camada de *driver* de dispositivo esconde diferenças entre controladoras de E/S do kernel
- Dispositivos variam em diversas dimensões
  - **Fluxo de Caracteres** ou **bloco**
  - **Acesso seqüencial** ou **aleatório**
  - **Compartilhável** ou **dedicado**
  - **Velocidade de Operação**
  - **Leitura/escrita, somente leitura, or somente escrita**





# Estrutura de E/S do Kernel





# 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 `read`, `write`, `seek`
  - Acesso de E/S bruto (*raw*) ou por sistema de arquivos
  - Acesso de arquivos mapeados na memória possível
  
- Dispositivos Caractere incluem teclado, mouse, porta serial
  - Comandos incluem `get`, `put`
  - Bibliotecas em camadas acima permitem edição em linha





# Dispositivos de Rede

---

- Variam suficientemente de bloco e caractere para ter sua própria interface
- Unix e Windows NT/9x/2000 incluem interface *socket*
  - Separa protocolo de rede de operação em rede
  - Inclui funcionalidade `select`
- Abordagens variam muito (pipes, FIFOs, streams, queues, mailboxes)





# Clocks e Timers

---

- Fornece a hora atual, tempo decorrido, cronômetro
- **Timer de Intervalo Programável** usado para cronometrar tempo, interrupções periódicas
- `ioctl` (no UNIX) cobre aspectos particulares de E/S tais como clocks e timers





# E/S Bloqueante e Não-bloqueante

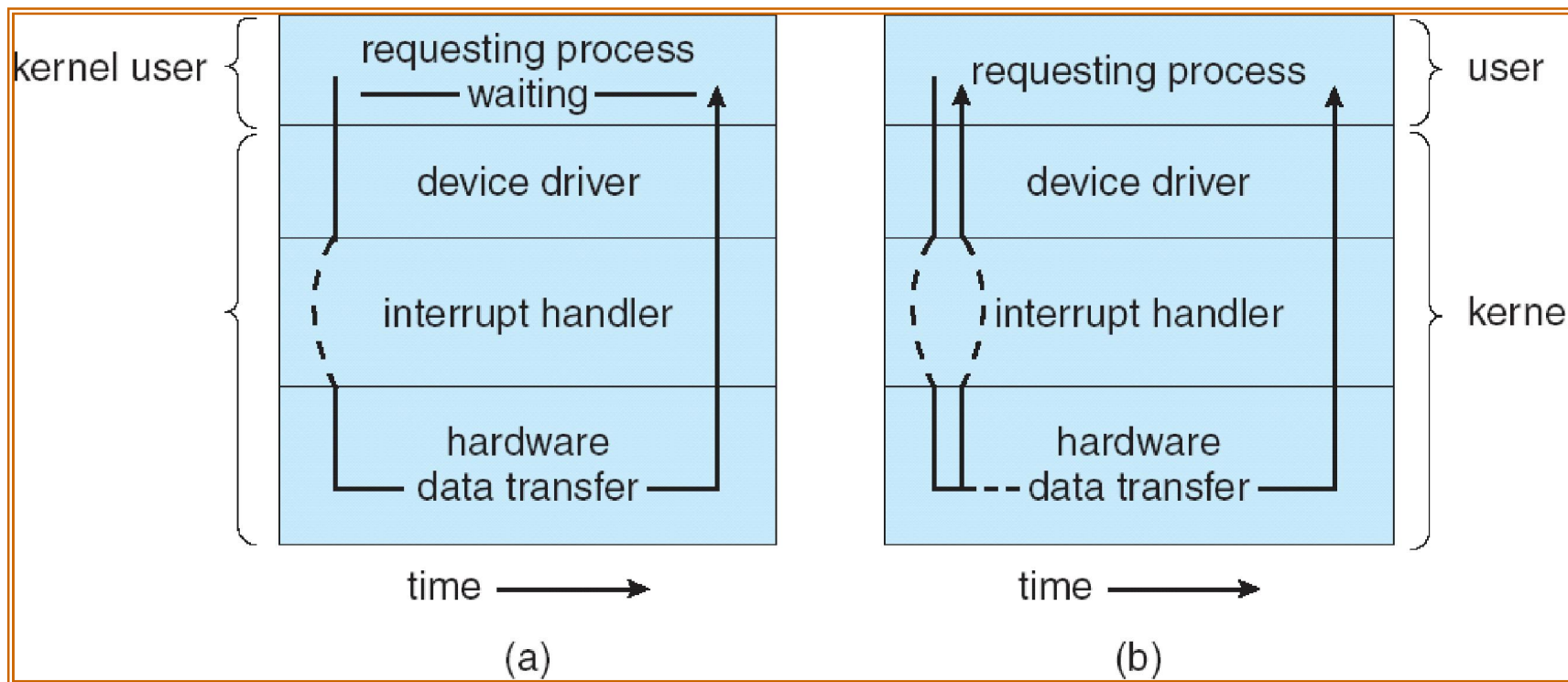
---

- **Bloqueante** - processo suspenso até o E/S completar
  - Fácil de usar e entender
  - Insuficiente para algumas necessidades
  
