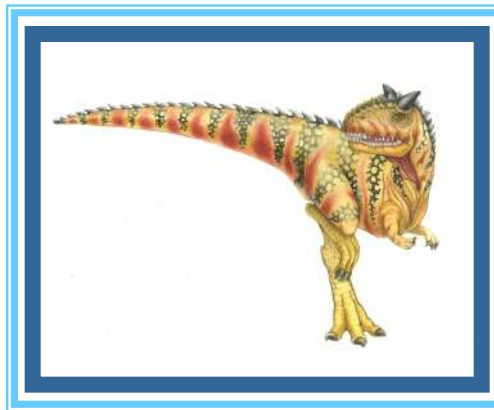


Chapter 13: I/O Systems



O que "ele" faz?

Gerencia toda a comunicação entre dispositivos e sistema

- Tratamento de interrupções
- Tratamento de erros
- Interface entre dispositivos e o “sistema”





Overview

- Gerenciamento de I/O é o “componente” mais abrangente tanto no projeto quanto na operação de um S.O.
 - Essencial para o funcionamento de qualquer sistema computacional
 - Grande variedade de dispositivos
 - Várias formas de realizar controle sobre eles
 - Gerenciamento de performance
 - Novos tipos de dispositivos desenvolvidos com frequência
- Portas, barramentos, controladores de dispositivos conectam a vários dispositivos;
- **Drivers de dispositivos** encapsulam detalhes dos dispositivos





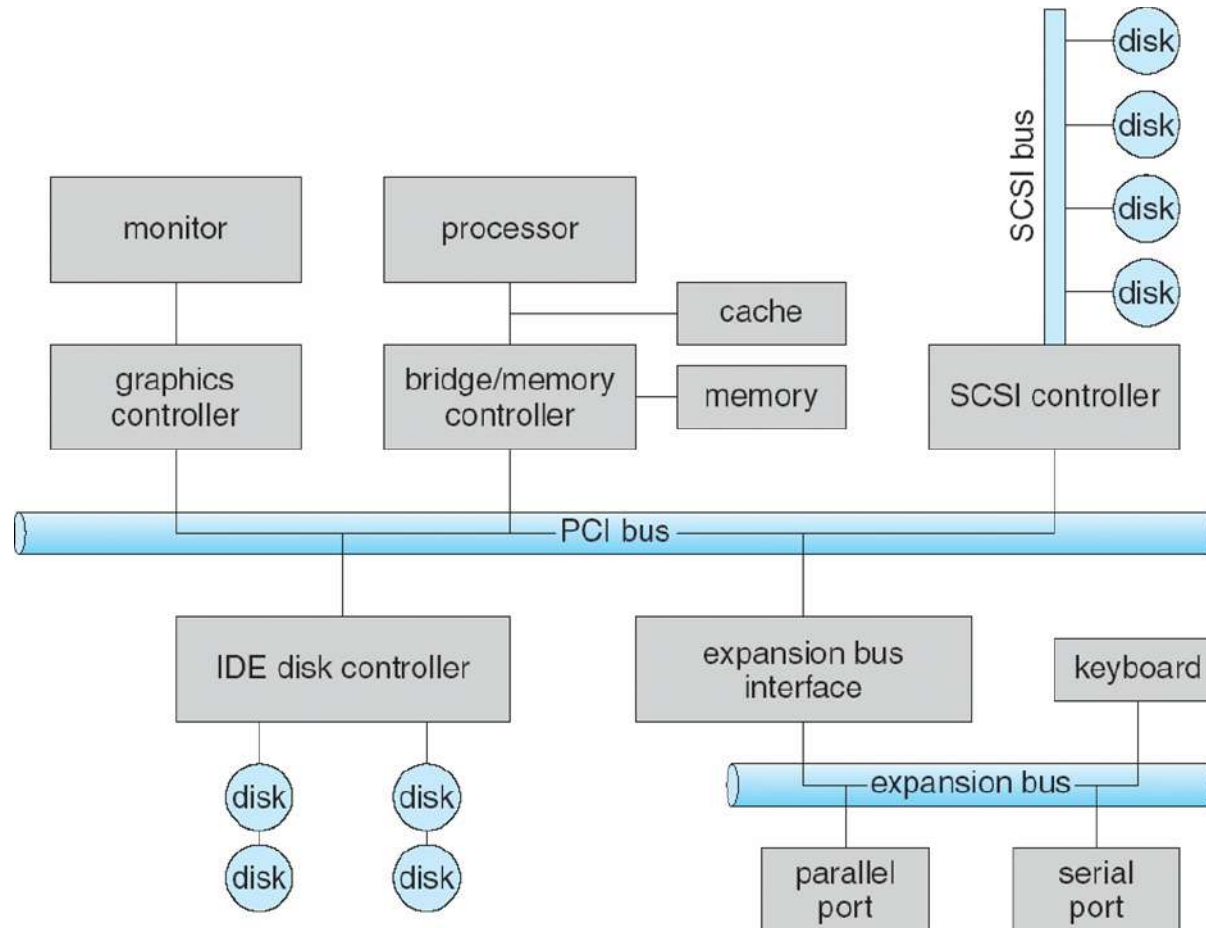
Hardware de E/S

- Grande variedade quanto a:
 - Armazenamento
 - Tx de transmissão
 - *Human-interface*
- Conceitos comuns
 - **Port** – ponto de conexão para o dispositivo
 - **Bus - daisy chain** or acesso compartilhado
 - ▶ **PCI** bus common in PCs and servers, PCI Express (**PCle**)
 - ▶ **expansion bus** interconecta dispositivos mais lentos
 - **Controlador (host adapter)** – dispositivo eletrônico que controla porta, barramento e dispositivos
 - ▶ Pode ser integrado à placa principal ou separado (placa própria)
 - ▶ Contém processador, memória (privada), controlador de barramento...





Estrutura de barramento do PC



**Dispositivos (I/O) “tratam”
da mesma forma a
informação?**



Blocos

x

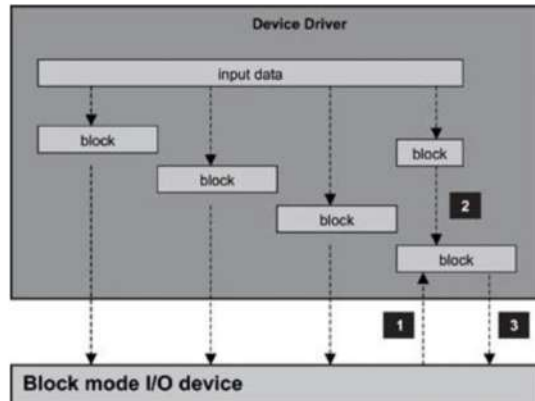
Caracteres

- Armazenamento da informação em blocos de tamanho fixo
- O endereçamento de cada bloco é único!
- Transferência, independente do sentido, acontece em blocos!
- Disco, floppy, pendrive, CD-ROM

Informação (envio e recepção) em caracteres

Não é endereçável;

Impressora, interface de rede, mouse, teclado



http://www.inf.pucrio.br/~emoreno/undergraduate/CC/sisop/class_files/Aula03



Dispositivos (I/O) se comunicam em velocidades diferentes?



Dispositivo	Taxa de dados
Teclado	10 bytes/s
Mouse	100 bytes/s
Modem 56 K	7 KB/s
Scanner	400 KB/s
Filmadora <i>camcorder</i> digital	3,5 MB/s
Rede sem fio 802,11g	6,75 MB/s
CD-ROM 52x	7,8 MB/s
Fast Ethernet	12,5 MB/s
Cartão flash compacto	40 MB/s
FireWire (IEEE 1394)	50 MB/s
USB 2.0	60 MB/s
Padrão SONET OC-12	78 MB/s
Disco SCSI Ultra 2	80 MB/s
Gigabit Ethernet	125 MB/s
Drive de disco SATA	300 MB/s
Fita Ultrium	320 MB/s
Barramento PCI	528 MB/s

Tabela 5.1 Algumas taxas de dados típicas de dispositivos, placas de redes e barramentos.



