
MAC0422 - Sistemas Operacionais

Daniel Macêdo Batista

IME - USP, 5 de Novembro de 2020

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

Mais sobre implementação de sistemas de arquivos

Backup e redundância

Mais sobre
implementação de
▷ sistemas de arquivos

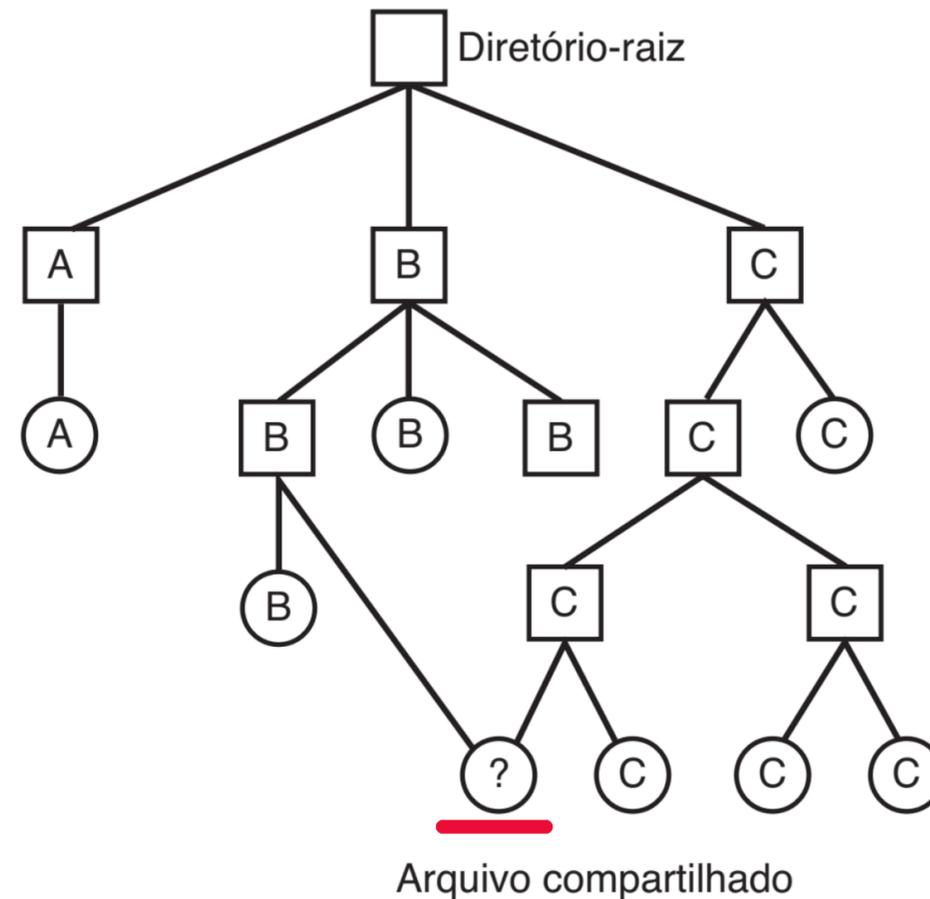
Backup e redundância

Mais sobre implementação de sistemas de arquivos

Arquivos compartilhados

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância



- O arquivo em destaque pertence ao usuário C mas tem uma ligação (link) em um diretório do usuário B
- A presença de arquivos compartilhados transforma a árvore de diretórios em um grafo acíclico orientado (DAG - Directed Acyclic Graph)
- Links são úteis para evitar múltiplas versões do mesmo arquivo

