

Aula 9: Sistema de Arquivos Parte I

Professor(a): Virgínia Fernandes Mota

<http://www.dcc.ufmg.br/~virginiaferm>

OCS (TEORIA) - SETOR DE INFORMÁTICA



- Parte do Sistema Operacional mais visível ao usuário
- Os arquivos de um sistema computacional são manipulados por meio de chamadas (system calls) ao Sistema Operacional;

- Três importantes requisitos são considerados no armazenamento de informações:
 - Possibilidade de armazenar e recuperar uma grande quantidade de informação;
 - Informação gerada por um processo deve continuar a existir após a finalização desse processo
 - Múltiplos processos podem acessar informações de forma concorrente → Informações podem ser independentes de processos;

- Para atender a esses requisitos, informações são armazenadas em discos (ou alguma outra mídia de armazenamento) em unidades chamadas arquivos;
- Processos podem ler ou escrever em arquivos, ou ainda criar novos arquivos;
- Informações armazenadas em arquivos devem ser persistentes, ou seja, não podem ser afetadas pela criação ou finalização de um processo;

- Arquivos são manipulados pelo Sistema Operacional;
- Tarefas:
 - Estrutura de arquivos;
 - Nomes;
 - Acessos (uso);
 - Proteção;
 - Implementação;

- Usuário: Alto nível
 - Interface: como os arquivos aparecem;
 - Como arquivos são nomeados e protegidos;
 - Quais operações podem ser realizadas;
- SO: Baixo nível
 - Como arquivos são armazenados fisicamente;
 - Como arquivos são referenciados (links);

- Nomes;
- Estrutura;
- Tipos;
- Acessos;
- Atributos;
- Operações;

- Quando arquivos são criados, nomes são atribuídos a esses arquivos, os quais passam a ser referenciados por meio desses nomes;
- Tamanho: até 255 caracteres;
- Letras, números, caracteres especiais podem compor nomes de arquivos:
 - Caracteres permitidos: A-Z, a-z, 0-9, \$, %, ?, @, {, }, , ' , !, #, (,), &
 - Caracteres não permitidos: ?, *, /, \, ?, |, <, >, :

- Alguns Sistemas Operacionais são sensíveis a letras maiúsculas e minúsculas (case sensitive) e outros não;
 - UNIX (e consequentemente, Linux) é case sensitive. Ex.: exemplo.c é diferente de Exemplo.c;
 - MS-DOS não é case sensitive. Ex.: exemplo.c é o mesmo que Exemplo.c;
- Win95/Win98/WinNT/Win2000/WinXP/WinVista herdaram características do sistema de arquivos do MS-DOS;
- No entanto, WinNT/Win2000/WinXP/WinVista/Win7/Win8 possuem um sistema de arquivos próprio NTFS (New Technology File System);

- Alguns sistemas suportam uma extensão relacionada ao nome do arquivo:
- MS-DOS: 1-3 caracteres; suporta apenas uma extensão;
- UNIX:
 - Extensão pode conter mais de 3 caracteres;
 - Suporta mais de uma extensão: Ex.: exemplo.c.Z (arquivo com compressão);
 - Permite que arquivos sejam criados sem extensão;

- Uma extensão, geralmente, associa o arquivo a algum aplicativo (associação feita pelo aplicativo).
- SO pode ou não associar as extensões aos aplicativos:
 - Unix não associa;
 - Windows associa;

- Arquivos podem ser estruturados de diferentes maneiras:
 - Sequência não estruturada de bytes
 - Sequência de registros de tamanho fixo
 - Árvores de registros (tamanho variado)

- Sequência não estruturada de bytes
 - Para o SO arquivos são apenas conjuntos de bytes;
 - SO não se importa com o conteúdo do arquivo: Significado deve ser atribuído pelos programas em nível de usuário (aplicativos);
 - Vantagem → Flexibilidade: os usuários nomeiam seus arquivos como quiserem;
 - Ex.: UNIX e Windows;

- Sequência de registros de tamanho fixo, cada qual com uma estrutura interna → leitura/escrita são realizadas em registros;
 - SOs mais antigos: mainframes e cartões perfurados (80 caracteres);
 - Nenhum sistema atual utiliza esse esquema;
- Árvores de registros (tamanho variado), cada qual com um campo chave em uma posição fixa:
 - SO decide onde colocar os arquivos;
 - Usado em mainframes atuais;

- **Arquivos regulares:** são aqueles que contêm informações dos usuários;
- **Diretórios:** são arquivos responsáveis por manter a estrutura do Sistema de Arquivos;
- **Arquivos especiais de caracteres:** são aqueles relacionados com E/S e utilizados para modelar dispositivos seriais de E/S;
 - Ex.: impressora, interface de rede, terminais;
- **Arquivos especiais de bloco:** são aqueles utilizados para modelar discos;