**Todos os dispositivos
realizam operações de
I/O da mesma forma?**



E/S mapeada na memória

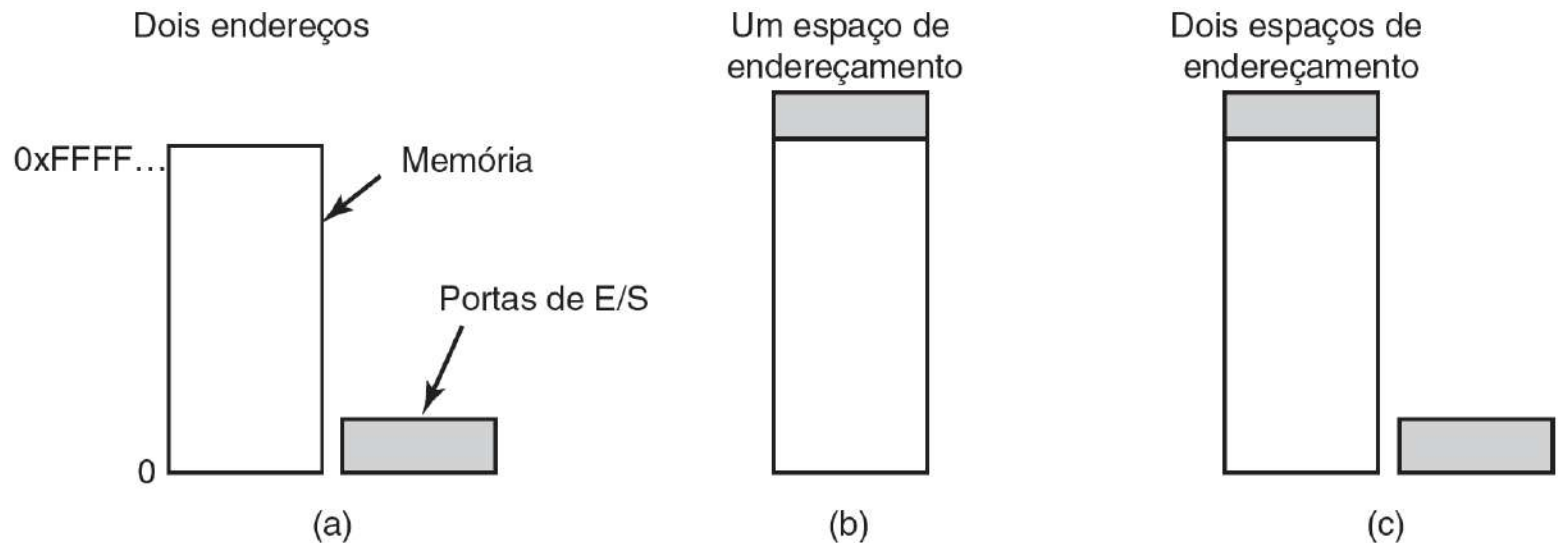
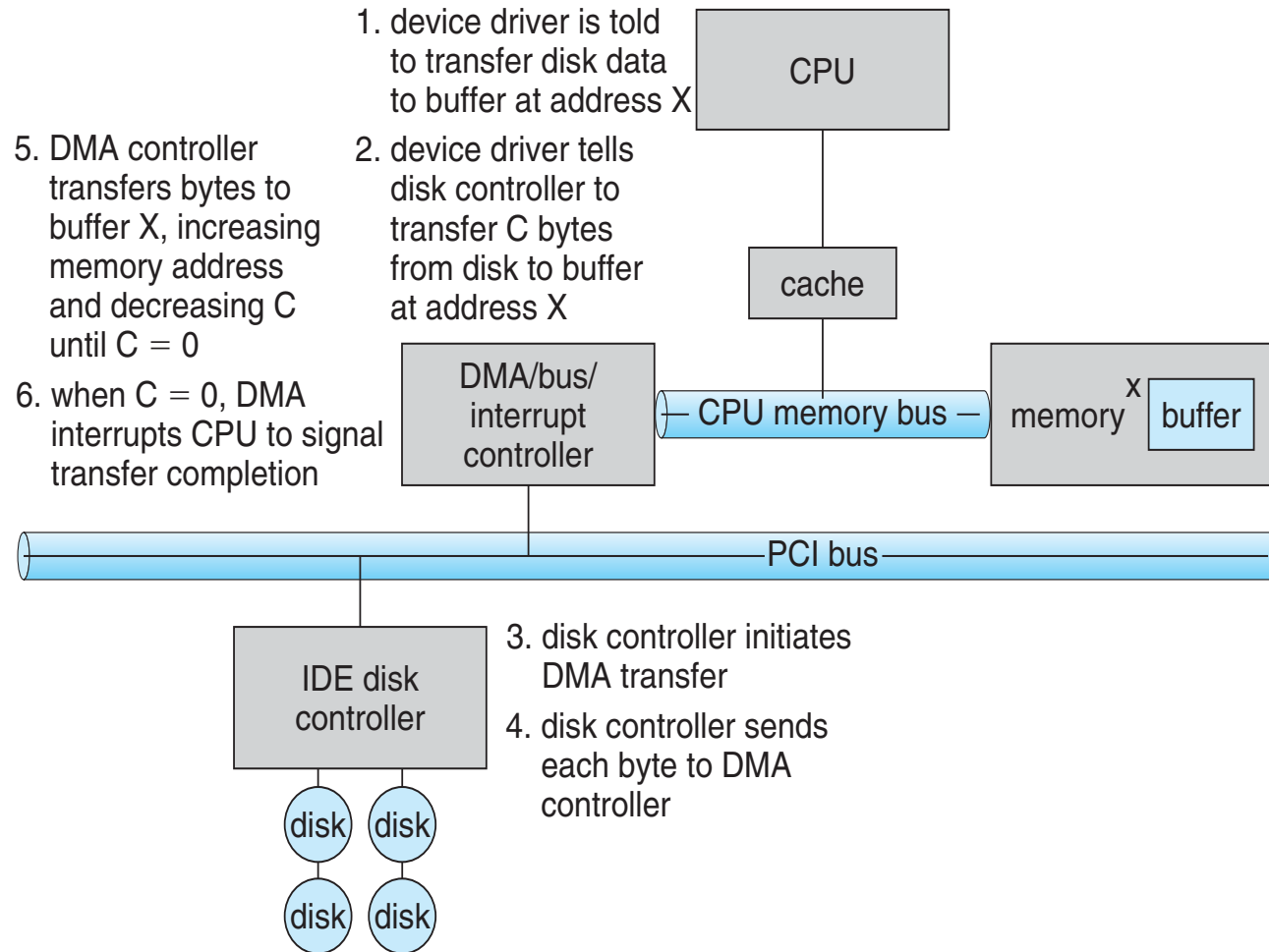


Figura 5.1 (a) Espaços de memória e E/S separados. (b) E/S mapeada na memória. (c) Híbrido.





A transferência por DMA (6 passos)





Acesso direto à memória (DMA)

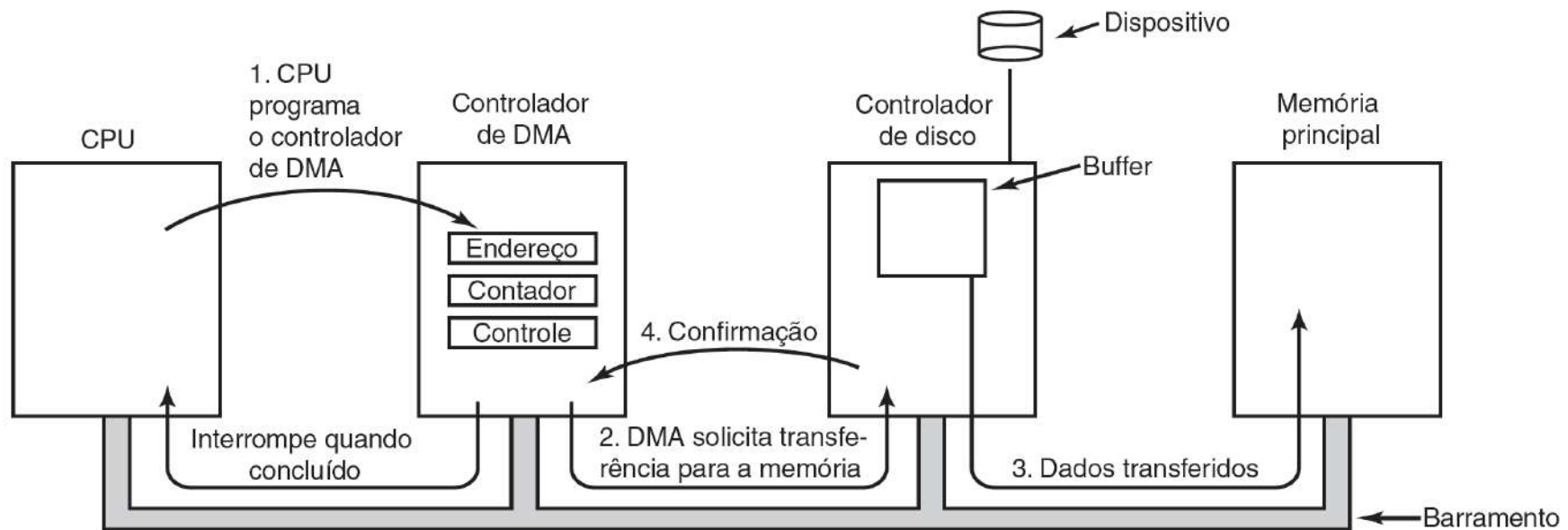


Figura 5.3 Operação de transferência utilizando DMA.





Interrupções

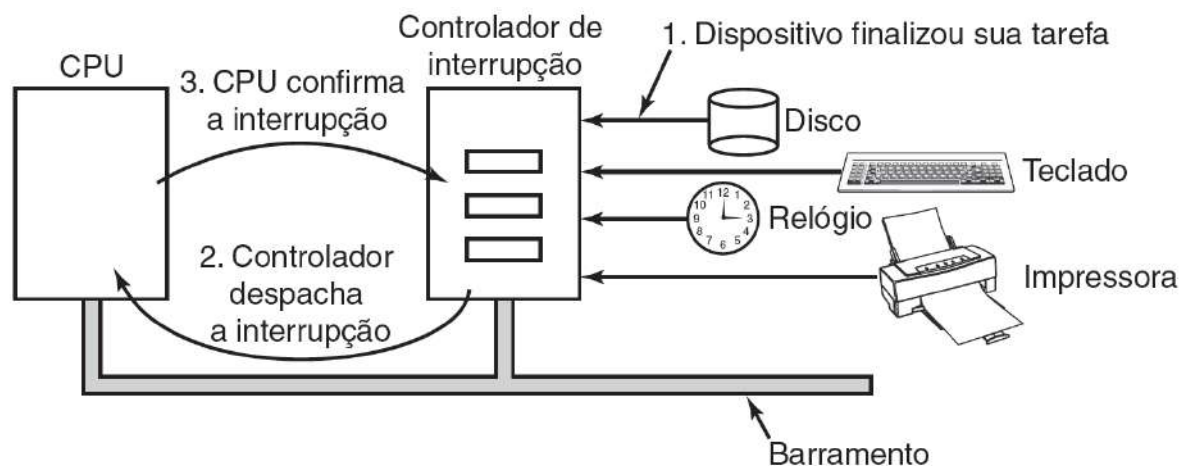


Figura 5.4 Como ocorre uma interrupção. As conexões entre os dispositivos e o controlador de interrupção atualmente utilizam linhas de interrupção no barramento, em vez de cabos dedicados.

