**Documentação – Implementação tabela FAT**

**Estrutura:**

Para implementar a estrutura corretamente foi inicializado structs para abstrair os principais objetos de manipulação de discos com tabela FAT, são eles:

* **Cluster**: A estrutura Cluster representa um cluster de dados no disco. Cada cluster possui dois campos:
  + **numeroCluster**: Armazena o número do cluster.
  + **dados**: Um ponteiro para um array de bytes que armazena os dados do cluster.
* **AreaDados**: A estrutura AreaDados representa a área física de dados do disco. Ela consiste nos seguintes campos:
  + **clusters**: Um ponteiro para um array de Cluster que armazena os clusters de dados.
  + **clustersLivres**: Armazena o número de clusters livres na área de dados.
  + **totalClusters**: Armazena o número total de clusters na área de dados.
* **TabelaFAT**: A estrutura TabelaFAT representa a tabela File Allocation Table (FAT) do disco. Ela possui os seguintes campos:
  + **tabela**: Um ponteiro para um array de inteiros que armazena os números de clusters e suas respectivas entradas na tabela FAT.
  + **clustersLivres**: Armazena o número de clusters livres na tabela FAT.
  + **totalClusters**: Armazena o número total de clusters na tabela FAT.
* **EntradaDiretorio**: A estrutura EntradaDiretorio representa uma entrada no diretório do disco. Ela possui os seguintes campos:
  + **nomeArquivo**: Um ponteiro para uma string que armazena o nome do arquivo.
  + **clusterInicial**: Armazena o número do cluster inicial onde o arquivo está armazenado.
* **Diretorio**: A estrutura Diretorio representa o diretório do disco, onde é armazenado informações sobre os arquivos. Ela possui os seguintes campos:
  + **entradas**: Um ponteiro para um array de EntradaDiretorio que armazena as entradas do diretório.
  + **numEntradas**: Armazena o número de entradas no diretório.
* **Disco**: A estrutura Disco representa o disco como um todo. Ela possui os seguintes campos:
  + **tabelaFAT**: Uma instância da estrutura TabelaFAT que representa a tabela FAT do disco.
  + **areaDados**: Uma instância da estrutura AreaDados que representa a área de dados do disco.
  + **diretorio**: Uma instância da estrutura Diretorio que representa o diretório do disco.

Essas são as estruturas desenvolvidas para a simulação de uma tabela FAT em C. A inicialização dessas estruturas é feita através de funções específicas, como **inicializarAreaDados**, **inicializarTabelaFAT** e **inicializarDiretorio**, garantindo que todas as propriedades sejam inicializadas corretamente.

**Inicialização das estruturas:**

Para inicializar o disco, é passado informações como o número total de clusters e a capacidade de cada cluster, nas funções inicializadoras é alocado os arrays com base nesse parâmetro. Além disso, a tabela FAT, a área de dados e as entradas de diretório são inicializadas em “branco”, ou seja, não há informações em suas entradas.

**Operações:**

Para simular um disco real, foi desenvolvido operações de gravação, leitura e remoção de arquivos utilizando todas as estruturas definidas:

* **Adicionar arquivo em disco:** Ao enviar um disco para ser gravado no disco, é realizado os seguintes passos:
  1. Verifica se há espaço disponível no disco para armazenar o arquivo verificando as propriedades “clustersLivres” na TabelaFAT e AreaDados.
  2. Dividi os dados do arquivo em clusters, definido pela configuração inicial de tamanho de cluster. Caso uma fração do arquivo não ocupe 100% da capacidade do cluster, esse cluster não estará disponível para armazenar informações de outros arquivos.
  3. Grava os dados nos clusters da Area de dados, buscando os clusters livres mais próximos e refletindo essas alterações na tabela FAT. Após isso é inserido na entrada de diretório o nome e o número do cluster inicial do arquivo.
* **Ler arquivo do disco:** Para leitura de um arquivo a partir de seu nome, é realizado os seguintes passos:
  1. Percorre o diretório, buscando pelo nome do arquivo, ao encontrar é armazenado o número do seu cluster inicial.
  2. Com o número do cluster inicial, é percorrido a tabela FAT para ler todos os clusters associados ao arquivo. Ele busca o cluster que deve ser lido na tabela FAT, e lê na área de dados do disco. Até encontrar uma marcação especial na tabela que indica o final do arquivo (numeroCluster = -1)
  3. Concatena os dados lidos de todos os clusters para reconstruir o arquivo completo.
* **Remover arquivos do disco:** Para remoção de um arquivo a partir de seu nome, é realizado os seguintes passos:
  1. Percorre o diretório, buscando pelo nome do arquivo, ao encontrar é armazenado o número do seu cluster inicial.
  2. Com o número do cluster inicial, é percorrido a tabela FAT para identificar todos os clusters associados ao arquivo. Ele busca o cluster que deve ser lido na tabela FAT, e lê na área de dados do disco. Até encontrar uma marcação especial na tabela que indica o final do arquivo (numeroCluster = -1)
  3. Libera os dados de cada cluster na área de dados e na tabela FAT, além de limpar a entrada de diretório com arquivo.

