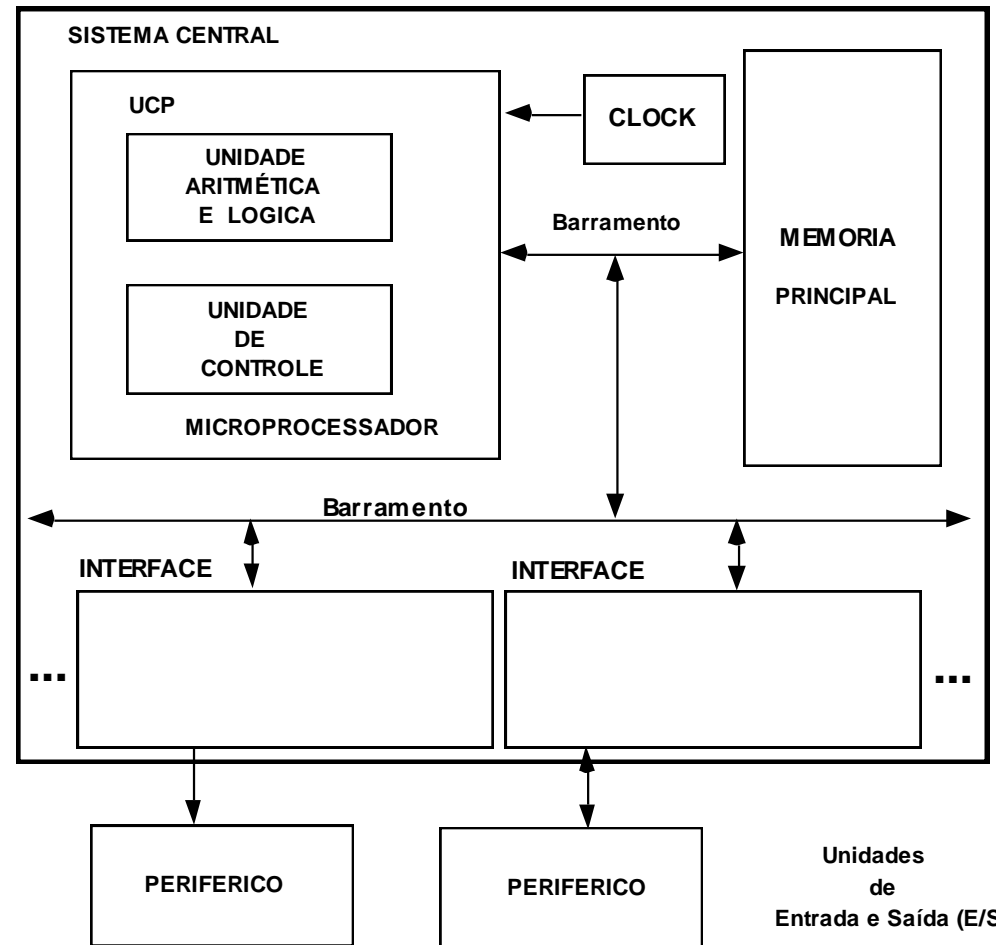


Gerência de E/S

- Relacionada com aspectos de HW de um computador
- Princípios básicos de HW
 - Interface
 - Controlador
 - Interação entre CPU e controladores de dispositivo
- Princípios básicos de SW
 - Drivers de dispositivo

Princípios básicos de HW: Interface

- Possibilita a conexão entre dispositivo e computador
- Interconectada aos barramentos internos do computador
- Para tratar a diversidade, complexidade e diferentes formas de operação, as interfaces contam com **controladores**



Princípios básicos de HW: **Controlador**

- Processador projetado especificamente p/ realizar uma função (ex.: controlar um disco rígido)
- Comunicação CPU-controlador:
 - Através de registradores (dado, status, comando)
- Função: implementar conjunto de comandos p/ seu periférico
 - Traduz cada ordem colocada no registrador de comando em uma sequência de ações

Princípios básicos de HW: dispositivos x interconexão às interfaces

- Interface serial
 - Uma única linha p/ transmitir dados
 - 1 byte é transferido como uma sequência de bits
 - Ex.: modems, alguns mouses e impressoras
- Interface paralela
 - Várias linhas p/ transmitir dados
 - Vários bits transferidos em paralelo entre dispositivo e interface
 - Ex.: impressoras paralelas