Possíveis inconsistências com arquivos compartilhados

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

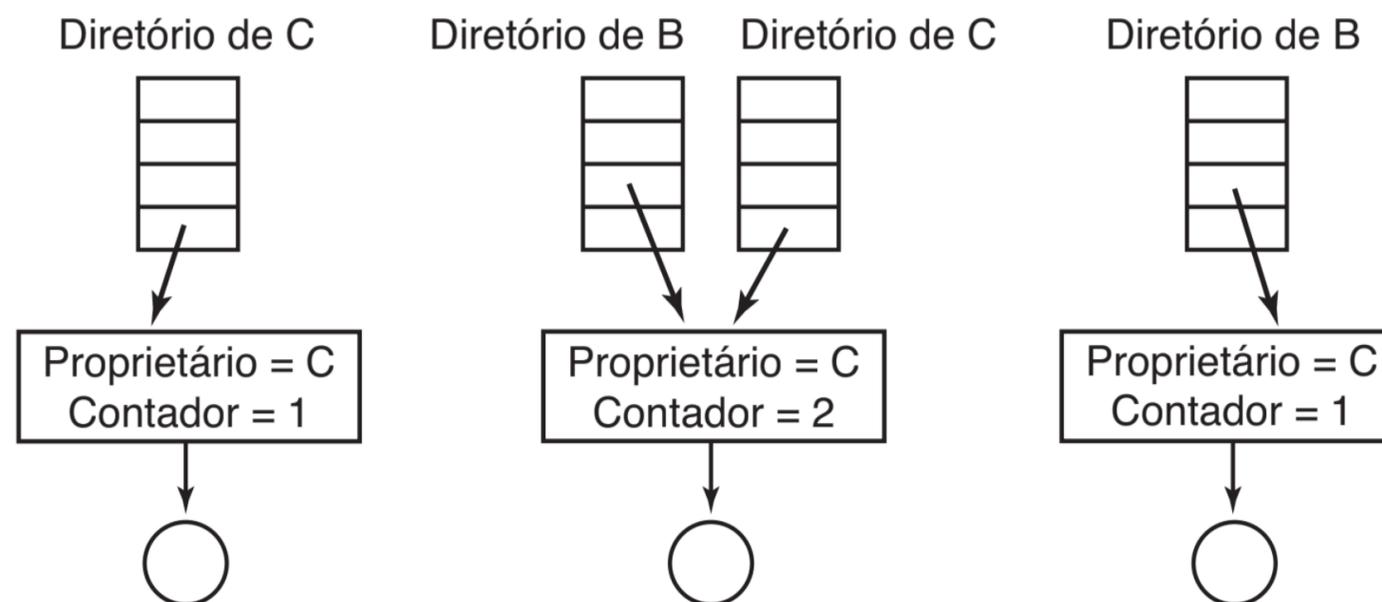
- Se diretórios só guardam os endereços dos conteúdos, como garantir que mudanças feitas no arquivo em um diretório serão vistas no arquivo do outro diretório se é um link?
 - i-nodes facilitam isso naturalmente (bastaria ter o mesmo i-node nos dois diretórios) → isso é um **link** (1n)
 - ter um tipo de arquivo especial que “aponta” para o nome do arquivo original → Isso é um **link simbólico** (1n -s)

Particularidades de links

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

- ❑ Criar um link não muda as permissões. Elas continuam valendo
- ❑ Criar um link aumenta o número de contador de ligações. Isso evita que o SO remova de fato o arquivo caso um link seja removido (Segunda coluna do `ls`) mas num sistema com quota, o usuário C continua “pagando a conta”



Particularidades de links simbólicos

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

- Criar um link simbólico não muda as permissões. Elas continuam valendo
- Criar um link simbólico não aumenta o número de contador de ligações
- Remoção de um arquivo com um link simbólico deixa o link simbólico “quebrado”
- Sobrecarga para acessar o conteúdo: **nome do link** → nome do arquivo original → conteúdo do arquivo
- Precisa de um i-node a mais por link
- Precisa ter cuidado na hora de copiar links simbólicos para sistemas de arquivos que não os suportem para evitar cópia duplicada

Sistemas de arquivos baseados em registros

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

- LFS – Log-structured File System
- Justificativa: CPUs mais rápidas, RAM mais rápidas e maiores, caches de disco maiores mas o acesso ao disco continua lento fazendo dele um gargalo
 - Em alguns casos seria possível acessar o conteúdo (lendo) apenas com o cache, se já foi lido em algum momento no passado
- Boa parte dos acessos à escrita são pequenos, o que é ineficiente

Sistemas de arquivos baseados em registros

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

- ❑ Ideia do LFS é manter operações de escrita apenas na memória e realizá-las no disco de tempos em tempos
- ❑ Escritas são feitas preferencialmente em sequência (1 segmento) no disco com um registro no início explicando o que está ali (i-nodes, blocos de diretório, blocos de arquivos, etc.)
- ❑ Assim, o trabalho de acessar o disco para escrever um conteúdo é melhor aproveitado (É muito mais rápido acessar o disco de uma vez só e escrever 100MB do que acessar o disco 100 vezes escrevendo 1MB em cada acesso)
- ❑ Tem que manter uma tabela de i-nodes informando onde cada um de fato está. Essa tabela pode ser mantida em cache
- ❑ Precisa, periodicamente, varrer o disco para compactá-lo removendo de fato segmentos não mais usados

