Curso de Engenharia de Computação
Sistemas Operacionais

Sistemas de Arquivo – Parte III

Slides da disciplina Sistemas Operacionais
Curso de Engenharia de Computação
Instituto Mauá de Tecnologia – Escola de Engenharia Mauá
Prof. Marco Antonio Furlan de Souza

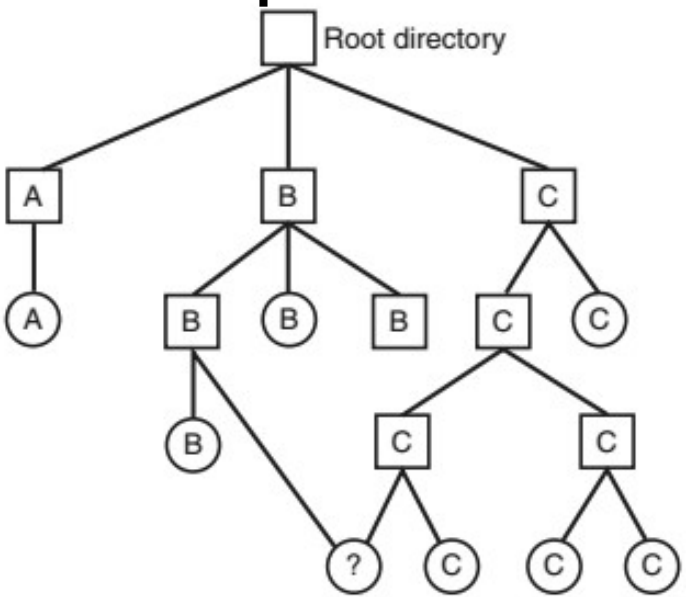
Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA

- **Arquivos compartilhados**
  - Um mesmo **arquivo** pode **aparecer simultaneamente** em **diferentes diretórios** pertencentes a **diferentes usuários**;
  - A **conexão** entre um **diretório** de **usuário** e um **arquivo compartilhado** é denominado **link**.

Neste exemplo, **um arquivo de C** está **compartilhado** em **B** (considerar que são de usuários B e C).

O **sistema** de **arquivo** agora assume uma **topologia** de **DAG** (*Direct Acyclic Graph* – Grafo Acíclico Direto) no lugar de árvore.



Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA

- **Arquivos compartilhados**
  - **Problema**
    - Como **diretórios** contém **endereços** de **arquivos**, então uma **cópia** de **endereços** de disco deverá **existir** tanto no **diretório B** (que tem o link) quanto no **diretório C** (original);
    - Se um dos **usuários** (B ou C) **altera** um **arquivo compartilhado**, as **alterações** serão **realizadas** apenas para o **usuário** que as **alterou** – isso **viola** o **princípio** de **compartimento**!

Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA

- **Arquivos compartilhados**
  - **Solução 1**
    - **Não listar** os **blocos** dos **arquivos** nos **diretórios** – mas em uma **estrutura** de dados associada com o **próprio arquivo** – usado no Unix/Linux com **i-nodes**;
  - **Solução 2**
    - Criar um **novo tipo** de **arquivo** – **link** – **contendo** apenas o **caminho** de **diretórios** do **arquivo compartilhado**. Esta é a solução **utilizada** em **links simbólicos** do Unix/Linux.

