



UNIVALI

Universidade do Vale do Itajaí
Escola do Mar, Ciência e Tecnologia
Engenharia de Computação

Sistemas de E/S

Agenda

- Visão geral
- Hardware de E/S
- Interface de E/S do aplicativo
- Subsistema de E/S do kernel
- Transformando solicitações de E/S em operações de hardware
- Desempenho

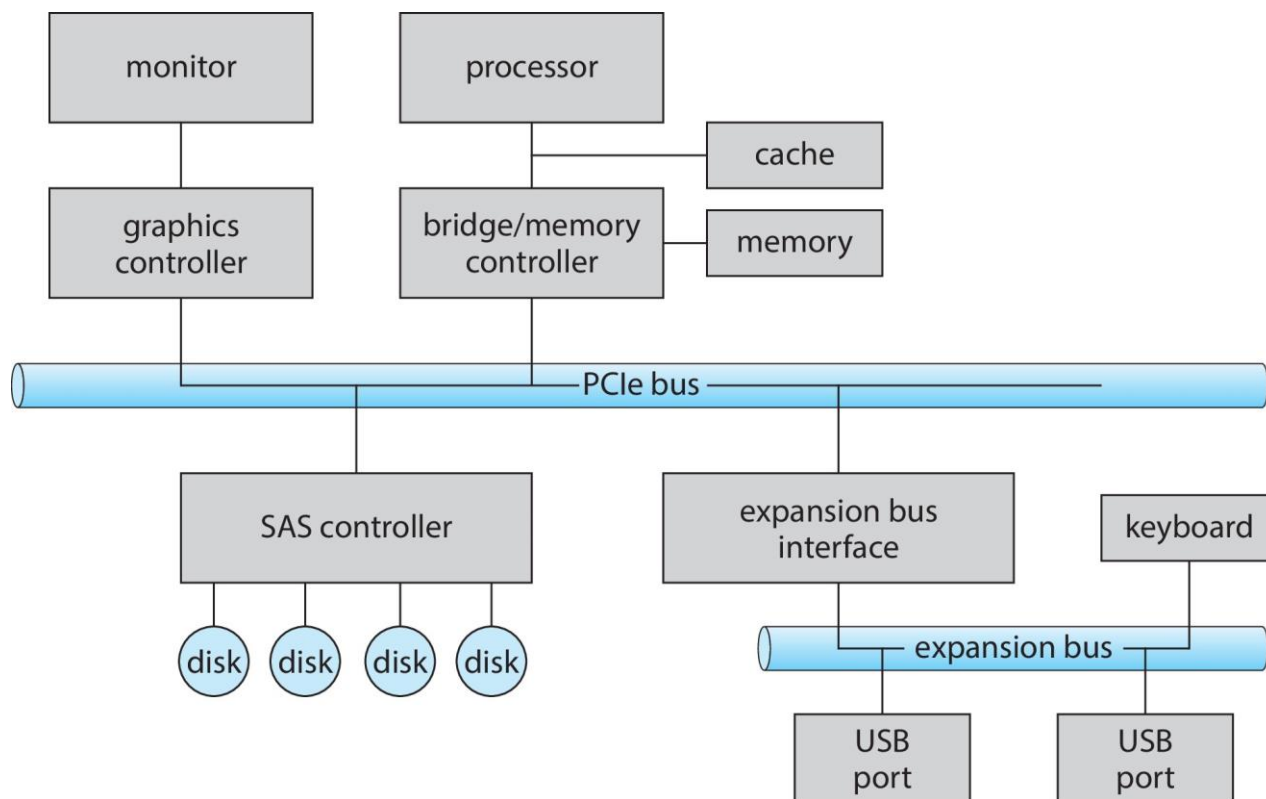
Visão geral

- O gerenciamento de E/S é um componente importante do projeto e da operação do sistema operacional
 - Aspecto importante da operação do computador
 - Os dispositivos de E/S variam muito
 - Vários métodos para controlá-los
 - Gerenciamento de desempenho
 - Novos tipos de dispositivos frequentes
- Portas, barramentos, controladores de dispositivo se conectam a vários dispositivos
- **Drivers de dispositivo** encapsulam detalhes do dispositivo
 - Apresentar uma interface uniforme de acesso ao dispositivo para o subsistema de E/S

Hardware de E/S

- Incrível variedade de dispositivos de E/S
 - Armazenamento, Transmissão e Interface humana
- Conceitos comuns – sinais da interface de dispositivos de E/S com o computador
 - **Porta** – ponto de conexão do dispositivo
 - **Barramento** – **lista encadeada** ou acesso direto compartilhado
 - **PCI** – barramento comum em PCs e servidores, PCI Express (**PCle**)
 - **barramento de expansão** – conecta dispositivos relativamente lentos
 - **Serial-attached SCSI (SAS)** – interface de disco comum
 - **Controlador (host adapter)** – eletrônicos que operam porta, barramento, dispositivo
 - As vezes integrado, as vezes placa de circuito separada (adaptador host), contém processador, microcódigo, memória privada, controlador de barramento, etc.

Uma estrutura típica de barramento de PC



Hardware de E/S (Conta.)

- **Ex: Fibre channel (FC)** é um controlador complexo, geralmente placa de circuito separada (**host-bus adapter, HBA**) conectando-se ao barramento
- Dispositivos de controle de instruções de E/S
- Os dispositivos geralmente têm registradores onde o driver de dispositivo coloca comandos, endereços e dados para gravar ou lê dados de registradores após a execução do comando
 - Registrador de entrada de dados, registrador de saída, registrador de estado, registrador de controle
 - Normalmente 1-4 bytes ou buffer FIFO

Hardware de E/S (Cont.)