- **Não-bloqueante** - chamada de E/S retorna tão logo possível
  - Interface com Usuário, cópia de dados (E/S bufferizado)
  - Implementado via multi-threading
  - Retorna rápido com contagem de bytes lido ou escrito
  
- **Asíncrono** - processo executa enquanto E/S executa
  - Difícil de Usar
  - Subsistema de E/S sinaliza processos quando E/S completa





# Dois Métodos de E/S



Synchronous

Asynchronous





# Subsistema de E/S do Kernel

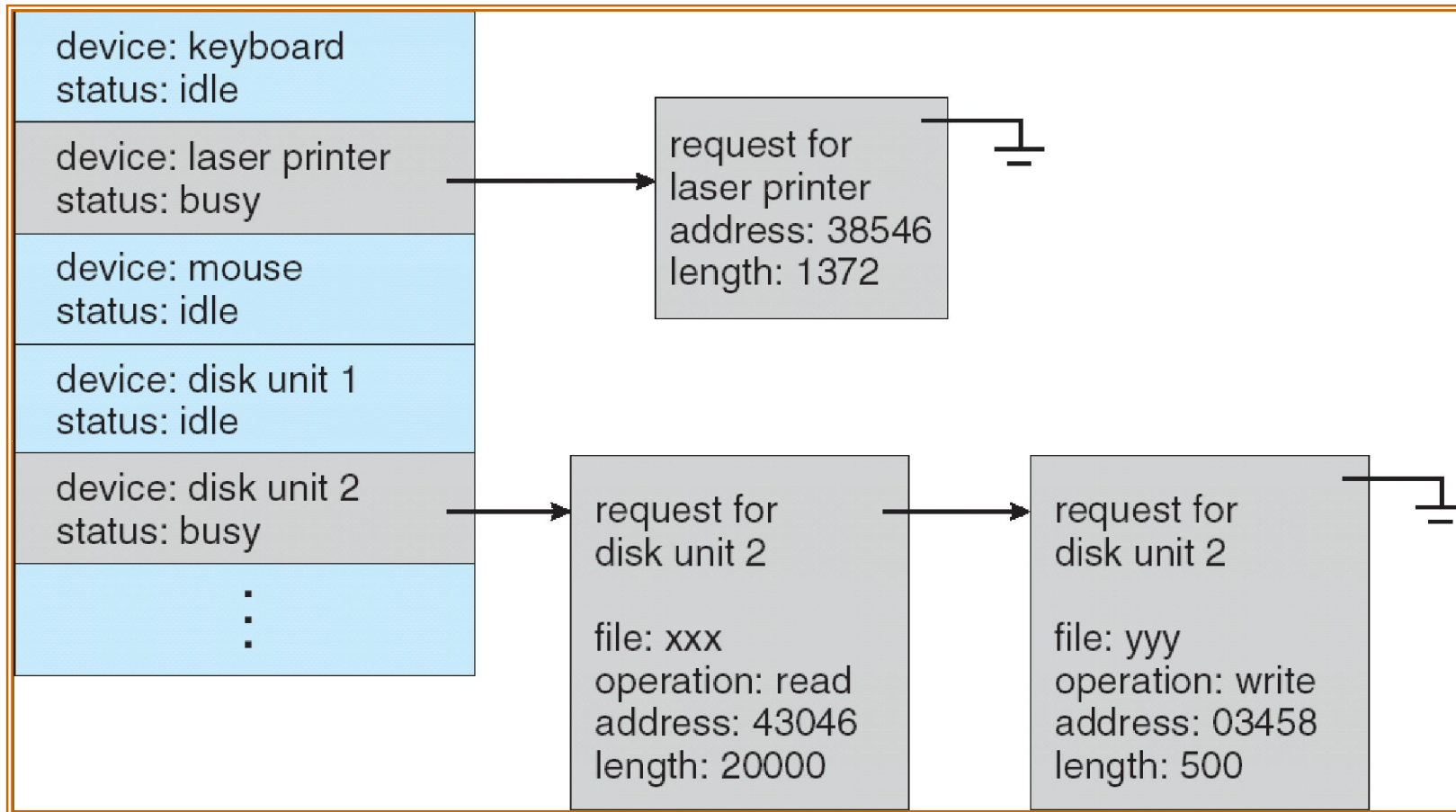
---

- Escalonamento
  - Algumas requisições de E/S são ordenadas por fila de cada dispositivo
  - Alguns SOs tentam justiça
  
- Bufferização - armazena dado na memória enquanto está transferindo entre dispositivos
  - Para lidar com diferença de velocidade entre dispositivos
  - Para lidar com a diferença de tamanho da transferência
  - Para manter a “semântica da cópia”



