Sistemas Operacionais

Entrada e Saída

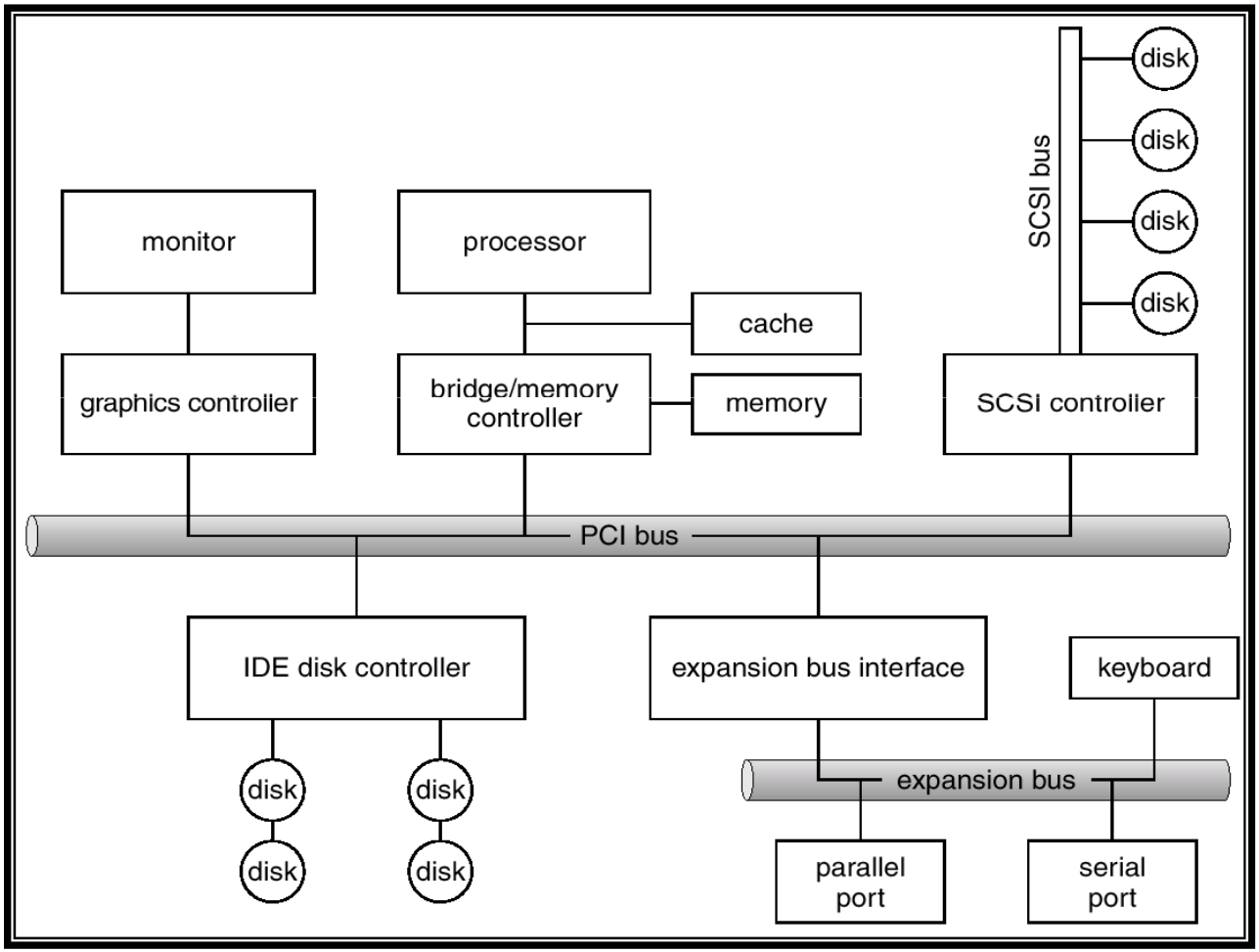
Edeyson Andrade Gomes

www.edeyson.com.br

Roteiro da Aula

- Entrada e Saída
  - Princípios
  - Classificação
  - Controladores
  - DMA
  - Software de E/S
  - Drivers

Estrutura Típica de Barramento



Portas de E/S

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)

Entrada e Saída

- Controle de dispositivos de entrada e saída
  - Sistema Operacional
    - Emitir comandos para os dispositivos, capturar interrupções e manipular erros;
    - Oferecer uma interface entre os dispositivos e o resto do sistema
      - Simples, fácil de usar e homogênea.
  - Parte significativa do código do SO.

Entrada e Saída

- Princípios
  - Visão de engenheiros eletrônicos
    - Chips, fios e componentes físicos.
  - Visão dos programadores
    - Interface com o software
      - Comandos que o hardware aceita;
      - Funções que tais comandos executam;
      - Erros que devem ser reportados nas diversas situações.

Entrada e Saída

- Classificação
  - Dispositivo de bloco
    - Armazena informações em blocos de tamanho fixo, cada um deles com seu próprio endereço.
    - Pode ler ou escreve blocos de forma independente
      - Exemplo: discos.
  - Dispositivo de caractere
    - Libera ou aceita um conjunto de caracteres, sem respeitar nenhuma estrutura de bloco.
    - Não é endereçável.
    - Exemplos: terminais, impressoras de linha, interfaces de rede e mouses.

Entrada e Saída

- Classificação
  - Problema:
    - Alguns dispositivos não se encaixam em nenhum dos dois grupos.
    - Os clocks, por exemplo, não são endereçáveis por blocos nem geram ou aceitam um fluxo de caracteres. Tudo o que eles fazem é gerar interrupções a intervalos de tempo muito bem definidos.

Classificação

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 readÐwrite	CD-ROM graphics controller disk

Controladores de Dispositivo

- As unidades de entrada/saída tipicamente são constituídas de duas partes distintas:
  - Uma parte mecânica
    - Dispositivo propriamente dito.
  - Uma parte eletrônica
    - Controlador de dispositivo ou adaptador.

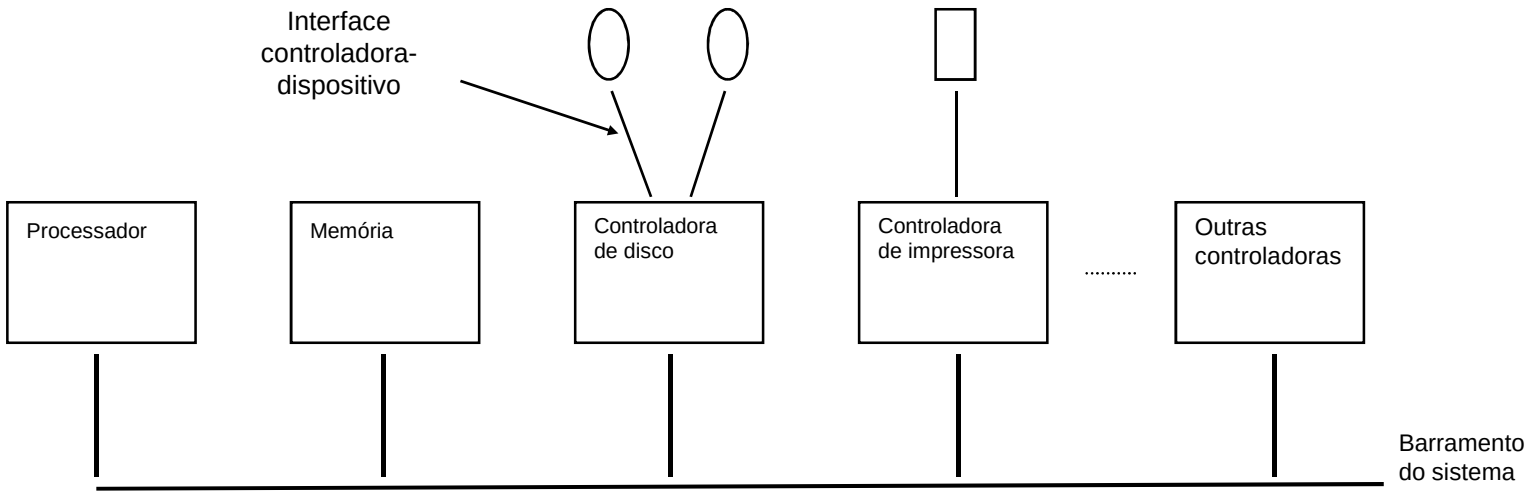
Controladores de Dispositivo

- A placa controladora tem geralmente um conector no qual deve ser ligado um cabo vindo do dispositivo. Muitas controladoras podem tratar de dois, quatro e até oito dispositivos idênticos.
- A ligação entre o dispositivo e placa controladora é feita através da interface controladora-dispositivo, que é uma interface de baixo nível.