Princípios básicos de HW: interação entre CPU e controladores de dispositivos

- Interação pode se dar de 3 maneiras:
 - E/S programada
 - Via interrupções
 - DMA

E/S programada (1)

- usada quando não há sistema de interrupção
 - interação CPU-controlador é de responsabilidade do programador
 - funcionamento: envia comando ao controlador e espera a realização
 - ex.: processador envia cmdo de leitura ao controlador e testa registrador de status (busy loop)
 - dado disponível ? sim → processador efetua a leitura
 - problema: E/S é lenta em relação ao processamento
-

E/S programada (2)

- busy loop → desperdício
 - alternativa: processamento entre verificações de status
 - ***polling***: verifica periodicamente o estado da op. E/S
 - reduz desperdício de tempo do processador
 - problema: frequência do polling
 - alternativa: dispositivo de E/S sinaliza quando o dado estiver disponível
 - **interrupção**

E/S via interrupção

- processador é responsável (via SW) apenas por iniciar a op. E/S enviando comandos ao controlador
- após, CPU passa a executar outra tarefa, e o controlador realiza a op. E/S
- E/S termina - controlador interrompe o processador
 - executa tratador de interrupção: aciona o driver do dispositivo

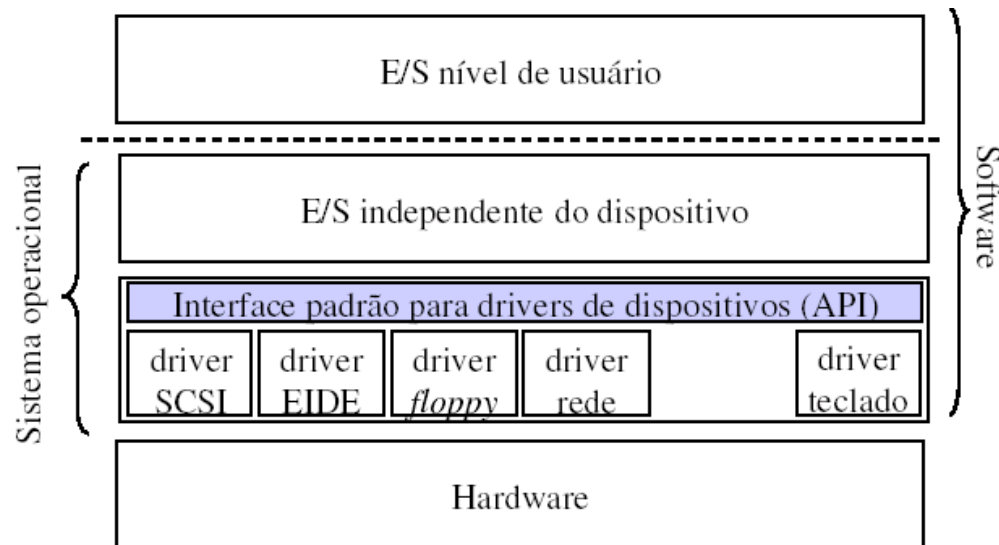
DMA (*Direct Memory Access*)

- transfere diretamente os dados da interface p/ memória
 - usa HW especial = controlador de DMA
 - acessa diretamente a memória
 - conectado fisicamente ao bus de dados e endereços
 - possui registradores usados pela CPU p/ programar a transferência de dados
 - após acionar o DMA, o processador pode se dedicar a outra tarefa
 - fim da transferência - controlador de DMA sinaliza o processador via interrupção de HW
-

Princípios básicos de SW (1)

