

# **Sistemas Computacionais Distribuídos**

**Prof. Marcos José Santana  
SSC-ICMC-USP**

**São Carlos, 2008**

# **Grupo de Sistemas Distribuídos e Programação Concorrente**

**Departamento de Sistemas  
de Computação - SSC**

# **Sistemas Computacionais Distribuídos**

**Aspectos Relevantes para a Organização  
de Servidores de Arquivos para uso em  
Sistemas Distribuídos**

# Conteúdo

- ◆ Aspectos Relevantes para a Organização de Servidores de Arquivos para uso em Sistemas Distribuídos
- ◆ Primitivas para Comunicação entre Servidores de Arquivos e Clientes
- ◆ Unidade de Dados para read-data e write-data
- ◆ Atualização de Arquivos
- ◆ Caso de Versões
- ◆ Serviço de Diretório
- ◆ Divisão de Responsabilidades
- ◆ Transações

# Aspectos Relevantes para a organização de Servidores de Arquivos para uso em Sistemas Distribuídos

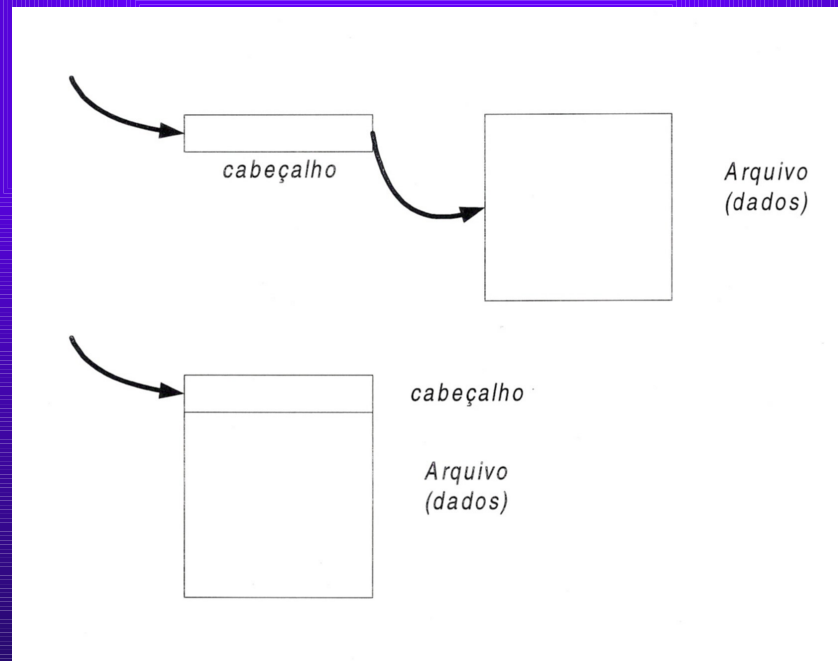
- ◆ **Arquivo:** é uma seqüência ordenada de “bytes” que pode ser “acessada” para escrita e para leitura
  - **“Header”** ou cabeçalho (informações sobre o arquivo):
    - data de criação;
    - tamanho;
    - ultima modificação;
    - proteção;
    - etc.
  - **Dados:** informação armazenada

# Aspectos Relevantes para a organização de Servidores de Arquivos para uso em Sistemas Distribuídos

- ◆ **Dados:** são organizados em blocos (não necessariamente contíguos);
- ◆ **Blocos** → número inteiro de setores do disco;
- ◆ **Estrutura dos blocos** → árvores com múltiplos níveis:
  - 1 ou mais níveis, dependendo do arquivo (ex. UNIX);
  - Conjunto de blocos de tamanho fixo (ex. TRICE).

# Aspectos Relevantes para a organização de Servidores de Arquivos para uso em Sistemas Distribuídos

- ♦ Obs: o cabeçalho é muitas vezes armazenado em uma estrutura diferente.



# Aspectos Relevantes para a organização de Servidores de Arquivos para uso em Sistemas Distribuídos

- ◆ **FID: Todo arquivo recebe um identificador (File Identifier) que deve ser único.**
- ◆ **O FID é gerado pelo sistema (servidor de arquivos) quando o arquivo é criado.**



# Aspectos Relevantes para a organização de Servidores de Arquivos para uso em Sistemas Distribuídos

- ◆ **Técnicas para geração de FID (há diversas)**
  - **Dependente da posição de armazenamento:**
    - fácil geração;
    - baixa flexibilidade;
    - arquivo fica preso ao meio.
  - **Independente da posição de armazenamento:**
    - maior dificuldade para geração;
    - alta flexibilidade → mobilidade do arquivo é fácil.

