

Especificação de Sistema de Arquivos

Diego Thedoldi, Gabriel Ramos e Lucas Becker

1. Arquitetura

- Alocação contígua
- Gerenciamento de blocos livres através de lista ligada.

2. Diretórios

A representação de diretórios no nosso sistema de arquivos é feita por estruturas em árvores, onde agrupa arquivos em diferentes diretórios, possuindo assim caminhos para cada diretório, para acessar diretórios acima do que está sendo acessado no momento é necessário utilizar o caminho “..” e para acessar caminhos mais abaixo é necessário especificar o caminho.

3. Detalhe de implementação

Nosso sistema armazena os arquivos de forma contígua, o conteúdo dos arquivos vão ser armazenados em clusters sequenciais na seção de dados. Já no gerenciamento de blocos livres, utilizaremos listas ligadas que definem através de índices quais são os clusters livres no sistema. Essa combinação pode gerar uma certa limitação caso não haja um sistema de desfragmentação da seção de dados, pois há a necessidade de encontrar blocos livres contíguos para a alocação.

A mídia de armazenamento do sistema é organizada em quatro áreas.

- **Boot record.**
- **Bloco de gerenciamento de espaço livre.**
- **Bloco de diretórios.**
- **Seção de dados.**

Boot record	Bloco de gerenciamento de espaço livre	Bloco de diretórios	Seção de dados
-------------	--	---------------------	----------------

3.1. Boot Record

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

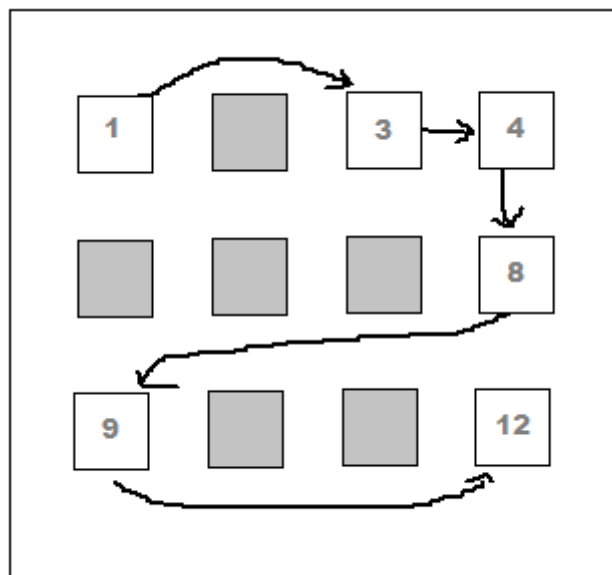
Descrição

Offset (decimal)	Offset (hex)	Tamanho (bytes)	Descrição
0	0x00	2	Número de bytes por setor.

2	0x02	1	Número de setores por cluster.
3	0x03	2	Número de setores reservados. O Boot Record está incluído nesse valor.
5	0x05	2	Número de setores no Bloco de Diretórios.
7	0x07	2	Número de setores na Seção de Dados.
11	0x0B	2	Número de setores no Bloco de Gerenciamento de Espaço Livre.
13	0x0D	4	Quantidade de arquivos armazenados no sistema.

3.2. Bloco de gerenciamento de espaço livre

O gerenciamento de espaço livre é disposto em formato de lista ligada, onde é necessário apenas que tenha o ponteiro armazenado para o início da lista, onde ele usará o byte correspondente ao cluster para identificar se está sendo usado ou não, caso esteja, esse byte será setado com 0x01, caso esteja vazio, estará setado como 0x00, porém, como a alocação é de forma contígua, para isso uma verificação é realizada para saber se o arquivo a ser alocado tem tamanho suficiente para esse espaço livre, caso contrário, deve-se procurar um outro espaço em que o tamanho do arquivo a ser alocado seja compatível com o espaço físico, ou então caso nenhum espaço seja suficiente para alocar o arquivo, uma mensagem de erro deverá aparecer.



No caso ilustrado na imagem, o ponteiro está apontando para o primeiro cluster livre, após isso aponta para os próximos clusters livres que estão a disposição para alocação, os blocos em cinza são os que já estão alocados.

3.3. Bloco de diretórios

Contém as informações sobre os diretórios/arquivos do sistema como nome do arquivo, extensão, e etc. Incluindo informações sobre sua alocação, como o primeiro cluster e seu tamanho para que possa ser calculado posteriormente quantos clusters ele ocupa na seção de dados.

Dados do arquivo 1 (status, nome, extensão, tamanho, primeiro cluster ...)	Dados do arquivo 2 (status, nome, extensão, tamanho, primeiro cluster ...)	...	Dados do arquivo N (status, nome, extensão, tamanho, primeiro, cluster...)
---	---	-----	---

3.3.1. Formato dos arquivos

Os arquivos dentro do bloco de diretórios são compostos por um total de 24 bytes, onde cada byte tem sua função específica conforme demonstra o quadro abaixo:

Offset (decimal)	Offset (hex)	Tamanho (bytes)	Descrição
0	0x00	1	Status. Caso o arquivo tenha sido apagado, o valor será 0xFF, caso contrário será 0x00.
1	0x01	12	Nome do arquivo.
12	0x0C	4	Extensão.
16	0x10	1	Atributo.
17	0x11	2	Primeiro cluster.
19	0x13	4	Tamanho do arquivo em bytes.

3.4. Seção de dados

Essa seção está reservada ao conteúdo dos arquivos contidos no sistema, os arquivos podem ocupar um ou mais clusters, porém como a alocação é realizada de forma contígua, eles devem sempre seguir a sequência em que serão alocados.

Cluster 1	Cluster 2	...	Cluster N
-----------	-----------	-----	-----------

4. Guia do programador

Essa seção ajudará durante a implementação do sistema, contém algumas estruturas e fórmulas básicas que serão usadas para verificação e testes no sistema de arquivos.

4.1. Estrutura do BootRecord

```
typedef struct BootRecord
{
    unsigned short    bytes_per_sector;
    unsigned char     sectors_per_cluster;
    unsigned short    reserved_sector;
    unsigned short    sector_directory_blocks;
    unsigned short    sector_section_data;
    unsigned short    sector_management_free_space;
} __attribute__((packed)) BootRecord;
```

4.2. Estrutura do Formato do Arquivo

```
typedef struct dir{
    unsigned char status;
    unsigned char name[12];
    unsigned char ext;
    unsigned char attribute;
    unsigned char first_cluster;
    unsigned int  fileSize;
} __attribute__((packed)) dir;
```

4.3. Fórmulas básicas

Nessa seção são encontradas algumas fórmulas básicas que podem ser usadas para facilitar durante a implementação.

4.3.1. Tamanho do Cluster

```
SizeCluster = bytes_per_sector * sector_per_cluster;
```

4.3.2. Tamanho do Bloco de Diretório

```
SizeDirectoryBlock = sector_directory_blocks *
bytes_per_sector;
```

4.3.3. Tamanho da Seção de Dados

```
SizeSectionData = sector_section_data * bytes_per_sector;
```

4.3.4. Tamanho do Gerenciamento de Espaço Livre

```
SizeManagementFreeSpace = sector_management_free_space *  
bytes_per_sector;
```