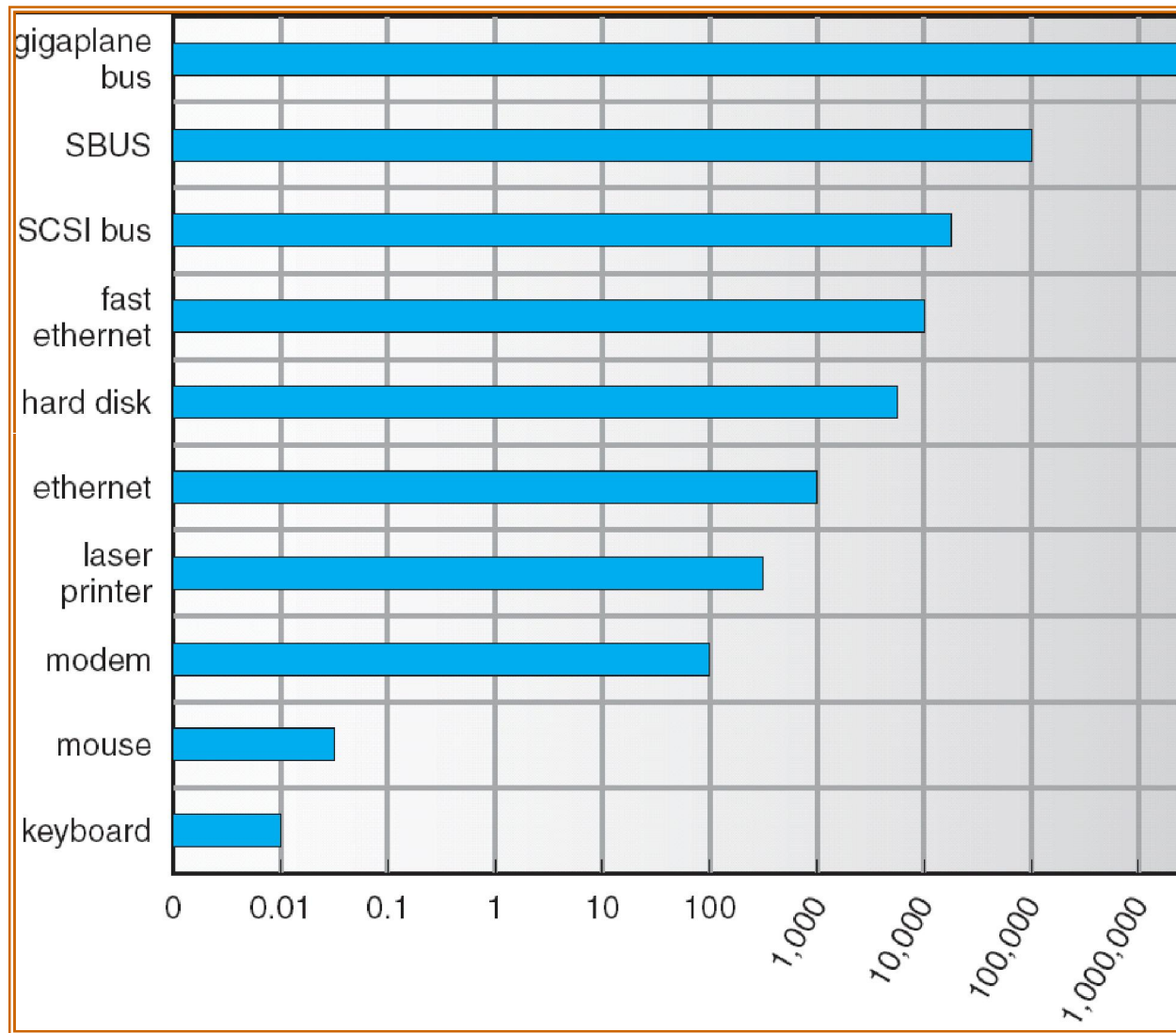


# Tabela de Status de Dispositivos





# Taxas de Transferência de Dispositivos na Sun Enterprise 6000





# Subsistema de E/S do Kernel

---

- **Caching** - memória rápida mantém cópia dos dados
  - Sempre somente uma cópia
  - Chave para o desempenho
  
- **Spooling** - guarda saída para um dispositivo
  - Se dispositivo pode atender somente uma requisição por vez
  - Ex.: impressão
  
- **Reserva de Dispositivo** - fornece acesso exclusivo para um dispositivo
  - Chamada de Sistemas para alocação e desalocação
  - Cuidadar com possíveis impasses (*deadlocks*)





# Manipulação de Erros

---

- SO pode recuperar de uma leitura de disco, disponibilidade de dispositivo, falhas transientes de escrita
- Grande parte retornam um número de erro ou código quando a requisição de E/S falha
- Rastros (logs) de erros de sistemas armazenam relatórios de problemas



