

# ESPECIFICAÇÃO SISTEMA DE ARQUIVOS SUPERMINI

Igor França Negrizoli, Lucas Veit de Sá e Mateus Karvat Camara

O sistema SuperMini foi parcialmente inspirado nos sistemas FAT e Ext2. O nome “SuperMini” é originado da aglutinação dos dois principais conceitos nesse sistema de arquivos: Superblocos e Miniblocos. Os superblocos dizem respeito a blocos de dados alocados contiguamente que podem ser acessados de forma sequencial, enquanto os miniblocos armazenam os metadados de arquivos presentes em pastas. Apesar dos superblocos serem compostos de blocos contíguos, o sistema SuperMini utiliza de indexação para acesso a blocos, utilizando superblocos como forma de otimizar o gerenciamento de espaços livres.

A estrutura geral de um disco formatado em SuperMini é apresentada na Figura 1. Nota-se a existência de um Boot Record (BR), seguido de informações do Diretório Raiz (Root), um Block Bitmap e os blocos de dados (Data Blocks). Assim como outros sistemas de arquivo, o SuperMini utiliza o padrão Little-Endian.

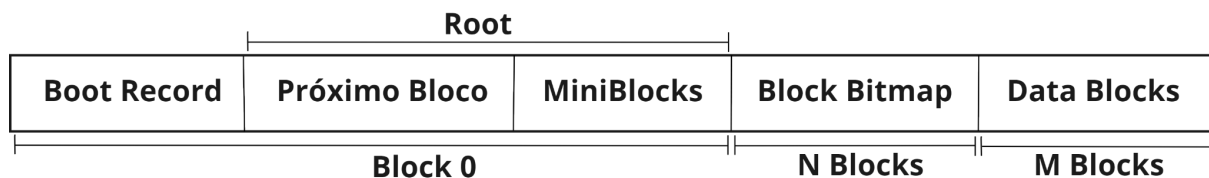


Figura 1 - Estrutura geral de um disco formatado em SuperMini.

## BOOT RECORD

O Boot Record é utilizado para armazenar informações necessárias à leitura adequada do disco. Ele segue a estrutura apresentada na Tabela 1.

Byte Inicial	Byte Final	Tamanho (Bytes)	Descrição
0	6	7	Identificador para o formato do sistema de arquivo sendo utilizado. SuperMini se inicia com 0x7375706D696E69.
7	14	8	Quantidade de blocos presentes no disco.
15	15	1	Tamanho dos blocos. O valor $n$ aqui apresentado deve ser utilizado na fórmula $2^n$ para ser obtido o tamanho em bytes.
16	23	8	Índice do superbloco que armazena as instruções de boot para sistemas operacionais cujo disco esteja formatado em SuperMini. Para sistemas de arquivos que não efetuam boot, tal campo é definido como zero.

Tabela 1 - Estrutura do Boot Record de um sistema de arquivos SuperMini.

O tamanho dos blocos é utilizado como potência de 2 para se obter o número total de bytes em cada bloco. Assim, caso o valor armazenado seja 0x0A, o número de bytes por

bloco será 1024. O tamanho mínimo de cada bloco em um sistema SuperMini é 512 (logo, o valor armazenado é 0x09) e o tamanho máximo é 4096 (valor armazenado 0x0C).

Dado que 8 bytes são utilizados para armazenar o número de blocos, o número máximo de blocos em um sistema SuperMini é  $2^{64}$ , equivalente a cerca de 18 exablocos, e o número mínimo é de 4 blocos. Visto que o tamanho máximo de cada bloco é de 4096 bytes, a capacidade máxima de um disco formatado em SuperMini é de 75.557 exabytes, a qual pode ser considerada bastante elevada.

O tamanho total do BR é de 24 bytes. Devido a isso, o Root é armazenado logo após o BR, no primeiro bloco do disco.

