

Alunos:

Leonardo Bednarczuk Balan de Oliveira
David Antonio Brocardo
Gabriel Santos da Silva

Docente:

Marcio Seiji Oyamada

Sistemas Operacionais

Documento de Especificação do Sistema de Arquivos

Alternativa escolhida:

a) alocação contígua + gerenciamento de blocos livres com mapa de bits

Sumário

Tópico	Conteúdo
1	Gerenciamento de Blocos Livres
2	Alocação
3	Especificação dos Blocos
4	Inserção de um arquivo
5	Exclusão de um arquivo
6	Formatador
7	Listagem de arquivos
8	Cópia de arqs sistema -> disco rígido
9	Cópia de arqs disco rígido -> sistema
10	Guia de Programação extra

1. Gerenciamento de Blocos Livres

Para o gerenciamento de alocação dos blocos de memória é utilizado um **mapa de bits**, dessa forma, é declarado um vetor para representar esse mapa, onde cada posição do vetor corresponde aos blocos de memória (exemplo: índice 0 = bloco 0; índice 1 = bloco 1; etc). Assim, temos:

0	1
<i>Bloco livre</i>	<i>Bloco alocado</i>

valor da posição *k* do vetor

2. Alocação

Para a alocação de arquivos foi-se utilizado o método da **alocação contígua**, ou seja, cada arquivo é armazenado/alocado de forma sequencial, sem ter 'buracos' de memória entre seus blocos de alocação. Assim temos uma facilidade para encontrar o arquivo no sistema, somente é preciso saber seu tamanho em blocos e seu bloco inicial de alocação. Exemplo: *Disco com 10 blocos*.

Arquivos alocados	Início (bloco)	blocos alocados	Alocação no disco (blocos)
teste.txt	0	3	0, 1 e 2
arquivo.txt	4	2	4 e 5
bin.img	8	2	8 e 9

Como o disco fica:

0	1	2	3	4	5	6	7	8	9
t1	t2	t3		a1	a2			b1	b2

Em branco temos os espaços que ficam 'perdidos' na memória (fragmentação externa), sendo diretamente uma consequência da alocação sequencial de arquivos.

3. Especificação dos Blocos

Cada bloco do sistema de arquivos terá 512 bytes, dessa maneira, para obtermos o tamanho total do disco, temos a seguinte fórmula:

$$tam_total_disco = bytes_por_bloco * qnt_de_blocos.$$

Logo abaixo temos a especificação da posição dos blocos no sistema de arquivos:

Bloco 0	Boot Record
Bloco 1 + n	Mapa de Bits (pode ocupar mais de um bloco)
Bloco (1 + n) + 1	Diretório Raiz
Bloco (1 + n) + 33	Área de Dados

Melhor dizendo:

Boot Record	Mapa de Bits	Diretório Raiz	Área de Dados
--------------------	---------------------	-----------------------	----------------------

3.1 Bloco 0 : Boot Record

Por padrão, no bloco 0 sempre terá o boot record. O boot record contém as informações essenciais para o processamento do sistema de arquivos em geral. Dessa forma, temos os seguintes dados no boot record:

Deslocamento (decimal)	Tamanho (em bytes)	Significado
0	3	Código de Inicialização (caso houver instruções de boot específicas)
3	2	Número de bytes por bloco
5	1	Número de mapas de bits alocados (padrão sempre 1). São cópias do mapa de bits
6	1	Número de blocos reservados (geralmente 1 para o boot record)
7	2	Número de entradas no diretório raiz (definir de forma que o diretório raiz ocupe blocos inteiros. Fixado em 512)
9	4	Número total de blocos
13	4	Tamanho total do Mapa de Bits em blocos
17	495	Preenchimento (vazio)

3.2 Bloco 1 + n : Mapa de Bits

No bloco 1 temos o mapa de bits previamente alocado, seu tamanho varia de acordo com o tamanho do disco. Para representar a quantidade de blocos a mais que o Mapa de Bits vai ocupar, temos a variável 'n'.

3.3 Bloco (1 + n) + 1 : Diretório Raiz

Após o(s) bloco(s) do mapa de bits, temos os blocos representando o diretório raiz, no qual tem-se as informações de todos os arquivos alocados. Podemos visualizar as entradas do diretório raiz através da seguinte estrutura:

Deslocamento (em bytes)	Tamanho (em bytes)	Significado
0	11	Nome do arquivo
11	4	Primeiro bloco do arquivo
15	4	Tamanho do arquivo (em bytes)
19	1	Tipo do arquivo (por padrão todos são do tipo '0' = arquivos comuns, já que não temos sub-diretórios) tipo de arquivo: 'E5' arquivos excluídos.
20	12	Preenchimento (vazio)

32 bytes

Como temos fixado por padrão 512 entradas no diretório raiz, seu tamanho em bytes se dá pela seguinte fórmula:

$$tam_dir_raiz_bytes = num_entradas_dir_raiz * 32$$

Assim temos que o diretório raiz ocupa 16384 bytes, ou seja, 32 blocos.

