****

**Unioeste - Universidade Estadual do Oeste do Paraná**

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

***Curso de Bacharelado em Ciência da Computação***

*Docente: Marcio Seiji Oyamada*

**Sistemas de arquivos - Especificação**

*Arthur Augusto Eich*

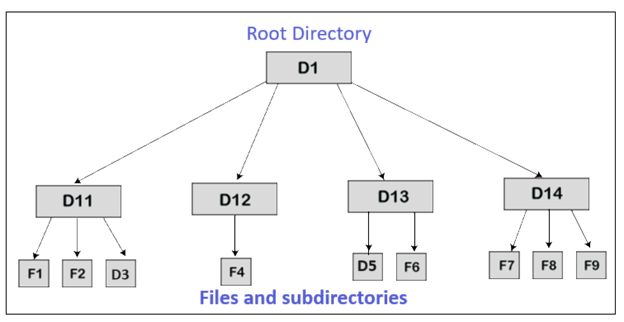
*Luiz Eduardo Garzon de Oliveira*

*Pedro Henrique de Oliveira Berti*

**CASCAVEL**

**2024**

1. **Arquitetura**
2. **Alocação contígua**
3. **Gerenciamento de blocos livres com mapa de bits**
4. **Diretórios**

****Os diretórios estão todos localizados na tabela de entradas do sistema, sendo no início da tabela o diretório raiz, com todas suas entradas. Os outros diretórios do sistema assim como o diretório raiz também estão localizados na tabela de entradas, e podem ser encontrados acessando seu bloco inicial e número de blocos utilizados, a partir de sua própria entrada, assim podem ser acessados e lidos as entradas desse próprio diretório, com todo sistema estruturado em formato de árvore.

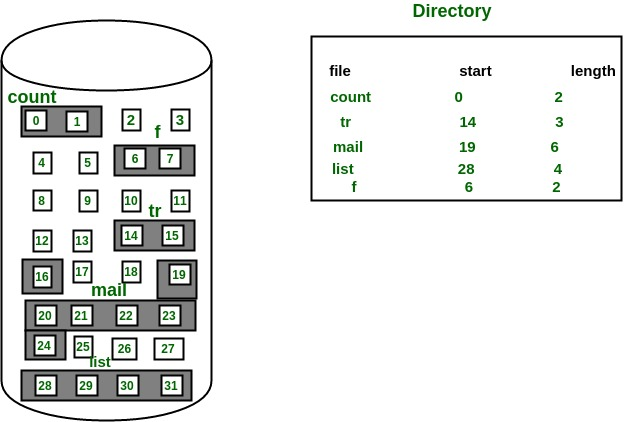
**Figura 1 - Árvore de Diretórios**

1. **Detalhes da implementação**

Na implementação do sistema de arquivos, cada arquivo é armazenado de forma contígua em blocos consecutivos, facilitando a implementação e o acesso aos dados. É utilizado o mapa de bits para controlar quais blocos estão livres ou ocupados. A abordagem utilizada proporciona uma organização simples dos arquivos, eficiente para sistemas onde os dados não são alterados com frequência.

Dessa forma, para determinar o espaço utilizado por cada arquivo é preciso armazenar apenas o bloco inicial de cada arquivo e o número de blocos ocupados, onde será atribuído um bit aos blocos correspondentes para determinar que o bloco está ocupado.

Para gerenciar a alocação é utilizado o mapa de bits, caso um bloco esteja livre, o bloco poderá ser utilizado para alocação caso o número de blocos contíguos em sequência livres seja suficiente para armazenar o arquivo todo, caso contrário, deve-se procurar uma outra sequência de blocos livres para armazenar o arquivo, caso nenhum espaço seja suficiente para alocar o arquivo seja encontrado, uma mensagem de erro deverá ser emitida.



**Figura 2 - Imagem de disco com alocação contígua.**

Na figura 2, pode-se reparar que o arquivo 'mail' na figura começa a partir do bloco 19 com comprimento igual a 6 blocos. Portanto, ocupa os blocos 19, 20, 21, 22, 23, 24.