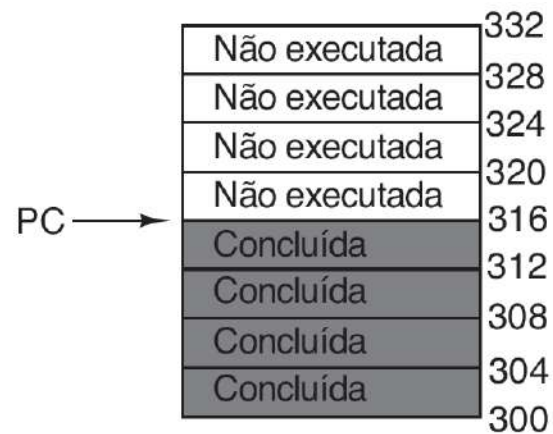




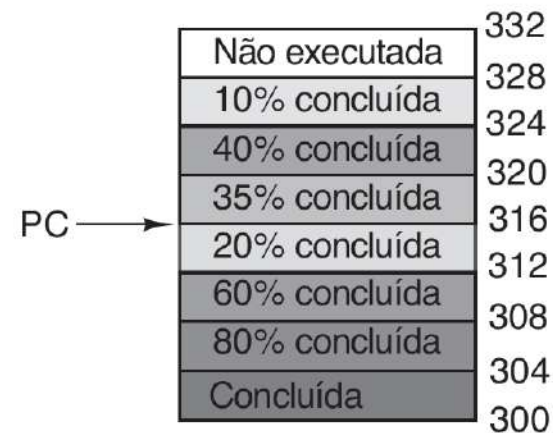
Interrupções precisas e imprecisas

- Propriedades de **uma interrupção precisa**:
 1. O contador de programa (Program Counter) é salvo em local conhecido.
 2. Todas as instruções anteriores à aquela apontada pelo PC foram executadas.
 3. Nenhuma instrução posterior à apontada pelo PC foi executada.
 4. O estado de execução da instrução apontada pelo PC é conhecido.





(a)



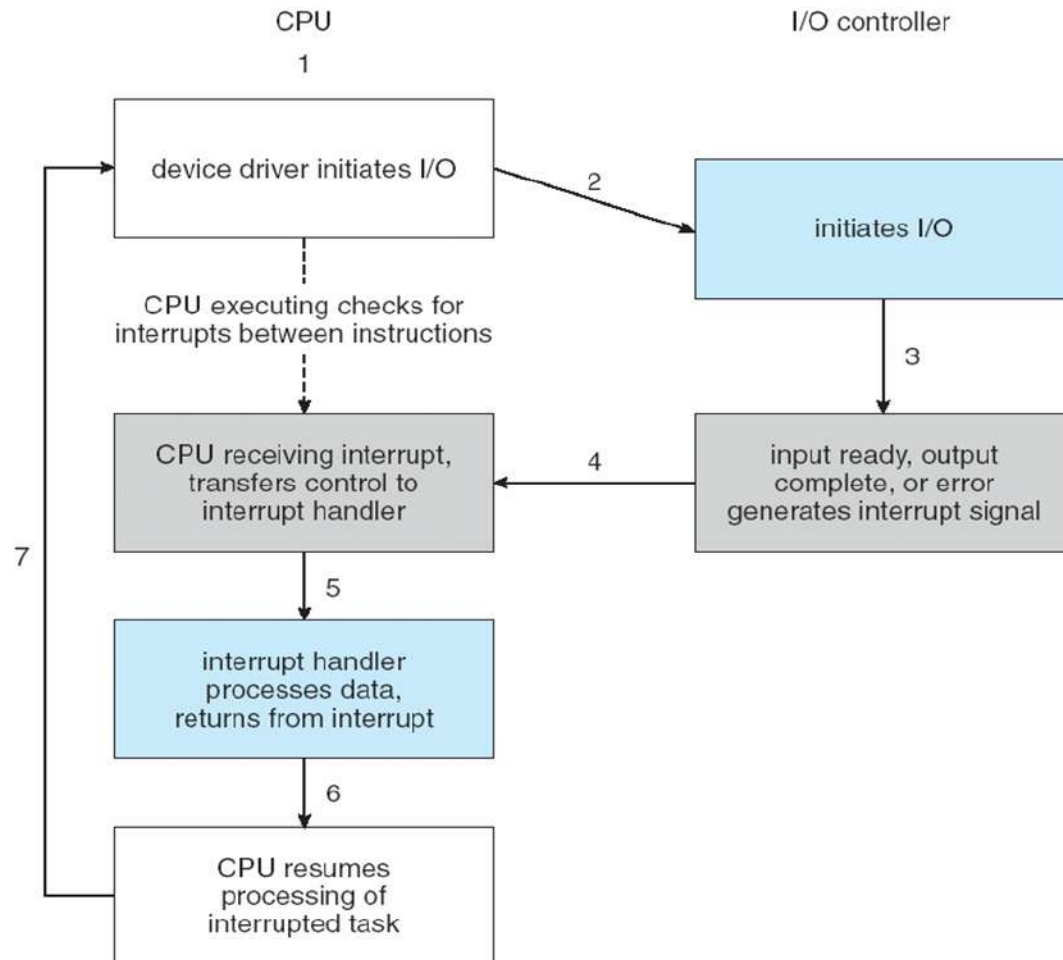
(b)

Figura 5.5 (a) Uma interrupção precisa. (b) Uma interrupção imprecisa.





Ciclo de I/O por **interrupção**






Interrupções (Cont.)

- O mecanismo de interrupções é também utilizado para **exceções**
 - Terminar processo, travamento de sistema devido a erro de hardware...
- Erro de página quando há erro de acesso a memória;
- CPU com vários núcleos pode processar interrupções de forma concorrente / paralela;





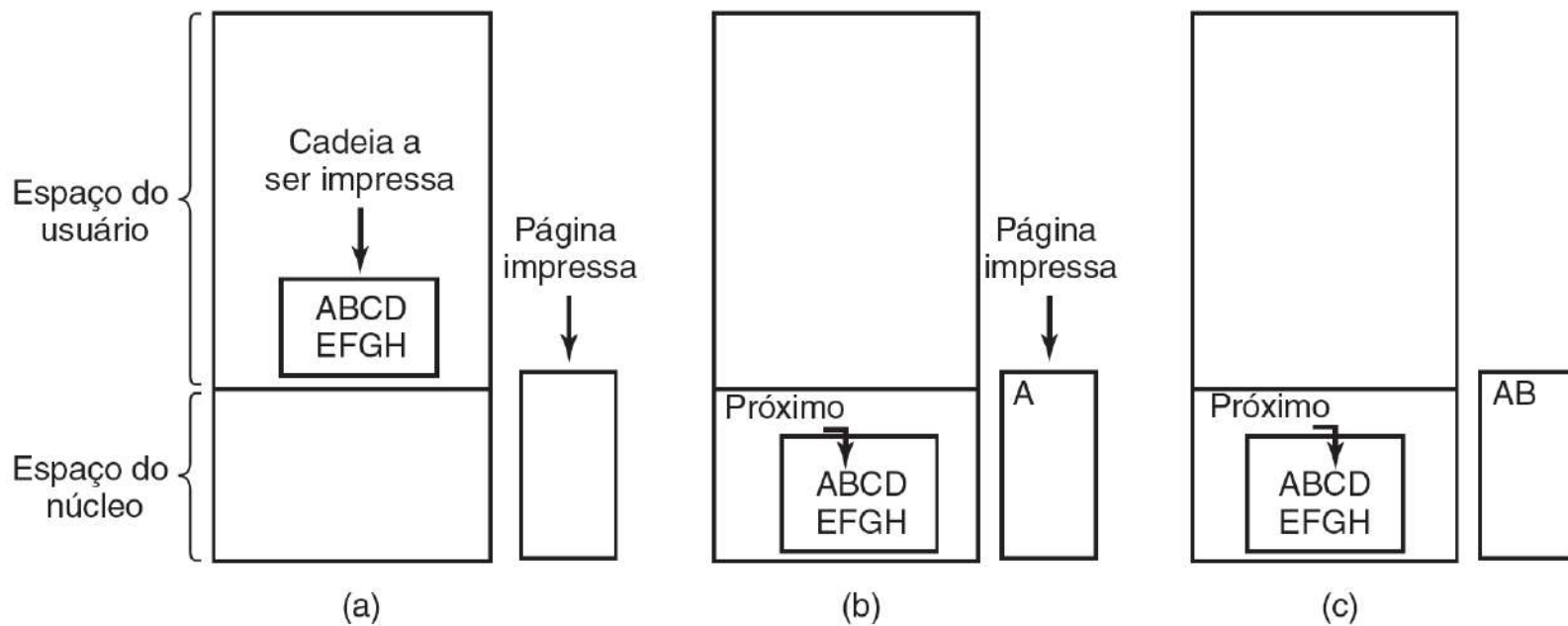
Tratadores de interrupção

- 
- Salva quaisquer registradores que não foram salvos pelo software de interrupção.
 - Estabelece um contexto para a rotina de tratamento da interrupção.
 - Estabelece uma pilha para a rotina de tratamento da interrupção.
 - Sinaliza o controlador de interrupção. Se não há um controlador de interrupção centralizado, reabilita a interrupção.
 - Copia os registradores de onde foram salvos para a tabela de processos.
 - Executa a rotina de tratamento da interrupção.
 - Escolhe o próximo processo a ser executado.
 - Estabelece o contexto da MMU para o próximo processo a ser executado.
 - Carrega os registradores do novo processo
 - Inicializa a execução do novo processo.





