

CUMULO

Anderson Bottega da Silva, Juliano Felipe da Silva, Maycon de Queiroz Oliveira

1 ÍNDICE

2	Resumo	1
3	Detalhes de implementação	2
3.1	Informações iniciais da unidade	2
3.2	Bitmap de indicação de ocupação	2
3.3	Dados	2
3.3.1	Diretórios	2
3.3.2	Arquivos	3
4	Guia de programação.....	3
4.1	Leitura das informações iniciais.....	3
4.2	Leitura de informações de diretórios	3
4.3	Cálculos para informações adicionais	4
4.3.1	Capacidade por bloco.....	4
4.3.2	Capacidade total de armazenamento.....	4

2 RESUMO

O sistema de arquivos Cumulo é um sistema de arquivos baseado em lista ligada para armazenamento de dados com um bitmap próximo ao início da unidade para indicar blocos de setores livres ou ocupados. Projetado inicialmente para unidades com 128 megabytes, pode ser usado em unidades desde 9232 bytes até 4096 megabytes em intervalos de 32 megabytes.

Caso a unidade seja formatada no menor tamanho, a quantidade de blocos será igual a dois, como um é reservado ao diretório raiz, resta apenas um bloco livre (um tamanho menor seria permitido, mas é inútil); apesar de matematicamente poder criar unidades de até 8160 megabytes, o formatador não permitirá tal formatação visto que são reservados apenas 4 bytes para o tamanho do arquivo, fazendo com que o maior arquivo que pode ser registrado tenha 4096 megabytes.

3 DETALHES DE IMPLEMENTAÇÃO

Como dito, o Cumulo é um sistema baseado em lista ligada que é composto por três partes principais:

- Informações iniciais da unidade;
- Bitmap para indicação de blocos de setores livres ou ocupados;
- Área de armazenamento de dados.

3.1 INFORMAÇÕES INICIAIS DA UNIDADE

Ocupa os 16 primeiros bytes da unidade, ficam armazenados nome da unidade, quantidade de blocos de setores e a quantidade de setores por bloco.

Início (bytes)	Tamanho (bytes)	Significado
0	13	Nome da unidade em formato ASCII
13	2	Quantidade de blocos de setores na unidade
15	1	Quantidade de setores por bloco de setores

Por definição do sistema, cada setor tem 512 bytes e o sistema possui 65536 blocos de setores, apesar de o valor de blocos de setores pode ser alterado pelo formatador. Caso a quantidade de blocos seja 65536, o valor da variável será 0.

3.2 BITMAP DE INDICAÇÃO DE OCUPAÇÃO

Ocupa os 8192 seguintes bytes da unidade imediatamente após os 16 primeiros bytes. Representa o estado de cada um dos blocos de setores da unidade, utilizando um mapeamento direto onde cada bloco é representado por um bit no mapa na ordem em que se recebem os dados da unidade utilizando 0 (zero) para livre ou 1 (um) para ocupado. Logo, o bloco 0 terá seu estado representado pelo primeiro bit do mapa, o bloco 1 pelo segundo bit do mapa e assim por diante. Caso a unidade possua menos blocos que os 65536 que o bitmap representa, o tamanho do bitmap não é alterado e os bits com referência inválida são marcados como ocupados.

3.3 DADOS

Na área destinada a dados, são colocadas as informações sobre o conteúdo dos diretórios e as informações dos arquivos da unidade. Por definição do sistema, o primeiro bloco é sempre reservado ao diretório principal da unidade e este bloco possui endereço 0. Portanto, se o valor zero for encontrado como ponteiro a um próximo bloco no final de um bloco, indica o fim do arquivo ou do diretório.

3.3.1 Diretórios

Os blocos de diretórios armazenam informações sobre arquivos e diretórios filhos (metadados) e o ponteiro para o primeiro bloco onde o arquivo ou diretório se inicia.

Início (bytes)	Tamanho (bytes)	Significado
0	17	Nome do arquivo ou diretório (13 primeiros bytes para o nome e os 4 seguintes para a extensão)
17	1	Tipo (0 = diretório; 1 = arquivo)

18	4	Tamanho do arquivo (Máximo de 4096 megabytes)
22	4	Data e hora de criação
26	4	Data e hora da última modificação
30	2	Ponteiro para o primeiro bloco de setores do arquivo ou diretório

A data e hora de criação e modificação tem o mesmo formato; sendo os primeiros 7 bits para o ano (que são acrescentados a 2000 para a constituição), os 4 bits seguintes para mês, os 5 bits seguintes para dia, os 5 seguintes para segundos (que devem ser multiplicados por 2 para o valor real), os 6 seguintes para minutos e os 5 últimos para horas.

Dentro de cada diretório existe um subdiretório com o nome “.” que é uma referência direta para o diretório local atual, também existe um subdiretório com o nome “..” que é uma referência direta para o diretório pai, no caso do diretório raiz, este subdiretório aponta para ele mesmo. Esses subdiretórios não podem ser movidos, copiados ou apagados e são criados junto com a criação de qualquer diretório pelo sistema de arquivos.

3.3.2 Arquivos

Os blocos de setores de arquivos armazenam os dados dos arquivos em sequência, destinando os dois últimos bytes para um ponteiro para o próximo bloco de setores.

4 GUIA DE PROGRAMAÇÃO

4.1 LEITURA DAS INFORMAÇÕES INICIAIS

Estrutura sugerida:

```
typedef struct IICumulo
{
    unsigned char  nome_unidade[13];
    unsigned short qtde_blocos;
    unsigned char  setores_por_bloco;
}__attribute__((packed)) IICumulo_t;
```

4.2 LEITURA DE INFORMAÇÕES DE DIRETÓRIOS

Estrutura sugerida:

```
typedef struct IDCumulo
{
    unsigned char  nome[13];
    unsigned char  extensao[4];
    unsigned char  tipo;
    unsigned int   tamanho;
    unsigned int   criacao;
    unsigned int   modificacao;
    unsigned short primeiro_setor;
}__attribute__((packed)) IDCumulo_t;
```

4.3 CÁLCULOS PARA INFORMAÇÕES ADICIONAIS

4.3.1 Capacidade por bloco

```
capacidade_bloco = boot.setores_por_bloco * 512;
```

4.3.2 Capacidade total de armazenamento

```
capacidade_total = boot.qtde_blocos * capacidade_bloco;
```