

# Sistemas de Ficheiros

Paulo Sérgio Almeida

Grupo de Sistemas Distribuídos  
Departamento de Informática  
Universidade do Minho

2005/2006



# Ficheiro

- Unidade lógica de armazenamento
- Espaço de endereçamento lógico contíguo
- Mapeado pelo SO em dispositivos de armazenamento
- Armazenamento persistente
- Pode conter programas ou dados
- Dados podem ser numéricos, texto, binários, ...



# Estrutura de ficheiros

- Ficheiros podem não ser estruturados: sequência de bytes
- Podem ter estruturados por records:
  - linhas
  - tamanho fixo
  - tamanho variável
- Podem ter estrutura complexa:
  - documento formatado
  - programa recolocável
- Estruturas complexas podem ser simuladas sobre sequências de bytes
- Quem decide estrutura:
  - programas ← o mais versátil
  - sistema operativo



# Atributos de ficheiros

## Exemplos de atributos

- **Nome**: identificador em forma legível por pessoas
- **Identificador**: etiqueta que identifica o ficheiro no sistema de ficheiros
- **Tipo**: em sistemas que suportam tipos de ficheiros
- **Localização**: apontador para a localização no dispositivo
- **Tamanho**: tamanho corrente de ficheiro
- **Protecção**: controla quem pode ler, escrever, executar
- **Tempo, data**: de criação, modificação ou acesso
- Informação sobre ficheiros é guardada na estrutura de directórios



# Operações sobre ficheiros

- Um ficheiro é um tipo abstracto de dados
- Operações para:
  - criar
  - escrever
  - ler
  - apagar
  - truncar
- Operações usam apontador para posição corrente:
  - diz onde leituras e escritas se fazem
  - pode ser reposicionado
- Operações delimitadas entre
  - **abrir**: localiza o ficheiro e prepara o acesso em memória
  - **fechar**: finaliza o acesso, gravando em disco



# Directórios

- Um disco pode conter vários sistemas de ficheiros em diferentes partições
- Cada sistema de ficheiros contém um directório
- Directório: informação sobre um conjunto de ficheiros
- Conjunto de entradas que associam nomes de ficheiros à informação sobre os ficheiros
- Operações sobre directórios:
  - inserir entrada
  - apagar entrada
  - procurar por nome
  - listar entradas
  - mudar nome de ficheiro



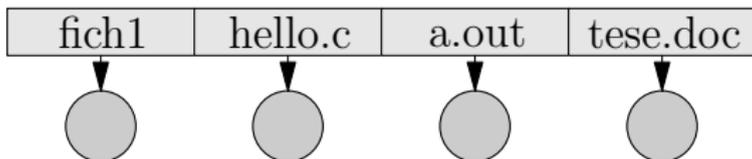
# Estrutura de directórios

- Um sistema de ficheiros pode conter muitos ficheiros
- Único directório monolítico inconveniente
- Útil estruturar directório:
  - procura eficiente
  - agrupar ficheiros logicamente
  - estruturar espaço de nomes de ficheiros
- Diferentes estruturas de directórios:
  - um nível
  - dois níveis
  - em árvore
  - grafo acíclico
  - grafo genérico



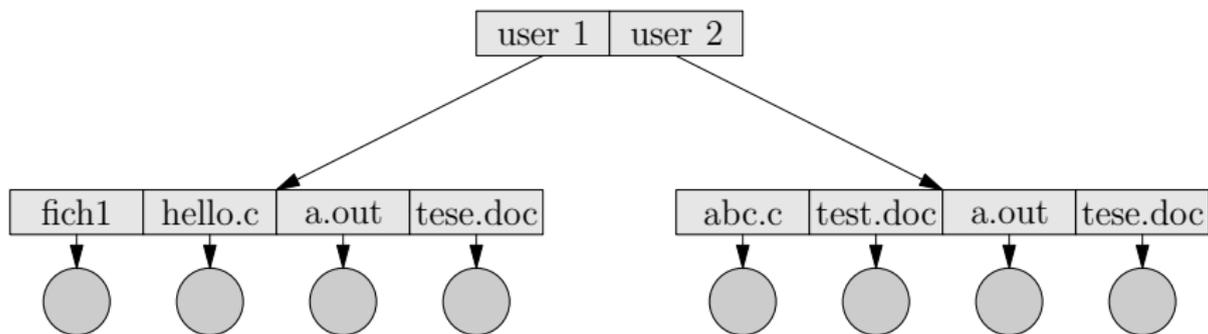
# Directório de um nível

- Um único directório usado para o sistema de ficheiros
- Problema: espaço de nomes único
- Problema: não permite agrupamento



