

# *Fundamentos de Sistemas de Operação*

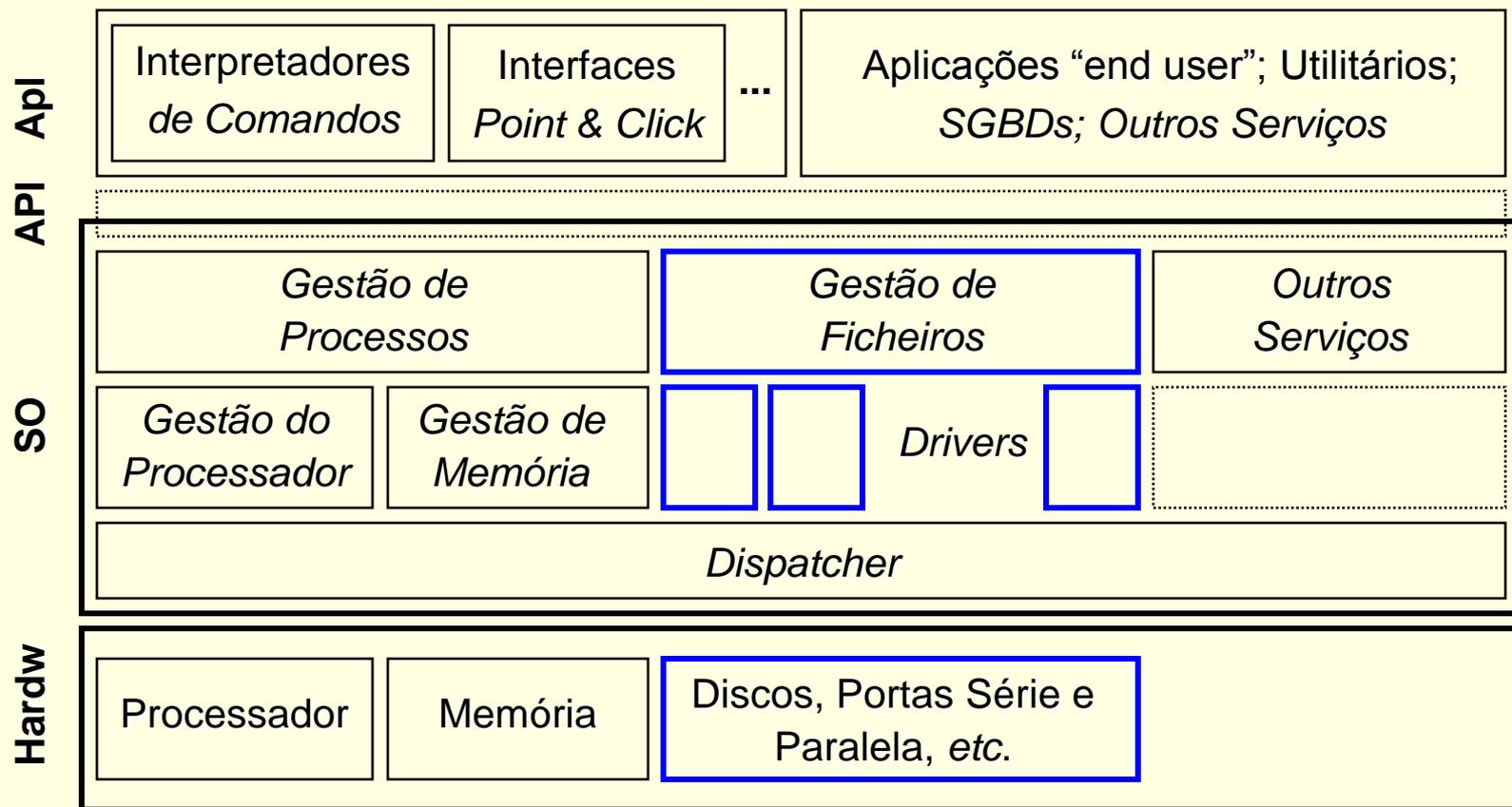
---

*Unix Windows NT Netware MacOS DOS/VS Vax/VMS  
Linux Solaris HP/UX AIX Mach Chorus*