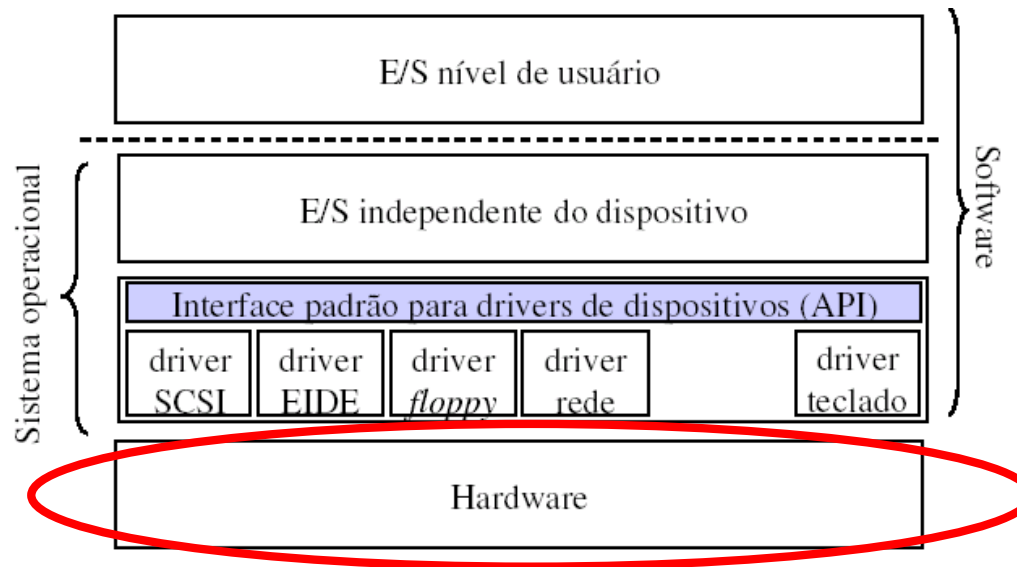
- subsistema de E/S de um SO é complexo
 - deve tratar diversos tipos de dispositivos
- objetivo: padronizar forma de acesso aos dispositivos

subsistema de E/S:
organizado em
camadas



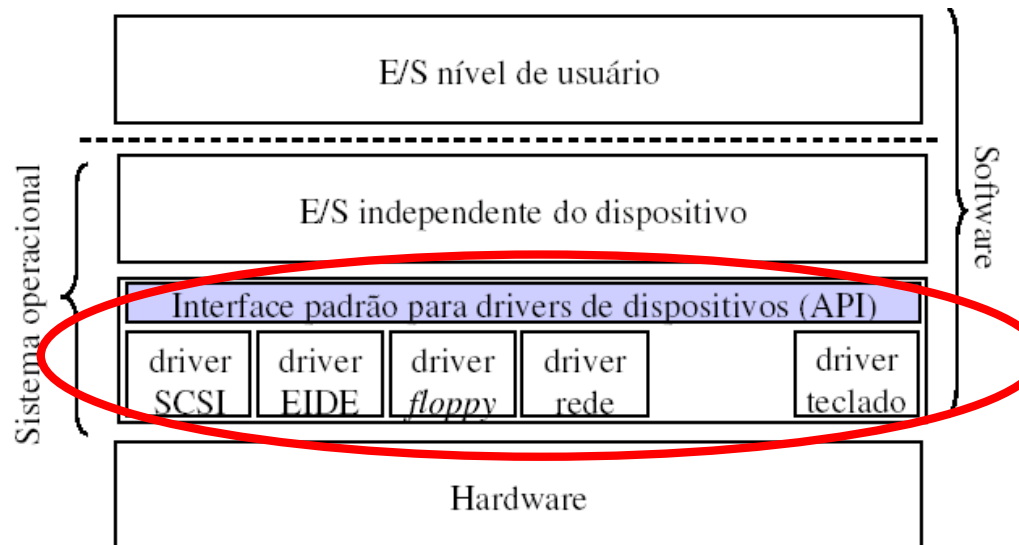
Camada de Hardware

- Composta pelo HW dos dispositivos de E/S
 - A interface que ela apresenta à camada superior (drivers de dispositivos) é composta pelos mecanismos discutidos anteriormente



Camada Device Drivers (1)

- Camada inferior de SW – drivers de dispositivos
 - Cada módulo implementa acesso a um dispositivo
 - Objetivo dos drivers: esconder as diferenças entre os vários dispositivos
 - Fornece à camada superior uma “visão uniforme” dos dispositivos através de uma interface de programação única