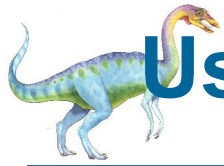


# Proteção de E/S

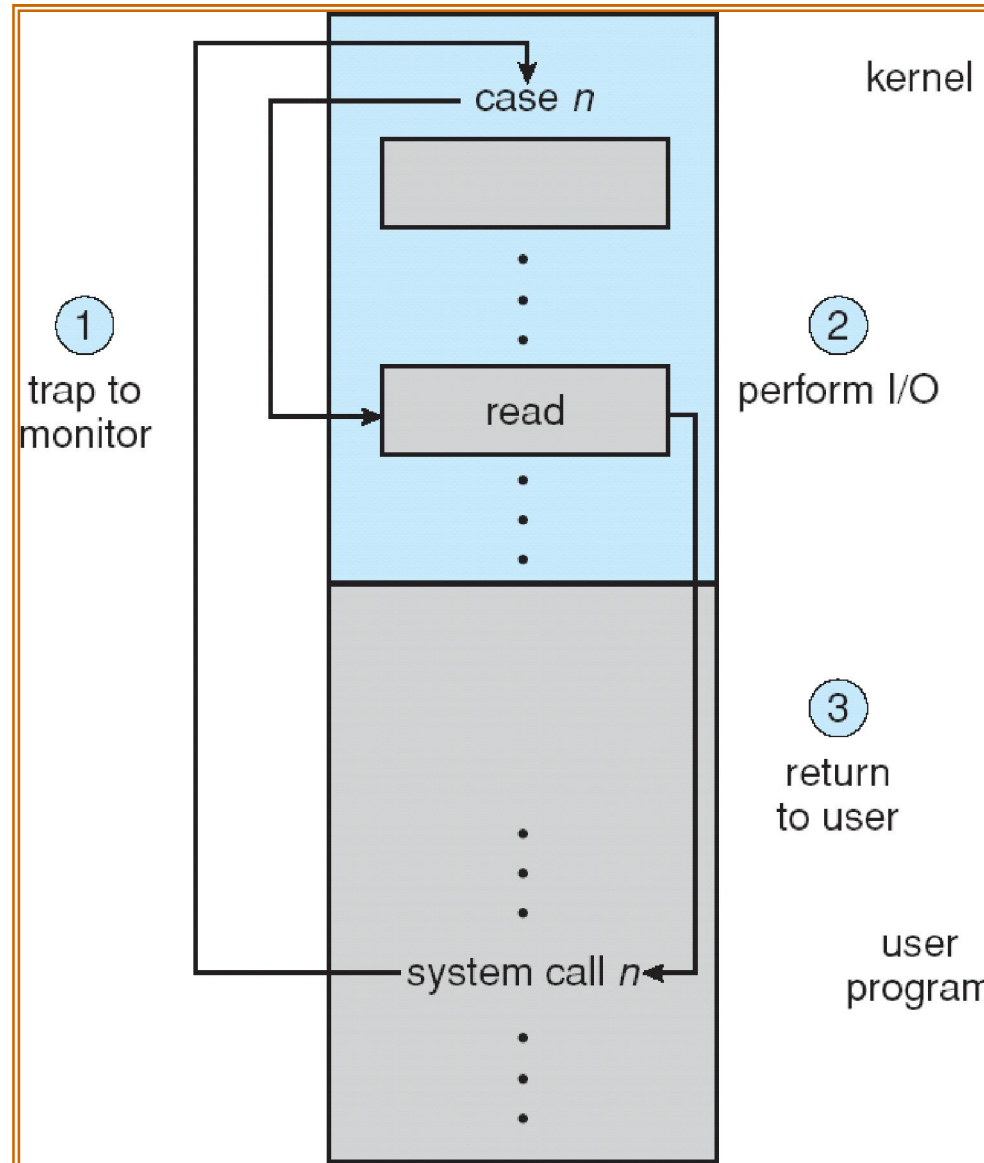
---

- Processo do usuário pode acidentalmente ou propositalmente tentar interromper a operação normal via instruções ilegais de E/S
  - Todas as instruções de E/S são definidas como privilegiadas
  - E/S deve ser realizado por chamadas de sistemas
    - ▶ Mapeamento de memória e localização em memória de portas de E/S devem ser protegidos também





