

Interface de Sistemas de Ficheiros

Paulo Sérgio Almeida

Grupo de Sistemas Distribuídos
Departamento de Informática
Universidade do Minho



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:
 - sistema operativo
 - programas ← o mais versátil



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



Ficheiros abertos

- O sistema operativo mantém uma tabela por processo e uma tabela do sistema relativamente a ficheiros abertos
- A tabela por processo mantém, para cada ficheiro aberto:
 - **file pointer**: apontador para a posição da última leitura/escrita
 - **direitos de acesso**: ao ficheiro pelo processo
 - **apontador para a tabela do sistema**: relativa ao ficheiro aberto
- A tabela do sistema mantém:
 - **tamanho**: do ficheiro
 - **datas**: em que o ficheiro foi acedido
 - **localização em disco**: permite saber onde está o ficheiro em disco, sem aceder ao disco
 - **contador de opens**: mantém número de vezes que o ficheiro foi aberto e ainda não fechado; quando chega a zero o SO pode remover entrada



Locking de ficheiros

- SOs oferecem interface para fazer **locking** de ficheiros
- Permite controlar o acesso por vários processos a ficheiros
- E.g. útil para **log files**
- Um processo obtém o lock; enquanto o detiver os outros ficheiros são impedidos de aceder ao ficheiro
- Pode ser especificada zona: e.g.com posição e tamanho
- Podem ser de dois tipos:
 - **mandatory**: o sistema impede o acesso a ficheiro com lock, mesmo que não seja programado para verificar locks
 - **advisory**: processo é impedido de aceder apenas se usar a interface de locking



Tipos de ficheiros

- Vários tipos de ficheiros: executável, objecto, código fonte, scripts, texto, biblioteca, impressão, arquivo, ...
- Tipos podem ser reconhecidos pelo SO ou apenas aplicações
- Alguns SOs reconhecem tipo por extensão: e.g. .com, .exe, .bat em MS-DOS são executáveis
- Extensão pode ser apenas sugestiva para a aplicação
- Alguns SOs reconhecem relações entre ficheiros; e.g.:
 - TOPS-20 recompila source se modificada aquando tentativa de correr executável
 - Mac OS X associa a ficheiro o criador; em double-click invoca criador
- Pode ser associado a cada ficheiro um **magic number**: sequência de bytes no início do ficheiro que determina o seu tipo



Métodos de acesso

- Acesso sequencial:
 - acesso começa no início e vai até ao fim do ficheiro
 - operações de read ou write avançam o file pointer
 - pode ser feito **rewind** e eventualmente avançar ou recuar n posições
 - funciona bem em todos os dispositivos
 - o mais frequente
- Acesso aleatório:
 - ficheiro constituído por records
 - records podem ser acedidos por qualquer ordem
 - usado e.g. em bases de dados
 - não apropriado para dispositivos de acesso sequencial; e.g fita
- Acesso sequencial pode ser simulado eficientemente com acesso aleatório; o contrário não



Acesso por índice

- Outros métodos podem ser feitos à custa de acesso aleatório
- Ficheiro índice:
 - ordenado, com nome do primeiro record de um bloco
 - contém apontadores para blocos
 - blocos contém records
 - ficheiro índice é guardado em memória
 - pode ser feita pesquisa binária no ficheiro índice
 - podem ser feitas hierarquias de índices