Controladores de Dispositivo

- Dispositivo X Controladora
  - SO manipula a controladora, não o dispositivo.
  - A comunicação entre o processador e as controladoras é feita basicamente de duas formas:
    - Barramento único.
      - PC
    - Vários barramentos e processadores que realizam somente operações com entrada/saída - canais de entrada/saída - processadores esses que absorvem parte do trabalho do processador central.
      - Mainframe

Barramento Único



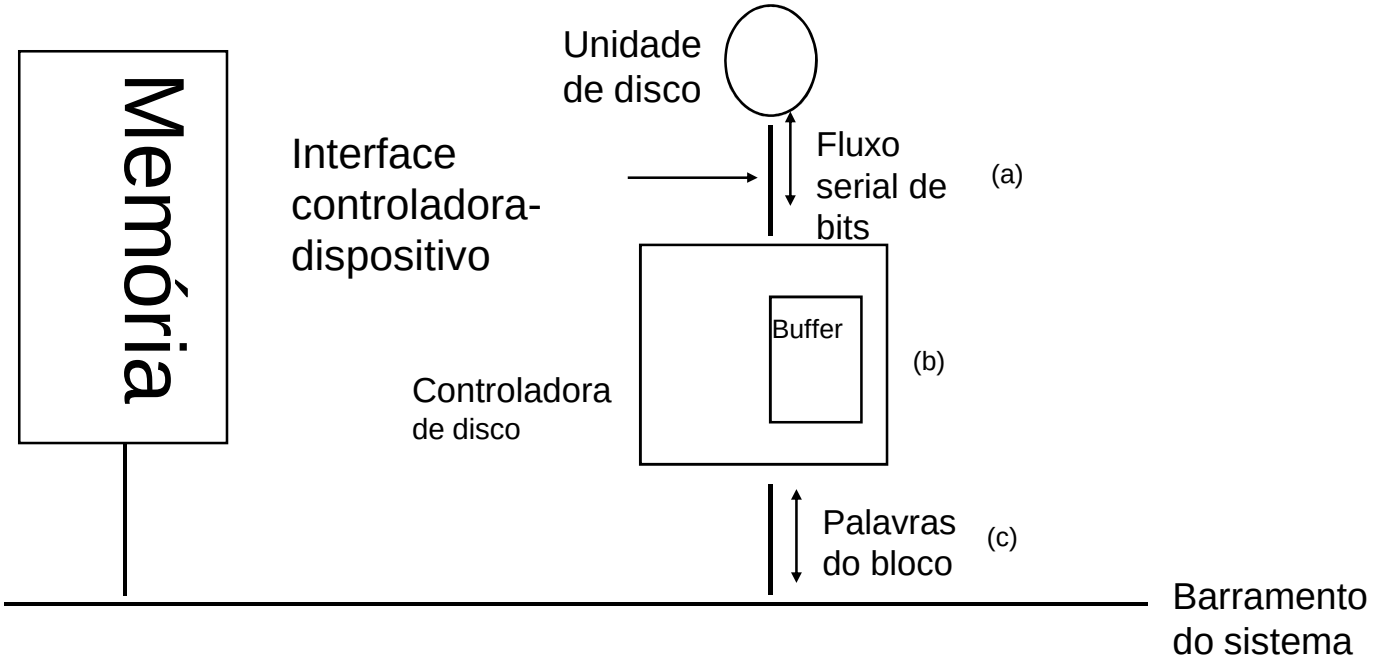
Controladora

- O que transita entre a unidade de disco e a controladora são fluxos seriais de bits. O trabalho da controladora é converter o fluxo serial de dados em blocos de bytes, realizando as correções de erros que se fizerem necessárias. O procedimento ocorre da seguinte maneira:

Controladora

- O fluxo serial de bits chega da unidade de disco, através da interface controladora-dispositivo. Em outras palavras, a controladora lê o bloco do *drive* (dispositivo) serialmente, um bit após o outro;
- O bloco de bytes é montado, bit a bit, em um buffer dentro da controladora;
- Calcula-se o *checksum* para verificar a ocorrência de erros;
- Se o bloco for declarado correto, ele poderá ser copiado na memória.

Controladora de Disco



Controladora de Disco

- Um disco pode ser formatado, por exemplo, com oito setores de 512 bytes por trilha (bloco de 4KB). Assim, a leitura de um bloco é composta por:
- um **cabeçalho**, o qual é composto de:
  - o número do cilindro
  - o número do setor
  - o tamanho do setor
  - outros dados.
- os 4.096 bits (8 x 512 bytes = 4.096 bits)
- um checksum ou código de correção de erros (ECC)

SO X Controladora

- Cada controladora precisa de uns poucos registradores que são usados na comunicação com o processador.
- O SO efetua o I/O escrevendo nos registradores das controladoras.
  - A controladora da unidade de disquete do PC, por exemplo, aceita 15 comandos diferentes, como READ, WRITE e SEEK. Muitos desses comandos têm parâmetros que também precisam ser carregados nos registradores da controladora.

SO X Controladora

- Os registradores das controladoras podem ser, genericamente, de dois tipos:
  - Registradores localizados na própria controladora.
  - Registradores localizados no espaço de endereçamento da memória. Tal esquema é denominado **entrada/saída mapeada na memória**.

SO X Controladora

- Interação SO x Controladora:
  - Quando um comando é aceito pela controladora, o processador deixa a mesma trabalhando sozinha (processamento assíncrono) e vai realizar outra atividade.
  - Quando a execução termina, a controladora **gera uma interrupção**, de forma a permitir a restauração do SO.

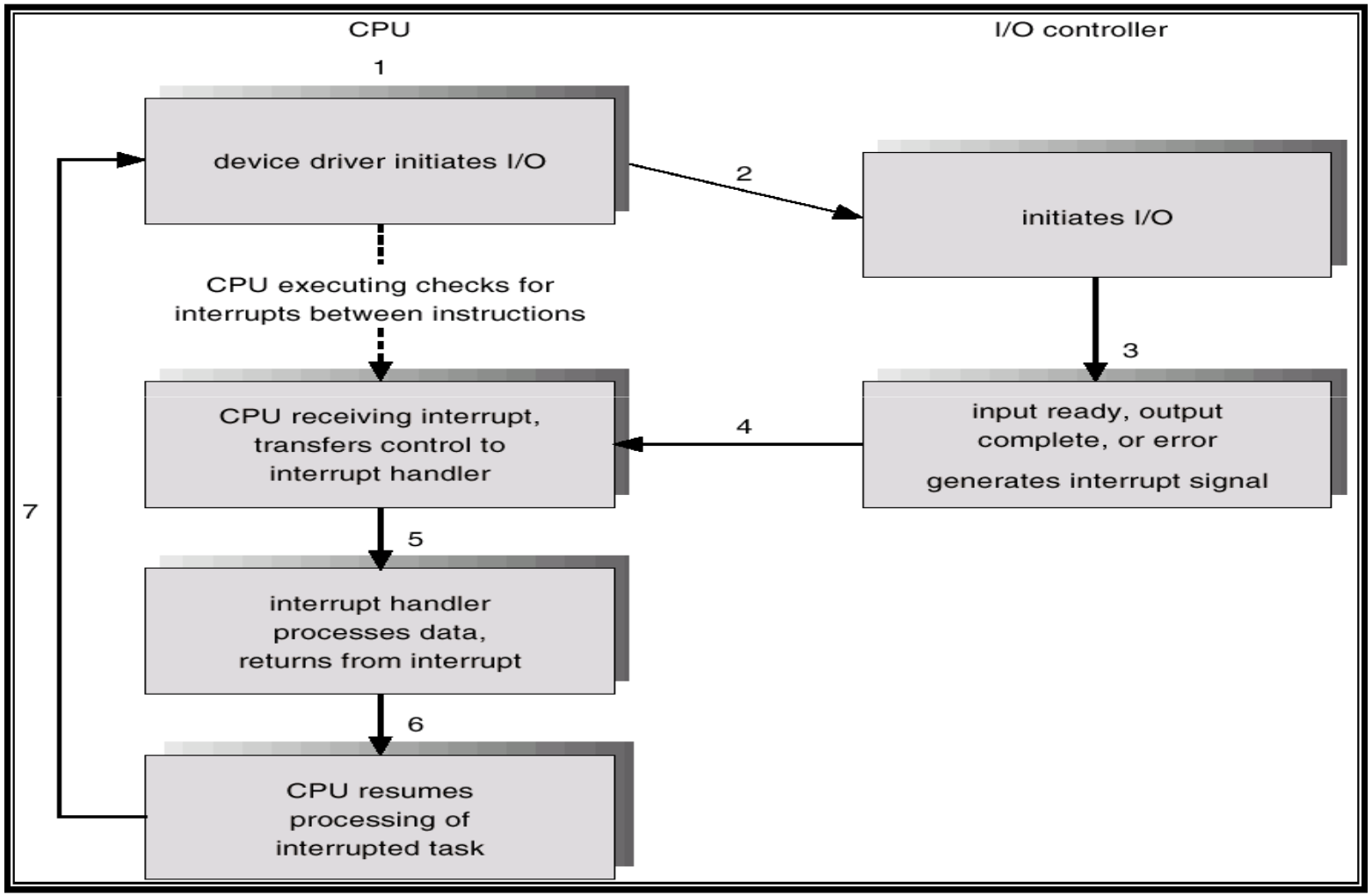
SO X Controladora

- Interação SO x Controladora:
  - O SO obtém o resultado da operação e pode ler o bloco do disco escrito no buffer da controladora, um byte ou uma palavra por vez.