# Uso de uma Chamada de Sistema para Realizar E/S





# Estruturas de Dados do Kernel

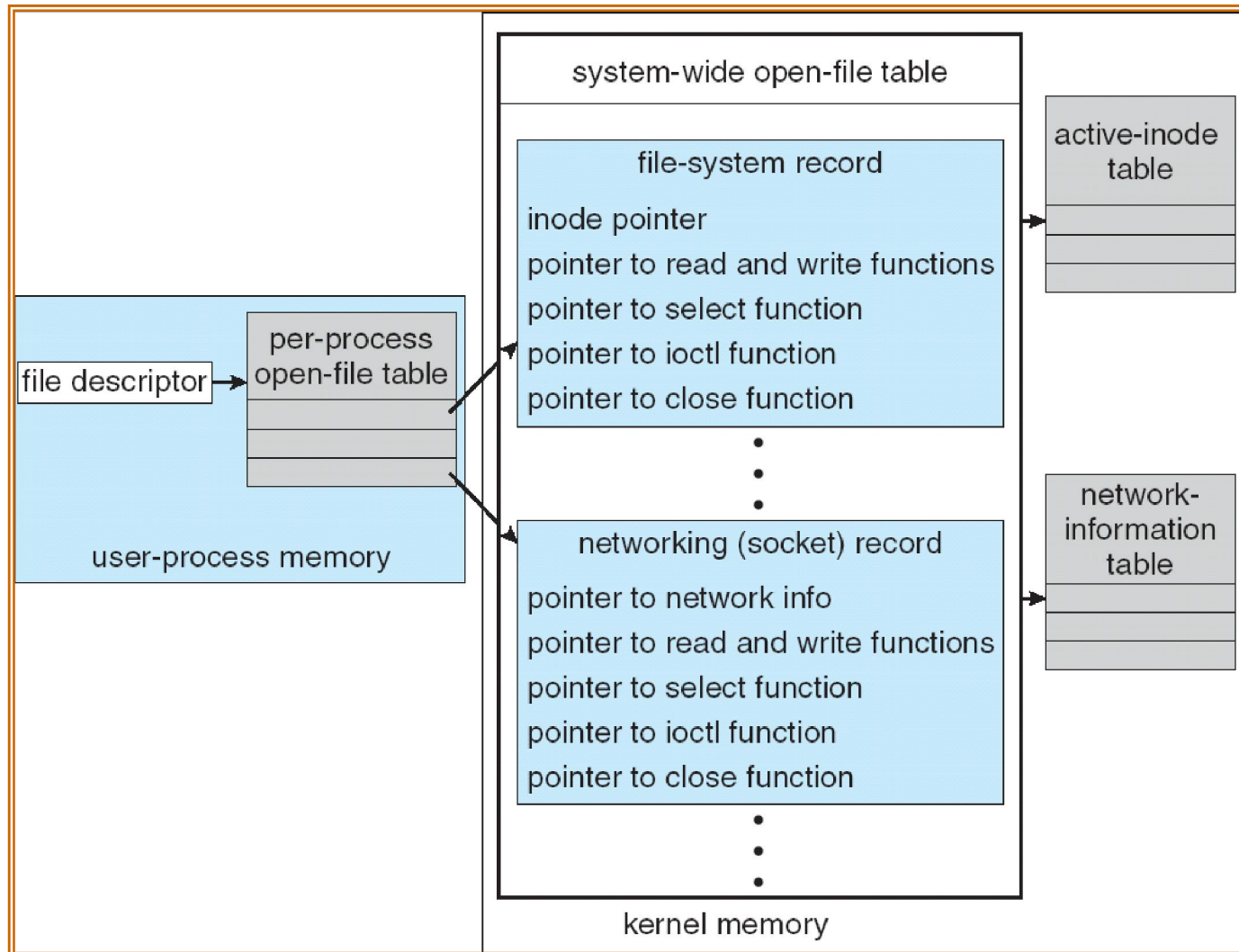
---

- O Kernel mantém informações de estado para componentes de E/S, incluindo tabelas de arquivos abertos, conexões de rede, estados de dispositivos de caractere
- Estruturas de dados muito, muito complexas para rastrear buffers, alocações de memória, blocos modificados (“dirty”)
- Alguns usam métodos orientados a objetos e troca de mensagens para implementar E/S