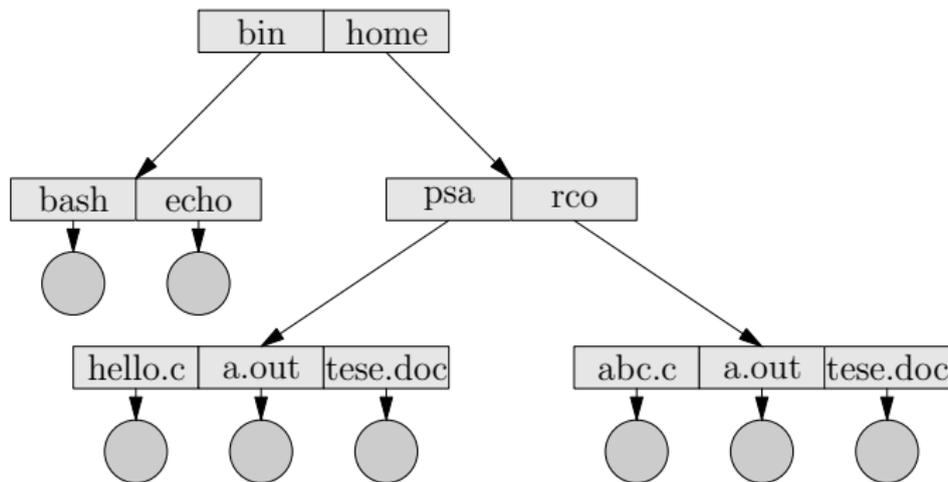
# Directório de dois níveis

- Atribuir directório para cada utilizador
- Diferentes utilizadores pode usar mesmos nomes de ficheiros
- Procura mais eficiente
- Não permite agrupamento



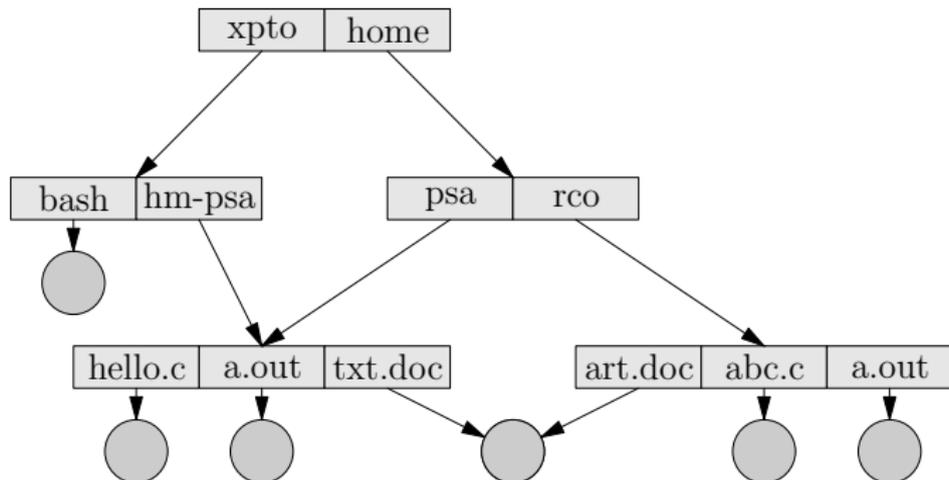
# Árvore de directórios

- Procura eficiente
- Possibilidade de agrupar ficheiros
- Espaço de nomes estruturado
- Noção de directório corrente



# Grafos acíclicos

- Permite partilha de ficheiros e sub-directórios
- **Aliasing**: vários nomes para a mesma entidade
- Obtido com o conceito de **link**
- Cópia de ficheiro versus vários nomes para o mesmo ficheiro



# Alocação de espaço em disco

- Espaço em disco dividido em blocos
- Vários blocos por ficheiro
- Vários métodos para atribuir blocos a ficheiros:
  - contígua
  - lista ligada
  - indexada



# Contígua

- Cada ficheiro ocupa conjunto contíguo de blocos
- Simples: necessário guardar bloco inicial e tamanho
- Acesso eficiente: pouco deslocamento cabeça do disco
- Permite acesso aleatório a ficheiros
- Desperdício de espaço: fragmentação externa
- Ficheiros não podem crescer ou grande fragmentação interna
- Compactação periódica para eliminar fragmentação; custosa



# Lista ligada

- Ficheiro constituído por lista ligada de blocos
- Blocos espalhados pelo disco
- Acesso pouco eficiente: deslocamento cabeça do disco
- Viável para acesso sequencial; acesso aleatório muito ineficiente
- Bom uso do espaço: apenas pequena fragmentação interna
- Ficheiros podem crescer sem problema



# File allocation table (FAT)

- Variante de lista ligada
- Tabela separada com uma entrada por bloco do disco
- Cada ficheiro é representado por uma lista ligada na FAT
- Pode ser feita cache da FAT para aumentar eficiência no acesso
- FAT usada pelo MS-DOS e OS2



# Alocação indexada

- Cada ficheiro tem um **index block**
- Contém apontadores para blocos; análogo a tabela de páginas
- Acesso aleatório a ficheiros
- Ausência de fragmentação externa
- Problema: tamanho do index block
  - grande – desperdício espaço para ficheiros pequenos
  - pequeno – problema: como guardar ficheiros grandes



# Alocação indexada

Como suportar ficheiros grandes:

- lista ligada de index blocks
- indexação multi-nível: hierarquia de index blocks
- esquema misto (e.g. inodes em UNIX):
  - primeiras entradas apontam para blocos;
  - uma entrada para index block;
  - uma entrada para hierarquia de index blocks com 2 níveis;
  - uma entrada para hierarquia de index blocks com 3 níveis;