E/S programada



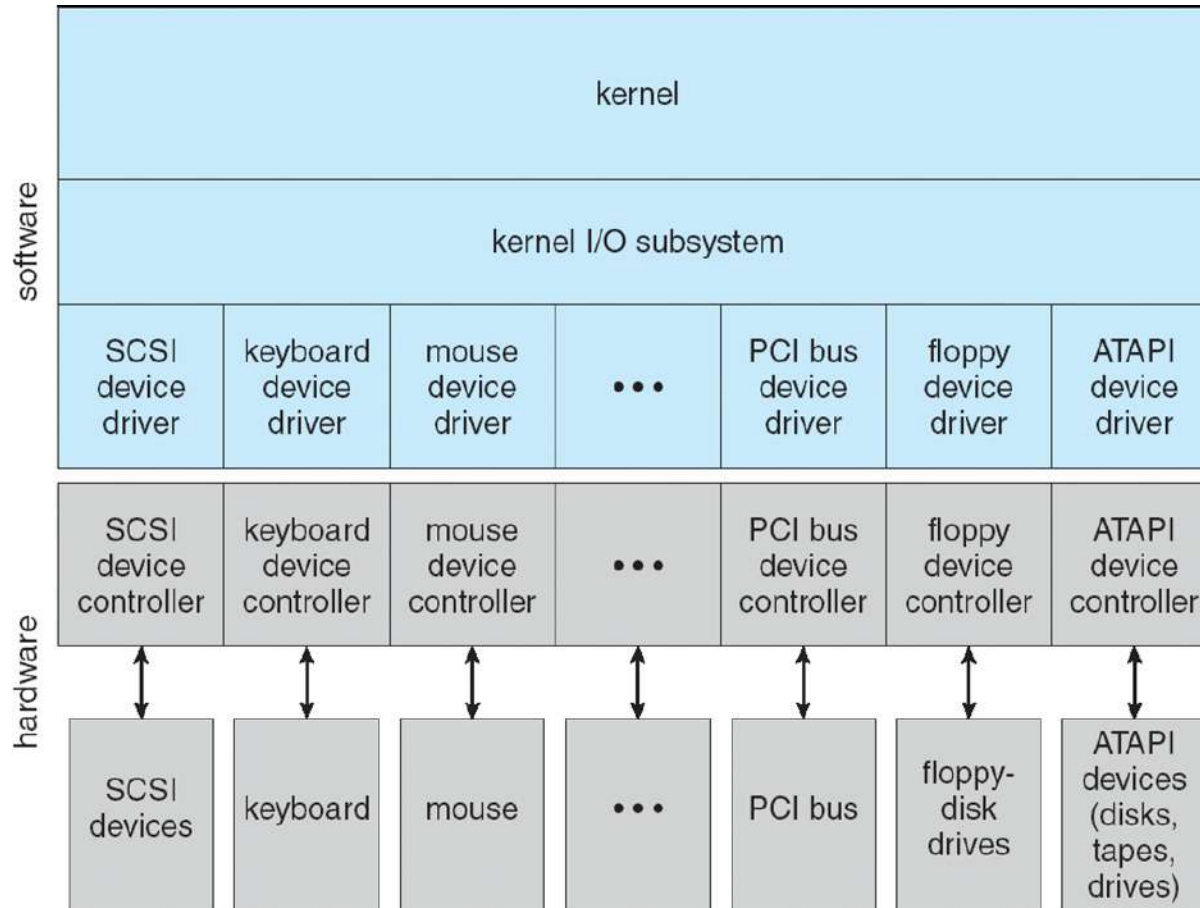
■ **Figura 5.6** Estágios da impressão de uma cadeia de caracteres.



**E/S é apenas
Hardware?**



A Kernel I/O Structure





Camadas de software de E/S

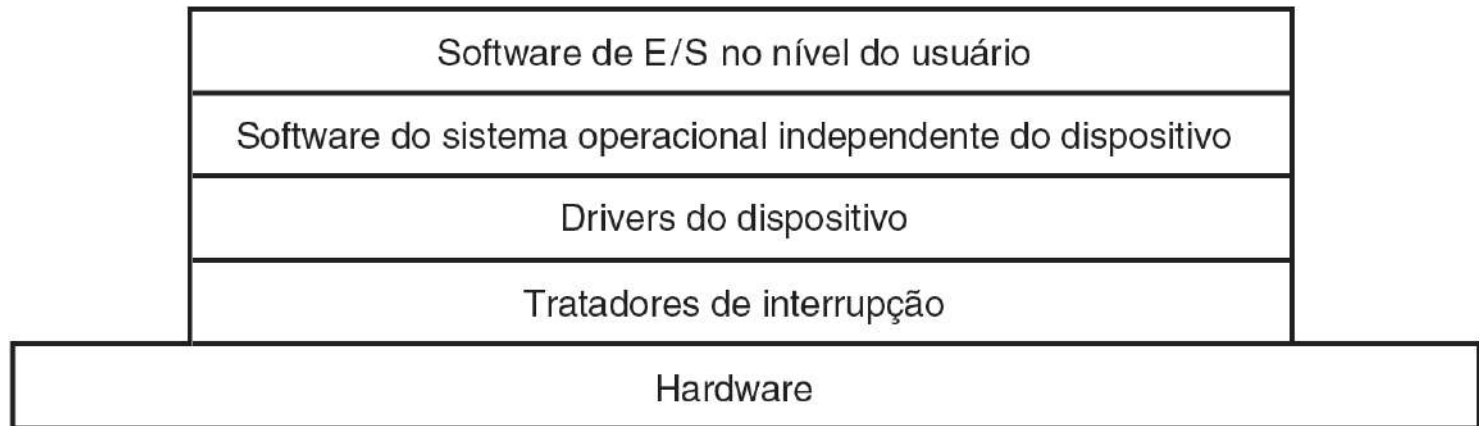


Figura 5.10 Camadas do software de E/S.





Drivers dos dispositivos

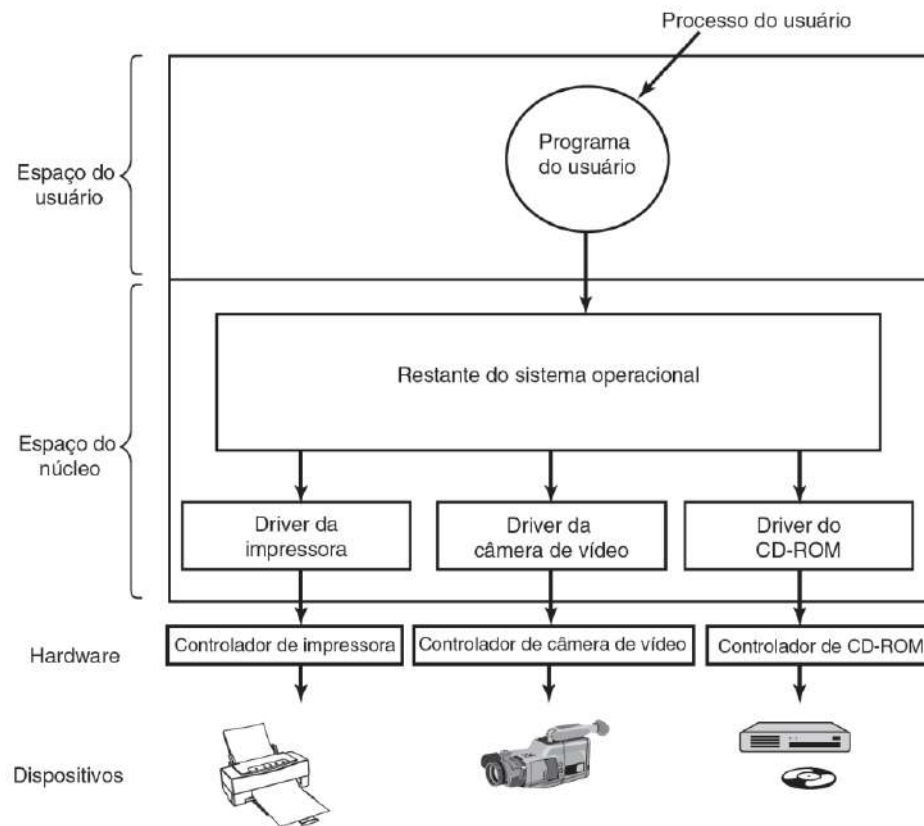


Figura 5.11 Posicionamento lógico dos drivers de dispositivos. Na verdade, toda comunicação entre os drivers e os controladores passa pelo barramento.