SO X Controladora

- Interação SO x Controladora:
  - Isso é feito executando-se um loop, com cada interação lendo um byte ou uma palavra a partir do registrador da controladora e armazenando tal informação na memória.
  - Perceba que a cada loop, um byte ou uma palavra é copiado do buffer para o registrador, de onde o dado é realmente lido pelo sistema operacional.

Ciclo de Interrupções



Vetor de Interrupções (Intel)

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)

- Muitas controladoras, especialmente aquelas desenvolvidas para dispositivos de blocos, suportam operações de **acesso direto à memória**.
  - Loop para cópia de conteúdo
    - Overhead

Acesso Direto à Memória (DMA)

- A DMA foi criada justamente para tirar do processador esse trabalho de baixo nível. Quando ela é usada, o processador fornece à controladora dois itens de informação, além do endereço do bloco do disco:
  - O endereço real de memória para onde o bloco deve ir, que é copiado no registrador de memória da controladora;
  - O número de bytes a serem transferidos, que é copiado no registrador contador.

Acesso Direto à Memória (DMA)

- Após a controladora ter lido todo o bloco do dispositivo, tê-lo colocado em seu buffer e ter verificado seu *checksum*, com a DMA, ela passa a agir da seguinte forma:
  - Copia o primeiro byte ou palavra para o endereço especificado no registrador de memória da controladora DMA;
  - Incrementa o valor de tal endereço;

Acesso Direto à Memória (DMA)

- Subtrai o número de bytes (ou palavras) que acabou de ser transferido do valor do contador de bytes (ou palavras).
- Este processo continua até que o contador chegue a zero, momento em que a controladora gera uma interrupção. Este processo é chamado de **bufferização de dois passos**. Quando o sistema operacional começa sua execução para tratar da interrupção, ele não vai mais precisar ler o bloco para a memória.

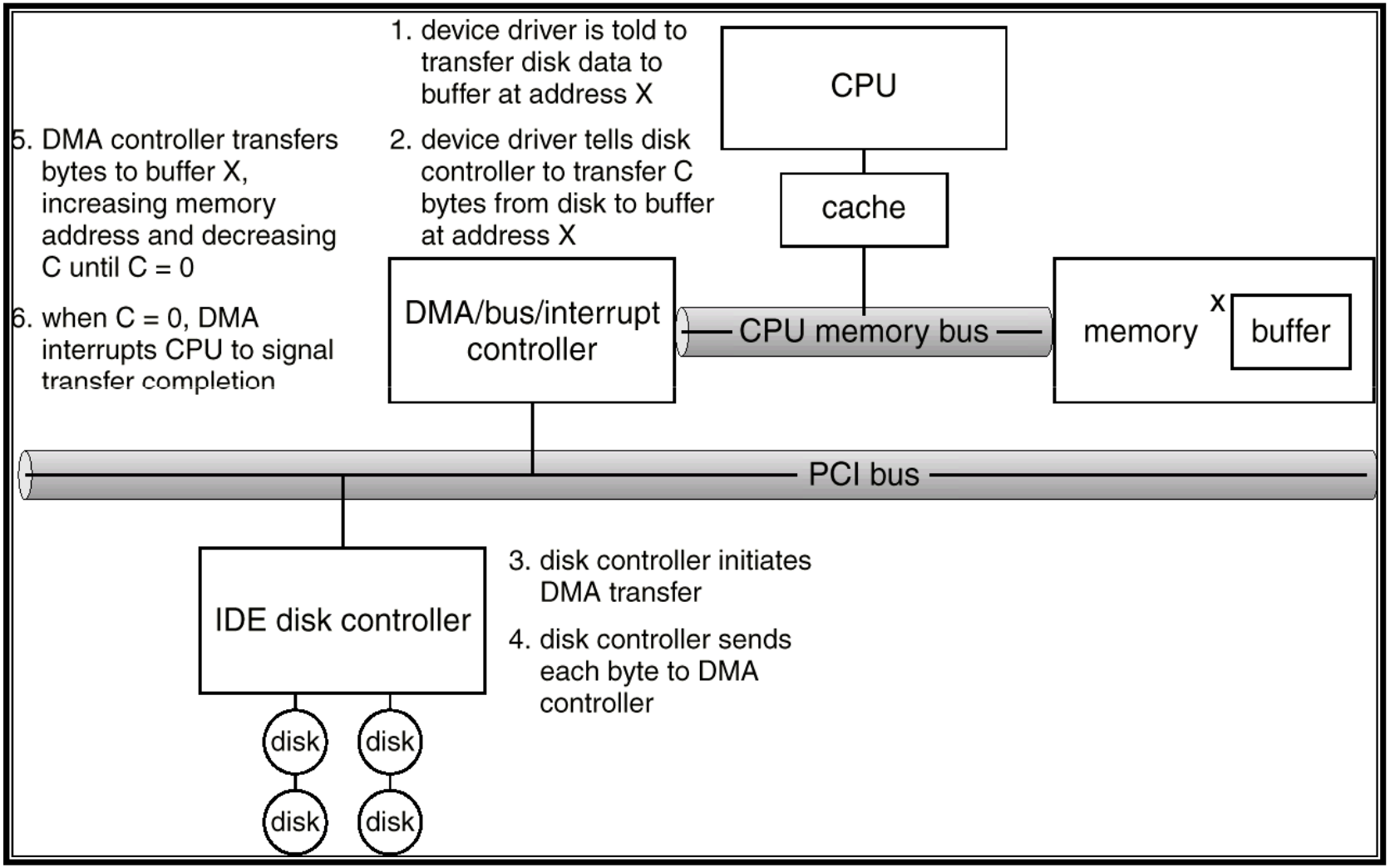
Acesso Direto à Memória (DMA)

- Concorrência ao Barramento
  - Quando o disco começa sua transferência, os bits vão chegando do disco a uma taxa constante, estando a controladora pronta ou não para recebê-los.
  - Escrita direta na memória
    - Acesso ao barramento a cada palavra transferida.
    - Concorrência x Espera

Acesso Direto à Memória (DMA)

- Concorrência ao Barramento
  - Espera para Gravar na Memória
    - Como ler novos dados do disco?
    - Fluxo contínuo
  - Quando o buffer é usado, o barramento não é acessado até que o bloco esteja todo na controladora.
    - Depois disso, ele pode ser transferido numa operação que não é crítica em relação a tempo como seria no caso de usar o barramento diretamente.

Acesso Direto à Memória (DMA)



Software de E/S

- O software que trata da entrada/saída é organizado em vários níveis, da seguinte forma:
  - O nível mais baixo é voltado a esconder do usuário as peculiaridades do hardware;
  - Os demais níveis são responsáveis por apresentar ao usuário uma interface boa e simples de usar.

Software de E/S

- Objetivos:
  - **Independência dos dispositivos**
    - HD x CD x Disquete
    - Arquivos e diretórios
    - SO:
      - Gerência através de *Drivers*
  - **Uniformidade da identificação**
    - Nomes de um arquivo ou de dispositivos
      - Strings ou Inteiro
    - Independente do dispositivo.

Software de E/S

- Objetivos:
  - **Manipulação de erros**
    - Erros tratados o mais próximo possível do hardware. Os níveis mais altos de software só devem tomar conhecimento de um erro se os mais baixos não puderem resolvê-lo.

Software de E/S

- **Tipo de transferência**
  - *Assíncrona* - **dirigida por interrupções**.
  - *Síncrona* / Bloqueante
  - Complexidade de Código
  - Escalabilidade

Software de E/S

- **Compartilhamento**
  - *Compartilháveis*
    - Alguns dispositivos de entrada/saída, como os discos, podem ser usados por vários usuários ao mesmo tempo. Não há nenhum problema pelo fato de vários usuários terem arquivos abertos num determinado disco ao mesmo tempo.