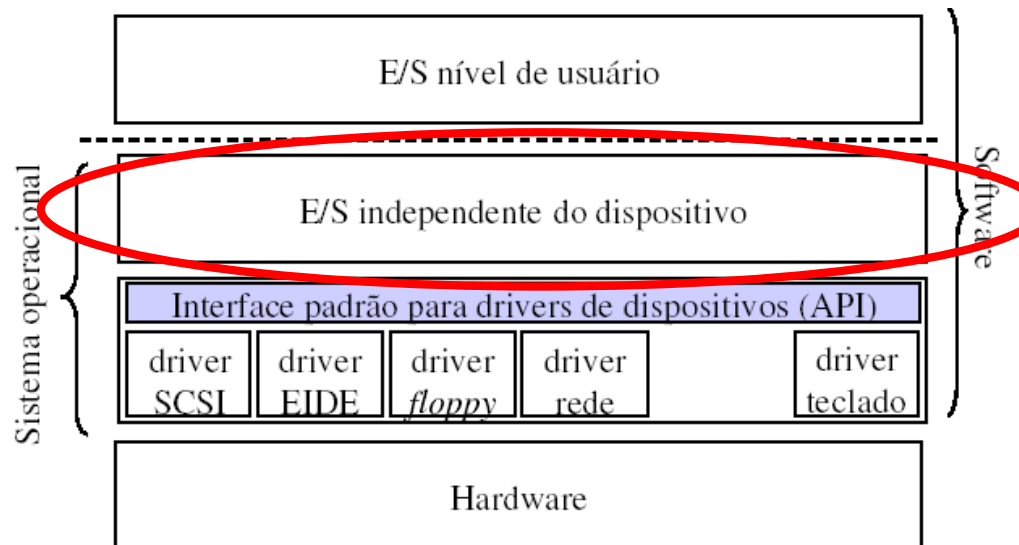


Camada Device Drivers (2)

- P/ fornecer a “visão uniforme”, os dispositivos são classificados de acordo com a transferência de dados em:
 - *Dispositivos orientados a bloco*: armazenam informações em blocos de tamanho fixo, cada um com seu próprio endereço
 - ex.: discos
 - *Dispositivos orientados a caractere*: realizam as transferências byte a byte, a partir de um fluxo de caracteres, sem considerar uma estrutura qualquer
 - ex.: impressoras, portas seriais

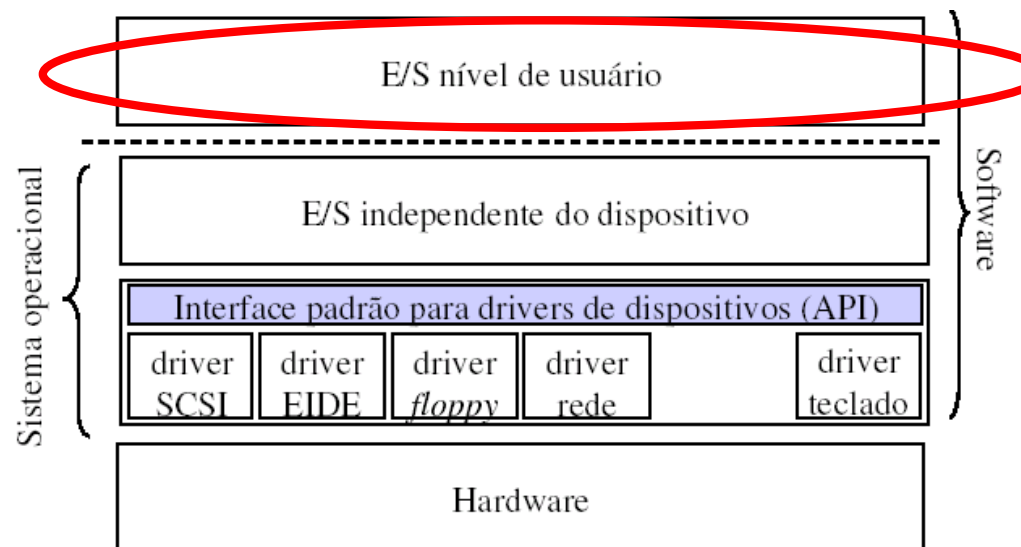
Camada E/S independente de dispositivo

- Camada independente de dispositivo: apenas usa a interface oferecida pelos drivers
 - Implementa funções genéricas e serviços gerais de E/S
 - ex.: função read – onde o 1º argumento indica o dispositivo a ser usado, e é mapeada p/ uma seq. de op. compatíveis com o dispositivo em questão