## *Parte II: Gestão de Ficheiros*

### a) Conceitos [Revisão]

# O Sistema de Ficheiros



# *Ficheiro: o Conceito*

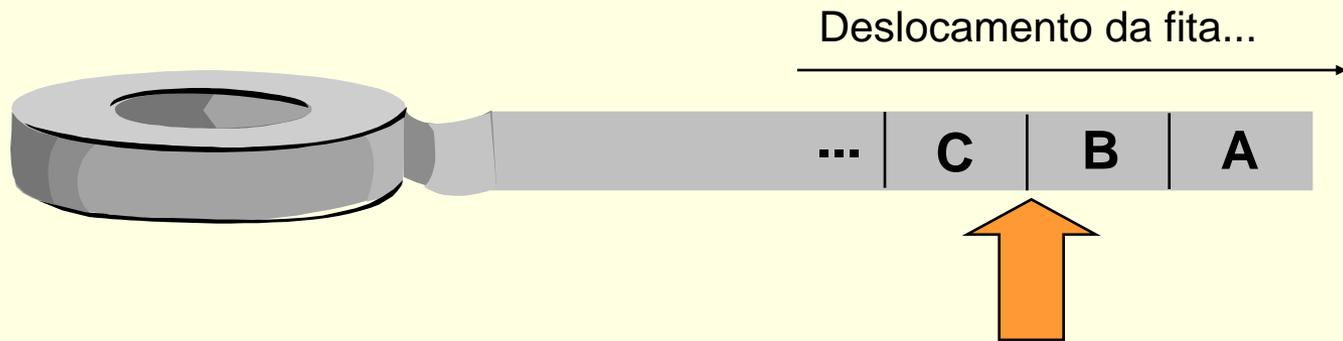
- O conceito de Ficheiro
  - O conceito pode ser simples, mas a sua implementação pelos diferentes SO's é muito diversa
  - Uma definição sucinta:
    - Entidade, referenciável por um <nome>, usada para armazenar dados, de uma forma persistente
    - Estrutura de dados, armazena 0 ... n ocorrências de um dado tipo de Registo
  - Operações elementares:
    - Criar ou Apagar um ficheiro; Abrir ou Fechar um ficheiro
    - Ler ou Escrever um “registo” num ficheiro

# Ficheiro: o Conceito

- O conceito de Ficheiro (continuação)
  - Foi fortemente influenciado pelas “necessidades” das Linguagens de Programação (e influenciou-as):
    - Ficheiros sequenciais, directos e indexados
    - Ficheiros “de texto”
    - Registos de comprimento fixo e variável
  - O “conceito minimalista” de ficheiro:
    - *Byte Stream*: um ficheiro é uma sequência não-estruturada de *bytes* - ou “quem o escreveu que o leia”

# Ficheiro: o Conceito

- O conceito de Ficheiro (continuação)
  - O *file pointer* (emulação da noção de cabeça de leitura)



- “Registos” lidos: A, B
- Próximo “registo” a ser lido: C

# File Pointer: o Conceito

- O conceito de *File Pointer*

- O ficheiro “está imóvel” e o *file pointer* desloca-se

Deslocamento do *file pointer*



Posição corrente ...  
do *File Pointer*

- “Registos” lidos: A, B
- Próximo “registo” a ser lido: C

# Ficheiro: Organização e Acesso

## ■ Organização Sequencial

- Os dados estão registados em posições fisicamente contíguas, e não é possível aceder à posição  $n$  sem ter “passado por cima” das  $n-1$  anteriores...