**O sistema de arquivos é dividido em quatro setores, sendo eles:**

1. **Boot Record.**
2. **Mapa de Bits.**
3. **Tabela de Entradas.**
4. **Seção de dados.**

|  |  |  |  |
| --- | --- | --- | --- |
| Boot Record | Mapa de Bits | Tabela de Entradas | Seção de Dados |

**3.1. Boot Record**

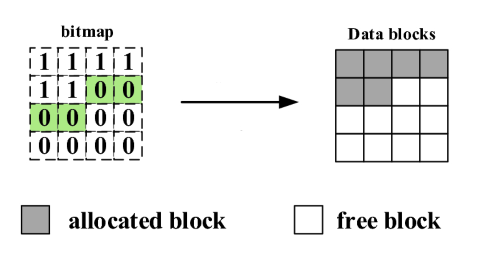
Contém as principais informações do sistema que vão ajudar na busca e acesso pelos dados.

**DADOS DO BOOT RECORD**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **Offset** | **Offset** | **Tamanho** | **Descrição** |  |
| **(decimal)** | **(hex)** | **(bytes)** |  |
|  |  |
|  |  |  |  |  |
| 0 | 0x00 | 2 | Número de bytes por bloco. |  |
|  |  |  |  |  |
| 2 | 0x02 | 2 | Número de blocos reservados. Boot Record incluído |  |
| 4 | 0x04 | 4 | Número de blocos do Mapa de Bits |  |
|  |  |  |  |  |
| 8 | 0x08 | 4 | Número de blocos da Tabela de Entradas. |  |
|  |  |  |  |  |
| 12 | 0x0C | 4 | Número de blocos na Seção de Dados. |  |
|  |  |  |  |  |
| 16 | 0x10 | 4 | Número de blocos totais do sistema. |  |
|  |  |  |  |  |
| 20 | 0x14 | 4 | Número de blocos para diretório raíz. |  |
| 24 | 0x18 | 4 | Quantidade de entradas do sistema. |  |
|  |  |  |  |  |
| 28 | 0x1C | 8 | Vazio, para completar 32 bytes. |  |
|  |  |  |  |  |

**3.2. Mapa de bits**

O mapa de bits é utilizado para localizar quais blocos estão ocupados ou livres, nele cada bit representa cada bloco do sistema, onde 1 representa um bloco ocupado e 0 um bloco livre.



**Figura 3 - Exemplo do uso de mapa de bits para gerenciamento de blocos.**

**3.3. Tabela de entradas**

Contém as informações sobre as entradas do sistema, nomes ou diretórios, contendo suas informações, como nome, extensão, tamanho, se é uma entrada removida ou não etc. Incluindo informações sobre sua alocação, como o primeiro bloco que ocupa e o número de blocos ocupados.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dados da entrada 1 | Dados da entrada 2 |  | Dados da entrada N |  |
| (status, nome, | (status, nome, | . . . | (status, nome, |  |
| extensão, tamanho, | extensão, tamanho, | extensão, tamanho, |  |
|  |  |
| primeiro bloco …) | primeiro bloco …) |  | primeiro bloco…) |  |
|  |  |  |  |  |

**3.3.1. Formato das entradas**

Os arquivos dentro do bloco de diretórios são compostos por um total de 32 bytes, onde cada byte tem seu dado conforme demonstra o quadro abaixo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Offset** | **Offset** | **Tamanho** | **Descrição** |  |
| **(decimal)** | **(hex)** | **(bytes)** |  |
|  |  |
|  |  |  |  |  |
| 0 | 0x00 | 1 | Status. 0xFF para entradas removidas. |  |
|  |  |  |  |  |
| 1 | 0x01 | 12 | Nome do arquivo. |  |
|  |  |  |  |  |
| 13 | 0x0D | 4 | Extensão. |  |
|  |  |  |  |  |
| 17 | 0x11 | 1 | Indicador de tipo. 0x01 = File, 0x02 = Directory |  |
|  |  |  |  |  |
| 18 | 0x12 | 4 | Primeiro Bloco. |  |
|  |  |  |  |  |
| 22 | 0x16 | 4 | Tamanho em bytes. |  |
|  |  |  |  |  |
| 26 | 0x1A | 4 | Número de blocos usados. |  |
| 30 | 0x1E | 2 | Vazio, para completar 32 bytes |  |