Sistemas de arquivos com journaling

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

- Aproveita a ideia do LFS de “anotar” o que vai ser feito antes de fazer
- Se houver falha nas ações, como tem elas anotadas num “diário” (journal), consegue refazer
- Útil principalmente numa escrita longa e quando há queda de energia
- Existente em vários sistemas de arquivos: JFS (AIX da IBM), NTFS (Windows NT da Microsoft), XFS (IRIX da SGI), ext3 (GNU/Linux), ext4 (GNU/Linux)

Sistemas de arquivos com journaling

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

- Sendo mais claro:
 1. Remoção de arquivo de diretório
 2. Liberar i-node do arquivo p/ o conjunto de i-nodes livres
 3. Liberar blocos de disco p/ o conjunto de blocos de disco livres
- Se há queda de energia entre os passos 1 e 2, os i-nodes continuam ocupados mas não são acessíveis por nenhum diretório
- Se há queda de energia entre os passos 2 e 3, os blocos continuam ocupados mas não são acessíveis por nenhum i-node
- Diversos problemas diferentes ocorrerão se a ordem das três ações mudar

Sistemas de arquivos com journaling

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

- As ações são escritas no disco
- Só depois de escrever as ações no disco, começa a fazê-las de fato
- Só remove as ações do disco depois de ter certeza que elas foram feitas
- Se houver falha no meio das operações, pode ler do disco e saber em que ponto parou (por exemplo, quando montar o disco de novo no próximo boot)

Mais sobre
implementação de
sistemas de arquivos

▶ Backup e
redundância

Backup e redundância

Backup

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

- Os dados salvos em um sistema de arquivos geralmente tem um valor muito maior do que o valor financeiro do sistema computacional
- Já que falhas acontecem (por exemplo por conta de queda de energia elétrica ou, mais recentemente, por ransomware), é melhor ter um mecanismo de tolerância a falhas
- Um mecanismo de tolerância a falhas bem difundido para sistemas de arquivos é backup → manter uma cópia do sistema de arquivos em outro local (preferencialmente em outro local físico. Não faz muito sentido manter o backup em uma outra partição no mesmo disco ou em outro disco na mesma máquina)

Backup

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

- Ideal é que seja um backup automático (confiar no ser humano para pressionar um botão todo dia num determinado horário não é uma boa) e que combine backups completos e backups incrementais
- Em sistemas críticos pode ser importante manter o backup em mídias com tecnologia diferente da mídia original dos dados (Fitas magnéticas ainda costumam ser usadas para backup, preferencialmente com braços robóticos)
 - De fato, backup em fita é algo bem antigo e ainda existente (tar vem de Tape ARchive por exemplo.)
 - Atualmente é comum ver sistemas híbridos com fitas mantendo dados pouco modificados e HDs ou SSDs mantendo dados muito modificados

Backup

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

- Tão importante quanto fazer o backup é recuperar o backup. Não são poucas as histórias de empresas que compram um sistema de gerenciamento de backup caríssimo, nunca testam a recuperação e levam muito tempo para recuperação em caso de falha
- Implantar um sistema de gerenciamento de backup exige pelo menos um teste de recuperação antes de um problema de fato acontecer
- Quem quiser passar a fazer backup independente de terceiros, pesquise pelo software AMANDA (Advanced Maryland Automatic Network Disk Archiver – <http://www.amanda.org/>).

Redundância

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

- O backup exige ações extra para salvar e recuperar um conteúdo. Seria interessante ter algum mecanismo de redundância transparente às ações do usuário no sistema de arquivos
- RAID (Redundant Array of Independent Disks) permite isso
- No geral, o mais confiável e eficiente é quando é implementado em hardware mas também pode ser implementado em software (Em hardware é comum vir acompanhado de sistema de hot-swap)
- A ideia é combinar múltiplos discos de modo que se houver falhas em um disco ainda seja possível recuperar a informação. Ou seja, os dados são espalhados pelo array de discos