Funções do software de E/S independente de dispositivo

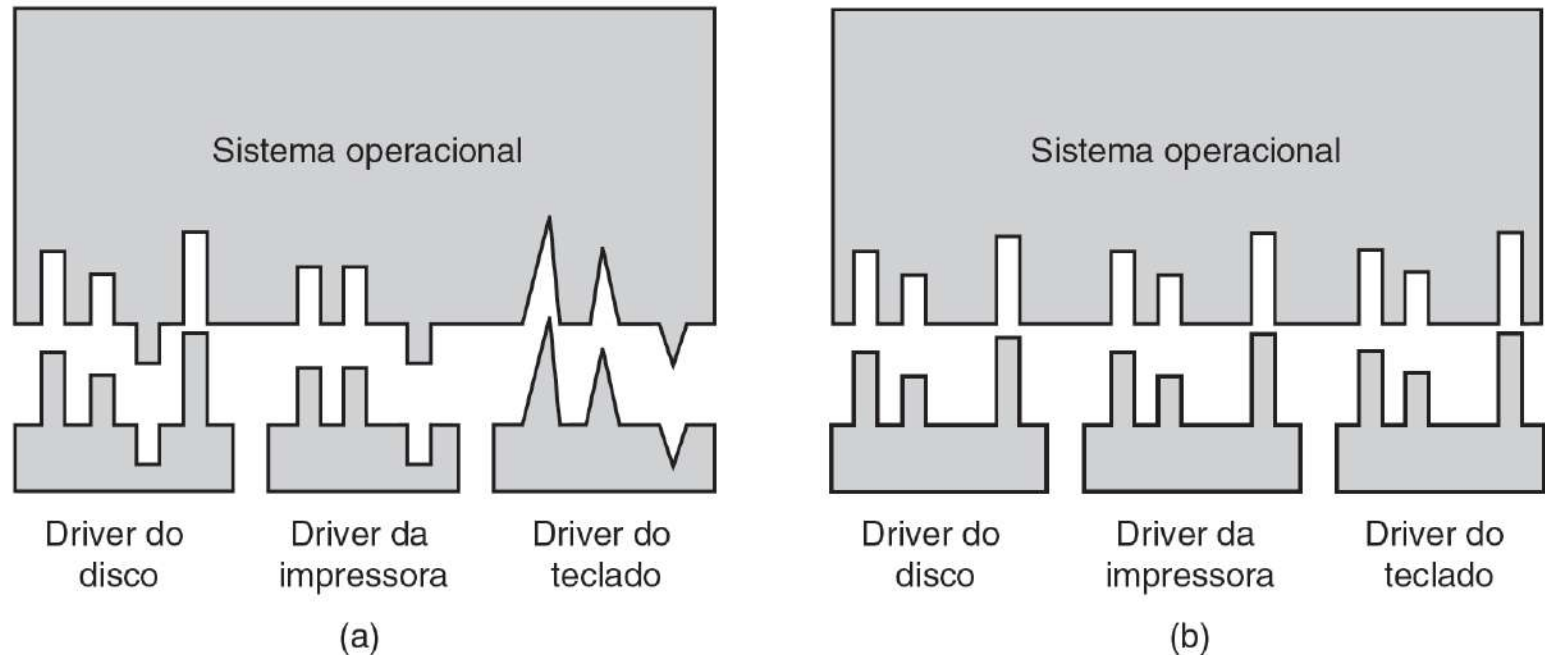
Uniformizar interfaces para os drivers de dispositivos
Armazenar no buffer
Reportar erros
Alocar e liberar dispositivos dedicados
Providenciar um tamanho de bloco independente de dispositivo

Tabela 5.2 Funções do software de E/S independente de dispositivo.





Interface uniforme para os drivers de dispositivo



■ **Figura 5.12** (a) Sem uma interface-padrão para o driver. (b) Com uma interface-padrão para o driver.





Uso de buffer

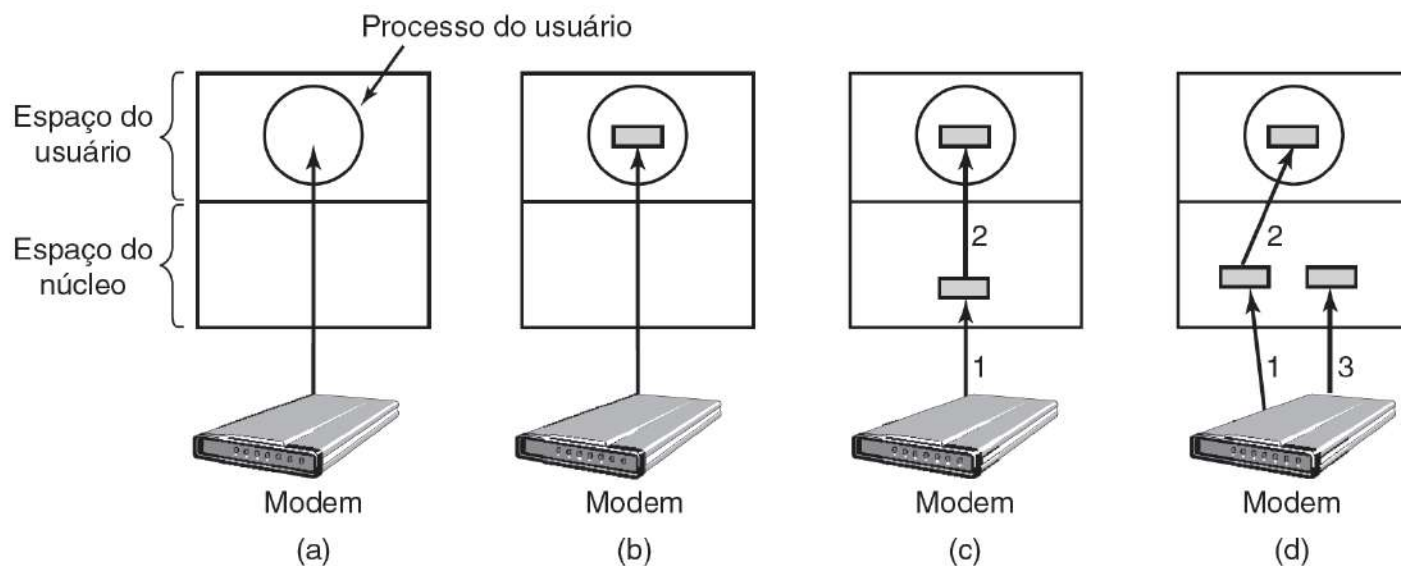
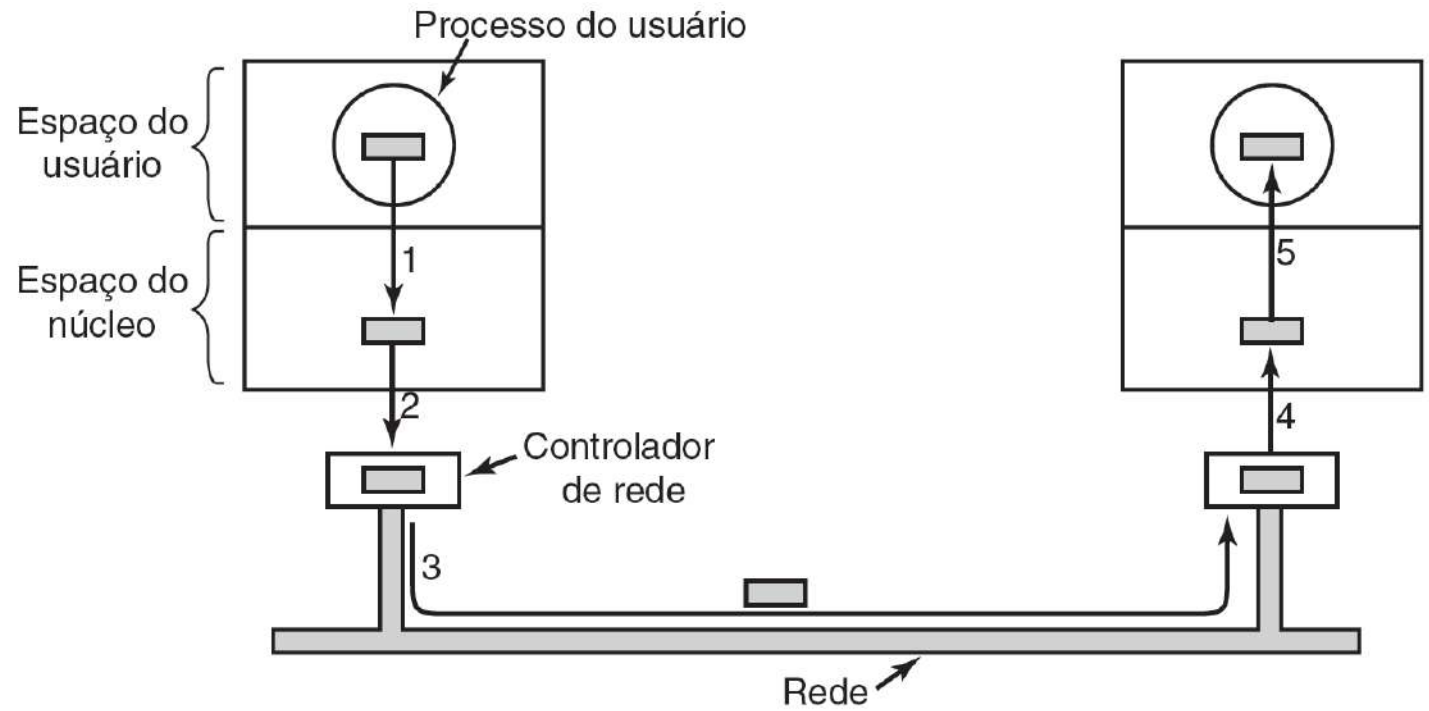


Figura 5.13 (a) Entrada não enviada para buffer. (b) Utilização de buffer no espaço do usuário. (c) Utilização de buffer no núcleo, seguido da cópia para o espaço do usuário. (d) Utilização de buffer duplicado no núcleo.





■ **Figura 5.14** O trânsito na rede pode envolver muitas cópias de um pacote.