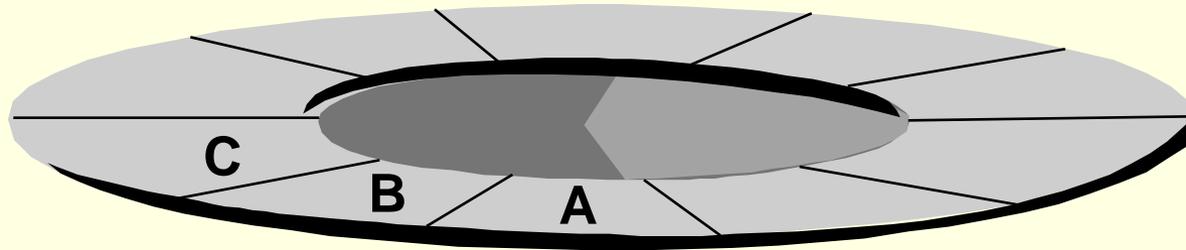


Numa “fita magnética” não é possível organizar os dados de outra forma...  
o **acesso** tem de ser sequencial

# Ficheiro: Organização e Acesso

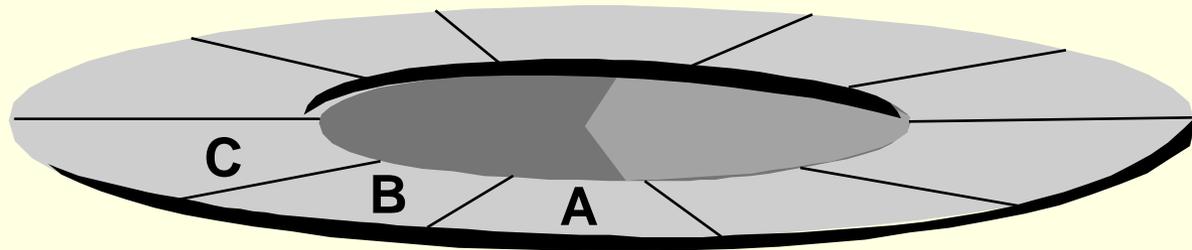
## ■ Organização Directa

- O dispositivo permite o acesso directo aos dados - *i.e.*, é possível aceder-se à posição  $n$  sem ter “passado por cima” das  $n-1$  anteriores...



# Ficheiro: Organização e Acesso

- Organização Directa: Acesso Directo vs. Sequencial
  - Se dispositivo permite o acesso directo aos dados, então é possível aceder-se directamente a um “registo” - desde que o Sistema de Ficheiros saiba como “calcular” a posição física correspondente ao item desejado...



- Claro que continua a ser possível o acesso sequencial...

# O File Pointer e os Métodos de Acesso

## ■ O File Pointer e o Acesso Directo

- Uma operação para deslocar o *file pointer* sem contudo transferir informação:

Seek ( *file pointer*, posição )



Nova  
posição do  
*File  
Pointer*

Posição  
anterior do  
*File  
Pointer*

- “Registos” lidos: nenhum
- Próximo “registo” a ser lido: C

# Acesso a Ficheiros: Primitivas

- Open
  - Associa um ficheiro a um identificador que será utilizado em todas as outras primitivas;
  - Posiciona o *File Pointer* no início ou, se este for aberto para acrescentar (*append*), no fim do ficheiro;
- Close
  - Dissocia o ficheiro do identificador; este não pode mais ser usado;
  - Garante que, se o ficheiro não está aberto por outros processos, toda a informação que possa existir em *buffers* é despejada (*flushed*) no suporte físico.

# Acesso a Ficheiros: Primitivas

## ■ Read/Write

- Lê/Escreve um objecto (um registo, um certo número de *bytes*) a partir da posição corrente do *file pointer*, actualizando-o depois

## ■ Seek

- Re-posiciona o file pointer no local indicado:
  - Princípio ou fim do ficheiro, ou
  - Deslocamento (*offset*) a partir de um ponto de referência

# Sistema de Ficheiros: a Implementação

- A implementação...
  - O ficheiro nos Sistemas de Ficheiros:
    - Regras para o nome (quantos e que caracteres, *etc.*)
    - Tipos de ficheiros (Existem? Quantos? São usados pelo SO ou apenas servem para os utilizadores “verem”?)
    - Extensões (sim ou não?; se sim, obrigatórias ou opcionais?)
  - O “ficheiro minimalista” vs. modelos disponibilizados pelas Linguagens de Programação:
    - Se o Sistema de Ficheiros não oferece os “tipos de ficheiros” necessários para uma dada Linguagem, então estes serão implementados numa Biblioteca de Suporte à Execução (a partir dos “tipos básicos”) a ligar (*link*) aos programas que são escritos nessa linguagem.