# Estrutura de E/S do Kernel UNIX





# Requisições de E/s para Operações de HW

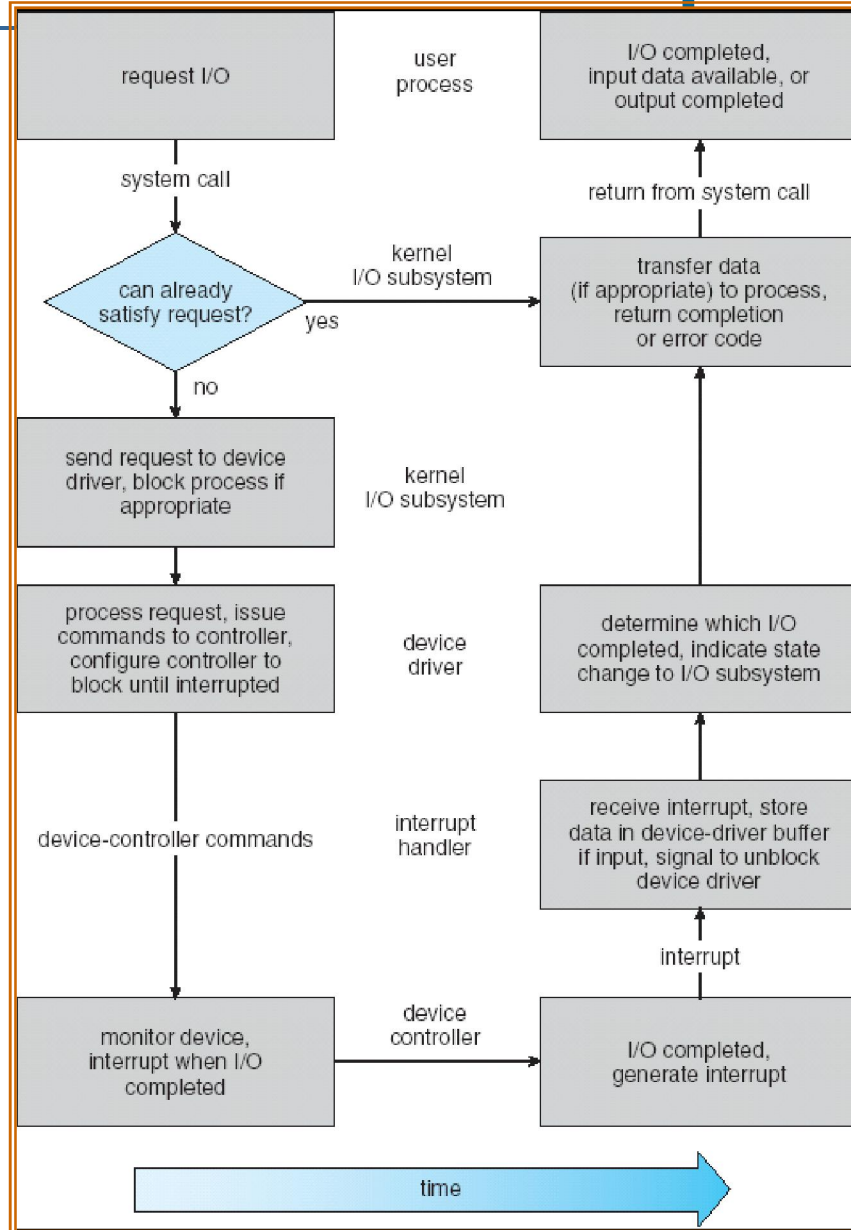
---

- Considere a leitura de um arquivo do disco por um processo:
  - Determinar dispositivo que detém o arquivo
  - Traduzir nome para representação no dispositivo
  - Ler dados fisicamente do disco no buffer
  - Tornar dados disponíveis para processo requisitante
  - Retornar controle ao processo