Software E/S do espaço do usuário

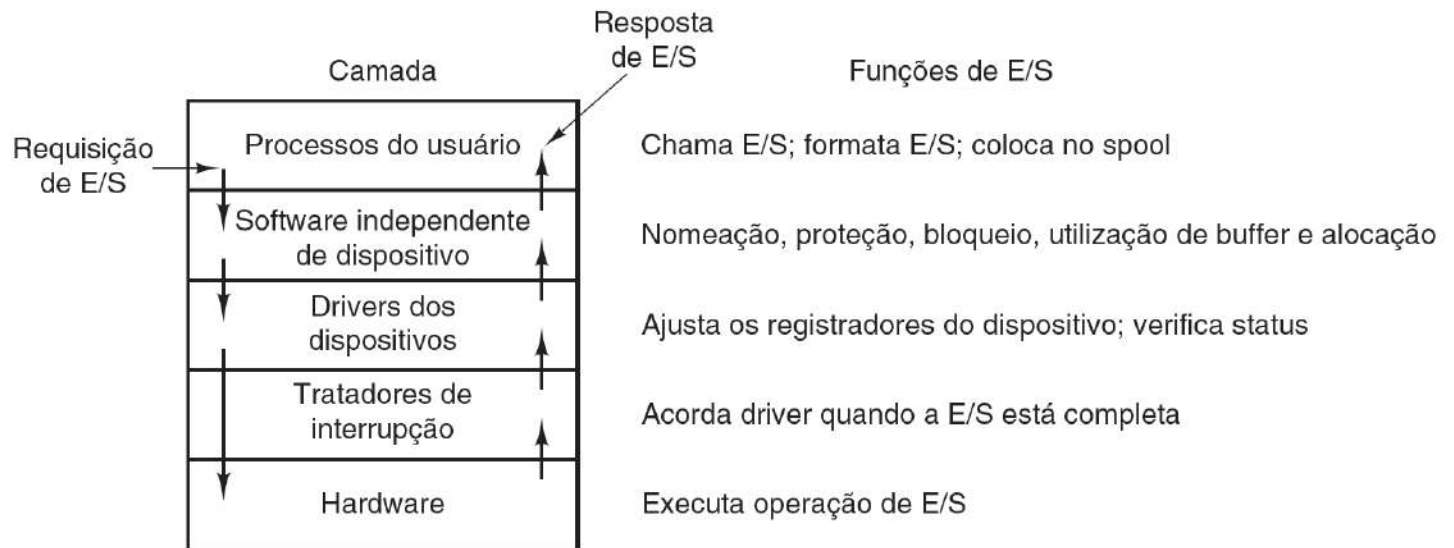


Figura 5.15 Camadas do sistema de E/S e as principais funções de cada camada.





Características de dispositivos de I/O

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





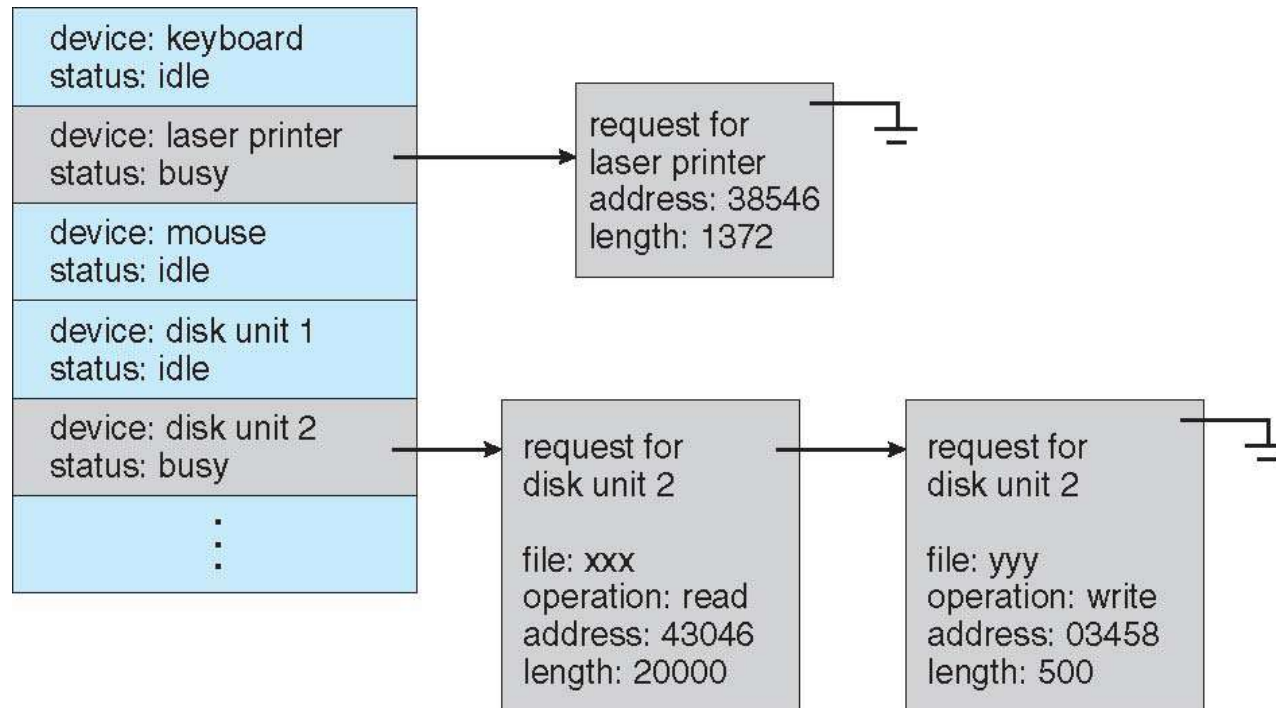
Subsistema de I/O do Kernel

- Scheduling/ Escalonamento
 - Ordenação de requisições (fila de dispositivos)
 - Implementação de QoS
- **Buffering** - armazena dados na memória enquanto a transferência entre dispositivos é realizada
 - Auxilia na diferença de velocidades de transferência dos dispositivos;





Device-status Table





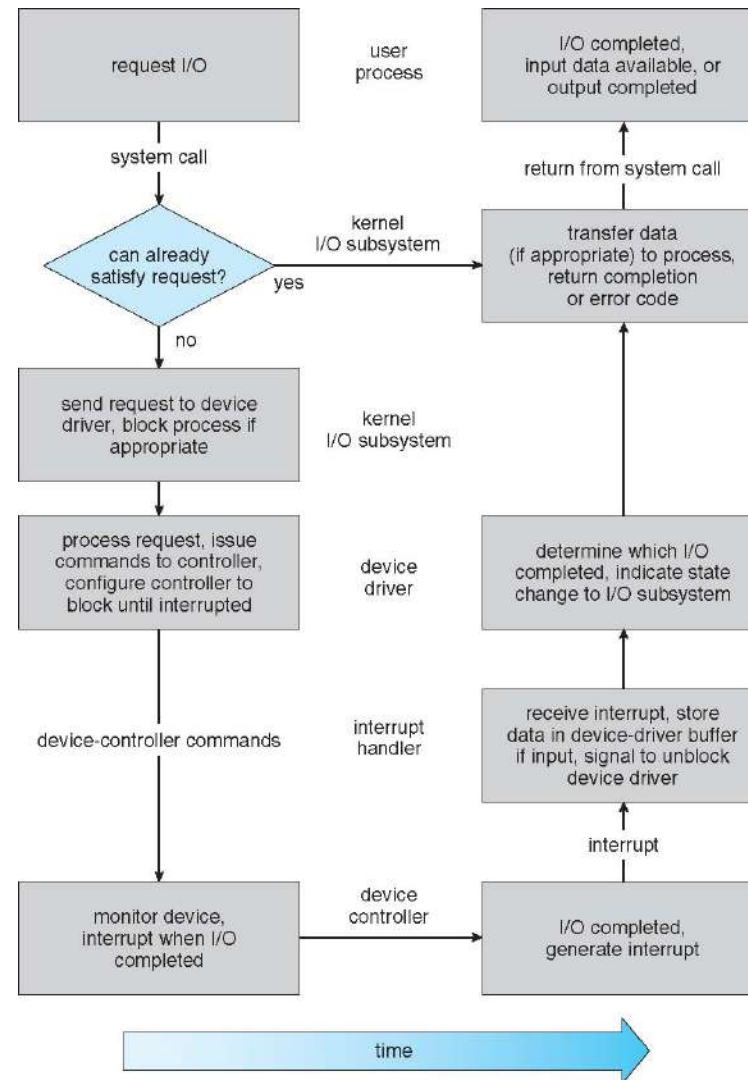
Requisições de I/O para operações de Hardware

- Considere a leitura de um arquivo em disco para um processo, é necessário:
 - Determinar o dispositivo que detém o arquivo;
 - “Traduzir” o nome para a representação utilizada pelo dispositivo;
 - Fisicamente, realizar leitura dos dados do disco para;
 - Disponibilizar os dados ao processo que requereu;
 - Retornar o controle ao processo.





Ciclo de vida de uma requisição de I/O





Relação de I/O com a Performance

- Demanda CPU para executar driver do dispositivo e código de I/O do kernel;
- Troca de contextos devido a interrupções;
- Cópia de dados;
- Tráfego de rede em alta demanda;



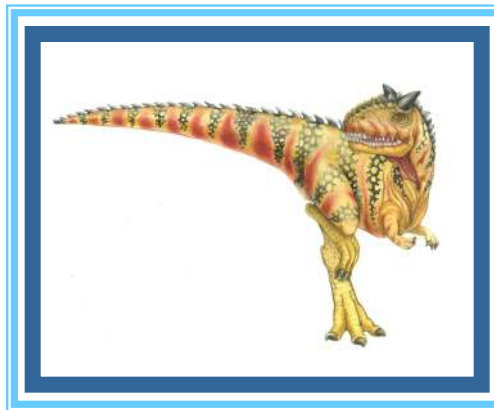


Melhoria na performance

- Redução do número de mudanças de contexto;
- Redução de cópia de dados;
- Redução de interrupção por meio de transferências maiores, controladoras inteligentes e *polling*
- Uso do DMA
- Balancear CPU, memória, barramento e performance de I/O para maior throughput
- “Transformar” processos de usuário para threads de kernel