Sistema de Arquivos - Tipos de Arquivos

- Arquivos regulares podem ser de dois tipos: ASCII e Binários
- ASCII
 - Consistem de linhas de texto;
 - Facilitam integração de arquivos;
 - Podem ser exibidos e impressos como são;
 - Podem ser editados em qualquer editor de texto;
 - Ex.: arquivos com extensão txt;
- Binários
 - Todo arquivo não ASCII;
 - Possuem uma estrutura interna conhecida pelos aplicativos que os usam;
 - Ex.: programa executável;

- SO's mais antigos ofereciam apenas acesso sequencial no disco
→ leitura em ordem byte a byte (registro a registro);
- SO's mais modernos fazem acesso aleatório aos arquivos, cujos bytes ou registros podem ser acessados em qualquer ordem;
 - Acesso feito por chave;
 - Métodos que podem ser usados para especificar onde iniciar leitura:
 - Operação *Read*: posição do arquivo em que se inicia a leitura;
 - Operação *Seek*: a operação seek estabelece a posição atual; depois de um seek, o arquivo pode ser lido sequencialmente a partir de sua posição atual. Usado no Linux e Windows.

Sistema de Arquivos - Atributos de Arquivos

- Além do nome e dos dados, todo arquivo tem outras informações associadas a ele: **atributos**
- A lista de atributos varia de SO para SO, mas com algumas categorias, como os de proteção, são comuns a todos;

Atributo	Significado
Proteção	Quem acesso o arquivo e de que maneira
Senha	Chave para acesso ao arquivo
Criador	Identificador da pessoa que criou o arquivo
Dono	Dono corrente
<i>Flag</i> de leitura	0 para leitura/escrita; 1 somente para leitura
<i>Flag</i> de oculto	0 para normal; 1 para não aparecer
<i>Flag</i> de sistema	0 para arquivos normais; 1 para arquivos do sistema
<i>Flag</i> de repositório	0 para arquivos com <i>backup</i> ; 1 para arquivos sem <i>backup</i>

Sistema de Arquivos - Atributos de Arquivos

Atributo	Significado
<i>Flag ASCII/Binary</i>	0 para arquivo ASCII; 1 para arquivo binário
<i>Flag de acesso aleatório</i>	0 para arquivo de acesso seqüencial; 1 para arquivo de acesso randômico
<i>Flag de temporário</i>	0 para normal; 1 para temporário
<i>Flag de impedido</i>	0 para arquivo desimpedido; diferente de 0 para arquivo impedido
Tamanho do registro	Número de bytes em um registro
Posição da chave	Deslocamento da chave em cada registro
Tamanho da chave	Número de bytes no campo chave (<i>key</i>)

Sistema de Arquivos - Atributos de Arquivos

Atributo	Significado
Momento da criação	Data e hora que o arquivo foi criado
Momento do último acesso	Data e hora do último acesso ao arquivo
Momento da última mudança	Data e hora da última modificação do arquivo
Tamanho	Número de bytes do arquivo
Tamanho Máximo	Número máximo de bytes que o arquivo pode ter

- Diferentes sistemas provêm diferentes operações que permitem armazenar e recuperar arquivos;
- Operações mais comuns (realizadas através de system calls):
 - Create; Delete;
 - Open; Close;
 - Read; Write; Append;
 - Seek;
 - Get attributes; Set attributes;
 - Rename;

- Alguns SO's permitem que arquivos sejam mapeados diretamente no espaço de endereçamento (virtual) de um processo em execução → acesso mais rápido;
- System Calls: Map e unmap;
- Funciona melhor em sistemas que suportam segmentação;

- **Problemas!!!!**
- Difícil prever o tamanho de arquivos de saída;
- Compartilhamento de arquivos entre diferentes processos → SO não deve permitir acesso a arquivos com dados inconsistentes;
- Arquivo pode ser maior que um segmento ou maior que o espaço virtual utilizado → mapear pequenas partes do arquivo;

- **Diretórios:** São arquivos responsáveis por manter a estrutura do Sistema de Arquivos;
- **Organização** pode ser feita das seguintes maneiras:
 - Nível único (Single-level);
 - Dois níveis (Two-level);
 - Hierárquica;

- Apenas um diretório contém todos os arquivos → diretório raiz (root directory);
- Computadores antigos utilizavam esse método, pois eram monousuários;
- Exceção: CDC 6600 - supercomputador que utilizava-se desse método, apesar de ser multiusuário;
- Vantagens: Simplicidade e Eficiência.
- Desvantagens: Sistemas multiusuários: Diferentes usuários podem criar arquivos como mesmo nome;