**3.4. Seção de dados**

Na seção de dados, os arquivos serão armazenados de forma contígua em blocos consecutivos no disco. Cada arquivo ocupará um ou mais blocos, seguindo uma ordem sequencial de alocação. O gerenciamento dos blocos livres será realizado por meio de um mapa de bits, onde para cada alocação será procurado uma sequência de blocos contíguos livres suficientes para o arquivo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bloco 1 | Bloco 2 | Bloco 3 | Bloco 4 | Bloco 5 |
| Arquivo A | Arquivo A | Livre | Arquivo B | Livre |

Na tabela, cada coluna representa um bloco no disco, onde os arquivos são armazenados de forma contígua em blocos consecutivos. O gerenciamento dos blocos livres é realizado pelo mapa de bits, embora não seja explicitamente mostrado na tabela. Os arquivos são identificados pelas letras A e B, ocupando os blocos correspondentes, enquanto os blocos livres são indicados como "Livre".

**3.5. Manipulação de Arquivos**

**3.5.1. Criar Arquivo**

Para criar um arquivo, o sistema primeiro verifica se há espaço suficiente disponível para armazenar o novo arquivo. Isso é feito verificando o mapa de bits para identificar os blocos livres no disco. Se houver espaço suficiente, o sistema procede atualizando o mapa de bits para marcar os blocos que serão utilizados pelo novo arquivo como ocupados. Além disso, o sistema atualiza a entrada de diretório correspondente ao novo arquivo.

**Ex:** **criar arquivo(nome, extensão, tamanho):** Cria um arquivo com o nome, extensão e tamanho especificados.

**3.5.2 Remover Arquivo**

Quando um arquivo é removido do sistema, o primeiro passo é atualizar o mapa de bits para marcar os blocos que estavam sendo usados pelo arquivo como livres. Isso permite que esses blocos sejam reutilizados para armazenar outros arquivos no futuro. Além disso, a entrada de diretório correspondente ao arquivo removido é marcada como "removida", definindo o status como 0xFF. Essa marcação indica que a entrada está disponível para ser reutilizada para armazenar informações de outros arquivos. Ao marcar a entrada de diretório como removida, o sistema também remove as referências ao arquivo removido, garantindo que ele não seja mais acessível.

**Ex:** **remover arquivo(nome, extensão):** Remove um arquivo com o nome e extensão especificados.

**3.5.3 Ler Arquivo**

Quando um arquivo precisa ser lido, o sistema primeiro localiza a entrada de diretório correspondente ao arquivo. Isso é feito procurando o nome e a extensão do arquivo na tabela de entradas. Uma vez que a entrada de diretório é encontrada, o sistema usa as informações contidas nela, como o primeiro bloco e o número de blocos ocupados pelo arquivo, para acessar os dados do arquivo na seção de dados.

**Ex:** **ler arquivo(nome, extensão):** Lê o conteúdo de um arquivo com o nome e extensão especificados.

1. **Referências bibliográficas**

FAT: Wiki OS Dev.org. Disponível em: https://wiki.osdev.org/FAT. Acesso em: 3 abr. 2024.

EXT2: Wiki OS Dev.org. Disponível em: https://wiki.osdev.org/Ext2. Acesso em: 3 abr. 2024.

Figura 1 retirada de: https://www.tutorialandexample.com/directory-structure-in-operating-system. **Directory Structure in Operating System.** Autor desconhecido.

Figura 2 retirada de: https://acervolima.com/metodos-de-alocacao-de-arquivos/. **MÉTODOS DE ALOCAÇÃO DE ARQUIVOS.** Autor desconhecido.

Figura 3 retirada de: https://link.springer.com/chapter/10.1007/978-3-030-05057-3\_49. **Acquiring Hidden Space via Modifying Block Bitmap for Android Devices,** por Wang Lianfang, Huang Hong, Li Yuanzhang & Zhang Li.