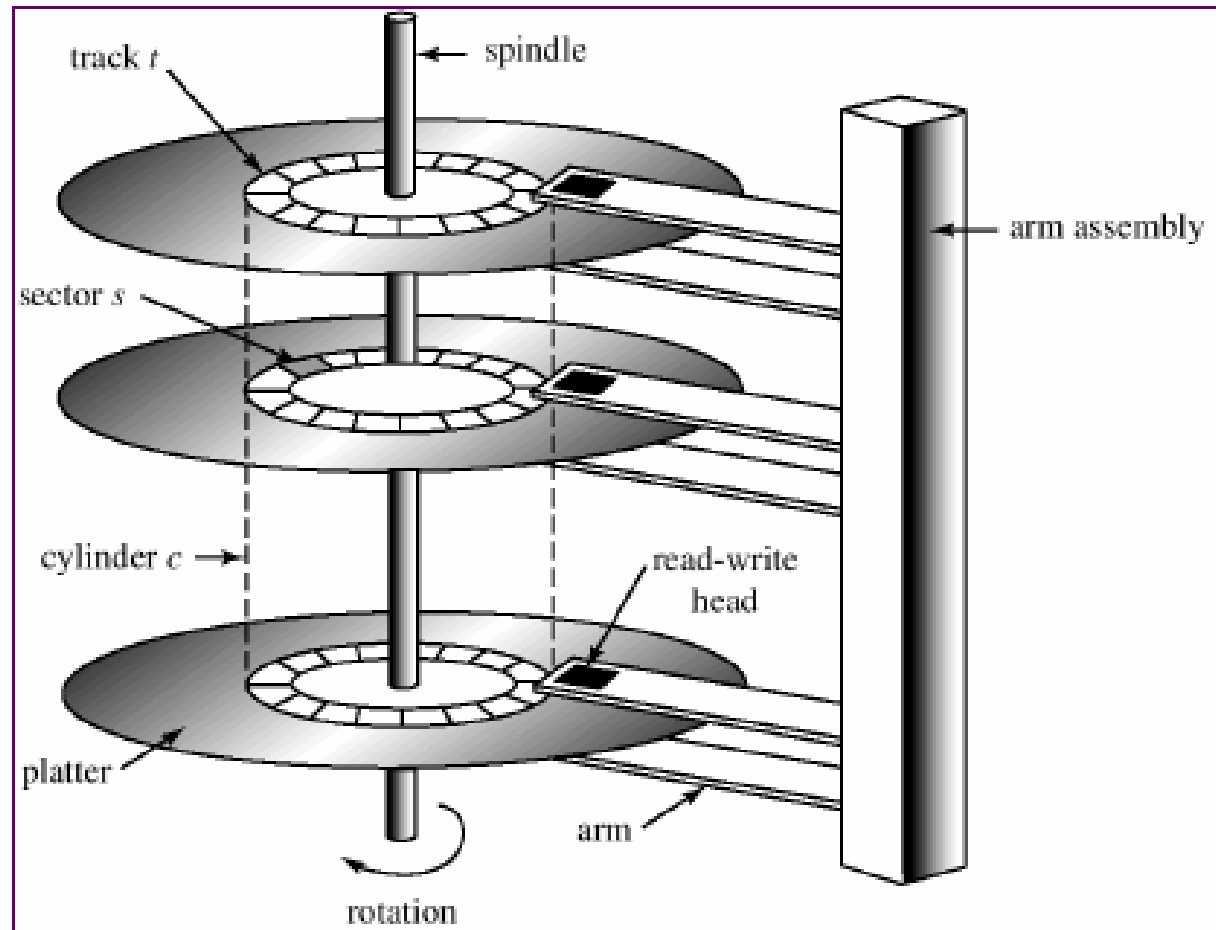


Camada E/S em nível de usuário

- Camada não pertencente ao SO
 - Usuário “vê” os dispositivos através de aplicativos e de LPs
 - Ex.: LPs oferecem rotinas de bibliotecas de E/S (ex.: printf do C)



Estudo de caso: periférico Disco



Disco: operação de E/S

- sistema multiprogramado: processos realizam pedidos concorrentemente e são bloqueados até a operação ser realizada
- escalonador do disco: ordena e atende os pedidos de E/S
 - objetivo: minimizar o tempo de bloqueio de processos