---

## ROOT

O Root ocupa o espaço remanescente do primeiro bloco do disco. Todavia, ele pode ser expandido e ocupar um número ainda maior de blocos. Assim, os primeiros 8 bytes do Root, conforme pode ser observado na Figura 1, representam o próximo bloco do Root, o qual estará localizado na área de dados do disco, em bloco não necessariamente inicial de tal área. Caso o Root não exceda o espaço remanescente do Bloco 0, não haverá próximo bloco e o valor de cada um dos bytes utilizados para indicar tal bloco será de 0xFF.

Em seguida, o Root armazena seus miniblocos, os quais dizem respeito aos nomes e metadados dos arquivos e pastas contidos em tal diretório. Caso o número de miniblocos seja grande, o Root ocupará múltiplos blocos, sendo dividido entre o Bloco 0 do disco e um ou mais blocos alocados à área de dados.

---

## MINIBLOCOS

Os Miniblocos armazenam os nomes ou os metadados referentes aos arquivos e pastas presentes em um diretório, sendo utilizados para descrever o conteúdo de diretórios. Eles têm duas estruturas distintas, uma utilizada para armazenar o tamanho e primeiro bloco do arquivo ou diretório e outra para armazenar o nome, sendo tais estruturas apresentadas nas Tabelas 2 e 3, respectivamente.

Byte Inicial	Byte Final	Tamanho (Bytes)	Descrição
0	0	1	Atributos do item sendo descrito. Os diferentes valores possíveis e significados são apresentados na Tabela 4.
1	7	7	Tamanho (em bytes).
8	15	8	Representa o bloco onde se inicia as informações referentes a esse item.

**Tabela 2** - Estrutura de um Minibloco de metadados do sistema SuperMini.

Byte Inicial	Byte Final	Tamanho (Bytes)	Descrição
0	0	1	Atributo. Para miniblocos de nome, é utilizado o valor 0x40.
1	15	15	Nome. Cada byte representa um caractere.

**Tabela 3** - Estrutura de um Minibloco de nome do sistema SuperMini.

A diferenciação entre tais estruturas é realizada a partir do primeiro byte de um minibloco, o qual tem valor 0x40 para miniblocos de nomes e valores distintos para miniblocos de metadados. Os possíveis valores para os atributos seguem codificação similar à utilizada pelo sistema 8.3, sendo apresentada na Tabela 4.

Valor Hex	Descrição
0x00	Minibloco desocupado (nunca utilizado).
0x01	Somente leitura.
0x02	Oculto.
0x04	Sistema.
0x08	ID Volume.
0x10	Diretório.
0x20	Arquivo.
0x40	Nome. Descreve que o minibloco é usado para representar um nome.
0x80	Minibloco desocupado (após remoção de arquivo).

**Tabela 4** - Codificação de atributos de um Minibloco do sistema SuperMini.

Para miniblocos que armazenam metadados, 7 bytes serão utilizados para indicar o tamanho em bytes do arquivo ou diretório correspondente (tal valor indica o tamanho dos dados, não o total de espaço utilizado pelo sistema SuperMini para armazená-lo). Devido a tal limitação, o tamanho máximo de um arquivo armazenado em um disco SuperMini é de  $2^{56}$  bytes, ou cerca de 72 petabytes. Os 8 bytes seguintes armazenam o índice do primeiro bloco de tal arquivo ou diretório, considerando endereçamento absoluto.

O minibloco subsequente a um minibloco de metadados representará o nome do arquivo/diretório correspondente. Seu primeiro byte terá valor 0x40, sendo os 15 bytes seguintes utilizados para armazenar os caracteres do nome, utilizando de codificação ASCII (1 caractere por byte). Caso o nome seja maior que 15 caracteres, outros miniblocos serão utilizados, não havendo limite de tamanho para o nome de arquivos ou diretórios. Em cada um desses miniblocos, o primeiro byte terá valor 0x40 e os 15 bytes seguintes serão utilizados para armazenar os caracteres.

Tendo isso em vista, um arquivo presente em um diretório tem suas informações divididas em N miniblocos, o primeiro armazenando seu tamanho e primeiro bloco de

dados, e os N-1 miniblocos subsequentes armazenando seu nome, o qual é formado pela concatenação dos caracteres presentes em cada minibloco.

