

# FSO - 31/10/2018

## Sumário:

- Discos de estado sólido (SSD)
- Utilização de discos: formatação física e lógica
- Algoritmos de escalonamento de pedidos de acesso ao disco

Bibliografia: OSTEP 44, 37, 38

# SSD

- Armazenamento de bits em células *cells* condensadores como uma carga eléctrica- n níveis  $\rightarrow \log_2 n$  bits
- Um grupo de células constitui um *bloco* (128 / 256 *Kbytes* ). Os blocos estão divididos em páginas de 4Kbytes

# Leitura e escrita em SSD

- A leitura é feita por páginas: Muito rápida (10 micro segundos). Não há movimento de cabeça
- Escrita de uma página
  - Exige o apagar de todo o bloco: alguns milisegundos
  - E a reescrita do bloco: da ordem das centenas de microsegundos

# Fiabilidade de um SSD

- A escrita dentro de uma página convém que seja ordenada
- Estima-se que um bloco pode ser lido ou escrito 10.000 vezes antes de se deteriorar
- Várias técnicas usadas para uniformizar o número de leituras / escritas em cada bloco físico

# Estrutura do disco

- Do ponto de vista lógico, as unidades de disco são endereçadas como grandes vectores de *blocos lógicos*, em que o bloco é a unidade mínima de transferência.
- Este vector de blocos lógicos é mapeado sequencialmente em sectores do disco.
  - O sector 0 é o primeiro sector da primeira pista do cilindro mais exterior.
  - O “mapping” continua nessa pista, seguem-se as outras pistas do mesmo cilindro, e depois todos os cilindros em direcção ao interior do disco.
- Os controladores escondem detalhes da formatação física:
  - N° de sectores por pista não constante
  - Optimizações na numeração de sectores na mesma pista
  - Substituição de blocos estragados

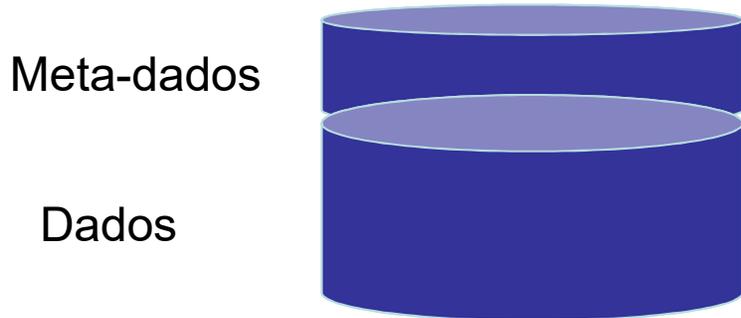
# Utilização de um disco

- Conter um sistema de ficheiros
  - mkfs/format: inicialização da parte dos metadados ...
- Conter uma partição de “swap” (ou disco de paginação)
- “Raw” – acesso bloco a bloco :Bases de dados, backups

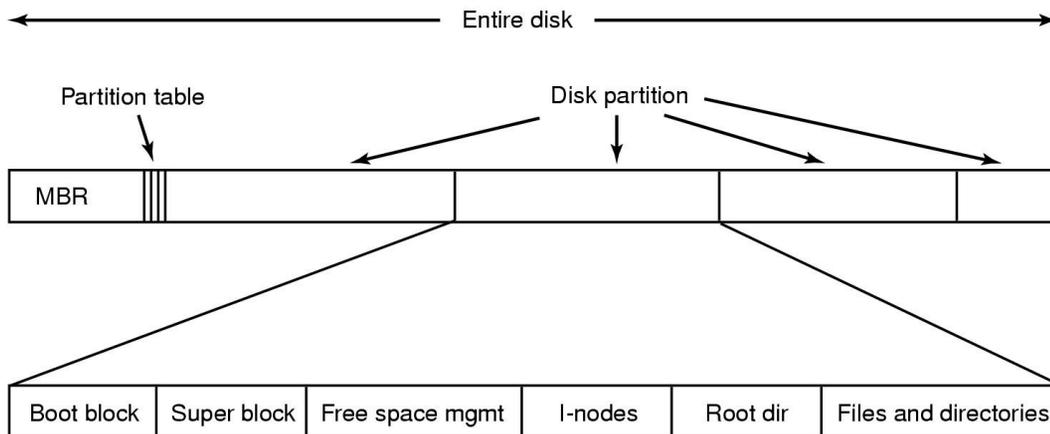
# Formatação física vs lógica

- *Formatação de baixo nível* , ou *formatação física* — Divisão do disco em sectores para que o controlador do disco possa ler e escrever.
- Para guardar ficheiros no disco, o SO tem de gravar nele as suas próprias estruturas de dados.
  - *Particiona* o disco em um ou mais grupos de cilindros (partições).
  - *Formatação lógica* ou “fazer um sistema de ficheiros”.
- Pode incluir um Boot block.