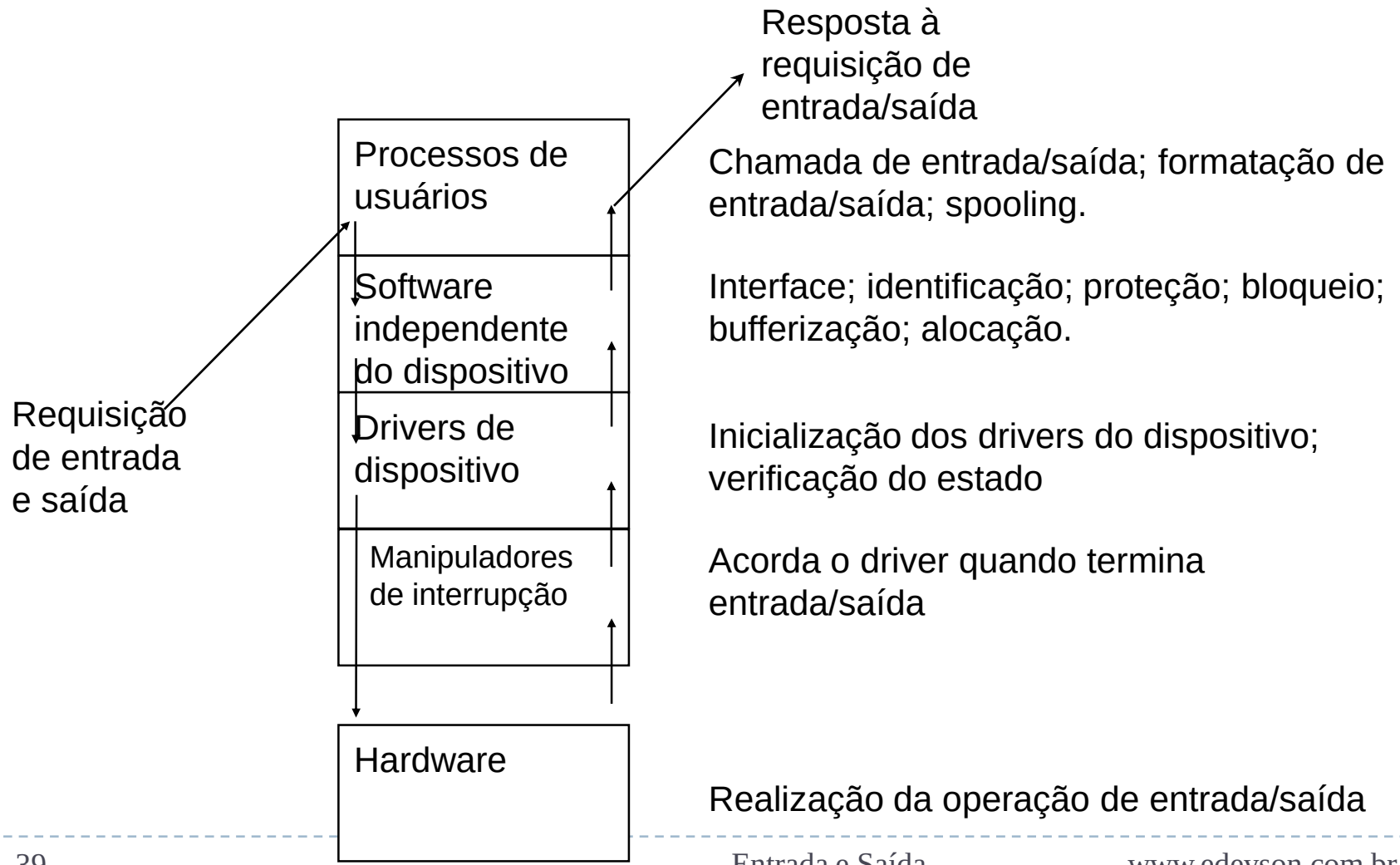
Software de E/S

- **Compartilhamento**
  - *Dedicados*
    - Alguns dispositivos, tais como impressoras, devem ser dedicados a um único usuário, até que este conclua seu trabalho.
    - Exclusão Mútua

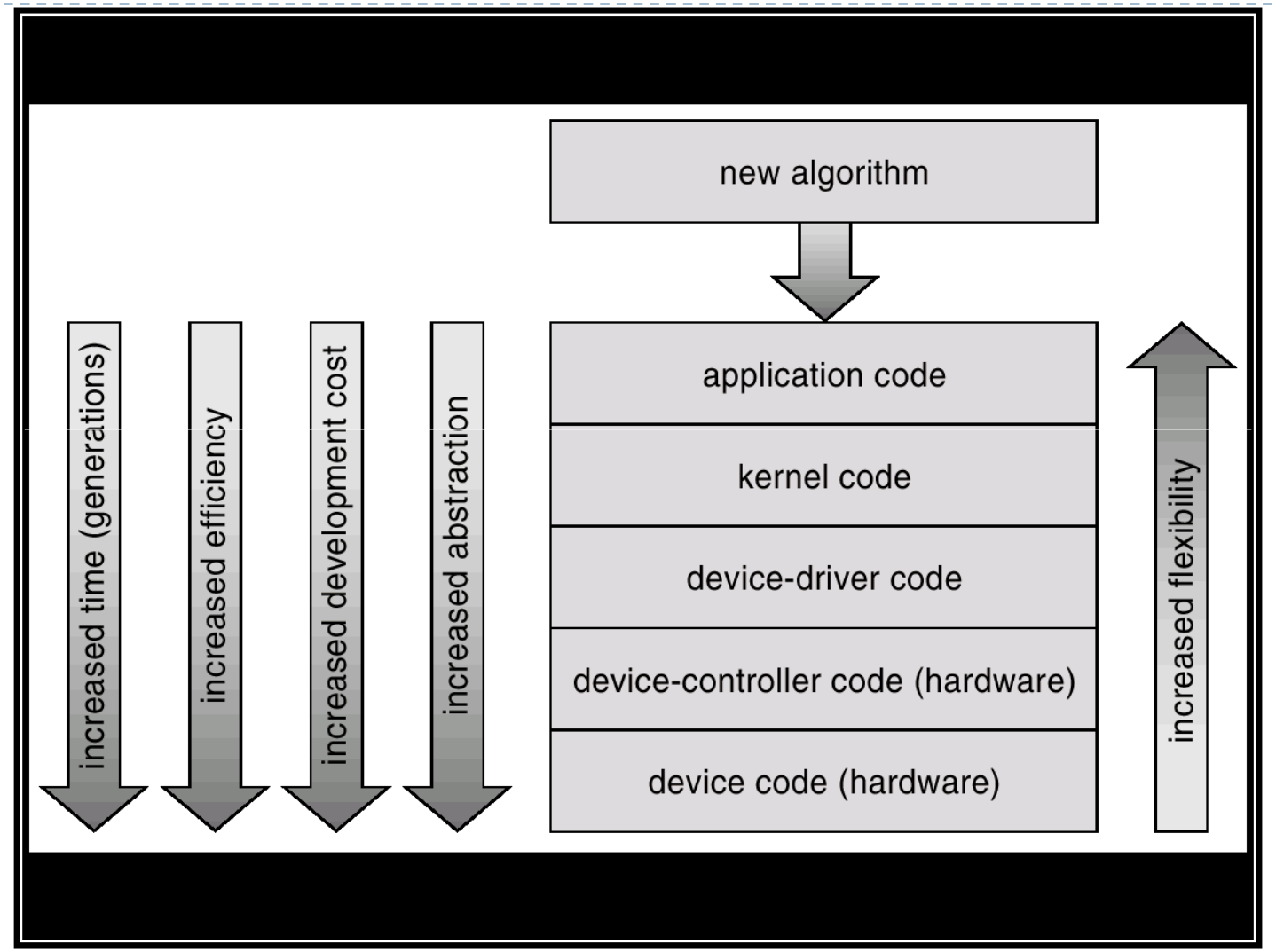
Software de E/S

- Os objetivos podem ser alcançados de forma eficiente estruturando o software de entrada/saída em quatro níveis:
  - Manipuladores de interrupção.
  - *Drivers* de dispositivo.
  - Software do sistema operacional independente do dispositivo.
  - Software do nível do usuário.