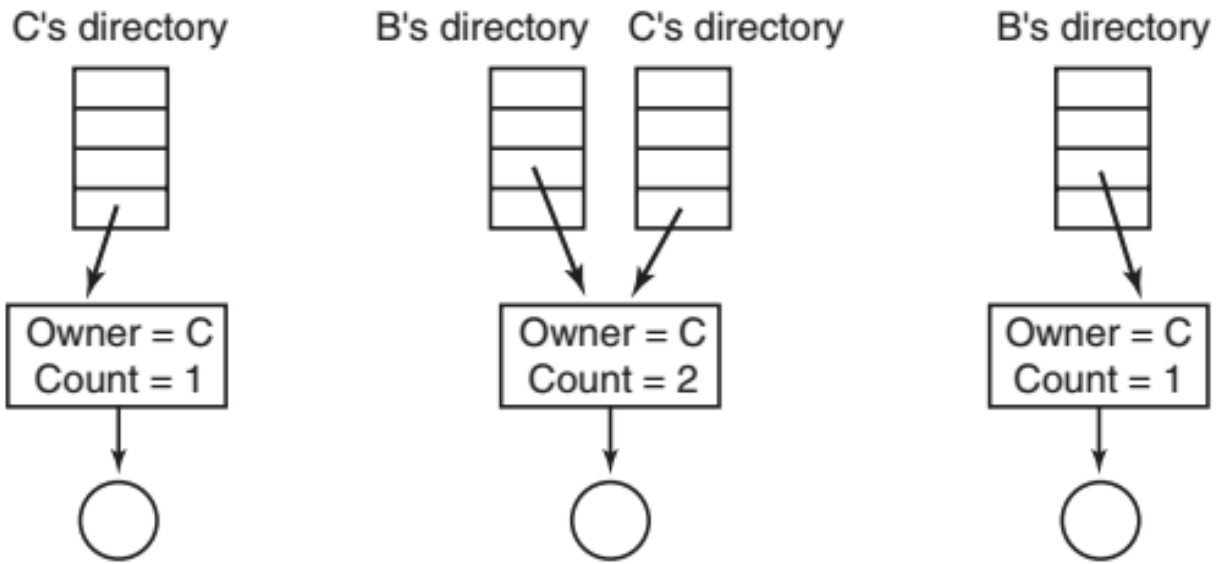
Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA

- **Arquivos compartilhados**
  - **Problemas com as soluções apresentadas**
    - Com **links**, se o **arquivo original** for **removido**, o **link** se tornará **inválido** – ao se tentar utilizá-lo, um erro deverá ser emitido – esse é o **comportamento** de **links simbólicos** do Unix/Linux ou ***soft links***;
    - No Unix/Linux existe o **conceito** de ***hard link***: quando se cria um hard link de um arquivo, uma **contagem** de **referência** é **criada** – **apenas quando** o **arquivo** tem **contagem** 1 ele **poderá** de fato **removido**!
    - **Outro problema**: quando se deseja fazer um **backup**, o **que fazer com links**? Se eles forem **copiados** no **backup**, **aumenta-se o espaço** necessário; **senão**, ao se **restaurar** o backup esses **links não existirão** – programas de backup mais inteligentes permitem configurar o que fazer.

Implementação de sistema de arquivos

- **Arquivos compartilhados**
  - **Solução de links com contagem de referência (*hard link*)**
    - **Exemplo**: depois que B criou um link para arquivo de C, C remove este arquivo.



Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA

- **Sistema de Arquivos estruturado para *log***
  - **LFS** (*Log-structured File System*) é baseado na **ideia** que conforme as **CPUs** se tornam **mais rápidas** e **memória RAM** são **maiores**, **caches** de **disco** também **aumentam significativamente**;
  - Então é **possível satisfazer** uma **boa fração** das **leituras diretamente** do **cache** do sistema de arquivos, **sem** a necessidade de realizar **acessos** a **discos**;
  - **Posteriormente**, a **maior parte** dos **acessos** a disco **serão** de **escrita** e então o **sistema** de **antecipar leituras não** trará **vantagens significativas** de desempenho;
  - Além disso, as **escritas** em **arquivos** tendem a ser **realizadas** em **agrupamentos pequenos** ($t_{escrita} + t_{busca} + t_{rotação}$), **reduzindo** o **desempenho**.

Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA

- **Sistema de Arquivos estruturado para *log***
  - A **abordagem LFS** considera todo o **disco como** um **grande** *log*;
  - **Periodicamente** e quando há uma **necessidade especial** para isso, **todas** as **escritas pendentes** sendo **buferizadas** em **memória** são **armazenadas** em um **segmento único** e que então é **escrito** ao **disco** como um **segmento contínuo** no **final** do **log**;
  - Este **segmento** pode conter **i-nodes**, **blocos** de **diretórios**, e **blocos** de **dados** – no **início** do **segmento** existe um **sumário** que **indica** o que **existe** no **segmento**;
  - A ideia é que o **tamanho** do **segmento** a ser escrito **ocupe** toda a **largura** de **banda** do **disco** na **escrita** (em MB/s).

Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA

- **Sistema de Arquivos estruturado para *log***
  - A organização dos i**-nodes** é a mesma, porém eles estão **espalhados** pelo **log**;
  - Um **mapa** de **i-node**, **indexado** pelo **número** do **i-node**, é **mantido**. A **entrada i** no **mapa aponta** para o **i-node i** no **disco**. O mapa é **mantido** em **disco** e **também** no **cache**, de modo que partes utilizadas com mais frequência estejam na memória a maior parte do tempo;
  - Ao se **abrir** um **arquivo**, consulta-se o **mapa** para **localizar** o **i-node** do **arquivo**. **Localizado** o i-node, seus **blocos** são **recuperados** dentro de **algum segmento** do log;
  - Para **blocos** que foram **removidos** do log, o **LFS** conta com um **processo** de **limpeza** que varre o **log** de modo **circular** e então o **compacta** o **segmento** – o **disco** se torna um **grande log circular** (mas não trivial).

Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA

- **Sistema de Arquivos com Journaling**
  - Uma abordagem que **utiliza log** para aumentar a robustez de um **sistema** de **arquivos** (sem alterá-lo como no LFS) **existentes** é chamada de **journaling file systems** e é empregada no Linux pelos sistemas ext3, ext4, ReiserFS, no Microsoft NTFS e também no MacOS;
  - A ideia é **manter** um **log** do que o **sistema** está para fazer de modo que, se o **sistema falha antes** da **realização** de **uma tarefa**, **após** o **boot** ele pode **recuperar do log** a **tarefa** que **estava** para ser **executada** antes da **falha** ocorrer e então se **recuperar**.

Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA

- **Sistema de Arquivos com Journaling**
  - **Exemplo**: quando se **remove** um **arquivo** no Unix/Linux ocorrem as seguintes **operações**:
    1) **Remover** o **arquivo** de seu **diretório**;
    2) **Liberar** o **i-node correspondente** do arquivo para o **grupo** de **i-nodes livres**;
    3) **Retornar** ao **disco** todos os **blocos** do **arquivo** ao **grupo** de **blocos** de disco **livres**.
  - O que acontece se uma falha provoca o término de operação do sistema operacional em cada um dos passos acima?

Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA
MAUÁ

- **Sistema de Arquivos com Journaling**
  - Um **sistema** de **arquivos** com **journaling** primeiro **adiciona** uma **entrada** no logo **com** as **três ações** a serem completadas;
  - Esta **entrada** então é **escrita** no **disco** (e **verificada** que foi escrita corretamente);
  - Apenas **quando** a **entrada** no **log** foi **escrita** é que as **operações** são **iniciadas**;
  - Quando as **operações** são **terminadas** com sucesso, a **entrada** no **log** é **apagada**;
  - **Se** o **sistema operacional falha** (*crash*), no processo de recuperação **após** o ***boot*** o **sistema** de **arquivos verifica** o **log** para **determinar** se **existe** alguma **operação pendente**;
  - Se **existir operações pendentes**, elas podem ser **reexecutadas até** que o **arquivo** seja corretamente **removido**.

Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA
MAUÁ

- **Sistema de Arquivos com Journaling**
  - As **operações** a serem **logadas** devem ser **idempotentes** – podem ser **repetidas quantas vezes necessário** sem problema – cabe ao sistema de arquivo com journaling organizar as operações no log de modo que elas sejam idempotentes;
  - Para aumentar a confiabilidade do sistema, pode-se adicionar o conceito de **transação atômica**, como em bancos de dados – um **grupo** de **ações** é **completamente executado ou nenhuma** das **ações** é **executada** (blocos begin transaction / end transaction com commit / rollback).