Sistema de Arquivos - Diretórios - Dois níveis

- Cada usuário possui um diretório privado;
- Sem conflitos de nomes de arquivos;
- Procedimento de login: identificação;
- Compartilhamento de arquivos → programas executáveis do sistema;
- Desvantagem: Usuário com muitos arquivos;

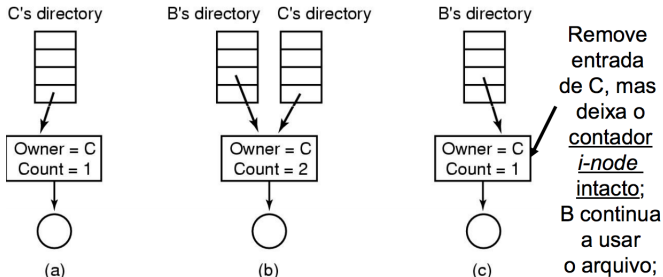
- Hierarquia de diretórios → árvores de diretórios;
- Usuários podem querer agrupar seus arquivos de maneira lógica, criando diversos diretórios que agrupam arquivos;
- Sistemas operacionais modernos utilizam esse método;
- Flexibilidade;

Sistema de Arquivos - Arquivos compartilhados

- Compartilhar arquivos é sempre conveniente, no entanto, alguns problemas são introduzidos!
- Se os diretórios tiverem endereços de disco, então deverá ser feita uma cópia dos endereços no diretório de B;
- Se B ou C adicionar blocos ao arquivo (append), os novos blocos serão relacionados somente no diretório do usuário que está fazendo a adição;
 - Mudanças não serão visíveis ao outro usuário, comprometendo o propósito do compartilhamento;
- Soluções???

- Primeira solução: os endereços de disco não estão relacionados nos diretórios, mas em uma estrutura de dados (i-node) associada ao próprio arquivo. Assim, os diretórios apontam para essa estrutura; (UNIX)
 - Ligação Estrita (hard link);
 - Problema com essa solução: o dono do arquivo que está sendo compartilhado apaga o arquivo;

Sistema de Arquivos - Arquivos compartilhados



- a) Antes da ligação;
- b) Depois da ligação;
- c) Depois de remover a entrada de C para o arquivo;

36

Ligação Estrita

- Segunda Solução: Ligação Simbólica → B se liga ao arquivo de C criando um arquivo do tipo link e inserindo esse arquivo em seu diretório;
 - Somente o dono do arquivo tem o ponteiro para o i-node;
 - O arquivo link contém apenas o caminho do arquivo ao qual ele está ligado: Assim, remoções não afetam o arquivo;
 - Problema: Sobrecarga → Geralmente um i-node extra para cada ligação simbólica;

- O método hierárquico requer métodos pelos quais os arquivos são acessados;
- Dois métodos diferentes:
 - Caminho absoluto (absolute path name): a partir da raiz;
 - Caminho relativo (relative path name): a partir do diretório determinado pelo usuário;

- Create; Delete;
- Opendir; Closedir;
- Readdir;
- Rename;
- Link (para um arquivo aparecer em mais de um diretório);
- Unlink;

Implementando o Sistema de Arquivos

- Implementação do Sistema de Arquivos deve estabelecer as formas e operações para as seguintes operações:
 - Como arquivos e diretórios são armazenados;
 - Como o espaço em disco é gerenciado;
 - Como tornar o sistema eficiente e confiável;

Implementando o Sistema de Arquivos - Layout

- Arquivos são armazenados em discos;
- Discos podem ser divididos em uma ou mais partições, com sistemas de arquivos independentes;
- Setor 0 do disco é destinado ao MBR - Master Boot Record; que é responsável pela tarefa de boot do computador;
 - MBR possui a tabela de partição, com o endereço inicial e final de cada partição;
 - BIOS lê e executa o MBR;

Implementando o Sistema de Arquivos - Layout

- Tarefas básicas do MBR (pode variar dependendo do SO):
 - 1 localizar a partição ativa;
 - 2 ler o primeiro bloco dessa partição, chamado bloco de boot (boot block);
 - 3 executar o bloco de boot;
- Layout de um Sistema de Arquivos pode variar... Mas como seria uma ideia geral?

- Armazenamento de arquivos: como os arquivos são alocados no disco;
- Diferentes técnicas são implementadas por diferentes Sistemas Operacionais:
 - Alocação contígua;
 - Alocação com lista encadeada;
 - Alocação com lista encadeada utilizando uma tabela na memória (FAT);
 - I-Nodes;
- Como são essas alocações?

Sistema de Arquivos Parte II