# Ciclo de Vida de um Requisição de E/S





# Fluxos (STREAMS)

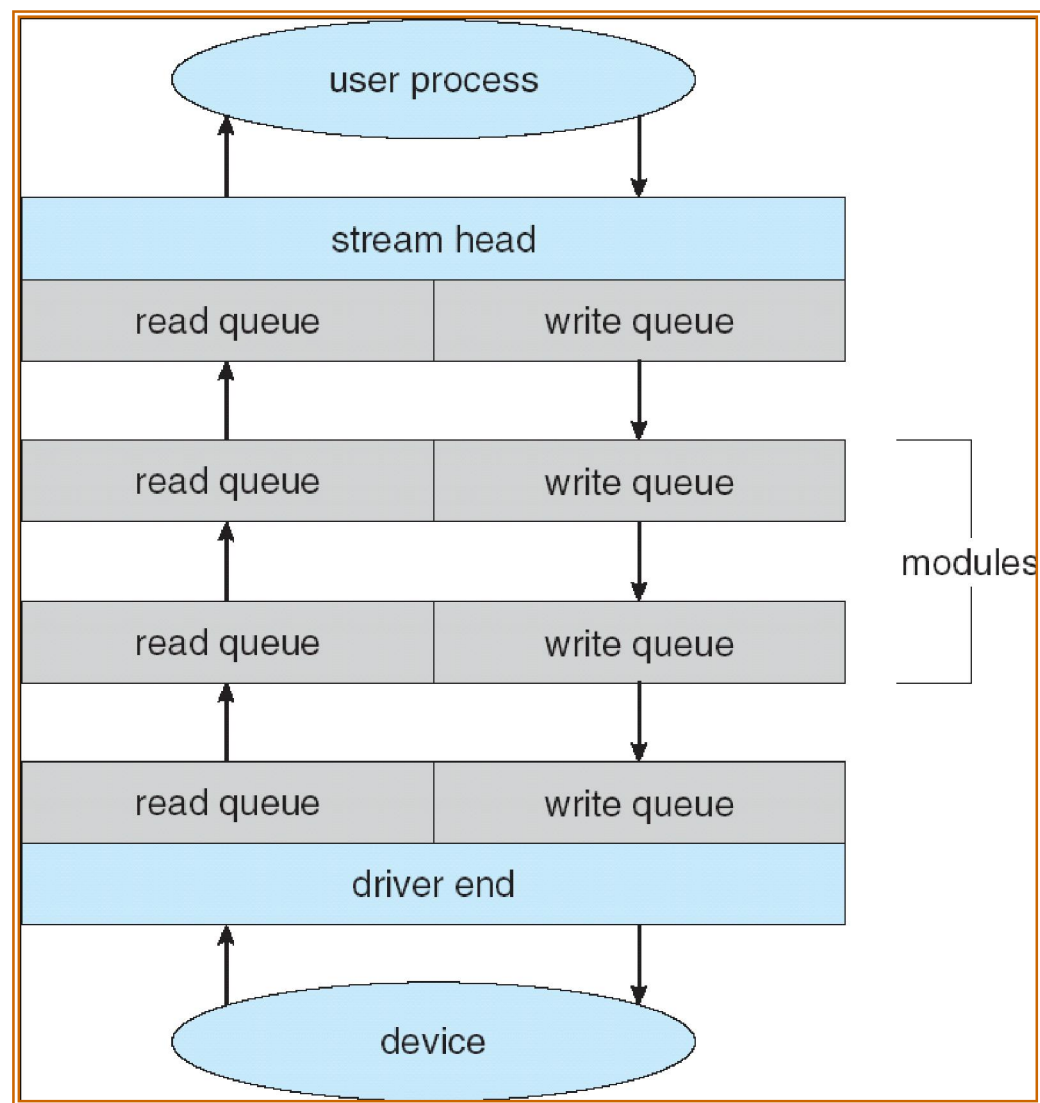
---

- **STREAM** – um canal de comunicação bi-direcional (full-duplex) entre um processo em nível de usuário e um dispositivo no Unix System V e posterior
  
- Uma STREAM consiste de:
  - *Stream head* interfaceia com o processo do usuário
  - *Driver end* interfaceia com o dispositivo
  - zero ou mais módulos STREAM entre eles.
  
- Cada módulo contém uma **fila de leitura** e uma **fila de escrita**
  
- Troca de mensagem é usada para comunicar entre filas





# A Estrutura de uma STREAM





# Desempenho

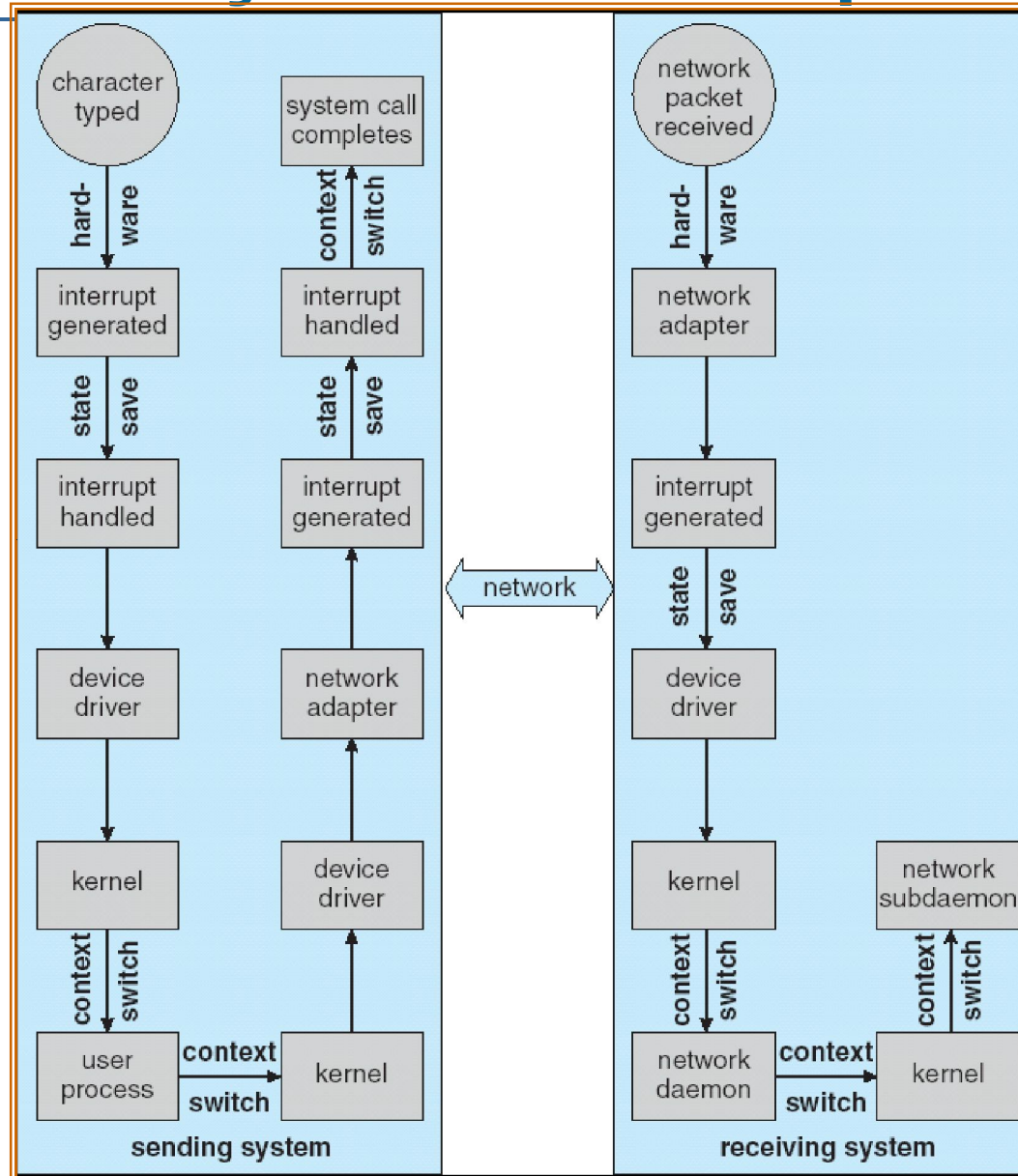
---

- E/S é um fator importante no desempenho do sistema:
  - Demanda CPU para executar driver de dispositivo, código de E/S do kernel
  - Trocas de Contextos devido a interrupções
  - Cópia de dados
  - Tráfego de Rede especialmente estressante