- Os dispositivos têm endereços, usados por
 - Instruções diretas de E/S
 - **E/S mapeada na memória**
 - Dados do dispositivo e registros de comandos mapeados para o espaço de endereço do processador
 - Especialmente para espaços de endereço grandes (gráficos)

Ex: Localizações das portas de E/S do dispositivo em 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)

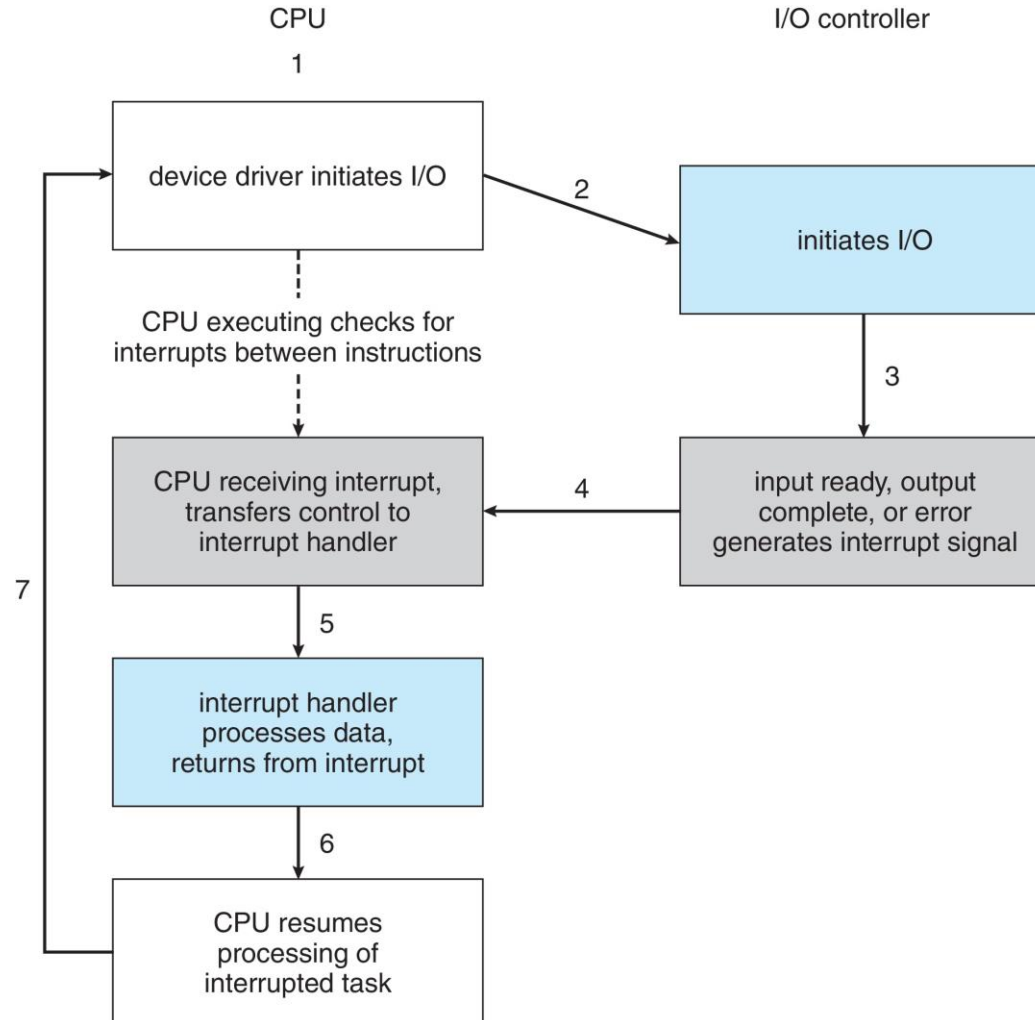
Polling

- Para cada byte de E/S
 1. Leia o bit ocupado do registro de status até 0
 2. O host define o bit de leitura ou gravação e, se a gravação copia os dados no registro de saída de dados
 3. O host define bit pronto para comando
 4. O controlador define o bit ocupado, executa a transferência
 5. O controlador limpa o bit ocupado, o bit de erro, o bit pronto para comando quando a transferência é concluída
- O passo 1 é um ciclo **espera ocupada** cycle para aguardar E/S do dispositivo
 - Razoável se o dispositivo for rápido
 - Mas ineficiente se o dispositivo ficar lento
 - CPU muda para outras tarefas?
 - Mas se perder um ciclo de dados substituídos / perdidos

Interrupção

- Polling pode acontecer em 3 ciclos de instrução
 - Ler status, AND lógico para extrair o bit de status, branch se não zero
 - Como ser mais eficiente se diferente de zero com pouca frequência?
- Linha de solicitação de interrupção da CPU acionada pelo dispositivo de E/S
 - Verificado pelo processador após cada instrução
- **Tratador de interrupção** recebe interrupções
 - **Mascarável** para ignorar ou atrasar algumas interrupções
- **Vetor de interrupção** para despachar a interrupção para corrigir o tratador
 - Mudança de contexto no início e no fim
 - Com base na prioridade
 - Algumas interrupções são **não mascarável**
 - Encadeamento de interrupção se mais de um dispositivo no mesmo número de interrupção

Ciclo de E/S controlado por interrupção



Interrupção (Cont.)

- Mecanismo de interrupção também usado para **exceções**
 - Encerrar processo, sistema de falha devido a erro de hardware
- Falha de página é executada quando há erro de acesso à memória
- A chamada do sistema é executada via trap para acionar o kernel para executar a solicitação
- Sistemas multi-CPU podem processar interrupções simultaneamente
 - Se o sistema operacional projetado para lidar com isso
- Usado para processamento sensível ao tempo, frequente, deve ser rápido

Latência

- Enfatizando o gerenciamento de interrupções porque mesmo os sistemas de usuário único gerenciam centenas de interrupções por segundo e os servidores centenas de milhares
- Por exemplo, um desktop macOS silencioso gerou 23.000 interrupções em 10 segundos

Windows:

Abra o Gerenciador de Dispositivos, vá em exibir e click em recursos por tipo, agora vá em IRQ

```
Fri Nov 25 13:55:59                                0:00:10
```

	SCHEDULER	INTERRUPTS
total_samples	13	22998
delays < 10 usecs	12	16243
delays < 20 usecs	1	5312
delays < 30 usecs	0	473
delays < 40 usecs	0	590
delays < 50 usecs	0	61
delays < 60 usecs	0	317
delays < 70 usecs	0	2
delays < 80 usecs	0	0
delays < 90 usecs	0	0
delays < 100 usecs	0	0
total < 100 usecs	13	22998