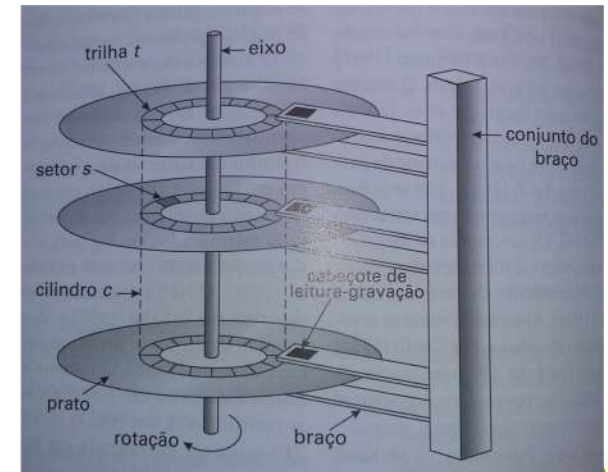
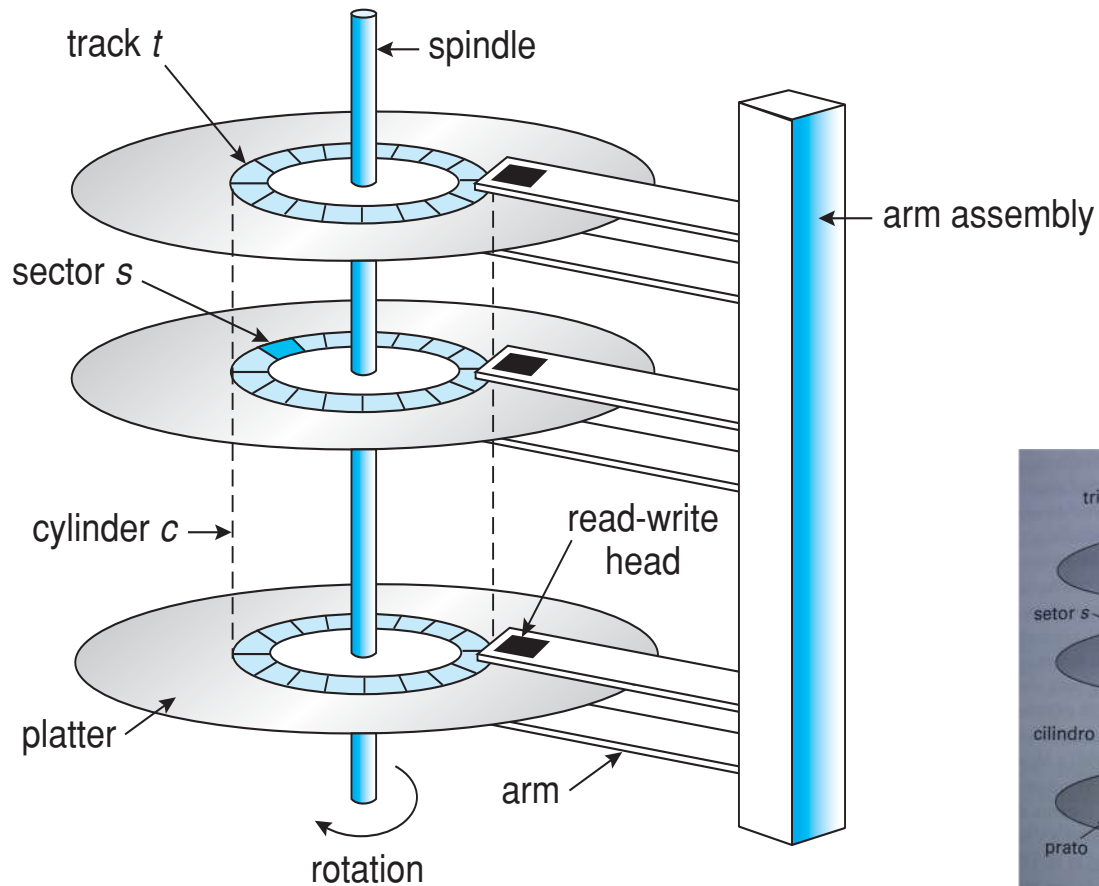


Dispositivos de armazenamento em massa





Disco





Overview

■ Discos magnéticos

- **Transfer rate** – taxa de transferência entre o dispositivos e o computador
- **Positioning time** (**random-access time**) é o tempo gasto para mover o braço do disco até o cilindro desejado (**seek time**) e o tempo para o setor desejado rodar/chegar até a cabeça de leitura (**rotational latency**)

■ Drives conectados ao computador por meio do **I/O bus**

- Incluindo **EIDE**, **ATA**, **SATA**, **USB**, **Fiber Channel**, **SCSI**, **Firewire**





Hard Disk Performance

- **Access Latency** = **Average access time** = average seek time + average latency
- Tempo médio para realização de I/O = average access time + (dados a serem transferidos/tx. de transferência) + atraso da controladora
- Minimizar tempo de busca (Seek time \approx seek distance)





Disk Scheduling

- Controladoras de dispositivos possuem *buffers* pequenos e estão aptas a gerenciarem uma fila de requisição de I/O
- Há vários algoritmos para gerenciar e escalonar requisições de I/O
- As análises dos algoritmos a seguir consideram requisições que podem variar de (0-199)

98, 183, 37, 122, 14, 124, 65, 67

Head pointer 53





Disk scheduling

- FIFO
 - Atendimento a requisições de acordo com ordem de chegada
- SSTF (starvation)
 - Requisições mais próximas a “cabeça de leitura” são atendidas primeiro, as mais distantes podem não ser atendidas (*starvation*)
- Elevator (SCAN)
 - O movimento da “cabeça de leitura” não depende de requisições. Ele busca requisições nas duas direções (fora para dentro -> dentro para fora) e a medida que são encontradas, as requisições são atendidas.
- Improved Elevator 1 (Circular SCAN)
 - O movimento da “cabeça de leitura” não depende de requisições. Ele busca requisições em uma única direção, de fora para dentro do cilindro (reinicia ciclo de fora para dentro). A medida que são encontradas, as requisições são atendidas.