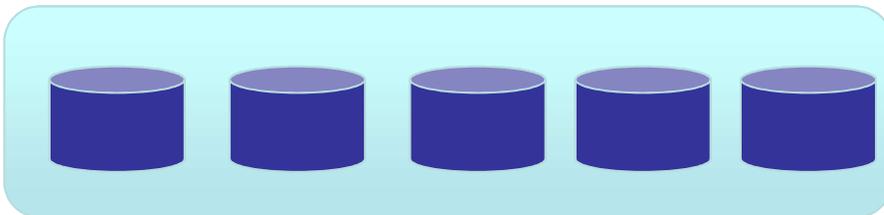
# Volume ou disco lógico



Correspondência directa disco lógico / disco físico (partição única)

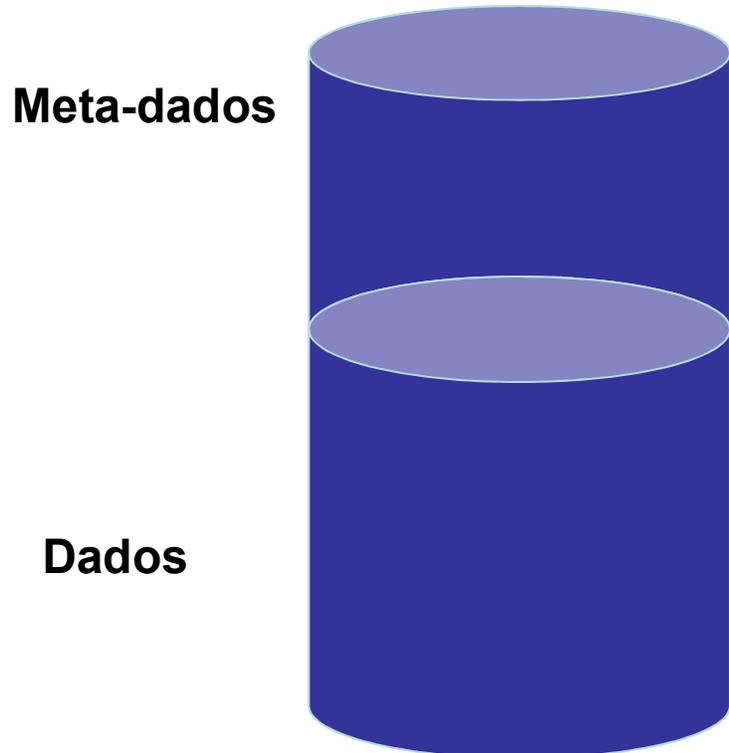


Vários volumes no mesmo disco físico (múltiplas partições)



Um volume que se estende por vários disco físicos (ex: RAID 0)

# Formatação de um disco lógico



- Mapa de ocupação dos blocos de dados
  - Bitmap (0- livre ; 1 ocupado)
  - Inicialmente tudo a zero (salvo os blocos da directoria raiz)
- Tabela Geral de Ficheiros (Tabela de i-nodes)
  - Todas as entradas livres, salvo a da directoria raiz (i-node 1)

Disco vazio, único ficheiro (no sentido lato) é a *directoria raiz*

- Format (Windows)
- Mkfs (Unix)

Tudo o que existia no disco é perdido !