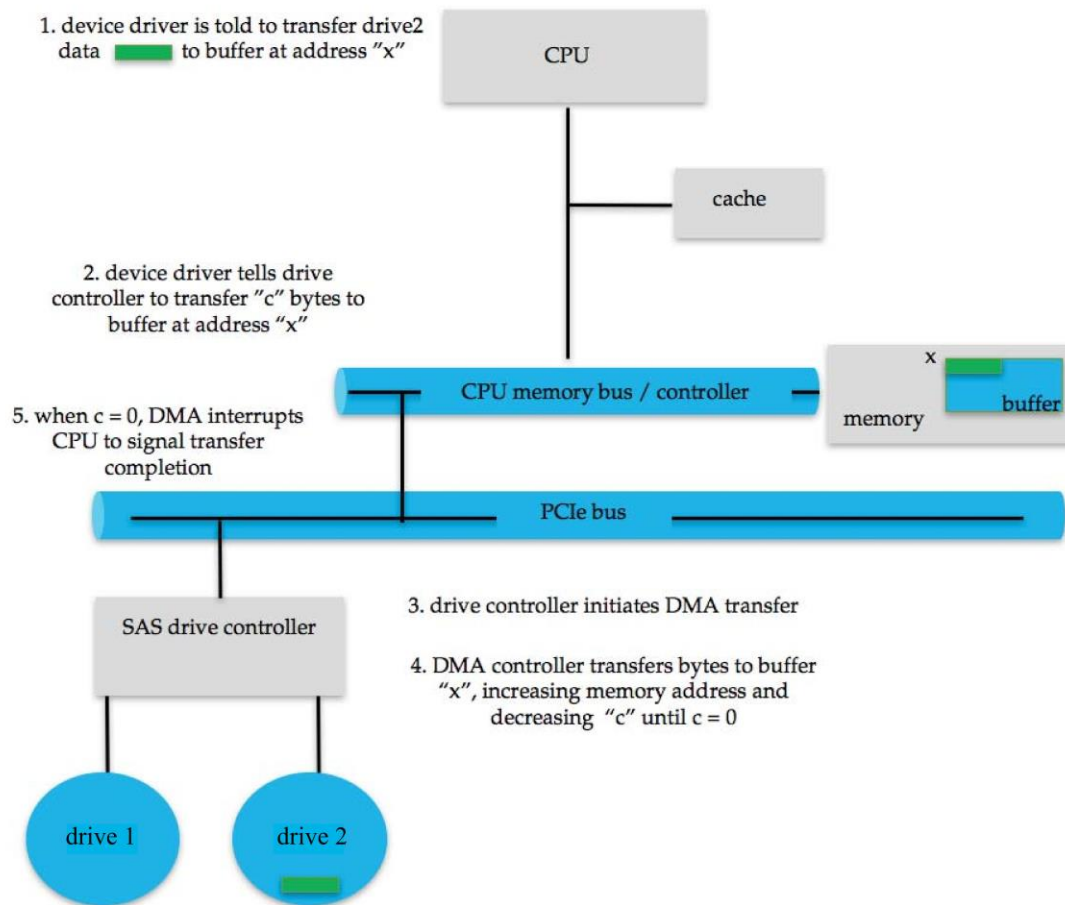
Tabela de vetores de eventos do processador Intel Pentium

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 à memória

- Usado para evitar **E/S programada** (um byte de cada vez) para grande movimentação de dados
- Requer controlador **DMA**
- Ignora a CPU para transferir dados diretamente entre o dispositivo de E/S e a memória
- O SO grava o bloco de comando DMA na memória
 - Endereços de origem e destino
 - Modo de leitura ou gravação
 - Contagem de bytes
 - Grava o local do bloco de comando no controlador DMA
 - Mestre de barramento do controlador DMA – pega o barramento da CPU
 - **Roubo de ciclo** da CPU, mas ainda muito mais eficiente
 - Quando terminar, interrompe para sinalizar a conclusão
- Versão que está ciente de endereços virtuais pode ser ainda mais eficiente - **DVMA**

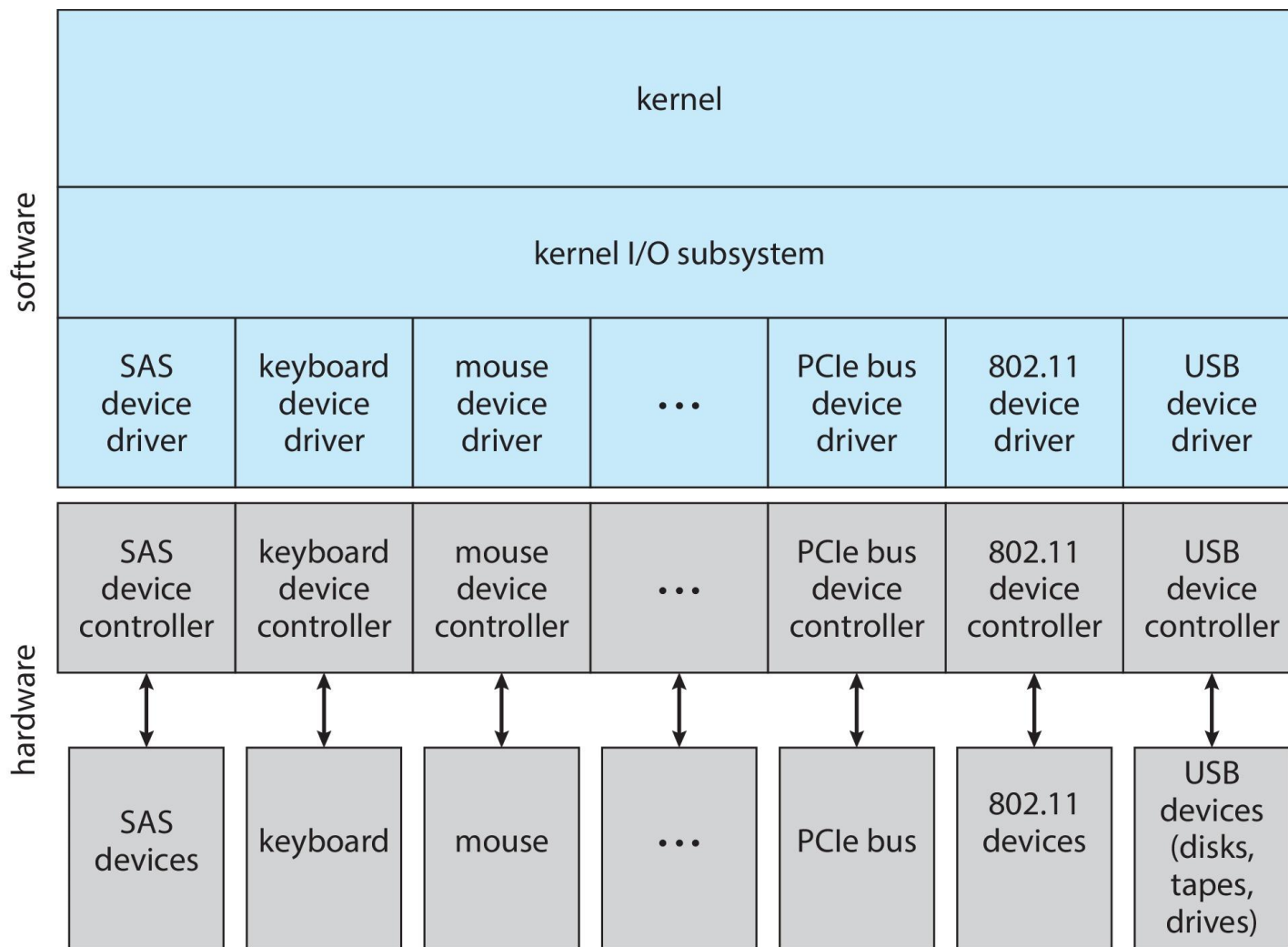
Processo de seis etapas para executar a transferência DMA