Implementação de sistema de arquivos

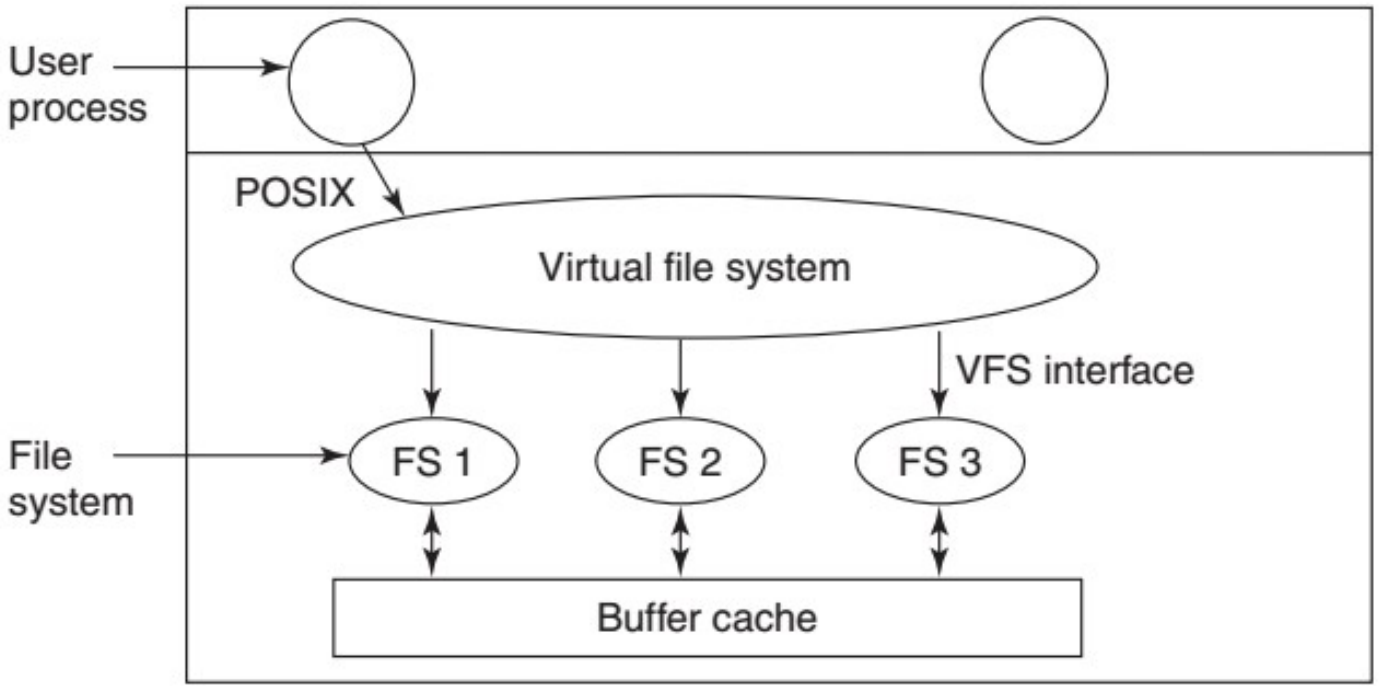
INSTITUTO MAUÁ DE TECNOLOGIA

- **Sistema de Arquivos Virtual**
  - Para pode **operar** com **diversos meios** de **armazenamento** contendo **sistemas** de **arquivos distintos**, a maior parte dos **sistemas UNIX/Linux** utilizam o conceito de **VFS** (*virtual file system* **– sistema de arquivos virtual**) para integrá-los em uma estrutura comum;
  - A i**deia-chave** é **abstrair** a **parte** do **sistema de arquivo** que é **comum** a **todos** os **sistemas** de **arquivos** e **colocar** o **código correspondente** em uma **camada separada** que **invoca** os **sistemas** de **arquivo concretos**, que de fato gerenciam os dados.