# Aspectos Relevantes para a organização de Servidores de Arquivos para uso em Sistemas Distribuídos

## ◆ Exemplo de técnicas para geração de FID:

- Dependente da localização:
  - Não é necessário tabela;
  - Ex: (CFS) endereço da raiz  $\langle ER \rangle$   
número aleatório  $\langle NA \rangle$

**FID**  $\rightarrow \langle \langle ER \rangle \langle NA \rangle \rangle$

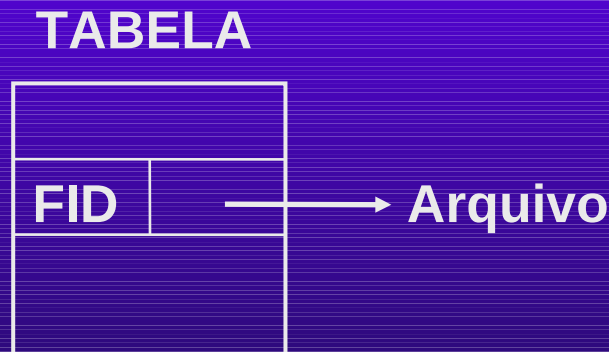
# Aspectos Relevantes para a organização de Servidores de Arquivos para uso em Sistemas Distribuídos

- ◆ Exemplo de técnicas para geração de FID:
  - Parcialmente dependente:
    - Precisa de tabela;
    - Ex: (XDFS)  $\Rightarrow$  identificador do disco  $\langle \text{IDK} \rangle$   
número aleatório  $\langle \text{NA} \rangle$
    - $\text{FID} = \langle \langle \text{IDK} \rangle \langle \text{NA} \rangle \rangle$

Isto força alterar o FID se o arquivo for movido de um disco para outro. Mas fica livre para ser re-locado dentro do mesmo disco.

# Aspectos Relevantes para a organização de Servidores de Arquivos para uso em Sistemas Distribuídos

- ◆ Exemplo de técnicas para geração de FID:
  - Independente:
    - Precisa de tabela;
    - Ex: (SWALLOW)
    - $FID = \langle NA \rangle$



# Primitivas para Comunicação entre Servidores de Arquivos e Clientes

- ◆ **create-file:** S.A. cria um arquivo e retorna ao cliente o seu FID;
- ◆ **read-data:** S.A. envia ao cliente uma unidade de “dados”;
- ◆ **delete-file:** S.A. apaga (logicamente) o arquivo indicado pelo cliente;
- ◆ **write-data:** o cliente envia ao S.A. uma unidade de dados e o S.A. escreve os dados no disco.

**Obs:** read-data; write-data e delete-file  $\Rightarrow$  requerem o FID.

# Unidade de Dados para read-data e write-data

## ◆ Arquivo:

- Solução mais simples;
- Arquivo completo é lido (ou escrito) na memória;
- Problemas de desempenho com arquivos grandes;
- Problemas de alocação podem ocorrer! (Na E.T.)

## ◆ Página (numero inteiro de blocos):

- Remove o problema de arquivos longos;
- Facilita a implementação de Memória Virtual;
- Algumas implementações permitem “sub-páginas” (ex. XDFS);

# Unidade de Dados para read-data e write-data

## ◆ Sub-parte do arquivo:

- Caso mais geral;
- Mais complexo de gerenciar;
- Permite que uma “parte arbitrária do arquivo seja lida/escrita”.



- Ex: CFS
  - Cliente define a palavra de início (*offset*) e o número de bytes (palavras).

# Atualização de Arquivos

- ◆ Operação de importância vital no S.A.
- ◆ Dependendo da operação tem-se:
  - Arquivos robustos → sobrevivem a falhas do dispositivo de armazenamento e decaimento do meio magnético;
  - Arquivos “recuperáveis” → o arquivo pode retornar ao seu último estado consistente se uma falha ocorrer;
  - Arquivos “comuns” (ordinários) → não se baseiam em nenhuma técnica que garanta integridade.

**Ex: no CFS ⇒ podem-se definir arquivos comuns e recuperáveis.**



# Caso de Versões

## ◆ Versão única:

- Caso mais simples;
- Caso menos confiável;
- Economiza espaço em disco;
- Ex: maior parte.

## ◆ Múltiplas versões:

- Mais completo de se gerenciar;
- Maior confiabilidade;
- Ex: TRICE.

# Serviço de Diretório

- ◆ **Desempenha funções importantes**
- ◆ **Pode ou não estar implementado no mesmo servidor que o de arquivos, isto é:**
  - S.A.  $\Rightarrow$  serviço de arquivos + serviço de diretórios  
OU
  - S.A.  $\Rightarrow$  serviço de arquivos
  - S.D.  $\Rightarrow$  serviço de diretórios
- ◆ **Funções**
  - Traduzir nomes de arquivos em FIDs
  - Pode conter informações sobre proteção de acesso
  - Prevenir perda de arquivos acidentalmente apagados

# Divisão de Responsabilidades

## Generalizando...

- ◆ Serviço de Diretórios  $\Rightarrow$  S.A.
- ◆ Serviço de Diretórios  $\Rightarrow$  cliente do S.A. (ex. XDFS, executando no S.A.)
- ◆ Serviço de Diretórios  $\Rightarrow$  cliente na E.T. com “arquivos diretórios” no S.A. (CFS tem recursos para isto)
- ◆ Serviço de Diretórios  $\Rightarrow$  servidor especial (Ex. TRICE com MIFIA)

# Transações

- ♦ Seqüência de operações aplicadas a algum dado que pode pertencer a um arquivo ou mesmo a um conjunto de arquivos
  - ♦ Primitivas
    - begin-trans ⇒ abre uma transação
    - end-trans ⇒ encerra uma transação com sucesso
    - abort-trans ⇒ encerra uma transação com insucesso
- Obs: quando operações são organizadas em transações, tem-se um TID para identificar a transação
- ♦ Construídas pelo cliente, mas o servidor pode abortar uma transação

Fim!