Aplicação da Interface de E/S

- As chamadas do sistema de E/S encapsulam comportamentos de dispositivo em classes genéricas
- A camada de driver de dispositivo oculta as diferenças entre os controladores de E/S do kernel
- Novos dispositivos que falam de protocolos já implementados não precisam de trabalho extra
- Cada sistema operacional tem suas próprias estruturas de subsistema de E/S e estruturas de driver de dispositivo
- Os dispositivos variam em muitas dimensões
 - **Fluxo de caracteres** ou **bloco**
 - **Sequencial** ou **acesso aleatório**
 - **Síncrono** ou **assíncrono** (ou ambos)
 - **Compartilhável** ou **dedicado**
 - **Velocidade de operação**
 - **leitura-gravação, somente leitura** ou **somente gravação**

Uma estrutura de E/S do kernel



Características dos 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

Características dos dispositivos de E/S (Cont.)

- Sutilezas de dispositivos manipulados por drivers de dispositivo
- De modo geral, os dispositivos de E/S podem ser agrupados pelo sistema operacional em
 - Bloco E/S
 - E/S de caracteres (Stream)
 - Acesso a arquivos mapeados na memória
 - Soquetes de rede
- Para manipulação direta de características específicas do dispositivo de E/S, geralmente uma porta de escape / traseira (famosa back door)
 - Chamada Unix `ioctl()` para enviar bits arbitrários para um registro de controle de dispositivo e dados para o registro de dados do dispositivo

```
brw-rw---- 1 root disk 8, 0 Mar 16 09:18 /dev/sda
brw-rw---- 1 root disk 8, 1 Mar 16 09:18 /dev/sda1
brw-rw---- 1 root disk 8, 2 Mar 16 09:18 /dev/sda2
brw-rw---- 1 root disk 8, 3 Mar 16 09:18 /dev/sda3
```

Dispositivos de bloqueio e caracteres

- Os dispositivos de bloco incluem unidades de disco
 - Os comandos incluem ler, gravar, procurar
 - **E/S Raw**, **E/S direta** ou acesso ao sistema de arquivos
 - Acesso a arquivos mapeados na memória possível
 - Arquivo mapeado para memória virtual e clusters trazidos via paginação de demanda
 - DMA
- Os dispositivos de caracteres incluem teclados, mouses, portas seriais
 - Os comandos incluem **get()**, **put()**
 - Bibliotecas em camadas na parte superior permitem a edição de linha

Dispositivos de rede

- Variando o suficiente de bloco e caracter para ter interface própria
- Linux, Unix, Windows e muitos outros incluem interface de soquete
 - Separa o protocolo de rede da operação de rede
 - Inclui funcionalidade `select()`
- As abordagens variam muito (pipes, FIFOs, streams, queues, mailboxes)