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



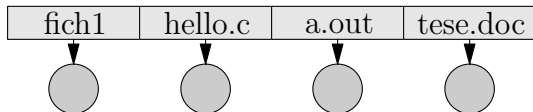
Estrutura de directórios

- 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



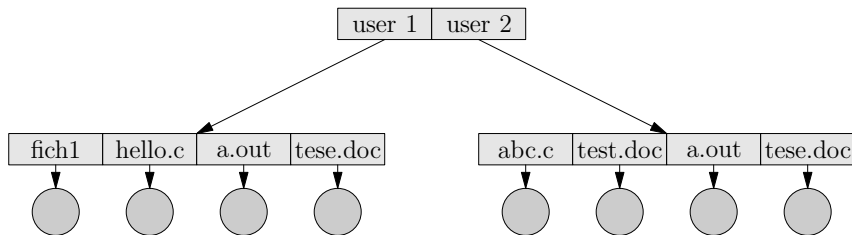
Directório de um nível

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



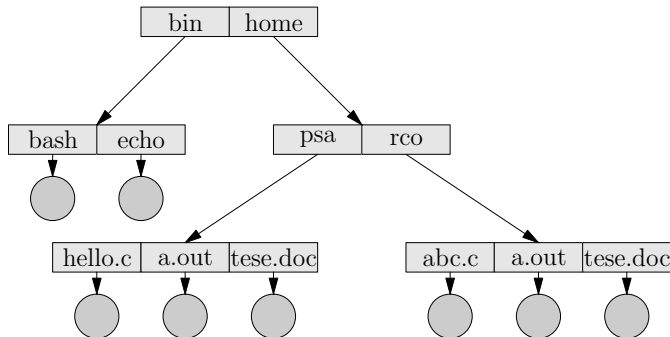
Directório de dois níveis

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



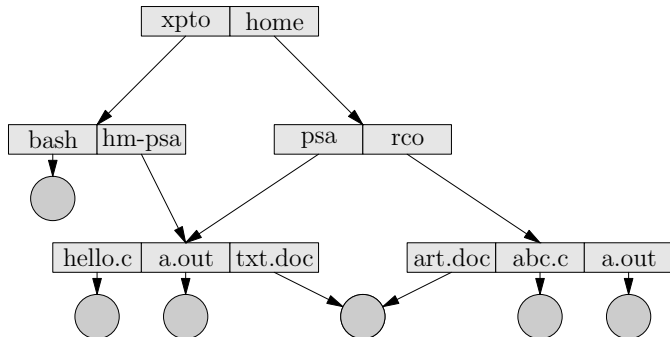
Árvore de directórios

- 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



Montar sistemas de ficheiros

- Um sistema de ficheiro necessita de ser montado (**mounted**) para ser acedido
- É escolhido um **mount point**: um local da árvore de directórios onde a raíz fica



Partilha de ficheiros

- Partilha é desejável em sistemas multi-utilizador
- Necessário implementar esquema de protecção
- Identificação de utilizadores é usada:
 - **user IDs** permitem protecção por utilizador
 - **group IDs** permitem protecção por grupo de utilizadores
- Em sistemas distribuídos, ficheiros são partilhados pela rede
- NFS (Network File System) é muito usado para partilhar ficheiros em sistemas distribuídos



Semântica de coerência

- Quando processos partilham ficheiros, escritas podem ou não ser vistas logo por outro processo; nomeadamente em sistemas de ficheiros distribuídos
- Semântica do UNIX:
 - escritas são visíveis imediatamente por outros utilizadores/processos que tenham o ficheiro aberto e façam read
 - um modo de partilha permite partilhar o file pointer
- Semântica de sessão; e.g em AFS (Andrew File System):
 - **sessão**: série de acessos entre open e close
 - escritas não são imediatamente vistas por outros utilizadores/processos que tenham o ficheiro aberto
 - escritas numa sessão só são vistas em sessões que comecem depois do fim daquela sessão



Protecção

- Criador de ficheiro deve controlar:
 - o que pode ser feito
 - por quem pode ser feito
- Tipos de acesso:
 - leitura
 - escrita
 - execução
 - adicionar
 - apagar
 - listar



Mecanismos de controlo de acesso

- Por ACL (access-control list):
 - lista de utilizadores especifica quem pode aceder
 - inconveniente: lista pode ser extensa e não conhecida à partida
 - entrada que descreve directório tem que ser de tamanho variável
- Versão condensada, usada em Unix:
 - é associado um utilizador e um grupo de utilizadores ao ficheiro
 - conjuntos de utilizadores são classificados como **user**, **group**, **other**
 - são definidos três tipos de acesso: **read**, **write**, **execute**
 - acesso é controlado em 3 vezes $3 \times 3 = 9$ bits