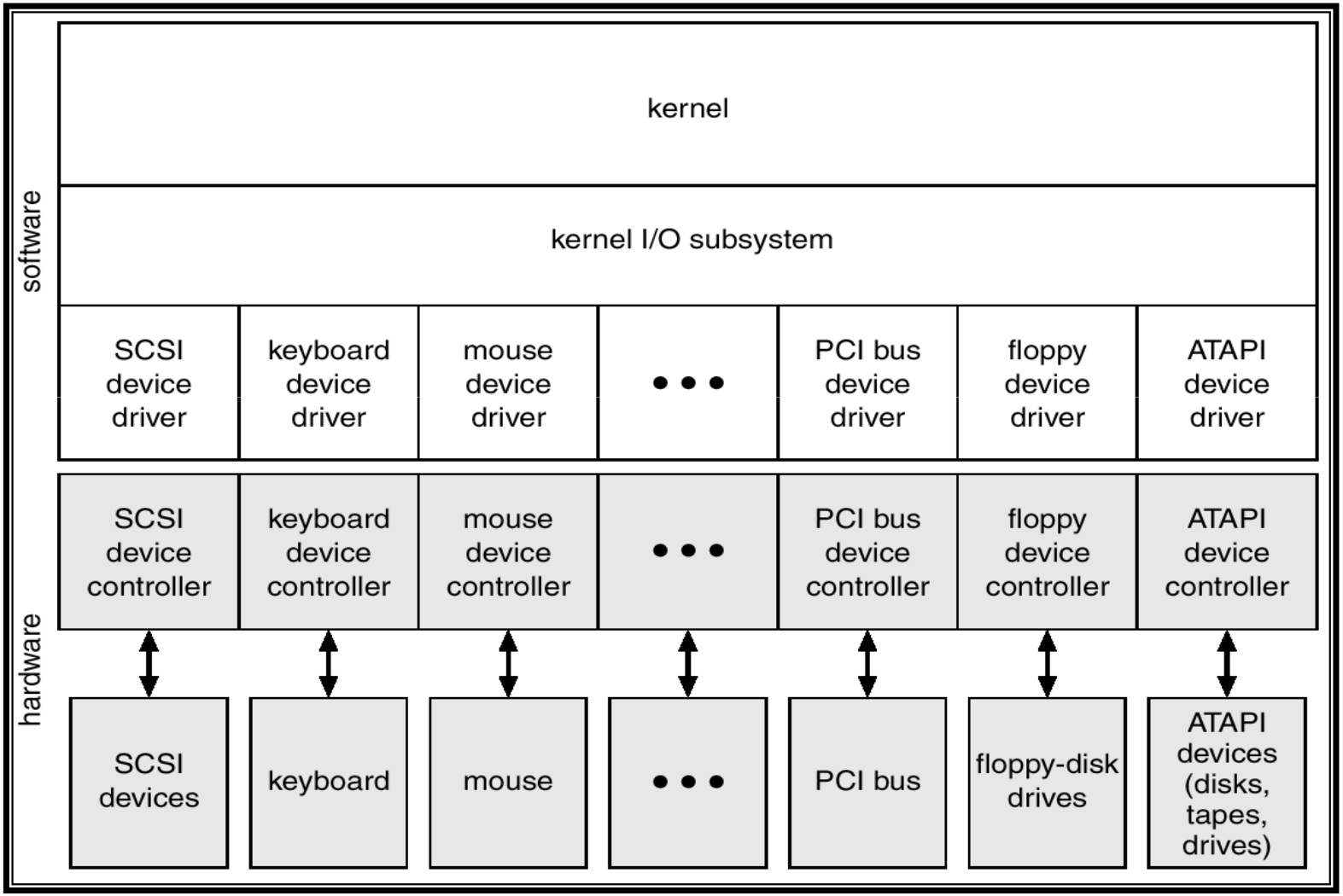
Camadas do Software de E/S



Camadas do Software de E/S



Estrutura de E/S do Kernel



Interrupções

- As interrupções devem ser tratadas sempre nos níveis mais baixos do sistema operacional, de forma que o mínimo possível de partes tenham que lidar com elas.

Interrupções

- Operações Síncronas (Bloqueio do processo). Exemplo:
  - DOWN num semáforo;
  - WAIT numa variável de condição de um monitor;
  - RECEIVE numa mensagem;
  - READ num arquivo.

Driver de Dispositivo

- Todo o código que depende do dispositivo está no *driver do dispositivo*. Cada *driver* manipula um tipo de dispositivo, ou, no máximo, uma classe de dispositivos muito semelhantes.
  - O *driver* do dispositivo é a única parte do sistema operacional que conhece quantos registradores tem a controladora do dispositivo e para que eles são usados.

Driver de Dispositivo

- Quando ocorre uma requisição de E/S, os seguintes passos são realizados pelo driver:
  - Traduzir a requisição de uma forma abstrata para uma forma mais concreta, ou seja, deve determinar quais operações da controladora são necessárias e em que ordem.

Driver de Dispositivo

- Quando ocorre uma requisição de E/S, os seguintes passos são realizados pelo driver:
  - Escrever os comandos nos registradores da controladora.
    - Algumas controladoras só podem tratar de um comando por vez.
    - Outras podem aceitar uma lista ligada de comandos, que elas podem executar sozinhas, sem nenhuma ajuda do sistema operacional.

Driver de Dispositivo

- Após a emissão de um ou mais comandos, duas situações são possíveis:
  - O driver do dispositivo deve esperar até que a controladora tenha feito algum coisa, isto é, ele se autobloqueia até a chegada da interrupção que o desbloqueia.
  - O driver não precisa se bloquear, isto é, a operação termina sem nenhum retardo.

Driver de Dispositivo

- Verificar a ocorrência de erro. Se não houve erro, o driver deve ter dados a serem passados para o software independente do dispositivo.
- Por fim, o driver deve retornar algumas informações de estado.

- Fila de requisições
  - Driver fica bloqueado esperando pela próxima requisição.

Software de E/S Independente do Dispositivo

- Apesar de parte do software de E/S ser específico para cada dispositivo, uma grande parte dele é independente.
  - Reuso x Eficiência
  - Independente x Específico

Software de E/S Independente do Dispositivo

- *Funções Comuns:*
  - *Interface uniforme para os drivers de dispositivo*
  - *Identificação do dispositivo*
  - *Proteção do dispositivo*
    - Usuários não autorizados.

Software de E/S Independente do Dispositivo

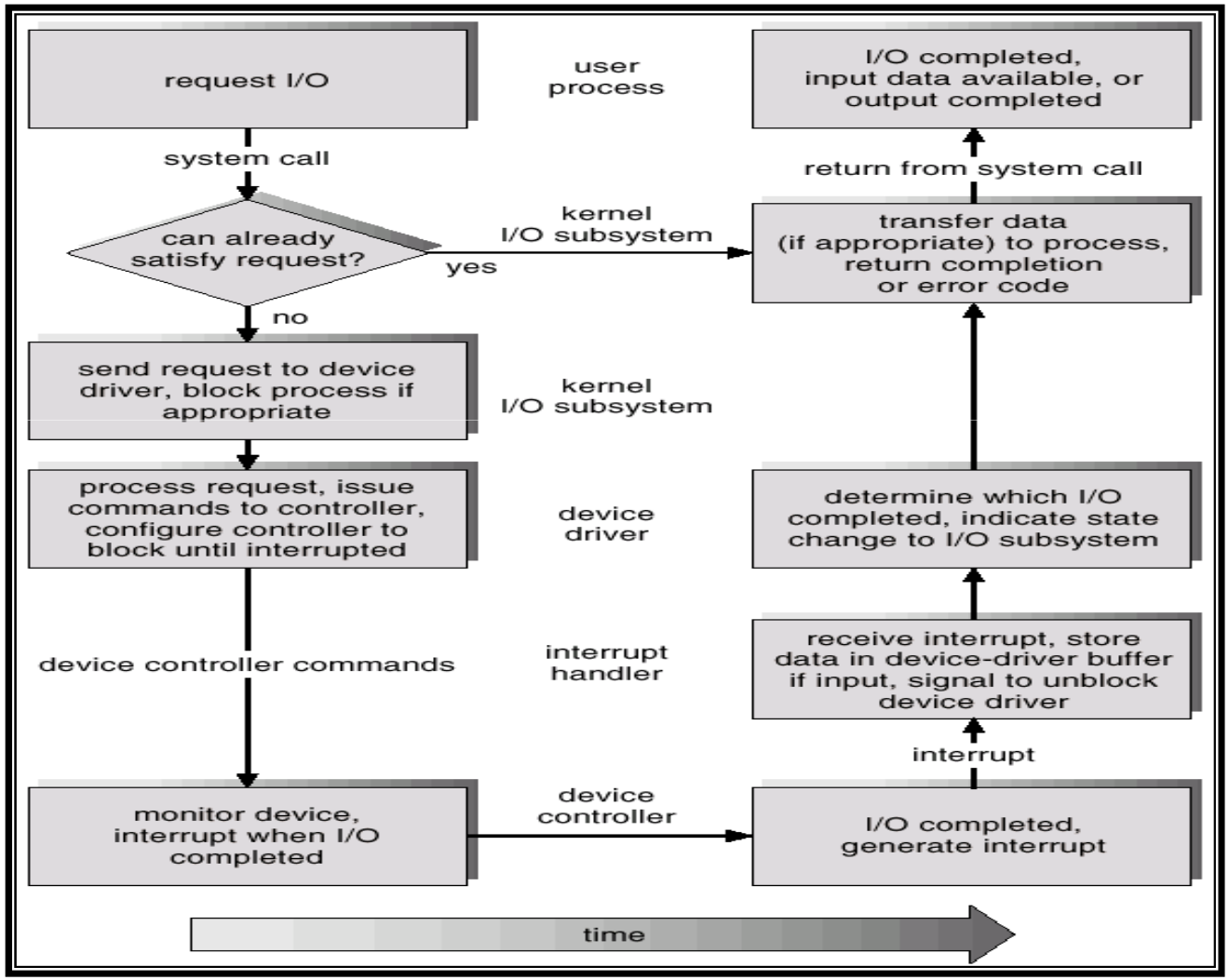
- *Fornecimento de um tamanho do bloco independente do dispositivo*
  - Discos diferentes com tamanhos de setor diferentes.
- *Bufferização*
  - Para os dispositivos de bloco;
  - Para os dispositivos de caractere.
- *Alocação de espaço para blocos*
  - Lista de Blocos Livres
- *Alocação e liberação de dispositivos dedicados*
  - Exclusão Mútua
- *Informação de erro*
  - Erros não resolvidos pelo *Driver*.