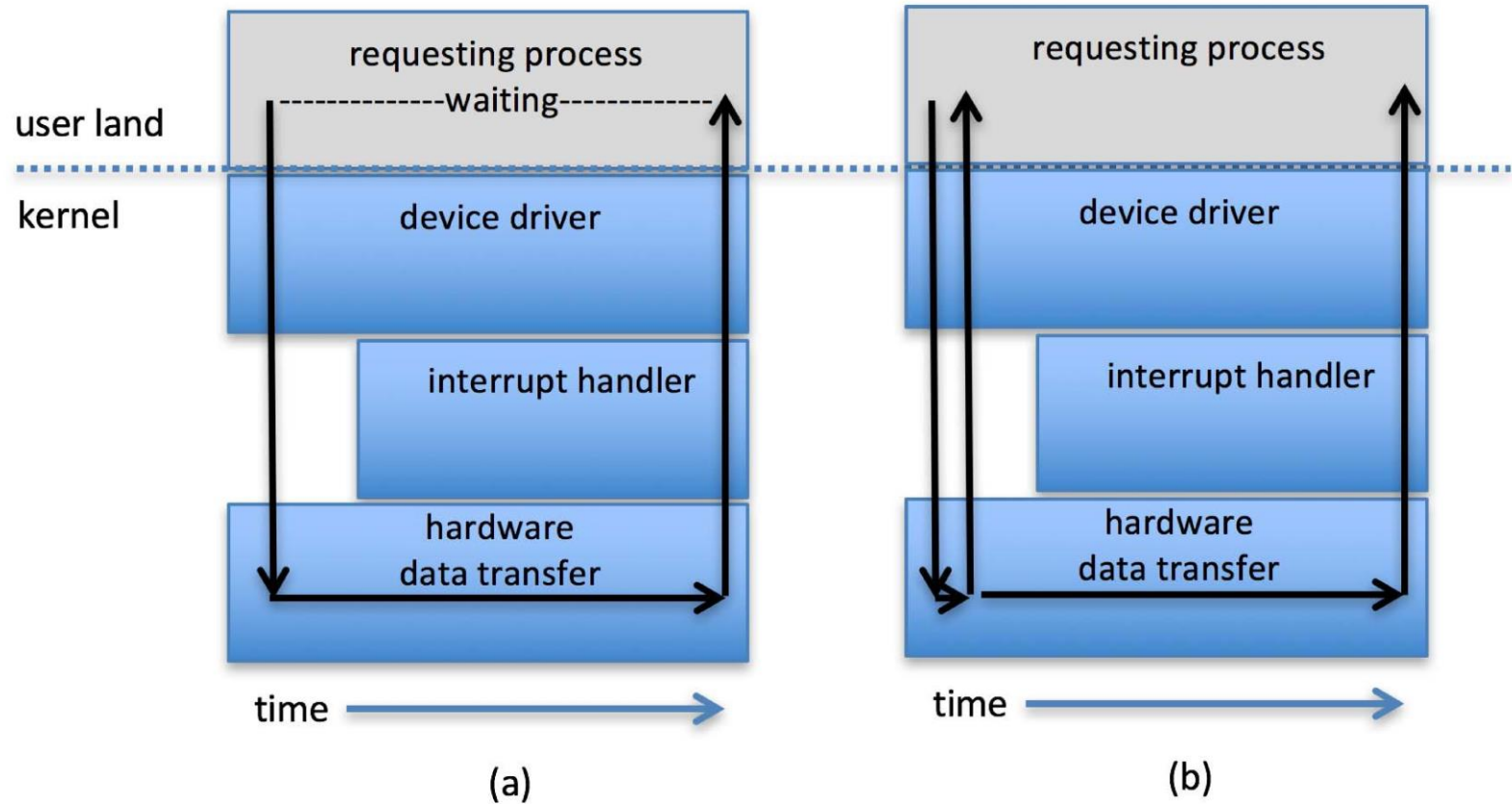
Clocks e Timers

- Fornecer tempo atual, tempo decorrido, temporizador
- Resolução normal de cerca de 1/60 segundo
- Alguns sistemas fornecem temporizadores de alta resolução
- **Temporizador de intervalo programável** usado para temporizações, interrupções periódicas

E/S sem bloqueio e assíncrona

- **Bloqueio** - processo suspenso até que a E/S seja concluída
 - Fácil de usar e entender
 - Insuficiente para algumas necessidades
- **Sem bloqueio** - A chamada de E/S retorna quando disponível
 - Interface do usuário, cópia de dados (E/S em buffer)
 - Implementado via multi-threading
 - Retorna rapidamente com a contagem de bytes lidos ou gravados
 - `select()` para descobrir se os dados estão prontos, em seguida, `read()` ou `write()` para transferir
- **Assíncrono** - processo é executado enquanto a E/S é executada
 - Difícil de usar
 - Os sinais do subsistema de E/S são processados quando a E/S é concluída

Dois métodos de E/S



Subsistema de E/S do kernel

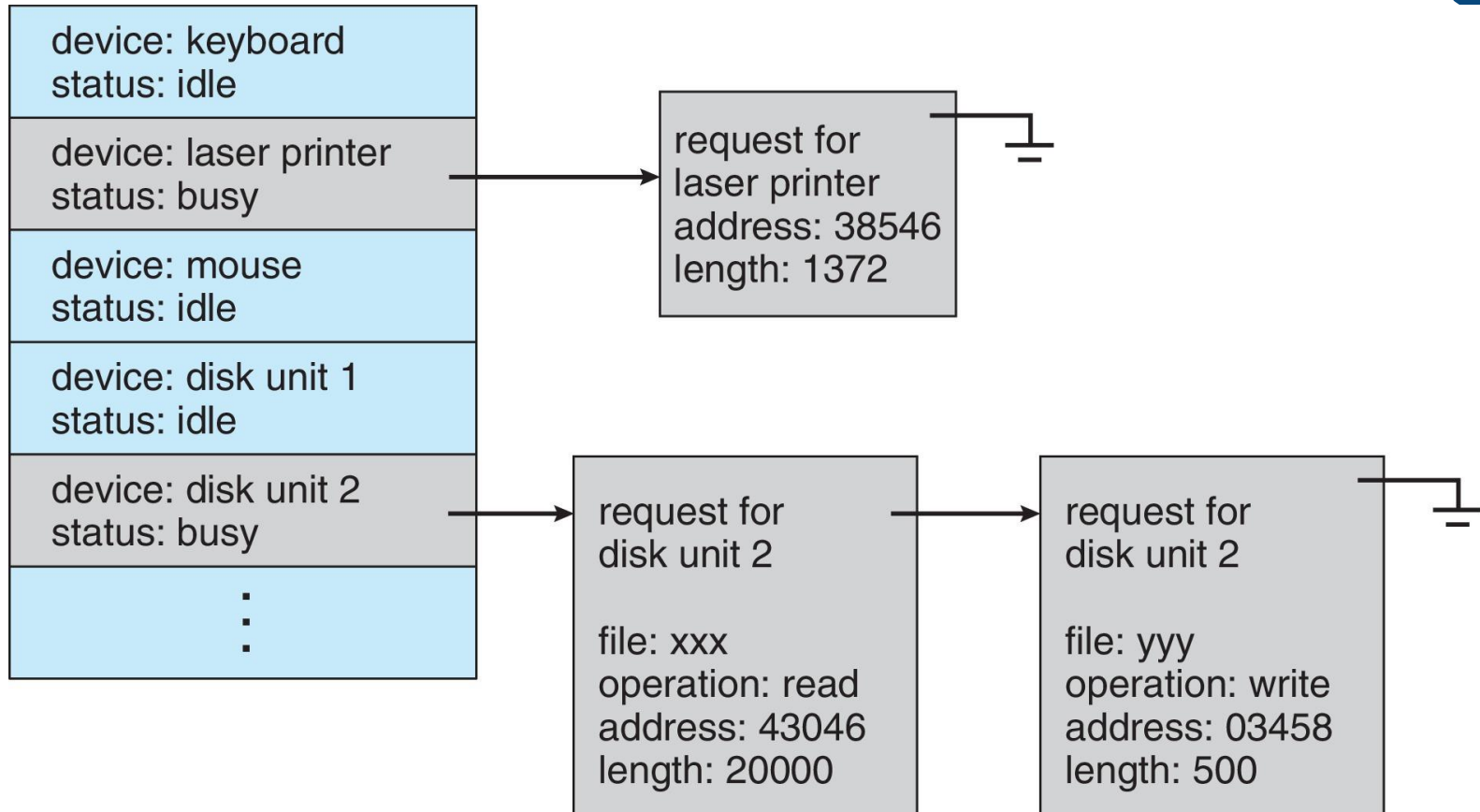
- Escalonamento

- Alguns pedidos de solicitação de E/S por meio de fila por dispositivo
- Alguns sistemas operacionais tentam a justiça
- Alguns implementam a Qualidade de Serviço (i.e. IPQOS)