# Dados e meta-dados contidos num disco lógico ou volume

- Dados
  - Blocos com o conteúdo dos ficheiros. Inicialmente apenas a directoria raiz
  - Irá ser ocupado com ficheiros “normais” e directorias
- Metadados
  - Super-bloco: replicado, em posição fixa no disco
    - informação sobre o tamanho (nº de blocos, dimensão de várias tabelas, endereços iniciais dos blocos que contêm as estruturas indicadas a seguir
  - Mapa de blocos ocupados e livres
  - Tabela geral de ficheiros
    - Para cada ficheiro informação sobre o tipo, tamanho, datas de acesso, dono, protecções, endereços da zona de dados com o conteúdo do ficheiro

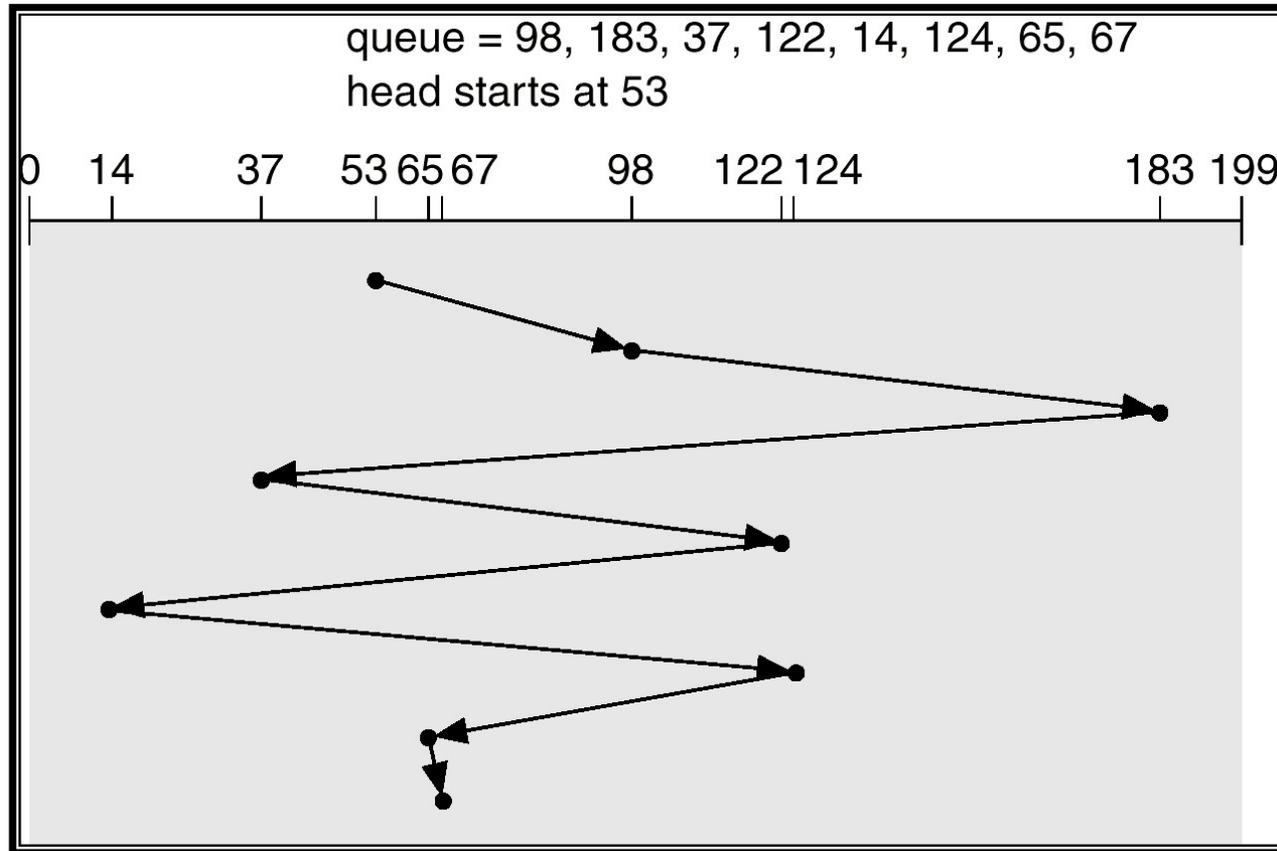
# Escalonamento de acesso ao disco

- O “Seek time” é o principal factor no tempo de acesso ao disco
- Para cada disco há um certo número de pedidos de acesso pendentess
- A reordenação da fila pode conduzir a tempos médios de acesso mais pequenos do que os conseguidos com FCFS (first come first served)
- As métricas que definem a qualidade dos algoritmos são:
  - Número de pistas atravessadas pela cabeça
  - Número de inversões de sentido de deslocamento

# Escalonamento dos pedidos de acesso ao disco

- A seguir ilustram-se vários algoritmos para escalonar (ordenar) a execução de pedidos de I/O sobre o disco
- Os algoritmos são ilustrados para:
  - um disco com pistas de 0 a 199
  - sequência de pedidos para as pistas 98, 183, 37, 122, 14, 124, 65, 67
  - posição corrente da cabeça: pista 53