# Comunicações entre Computadores





# Melhorando Desempenho

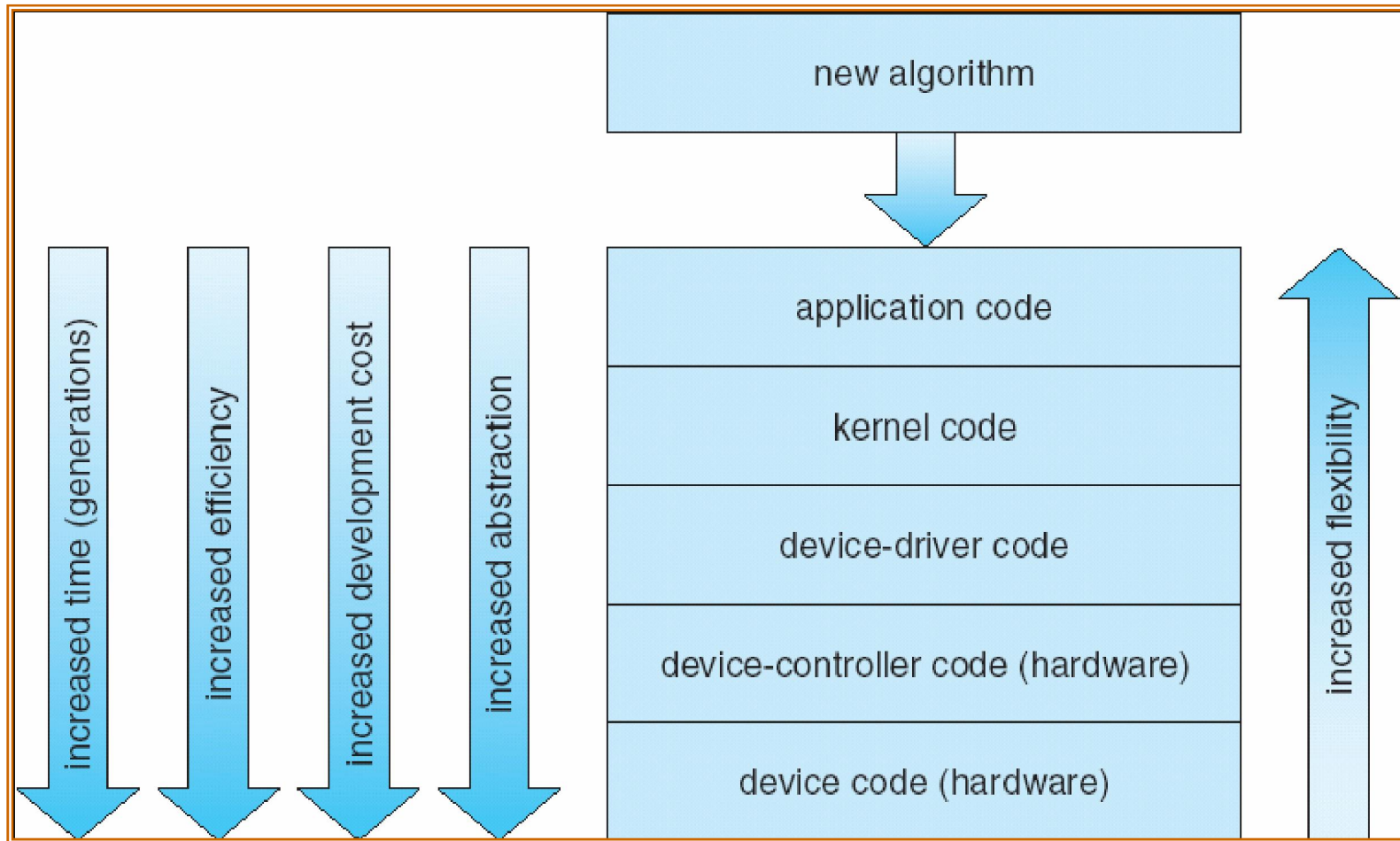
---

- Reduzir o número de trocas de contexto
- Reduzir a cópia de dados
- Reduzir interrupções através do uso de transferências maiores, controladores inteligentes, polling
- User DMA
- Equalizar desempenho de CPU, memória, barramento e E/S para maior vazão





# Progressão da Funcionalidade de Dispositivos



# Fim do Capítulo 13

---