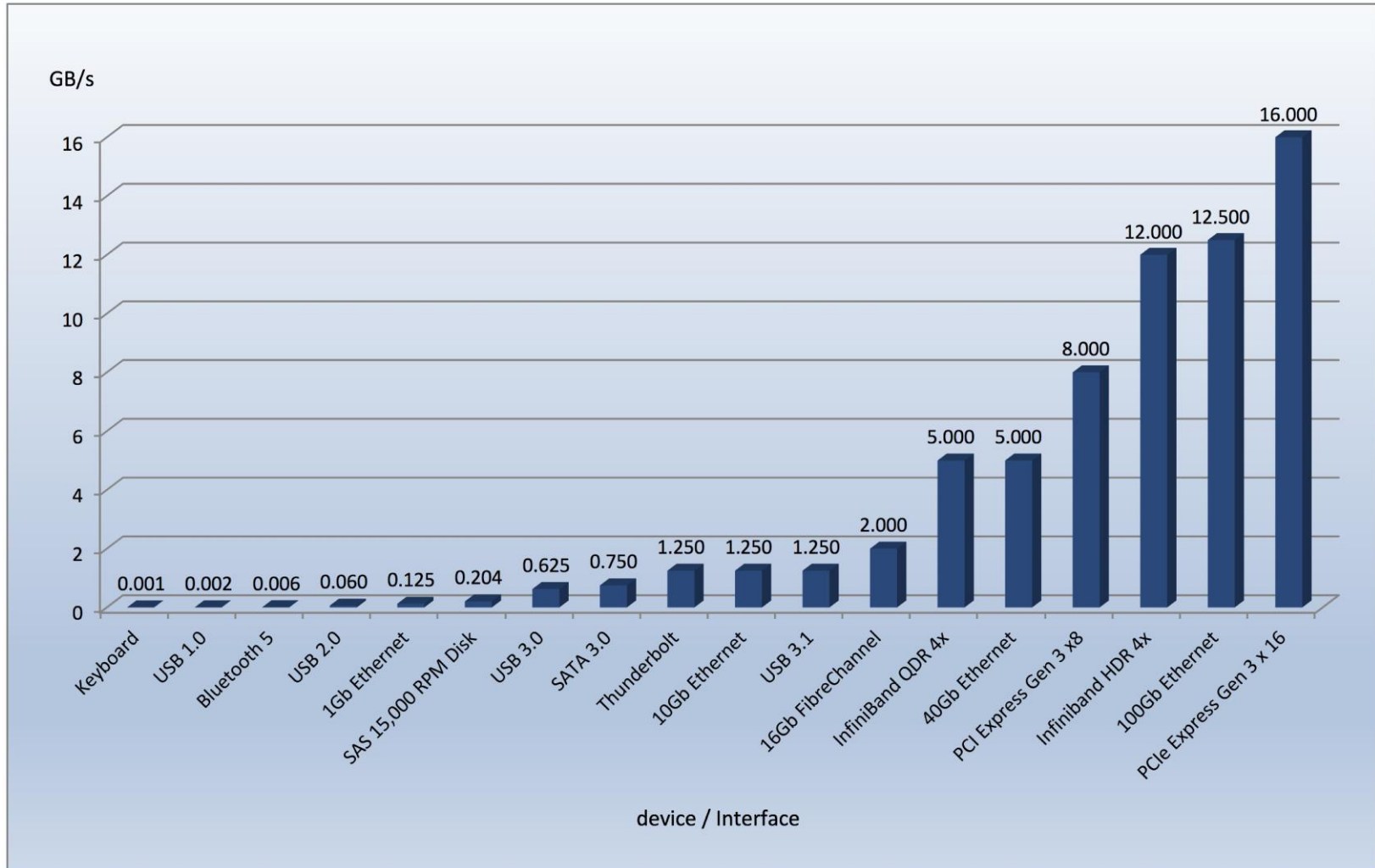
- **Buffering** - armazenar dados na memória durante a transferência entre dispositivos

- Para lidar com a incompatibilidade de velocidade do dispositivo
- Para lidar com a incompatibilidade de tamanho de transferência de dispositivo
- Para manter a "semântica de cópia"
- **Buffer duplo** – duas cópias dos dados
 - Kernel e usuário
 - Tamanhos variados
 - Completo / sendo processado e não cheio / sendo usado
 - Copy-on-write pode ser usado para eficiência em alguns casos