Tempo de acesso

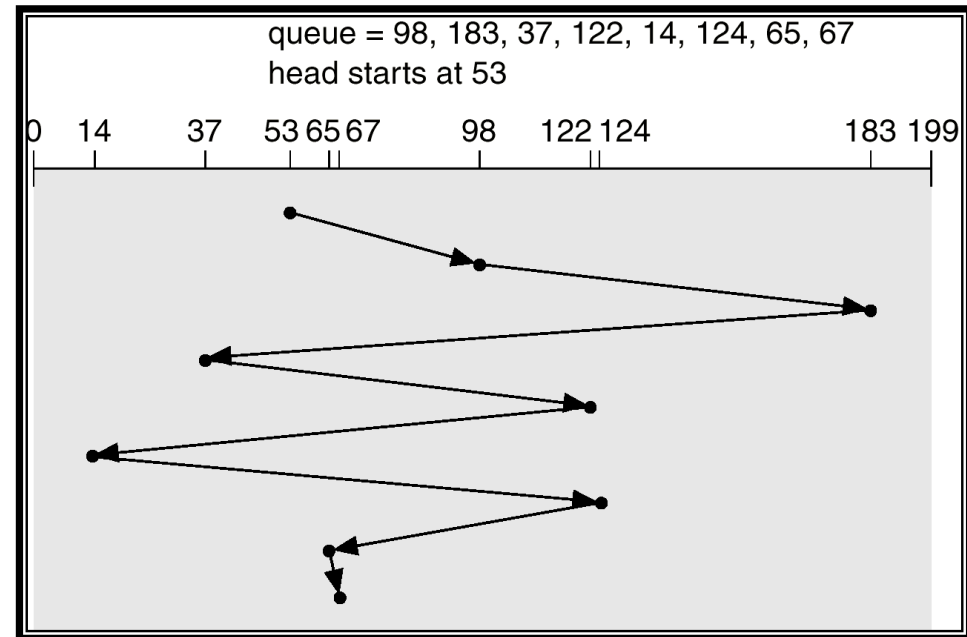
- acesso a disco: posiciona cabeçote de leitura/escrita no setor onde o dado será lido/escrito
 - **tempo de acesso** é formado pela soma de:
 - tempo de *seek*
 - t p/ deslocar os cabeçotes até o cilindro onde está a trilha a ser acessada
 - tempo de latência (atraso rotacional)
 - t p/ cabeçote se posicionar no início do setor a ser lido/escrito
 - tempo de transferência
 - t p/ efetuar a transferência dos dados (leitura/escrita)
-

Escalonamento de acesso a disco

- tempo p/ uma operação de E/S em disco tem como principal componente o **tempo de seek**
- algoritmos de escalonamento:
 - ideia: minimizar o t médio de seek (exceto o FCFS)
 - **FCFS**
 - **SSTF**
 - **SCAN, C-SCAN**
 - ...

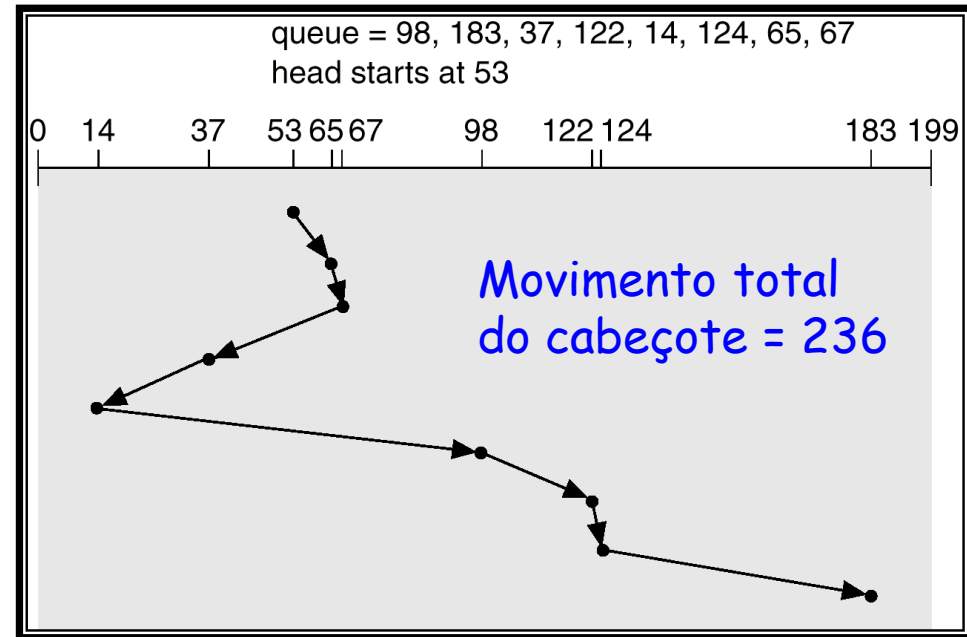
FCFS (First-Come-First-Served)