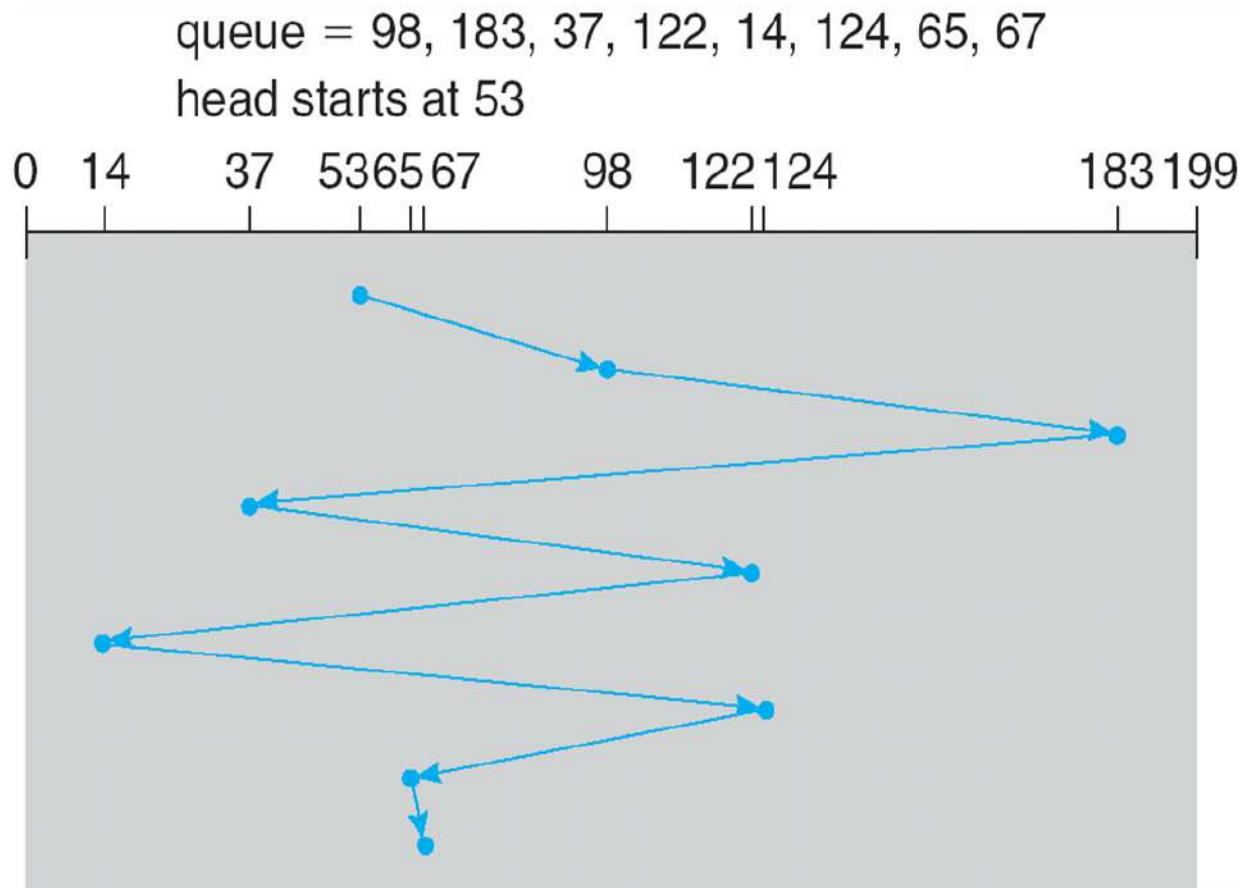




FCFS

Atendimento a requisições de acordo com ordem de chegada

Illustration shows total head movement of 640 cylinders

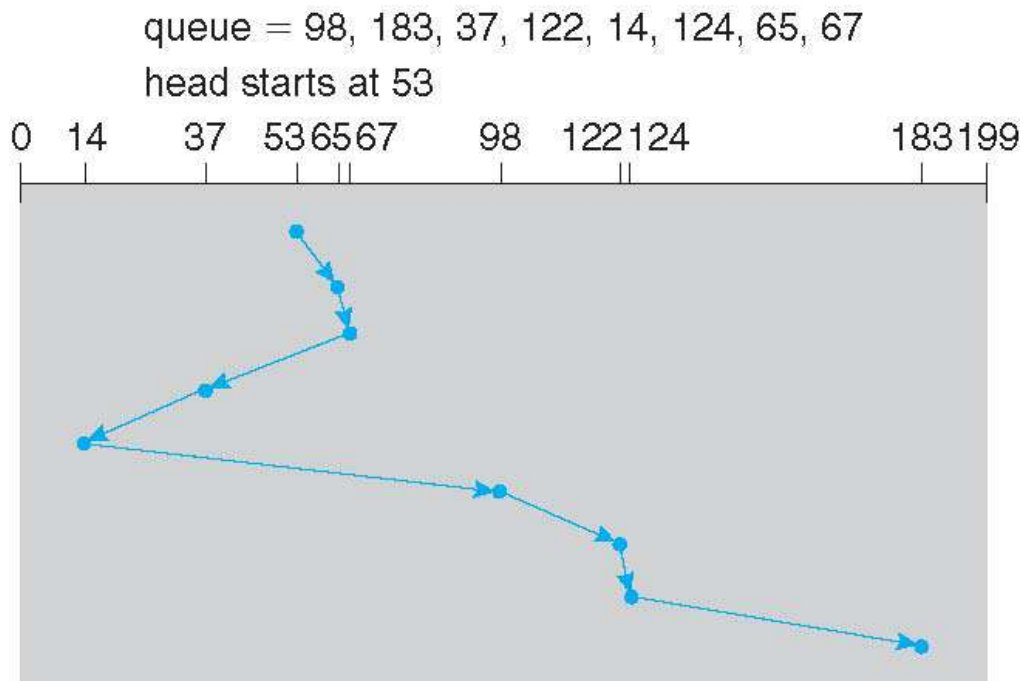




SSTF

Requisições mais próximas a “cabeça de leitura” são atendidas primeiro, as mais distantes podem não ser atendidas (*starvation*)

- Shortest Seek Time First seleciona a requisição que possui menor tempo de busca em relação a posição corrente.
- Similar ao SJF -> *starvation*
- Abaixo, o movimento total da cabeça de leitura por 236 cylinders





SCAN

O movimento da “cabeça de leitura” não depende de requisições. Ele busca requisições nas duas direções (fora para dentro -> dentro para fora) e a medida que são encontradas, as requisições são atendidas.

- **SCAN também conhecido por algoritmo Elevador**
- A ilustração mostra a movimentação por 208 cilindros

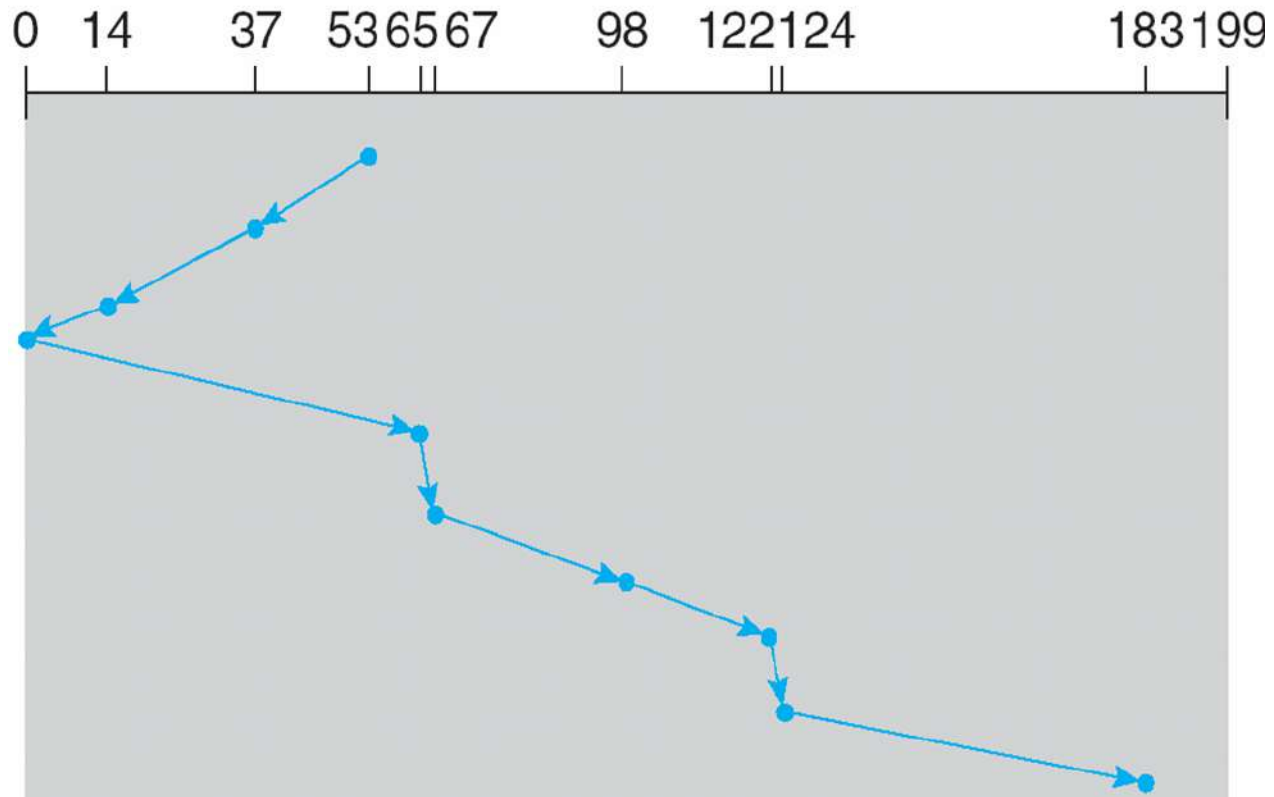




SCAN (Cont.)

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53





C-SCAN

O movimento da “cabeça de leitura” não depende de requisições. Ele busca requisições em uma única direção, de fora para dentro do cilindro (reinicia ciclo de fora para dentro). A medida que são encontradas, as requisições são atendidas.

- Provê um tempo mais uniforme que SCAN
- Trata os cilindros como uma lista circular

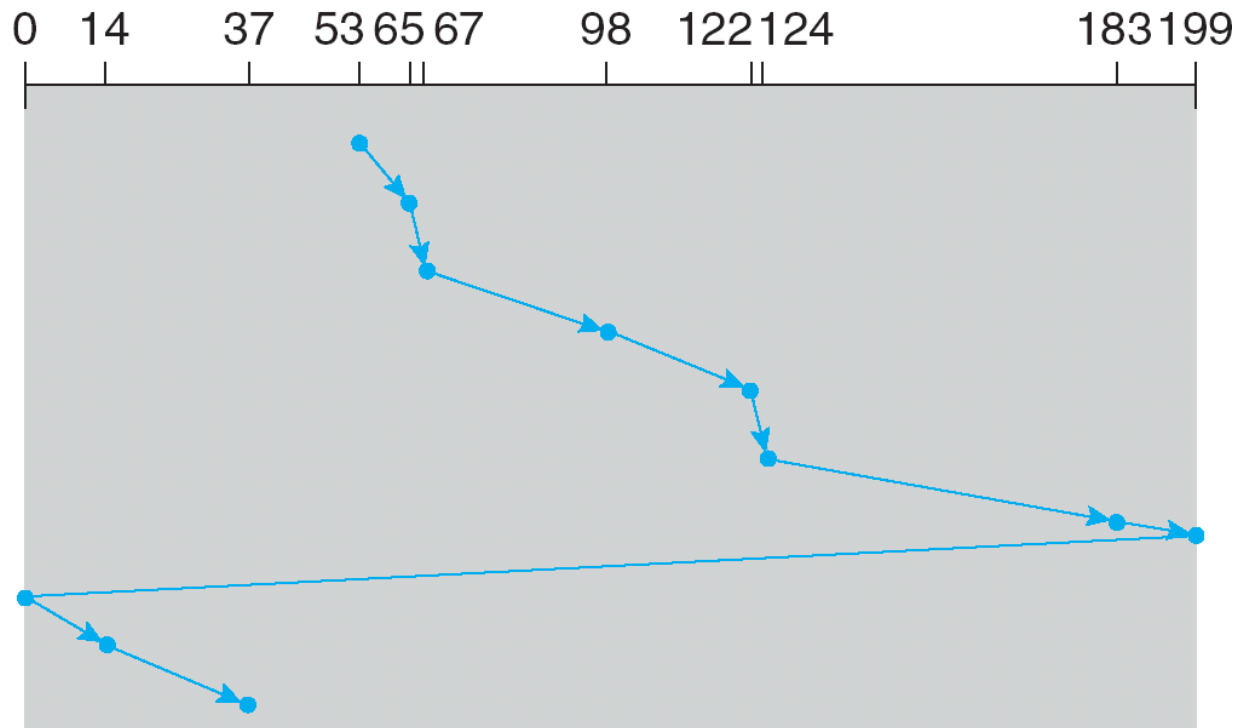




C-SCAN (Cont.)

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53





Selecionando um algoritmo

- SSTF é mais comum e uma abordagem mais prática;
- SCAN e C-SCAN tem melhor desempenho para sistemas com alta carga no disco (menor *starvation*)
- A performance dependerá do número e do tipo de requisições





Gerenciamento de Disco

- Para utilizar um disco para armazenar arquivos, o S.O necessita ainda armazenar sua estrutura de dados no disco
 - Partição (disco lógico)
 - Formatação lógica