Software de E/S no Espaço do Usuário

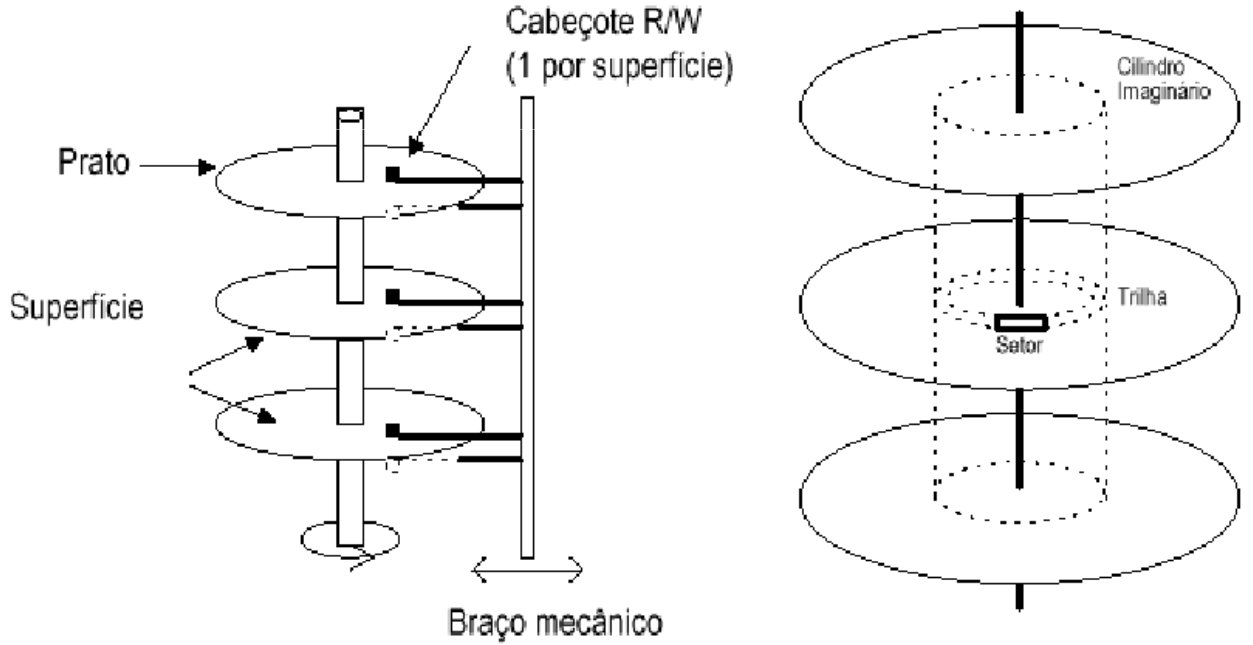
- *Rotinas de biblioteca, ligadas junto com os programas de usuário*
  - Procedure CALL

- *Sistemas de spooling*
  - Impressoras

Ciclo de Vida de um I/O



Estrutura de Disco



Estrutura de Disco

- discos são organizados em **trilhas**, que por sua vez dividem-se em **setores**
- **cilindro** é o conjunto de trilhas de todas as superfícies do disco que ficam exatamente à mesma distância do eixo central
- para acessar as informações armazenadas no disco é necessário especificar cilindro, cabeça (trilha do cilindro) e setor (dentro da trilha)
- a **formatação física** é um procedimento realizado pelo fabricante que define as trilhas e setores de um disco

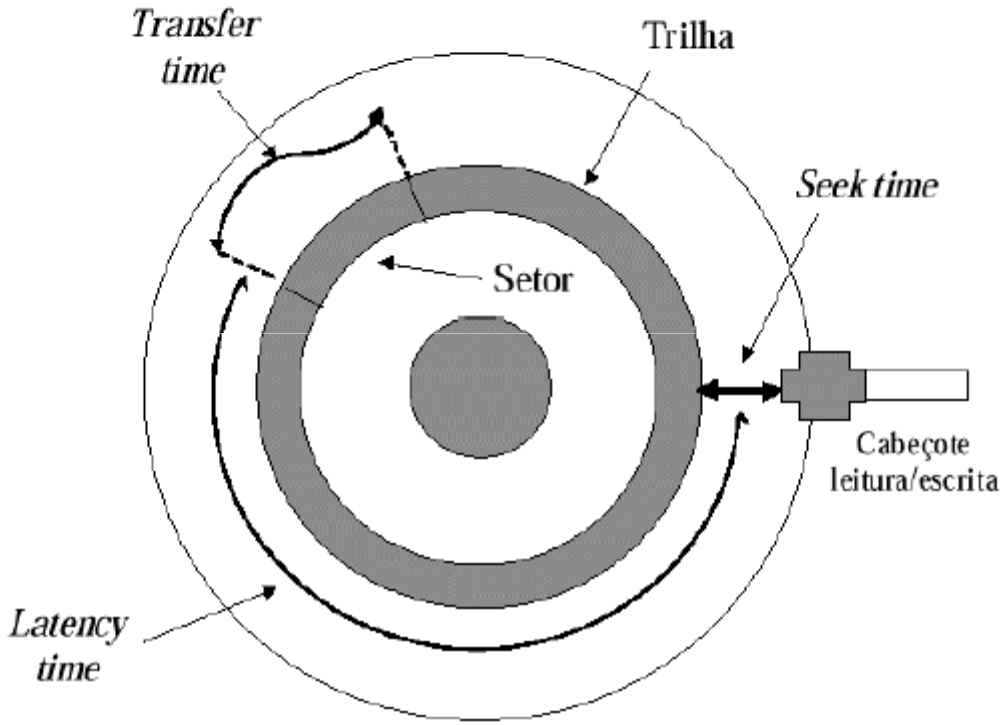
Estrutura de Disco

- O disco pode ser acessado pelos seguintes métodos:
  - CHS: deve-se informar cilindro (C), cabeça (H) e setor (S)
  - LBA (*Linear Block Addressing*): permite ver o disco como um conjunto linear de blocos

Estrutura de Disco

- O tempo de acesso a um disco é composto de:
  - tempo de busca (*seek*)
  - tempo de latência ou rotação (*latency*)
  - tempo de transferência

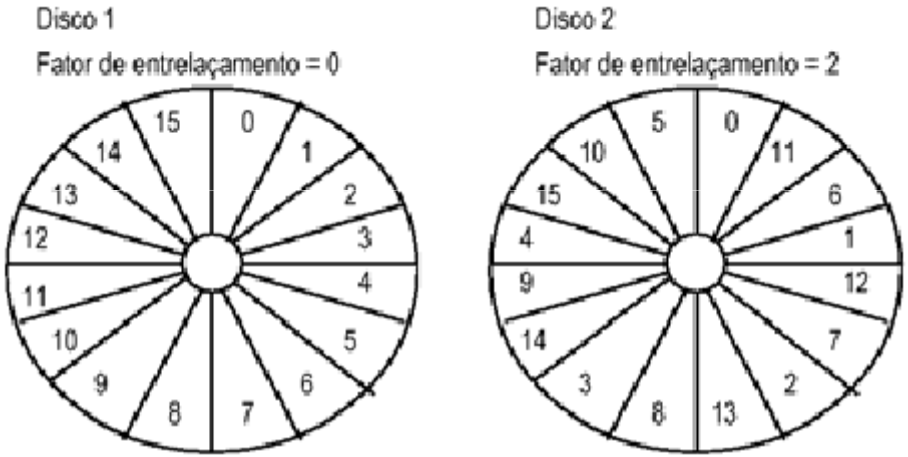
Estrutura de Disco



Estrutura de Disco

- Entrelaçamento (interleaving)
  - Corresponde à definição da numeração dos setores de forma que um setor k não fique contíguo no disco a um setor k+1
- Fator de entrelaçamento é a distância física entre os setores k e k+1
- O entrelaçamento permite que dois setores contíguos sejam lidos no mesmo giro

Estrutura de Disco



Estrutura de Disco

- Discos são endereçados como vetor de blocos lógicos.
- Vetor de blocos é mapeado em setores sequenciais
  - O setor 0 é o primeiro setor da primeira trilha do cilindro mais externo
  - O mapeamento é feito de fora para dentro do disco

Escalonamento de Disco

- Em sistemas multiprogramados há um número grande de acessos a disco
  - Alto tempo de espera por E/S
    - ➔ Ordenar e atender as requisições de forma a minimizar o tempo de espera dos processos
  - Como: diminuindo a movimentação das cabeças de leitura e gravação consegue-se atender mais requisições por unidade de tempo

Escalonamento de Disco

- O SO é reponsável pelo uso eficiente do hardware
  - Para discos isso significa acesso rápido
  - O tempo de acesso é determinado por:
    - *Seek time* – tempo para mover a cabeça de leitura para o cilindro que contém o setor desejado.
    - *Rotational latency* – tempo para o disco girar e posicionar o setor desejado

Escalonamento de Disco

- O SO é reponsável pelo uso eficiente do hardware
  - Objetivos:
    - Minimizar seek time
      - Seek time ≈ seek distance
    - Throughput de disco é o total de bytes transferidos dividido pelo tempo total entre a primeira requisição e o fim da última transferência