- ☺ simples de executar
- ☺ justo
- ☹ padrão de busca aleatório (movimentação mecânica)
- ☹ comportamento ruim sob carga altas (fila de requisição grande)



SSTF (Shortest-Seek-Time-First)

- atende solicitação c/ menor tempo de busca (seek) a partir da posição atual do cabeçote
- similar ao escalon. SJF
- ☺ redução do t de busca
- ☺ t médio tende a ser mais baixo
- ☹ não garante justiça (“fura” fila)
- ☹ postergação indefinida
- ☹ alta variância
 - ruim p/ sistema interativos, aceitável p/ batch

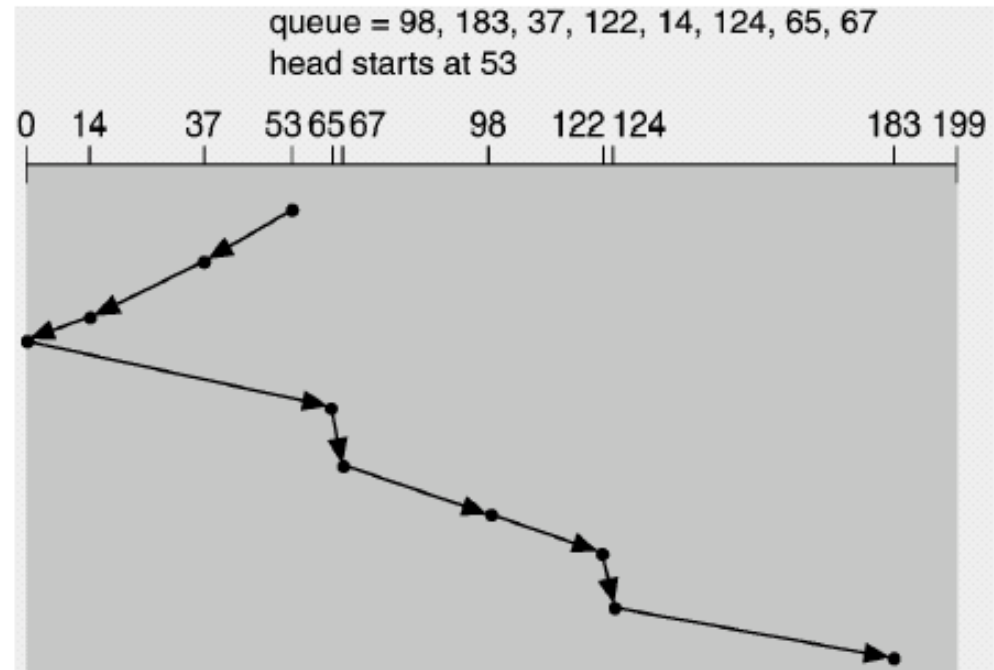


SCAN (1)

- variação do SSTF
 - algoritmo do elevador
 - estabelece um sentido preferencial p/ movimento do cabeçote
 - enquanto há pedidos no sentido atual, o braço do disco continua nesse sentido, atendendo os pedidos mais próximos
 - se não há mais pedidos no sentido atual, o sentido do braço é invertido e inicia uma varredura
 - elimina a possibilidade de *starvation*
-

SCAN (2)

- ☺ oferece bons tempos médios de resposta
- ☺ bom rendimento
- ☺ variância menor que o SSTF
- ☹ não justo
 - trilhas das extremidades são “visitadas” c/ menor frequência que as trilhas internas



C-SCAN (Circular SCAN)

- pedidos atendidos em um só sentido da varredura
 - elimina defeito do algoritmo SCAN:
 - após inverter a varredura, o SCAN privilegia pedidos dos cilindros recém servidos
- ☺ oferece um tempo médio de acesso mais uniforme que o SCAN