Redundância

Mais sobre
implementação de
sistemas de arquivos

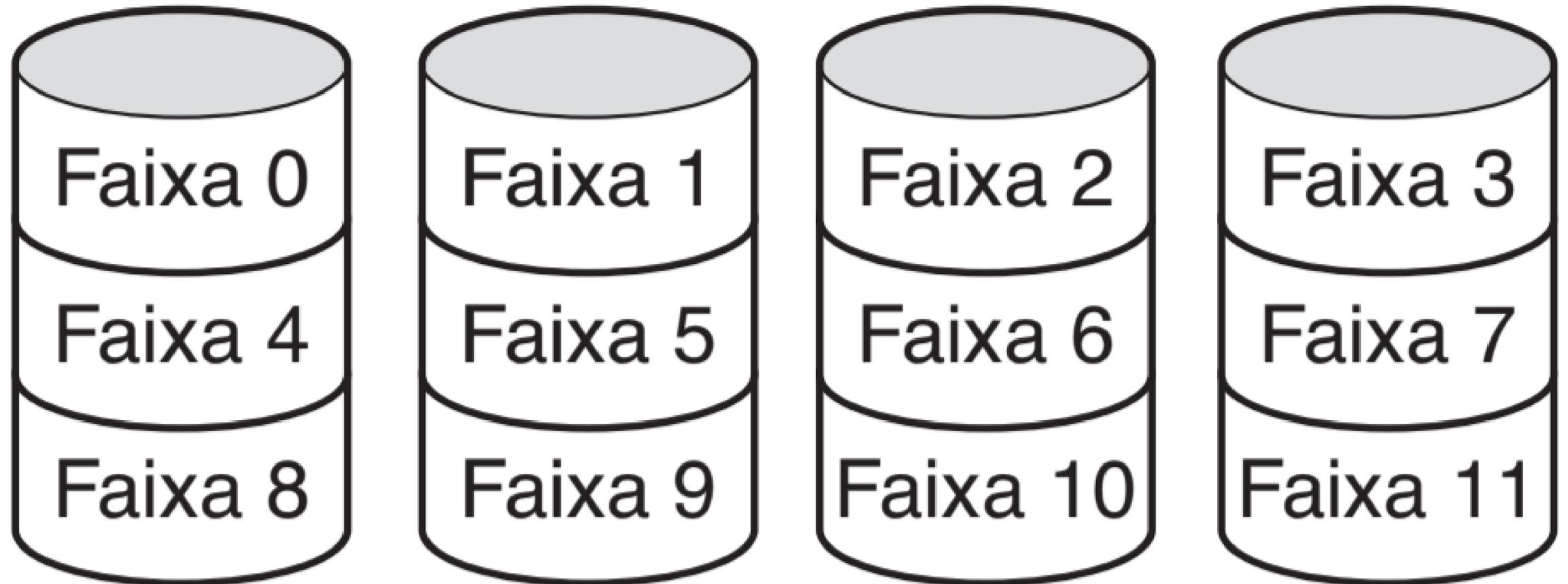
Backup e redundância

- Cada nível de RAID define o nível de desempenho e redundância desejado
- Usa espalhamento (espalha os blocos por vários discos), espelhamento (espelha os blocos por vários discos) ou paridade (espalha os blocos por vários discos mas faz paridade e consegue recuperar caso algum bloco seja perdido)
 - RAID 0: Faz espalhamento – Desempenho (2+ discos)
 - RAID 1: Faz espelhamento – Redundância (2+ discos)
 - RAID 5: Faz espalhamento e paridade distribuída – Mais redundância – funciona até com 1 disco a menos (3+ discos)

RAID 0

Mais sobre
implementação de
sistemas de arquivos

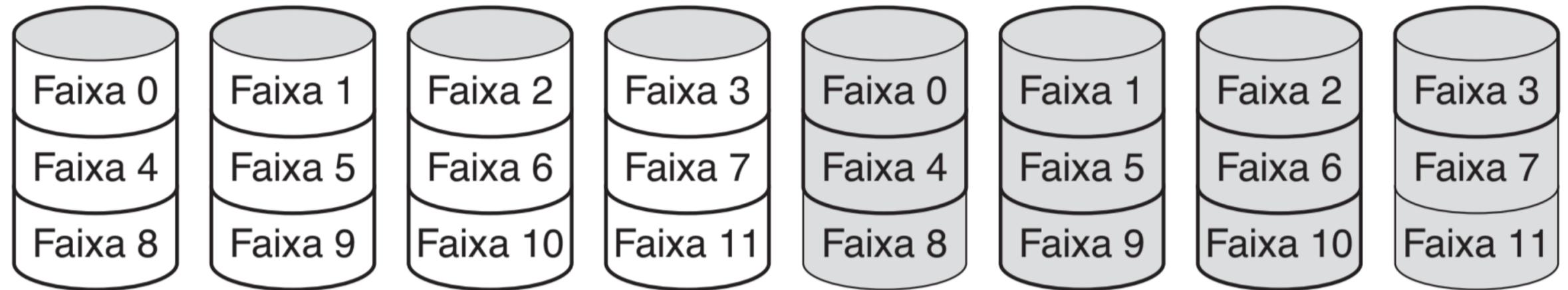
Backup e redundância



RAID 1

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância



RAID 5

Mais sobre
implementação de
sistemas de arquivos

Backup e redundância