Tabela de status do dispositivo



Dispositivos comuns de E/S de PC e Data-center e velocidades de interface



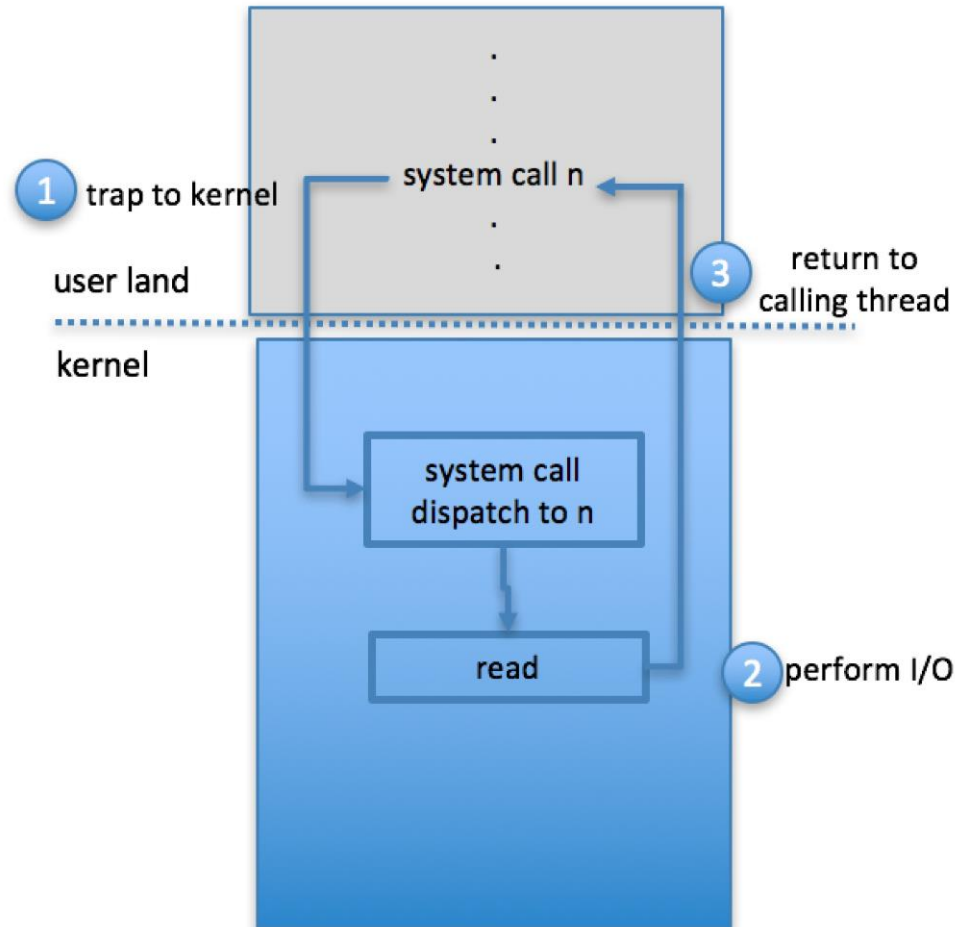
Subsistema de E/S do kernel

- **Caching** - dispositivo mais rápido que contém cópia de dados
 - Sempre apenas uma cópia
 - Chave para o desempenho
 - Às vezes combinado com buffering
- **Spooling** - segurar saída para um dispositivo
 - Se o dispositivo puder atender a apenas uma solicitação por vez
 - ou seja, Impressão
- **Reserva de dispositivo** - fornece acesso exclusivo a um dispositivo
 - O sistema exige a atribuição e a desalocação
 - Cuidado com o impasse

Proteção de E/S

- O processo do usuário pode acidentalmente ou propositalmente tentar interromper a operação normal por meio de instruções de E/S ilegais
 - Todas as instruções de E/S definidas como privilegiadas
 - A E/S deve ser executada por meio de chamadas do sistema
 - Os locais de memória mapeada na memória e na porta de E/S também devem ser protegidos

Uso de uma chamada do sistema para executar 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, estado do dispositivo de caracteres
- Muitas, muitas estruturas de dados complexas para rastrear buffers, alocação de memória, blocos "sujos"
- Alguns usam métodos orientados a objetos e passagem de mensagens para implementar E/S
 - O Windows usa a passagem de mensagens
 - Mensagem com informações de E/S passadas do modo de usuário para o kernel
 - Mensagem modificada à medida que flui para o driver de dispositivo e de volta para o processo
 - Prós / contras?

