

T2FS

Sistemas Operacionais I N

Apresentação

Os Sisopeiros

Yuri Jaschek (G)

Giovane Fonseca

Humberto Sagave

Matheus F. Kovaleski



Características

Alocação indexada combinada

Estilo *inodes*

- inodes (nodos de índice) são estruturas que guardam, além de metadados, a informação sobre quais blocos do disco são usados para compor o arquivo
- Os ponteiros para blocos de dados são diretos e indiretos, com indireção de 1 a 3 níveis. Sendo permitido calcular e acessar rapidamente qualquer parte do arquivo
- A alocação indexada combinada flexibiliza a indexada multinível, permitindo acesso mais rápido para arquivos pequenos

Alocação indexada combinada

Estilo *inodes*

Vantagens

- Não é passível de fragmentação externa, como alocação contígua
- Acesso a qualquer dado é garantido em até cinco requisições ao disco, e não dependente do número de blocos alocados, como a encadeada
- Como as entradas de diretório guardam apenas o *inode* do arquivo, o sistema de arquivos possui uma razão de entradas por bloco de diretório muito superior
- Tem-se, de graça, a possibilidade de criação e gerência de *hard links*

Alguns detalhes

- O número máximo de arquivos depende, além da capacidade do disco, do número de *inodes*
- O tamanho máximo de um arquivo depende do número de ponteiros no *inode*, do tamanho do bloco e dos níveis de indireção
- Com o padrão de 12 ponteiros diretos e 1 indireto para cada nível de indireção (1 a 3), temos
 - ~ 65 MiB com 1 setor por bloco
 - ~ 1 GiB com 2 setores por bloco
 - ~ 16 GiB com 4 setores por bloco

Bitmap (vetor de bits) para gerência de espaço livre

- *Bitmap* tanto para o conjunto de blocos livres quanto para *inodes* livres
- *Bitmaps* são compactos e de fácil gerência
 - As funcionalidades de marcar um bloco (ou *inode*) como livre ou usado possuem tempo constante de 1 acesso ao disco
- A busca por blocos (ou *inodes*) livres é linear, mas o desempenho pode ser melhorado guardando índices de interesse, como o índice do primeiro bloco (ou *inode*) livre

Suporte a vínculos simbólicos e estritos

Soft links e hard links

- Vínculos simbólicos permitem que um arquivo aponte para outro, um sendo o *link* e outro sendo o original
 - E se o original for deletado?
- Vínculos estritos fazem com que múltiplas entradas apontem para um mesmo arquivo, não havendo diferença sobre qual é o original e qual é o *link*
 - Então por que vínculos simbólicos?
- Vínculos estritos não podem ser criados para diretórios e nem quando os arquivos pertencerem a sistemas de arquivos diferentes, enquanto os simbólicos permitem isso.

Hierarquia de diretórios como um DAG

Directed acyclic graph

- Permite que um mesmo arquivo seja acessado de pontos distintos da hierarquia
- É uma consequência da existência de *hard links*
 - Entretanto, *hard links* para diretórios não são permitidos, uma vez que isso tornaria o grafo cíclico, causando problemas com algoritmos que percorrem a hierarquia
 - `.` e `..` são, tecnicamente, *hard links*, mas são sempre entradas existentes e fixas de todo diretório, permitindo que algoritmos as tratem de maneira especial
- Hierarquia começa com o diretório `/` (raiz), cuja localização será, por convenção, o *inode* 1

Registros de diretórios

- Os registros de diretório terão tamanho fixo, sendo compostos do nome do arquivo e do seu *inode* correspondente
- Os registros de diretório serão organizados como uma lista linear
 - Com tamanho fixo, as entradas podem ser facilmente indexadas. Torna-se fácil, também, a adição de novos registros, bastando apenas procurar por um espaço de registro vago

Suporte a caminhos relativos e absolutos

- Suporte a caminhos relativos e absolutos proporcionam uma maior flexibilidade ao usuário e a aplicações que fazem uso do sistema de arquivos
 - Caminhos relativos permitem que aplicações que têm atuação apenas a partir de certo diretório não precisem saber onde está tal diretório na hierarquia do sistema
 - Caminhos absolutos são altamente desejáveis para aplicações cuja atuação é, por exemplo, acionada pelo usuário, mas aplicada sobre diretórios do sistema. Nestes casos, caminhos absolutos fornecem o caminho direto (desde a raiz) até os arquivos ou diretórios

File system – Overview

Característica	Opção escolhida
Alocação do espaço em disco	Indexada combinada (estilo <i>inodes</i>)
Gerência de espaço livre	Bitmap (vetor de bits)
Hierarquia de diretórios T2FS	Grafo acíclico
Localização do diretório raiz	Local fixo (convencionado)
Registro de diretório	Tamanho fixo
Organização interna dos registros de diretório	Organização linear
Arquivos	Arquivos regulares (binário e ASCII)
Vínculos (<i>links</i>)	Simbólicos e estritos (<i>soft</i> e <i>hard</i> links)
Caminho	Relativo e absoluto



Layout após formatação

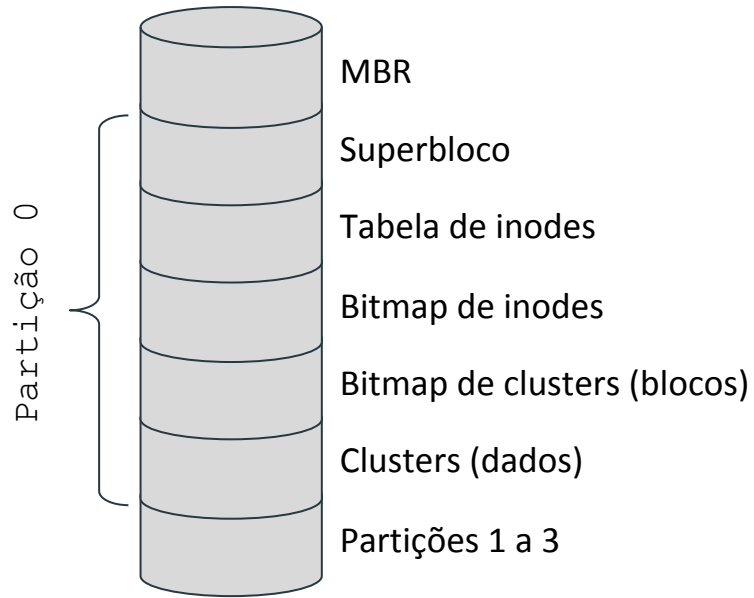


Imagem meramente ilustrativa

- O superbloco guarda informações necessárias à gerência do sistema de arquivos.
- A tabela de inodes contém as informações de cada inode do sistema de arquivos.
- Os bitmaps indicam se o inode ou cluster está livre ou está sendo usado.



Obrigado!