Implementação de sistema de arquivos

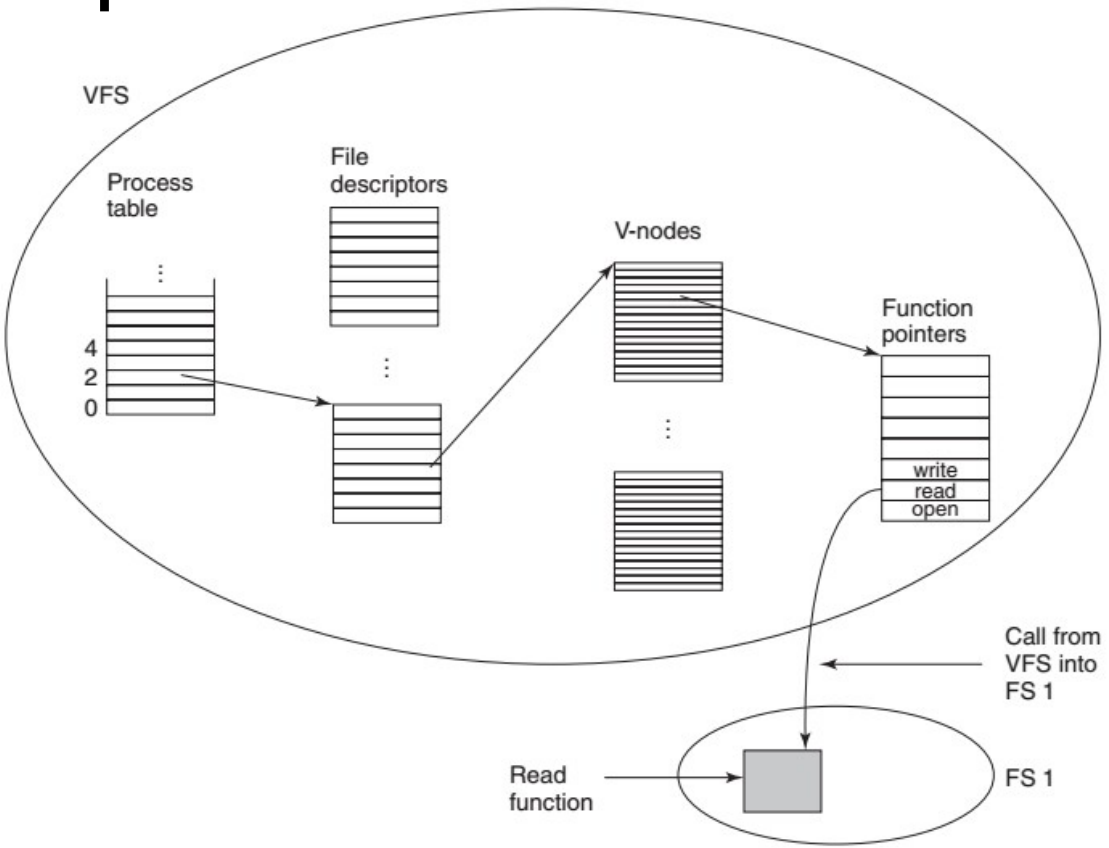
INSTITUTO MAUÁ DE TECNOLOGIA

- **Sistema de Arquivos Virtual**



Implementação de sistema de arquivos

- **Sistema de Arquivos Virtual**



Implementação de sistema de arquivos

INSTITUTO MAUÁ DE TECNOLOGIA

- **Sistema de Arquivos Virtual**
  - O VFS possui **dois tipos** de **interface**: uma para os **processos** de **usuário** e outra para os **sistemas** de **arquivo concretos**;
  - Existem **sistemas** tais como o **NFS** (network file system) que se trata de uma **interface VFS** onde os **arquivos** não estão em uma partição do disco e sim em alguma **máquina** de **rede**.

Referências bibliográficas

TANENBAUM, Andrew S. **Sistemas operacionais modernos**. 3. ed. São Paulo: Pearson, 2013. 653 p.