---

## BLOCK BITMAP

O Block Bitmap é utilizado para gerenciar os espaços livres no disco. Para cada bloco do disco, um bit indica se o bloco está em uso (1) ou não (0). Seu tamanho, contudo, é variável, dependendo do número de blocos existentes no disco. Assim, o número de blocos utilizados pelo Block Bitmap pode ser obtido a partir da Equação 1.

Os bytes “sobressalentes” que apontam para blocos além do número total de blocos do disco são definidos como em uso (1) para que não sejam acessados acidentalmente.

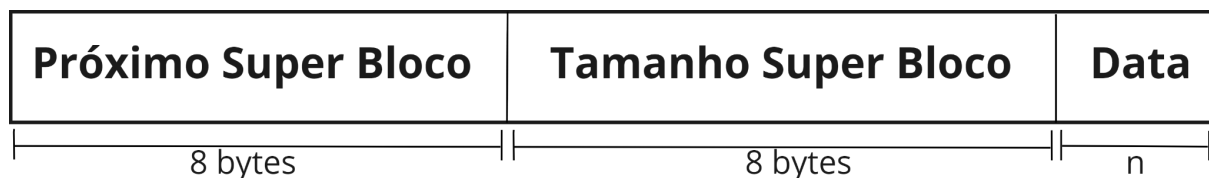
$$NUM\_BLOCOS\_BITMAP = \left\lceil \frac{NUM\_BLOCOS}{8 * BYTES\_POR\_BLOCO} \right\rceil$$

**Equação 1** - Número de blocos ocupados pelo Block Bitmap.

---

## FILE BLOCK

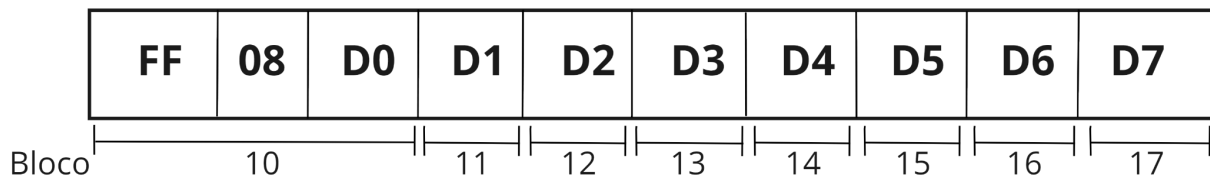
Um bloco de arquivo segue a estrutura apresentada na Figura 2. Os primeiros 8 bytes são utilizados para indicar o índice do próximo superbloco (endereçamento absoluto), enquanto os próximos 8 bytes são utilizados para indicar o tamanho do superbloco atual. Caso não haja um próximo superbloco, cada um dos 8 primeiros bytes terá valor 0xFF.



**Figura 2** - Estrutura de um bloco de arquivo no sistema SuperMini.

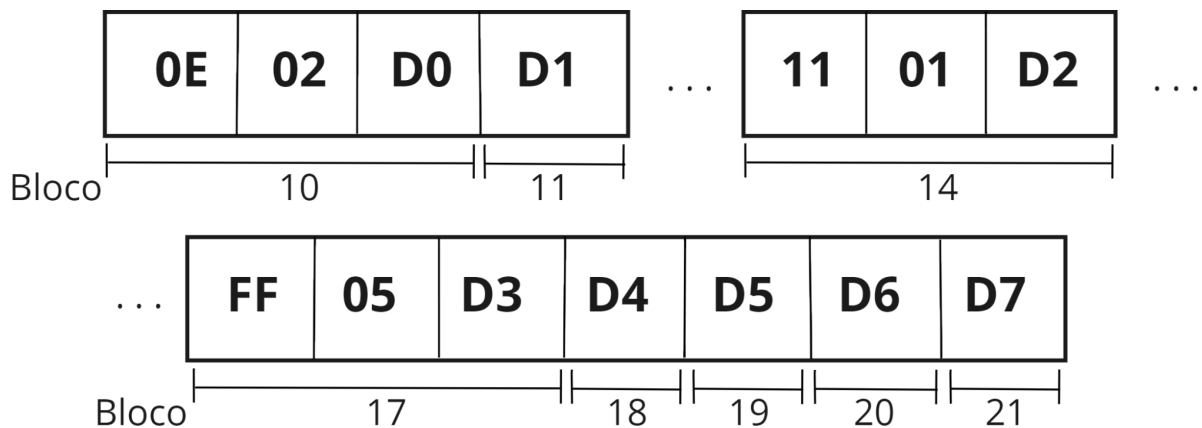
Um superbloco é composto por 1 a  $2^{64}$  blocos alocados contiguamente no disco. Apenas o primeiro bloco de um superbloco tem seus primeiros 16 bytes utilizados conforme descrito previamente, sendo tais bytes utilizados para armazenar dados para todos os blocos subsequentes. Cada superbloco tem índice igual ao de seu primeiro bloco.