# FCFS

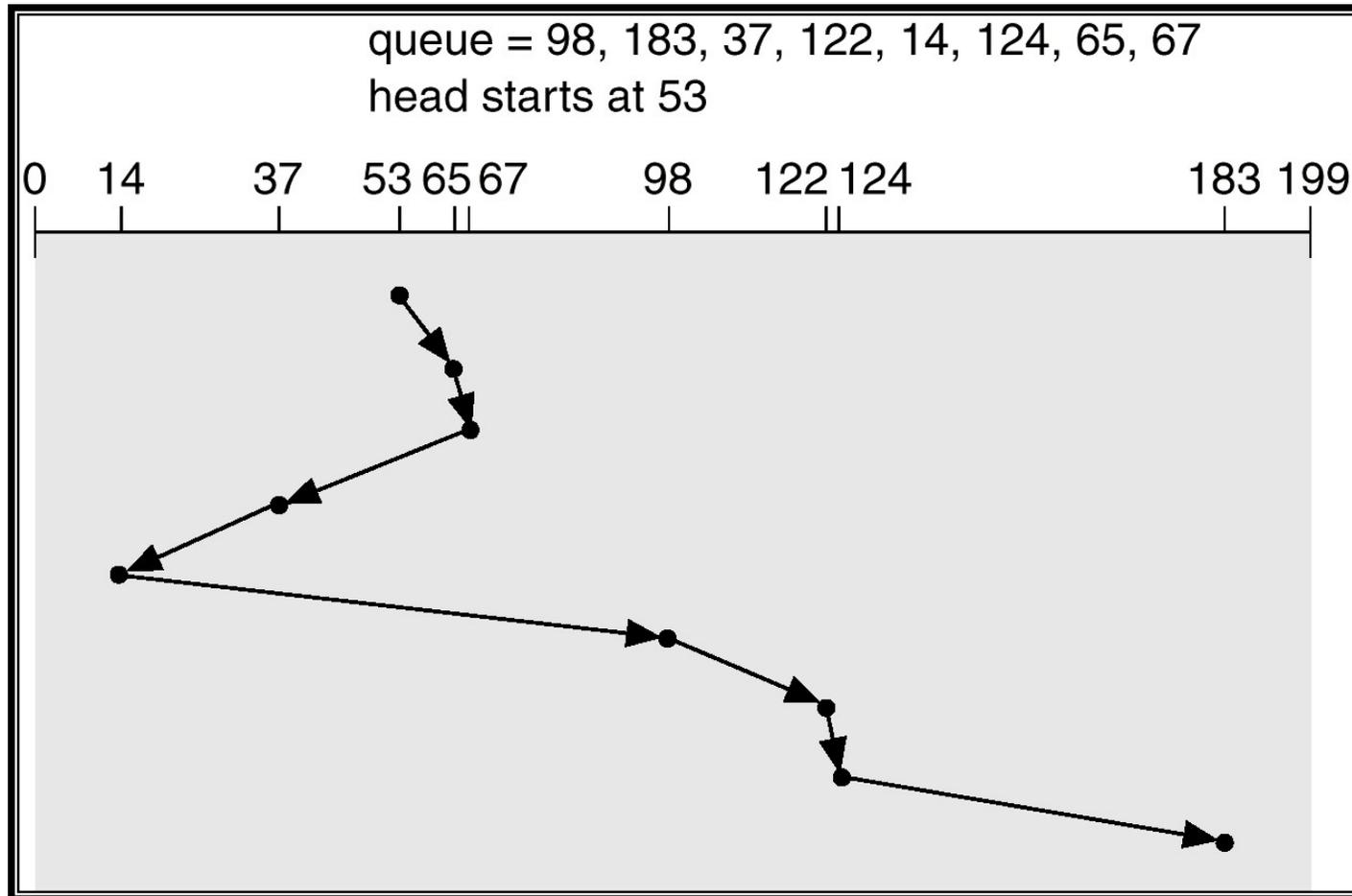


A cabeça movimenta-se sobre um total de 640 cilindros

# SSTF (shortest seektime first)

- Selecciona o pedido que corresponde à pista mais próxima da posição corrente da cabeça.
- SSTF pode causar esperas muitas longas a alguns pedidos.