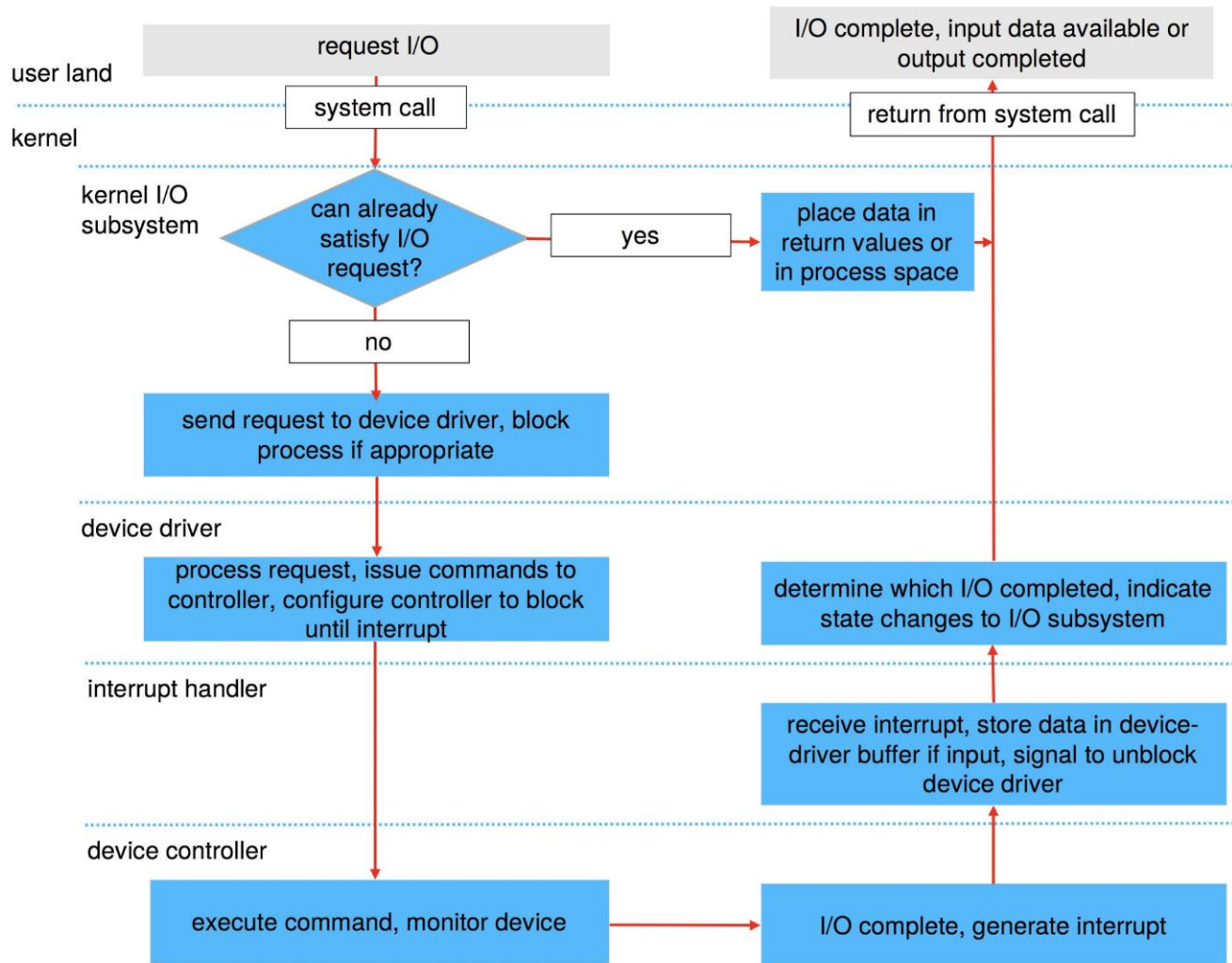
Resumo do subsistema de E/S do kernel

- Em resumo, o subsistema de E/S coordena uma extensa coleção de serviços que estão disponíveis para aplicativos e para outras partes do kernel
 - Gerenciamento do espaço de nome para arquivos e dispositivos
 - Controle de acesso a arquivos e dispositivos
 - Controle de operação (por exemplo, um modem não pode procurar())
 - Alocação de espaço do sistema de arquivos
 - Alocação de dispositivos
 - Armazenamento em buffer, cache e spool
 - Agendamento de E/S
 - Monitoramento do status do dispositivo, tratamento de erros e recuperação de falhas
 - Configuração e inicialização do driver de dispositivo
 - Gerenciamento de energia de dispositivos de E/S
- Os níveis superiores dos dispositivos de acesso ao subsistema de E/S através da interface uniforme fornecida pelos drivers de dispositivo

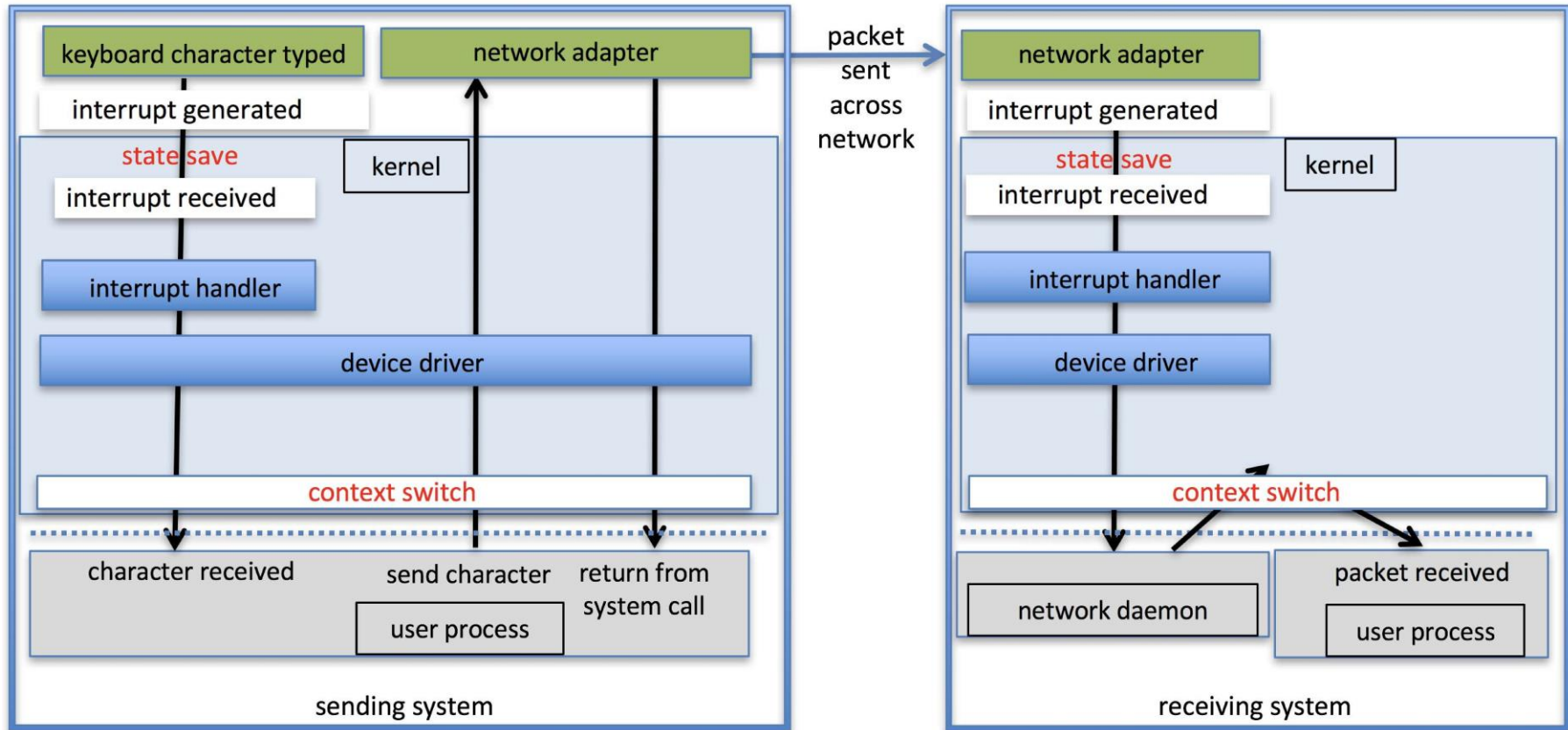
Transformando solicitações de E/S em operações de hardware

- Considere ler um arquivo do disco para um processo:
 - Determinar o arquivo de retenção do dispositivo
 - Traduzir nome para representação do dispositivo
 - Ler fisicamente dados do disco para o buffer
 - Disponibilizar dados para o processo de solicitação
 - Retornar o controle ao processo

Ciclo de vida de uma solicitação de E/S



Comunicação entre computadores



Melhorando o desempenho

- Reduzir o número de opções de contexto
- Reduza a cópia de dados
- Reduza as interrupções usando grandes transferências, controladores inteligentes, sondagem
- Usar DMA
- Use dispositivos de hardware mais inteligentes
- Equilibre o desempenho da CPU, da memória, do barramento e de E/S para obter a mais alta taxa de transferência
- Mova processos/daemons de modo de usuário para threads do kernel

Progressão da funcionalidade do dispositivo