Suponhamos um arquivo cujo tamanho requer 8 blocos para ser armazenado. Ao ser gravado em disco, o sistema SuperMini tentará alocar todos os seus 8 blocos de forma contígua, conforme Figura 3, gerando um único superbloco.



**Figura 3** - Alocação de um arquivo de 8 blocos em um único superbloco. Os primeiros 16 bytes do primeiro bloco indicam que este é o último superbloco do arquivo e que tal superbloco tem tamanho de 8 blocos. Os demais bytes dos 8 blocos apresentados armazenam os dados do arquivo (D0-D7).

Todavia, caso tal alocação não seja possível, o arquivo será dividido em diferentes superblocos. Cada superbloco utilizará seus primeiros 16 bytes para armazenar informações referentes ao próximo superbloco e a seu tamanho. Assim, o arquivo poderá ser alocado conforme Figura 4.



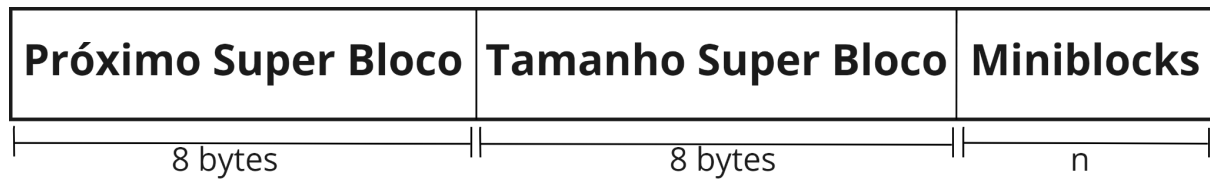
**Figura 4** - Alocação de um arquivo de 8 blocos em 3 superblocos distintos. Cada superbloco utiliza os primeiros 8 bytes para armazenar a posição do próximo superbloco do arquivo e os 8 bytes seguintes para armazenar seu tamanho.

A partir disso, a exclusão de arquivos é realizada pela leitura em disco de todos os bytes iniciais de cada superbloco de um arquivo, ocorrendo o armazenamento em memória dos índices de tais superblocos, os quais são utilizados posteriormente para modificar o Block Bitmap, zerando os bits relevantes. Nota-se que, em caso de grande fragmentação de um arquivo, tal processo torna-se bastante ineficiente, podendo exigir a leitura de N blocos do disco. Todavia, caso todo o arquivo seja gravado em um único superbloco, uma única leitura será suficiente para se obter todos os índices que devem ser zerados no Block Bitmap.

## FOLDER BLOCK

Assim como um bloco de dados, um bloco que contém uma pasta será inicializado com 8 bytes representando o próximo superbloco (endereçamento absoluto) e 8 bytes representando o tamanho do superbloco atual. Logo após tais campos, constam os miniblocos de cada arquivo e diretório presente em tal pasta, conforme Figura 5.

Devido a isso, uma pasta pode exigir que múltiplos blocos sejam utilizados para armazená-la, de modo que seus blocos seguirão a mesma lógica dos superblocos utilizada por blocos de arquivos.



**Figura 5** - Estrutura de um bloco de diretório no sistema SuperMini.