Escalonamento de Disco

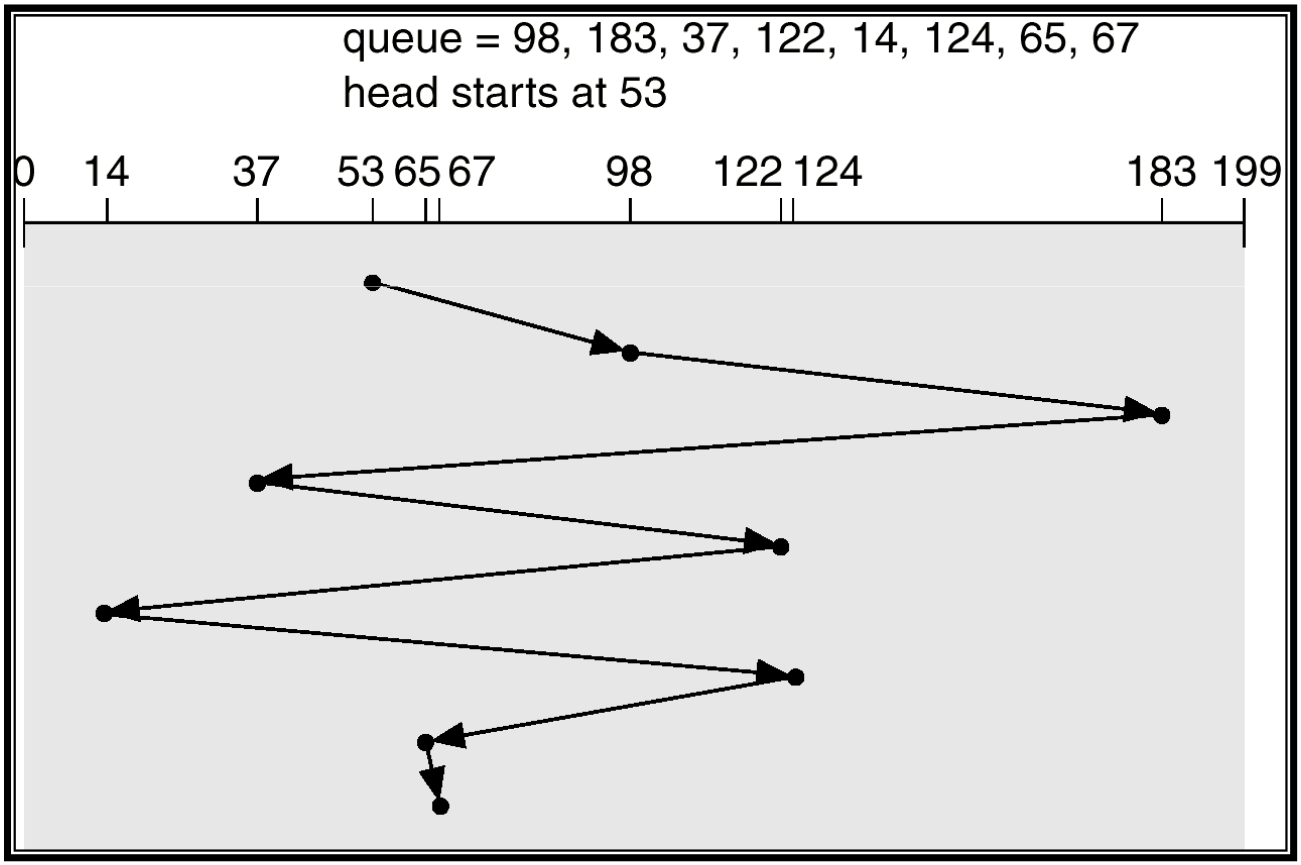
- Algoritmos
  - Fila de requisições (cilindros de 0-199).

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

Cabeça de leitura posicionada no cilindro 53

FCFS

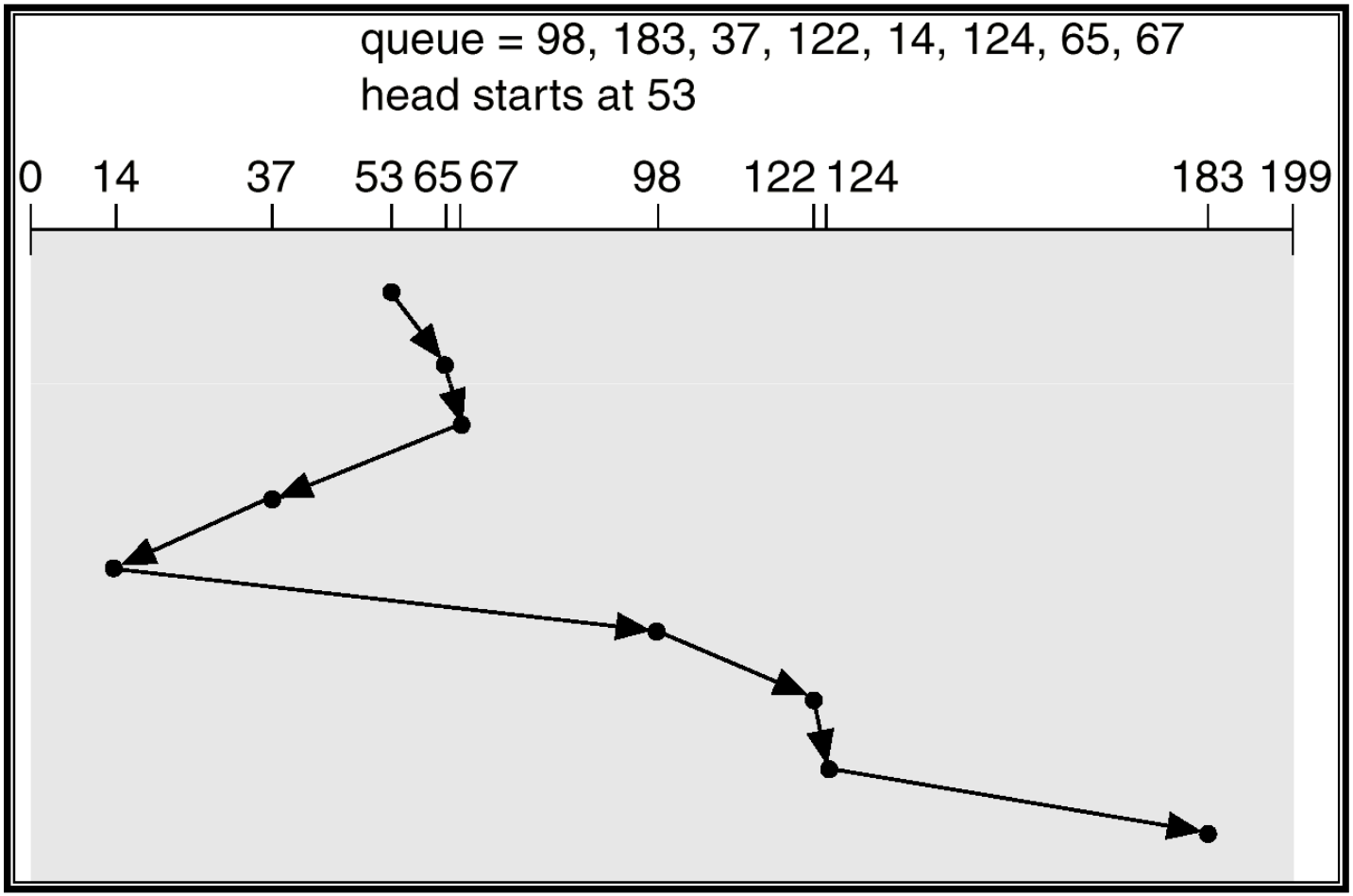
640
cilindros
percorridos



SSTF

- Seleciona a requisição com o menor tempo de busca com base na posição atual da cabeça de leitura
- SSTF é baseado em SJF; pode causar starvation de algumas requisições
- Total de movimentos da cabeça: 236 cilindros

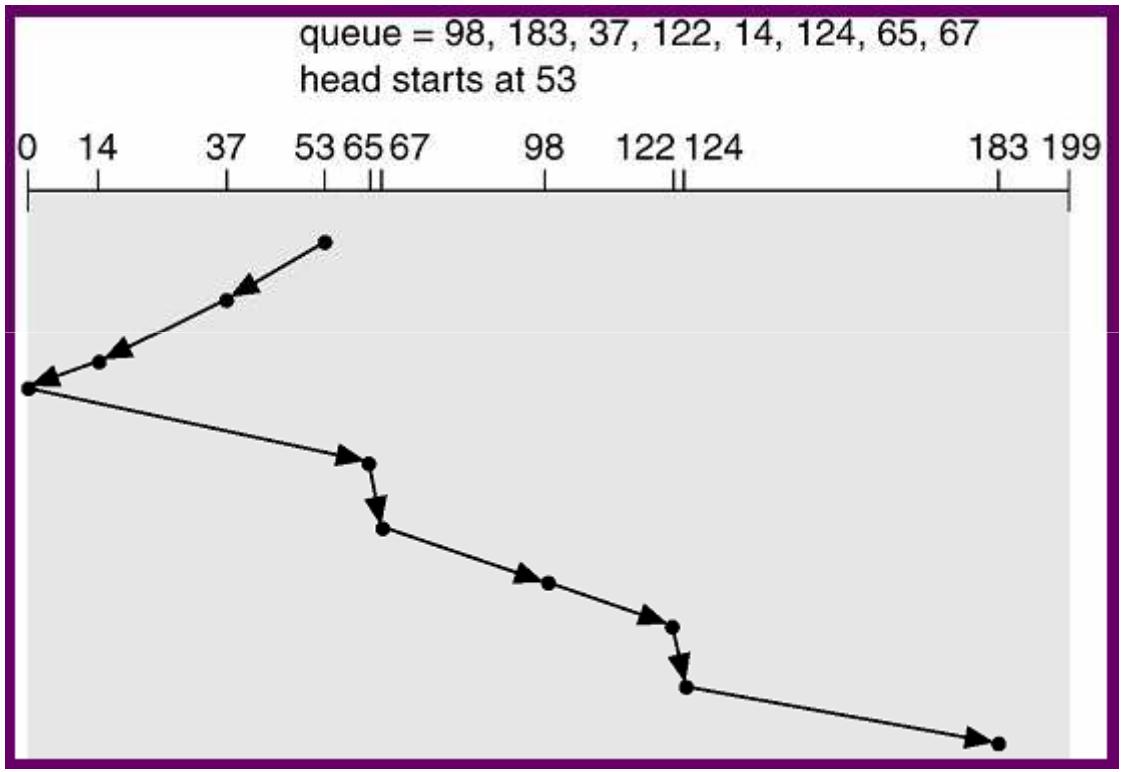
SSTF


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

SCAN

- O braço inicia num extremo do disco e se move ao outro extremo, atendendo as requisições no caminho. Vai e volta atendendo requisições.
- Algoritmo do Elevador
- 236 cilindros percorridos.