# SSTF (Cont.)



No exemplo o total de movimentos da cabeça é 236 cilindros

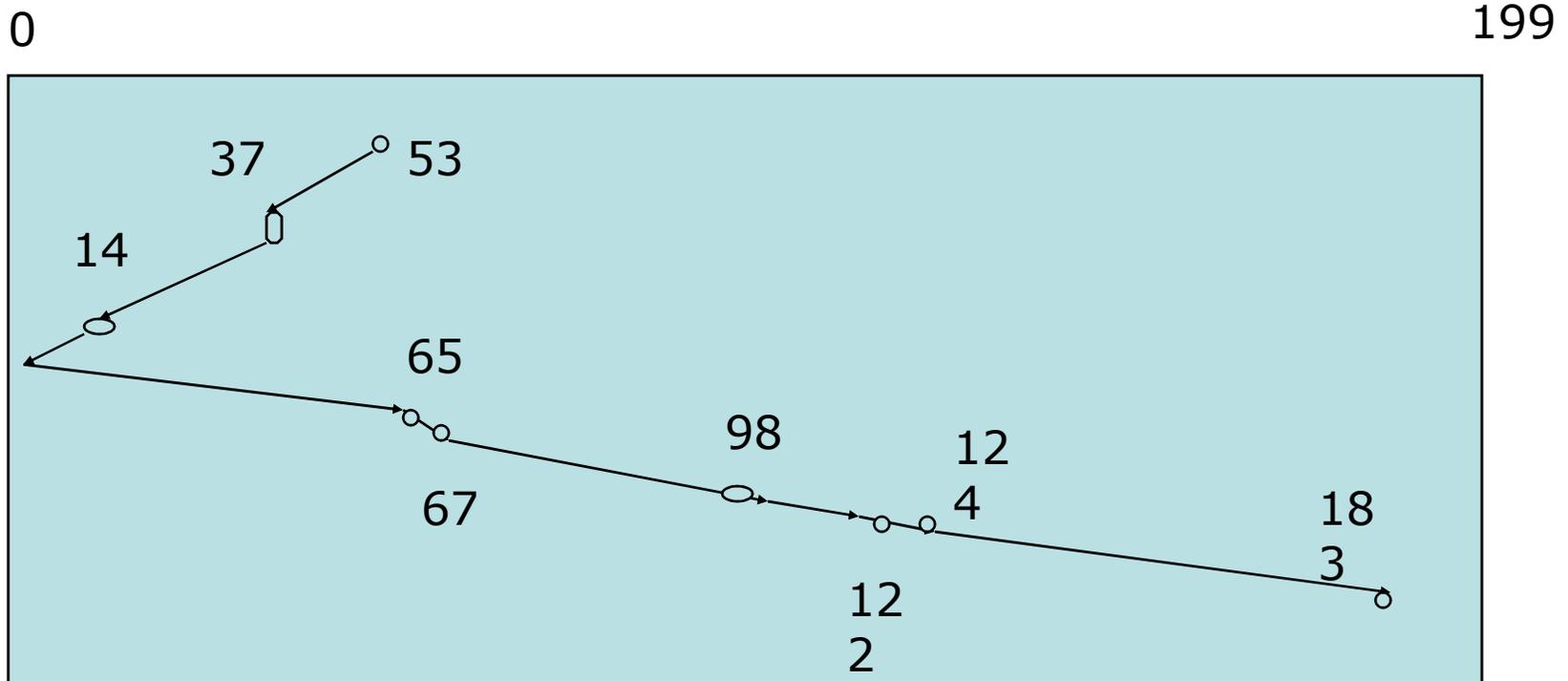
# SCAN

- A cabeça começa num extremo do disco e move-se em direcção à outra ponta, executando pedidos até chegar à outra ponta do disco, em que o movimento da cabeça inverte o sentido.
- Chamado às vezes *algoritmo do elevador*.

# SCAN (Cont.)

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

Cabeça = pista 53



- No exemplo, o total de movimentos da cabeça é de 208 cilindros.

# Outras optimizações

- Prioridade à reordenação feita pelo controlador do disco (sabe a posição da cabeça, sabe onde estão os blocos)
- SO pode fazer fusão (merging) de pedidos contíguos.
- Não fazer o acesso imediatamente, aguardar algum tempo para ver se aparecem pedidos para blocos contíguos
- Prioridade às leituras