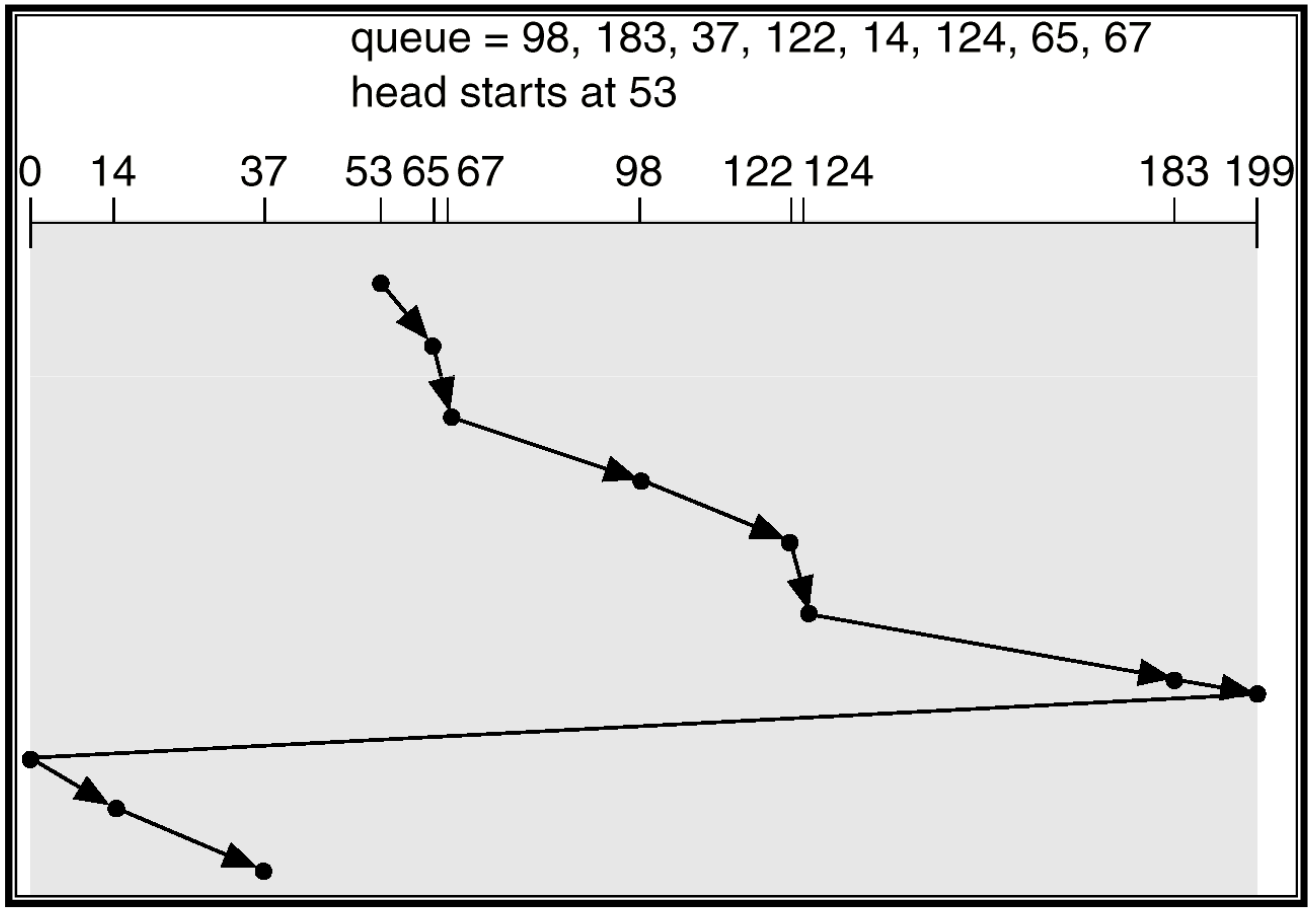
SCAN


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0 14 37 53 65 67 98 122 124 183 199

C-SCAN

- Tempo de espera mais uniforme que SCAN.
- O braço inicia num extremo do disco e se move ao outro extremo, atendendo as requisições no caminho. Vai atendendo requisições e volta sem atender.
- Trata os cilindros como uma lista circular

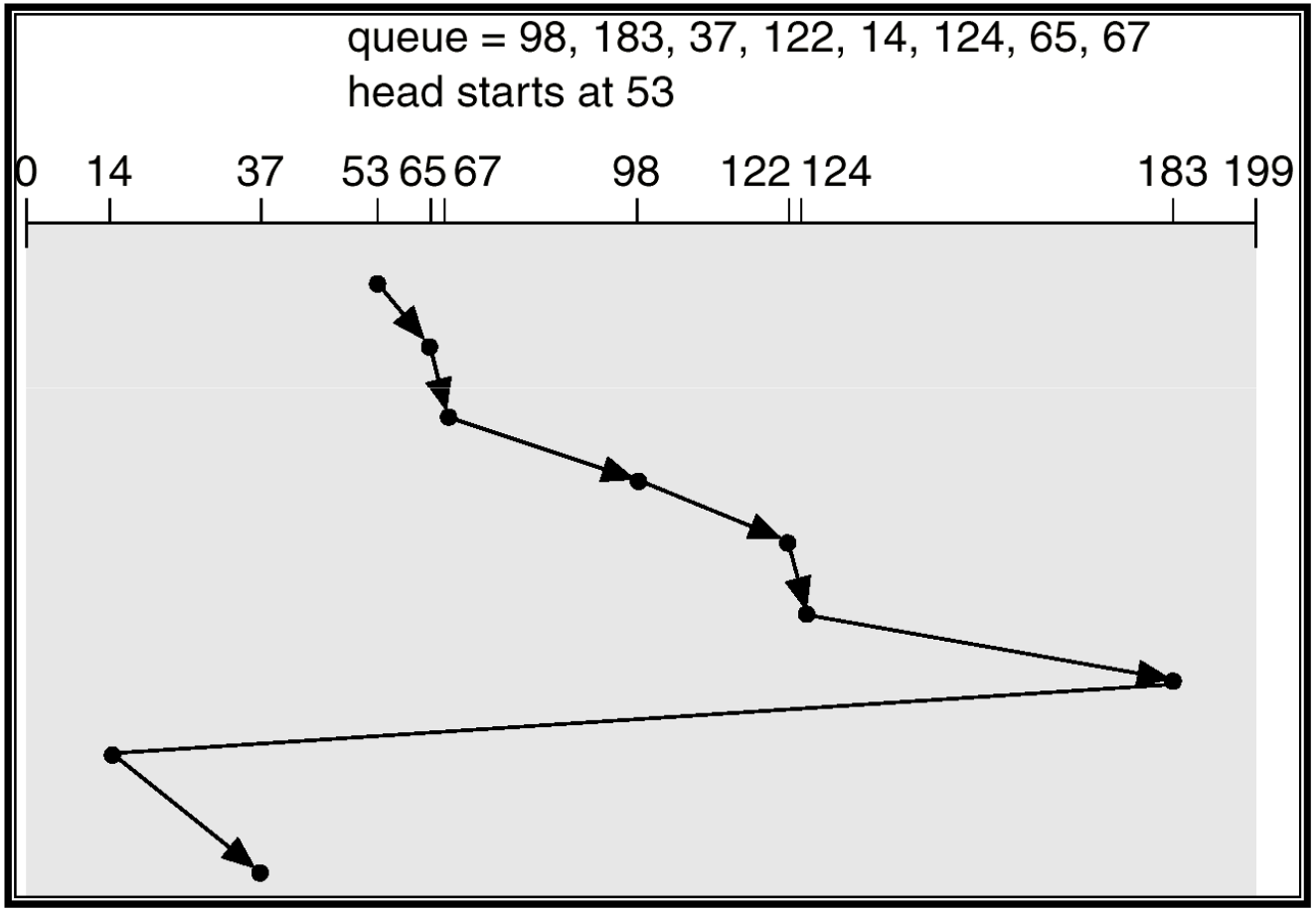
C-SCAN


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

C-LOOK

- Versão do C-SCAN
- O braço inicia num extremo das requisições e se move ao outro extremo, atendendo as requisições no caminho. Vai atendendo requisições e volta sem atender.

C-LOOK


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199