3.4 Bloco (1 + n) + 33 : Área de Dados

Por fim temos os blocos da Área de Dados, onde o conteúdo dos respectivos arquivos referenciados no Diretório Raiz está localizado. Logo, esses blocos de dados preenchem todo o restante dos blocos do disco.

4. Inserção de um arquivo

Para inserir um arquivo no sistema de arquivos, temos que percorrer o mapa de bits sequencialmente no intuito de achar um espaço contíguo disponível para alocar esse arquivo, lembrando que esse espaço tem que ser maior ou igual ao tamanho em blocos/bytes do arquivo, ou seja: *espaco_disponivel* \geq *tam_arquivo*. Dessa forma, com o espaço selecionado, marcamos no mapa de bits esses blocos como alocados e devidamente fazemos a inserção do arquivo no sistema. Caso ocorrer algum erro e a inserção for impossibilitada, deve-se mostrar algum código de erro, dependendo da implementação.

Número total de blocos para armazenar um arquivo:

$$\text{ceil} (qnt_blocos_arquivo = tamanho_arquivo / bytes_por_bloco)$$

- *ceil*: função que arredonda pra cima um valor quebrado.
- *qnt_blocos_arquivo*: Número total de blocos necessários para armazenar o arquivo.
- *tamanho_arquivo*: Tamanho do arquivo em bytes.
- *bytes_por_bloco*: Número de bytes por bloco.

5. Exclusão de um arquivo

- Para excluir um arquivo temos que marcar no diretório raiz em sua respectiva entrada, no campo tipo do arquivo, o valor 'E5' em hexadecimal.
- Depois, marcar os índices dos blocos que estavam sendo ocupados no mapa de bits por com o valor '1', para '0'.
- Por fim, caso for necessário na implementação, pegar a informação de quais blocos o arquivo ocupa no mapa de bits, e, após isso, limpar/excluir o conteúdo desses blocos. Ou simplesmente deixar esses blocos com o conteúdo do arquivo excluído, e, numa futura inserção, armazenar 'por cima' desses dados um futuro arquivo.

Não necessariamente esses passos têm que ser implementados na ordem, logo, depende da escolha de implementação.

6. Formatador

Para a formatação do disco, teremos alguns dados já pré-preenchidos no Boot Record como citado anteriormente nos tópicos acima. Esses dados são:

- *Número de bytes por bloco = 512 bytes;*
- *Número de mapas de bits alocados = 1;*
- *Número de blocos reservados = 1;*
- *Número de entradas no Diretório Raiz = 512;*
- *Preenchimento = 12 bytes vazios;*

Com esses dados preenchidos, temos a cara principal do nosso sistema de arquivos, agora vamos para as informações que podem alterar de acordo com a escolha do usuário em relação à quantidade de blocos do disco. O usuário inserirá a quantidade de blocos que deseja, mas caso seja necessário calcular essa quantidade de blocos totais, temos a seguinte fórmula:

$$qnt_de_blocos = tam_disco / bytes_por_bloco$$

- *qnt_de_blocos*: Quantidade total de blocos no disco.

- *tam_disco*: Tamanho total do disco (em bytes).

- *bytes_por_bloco*: Quantidade de bytes por bloco.

Após o usuário inserir a quantidade de blocos que deseja, deve-se ser feita uma verificação no intuito de garantir que a quantidade especificada seja suficiente para alocar o sistema de arquivos, e, também, se essa quantidade for muito alta, verificar se não extrapola o limite de tamanho do disco. Essa verificação fica a cargo das decisões de implementação.

Depois dessa verificação, vamos para a parte de cálculo do tamanho do disco a ser alocado, juntamente com o tamanho do mapa de bits armazenado.

Tamanho total do disco:

$$tam_total_disco = bytes_por_bloco * qnt_de_blocos$$

- *tam_total_disco*: Tamanho total do disco em bytes.

- *qnt_de_blocos*: Quantidade total de blocos no disco.

Com a quantidade de blocos especificada pelo usuário, basta multiplicar pelos *bytes_por_bloco* (512).

Tamanho total do Mapa de Bits em blocos:

$$tam_mapa_bits_blocos = \text{ceil}(qnt_total_blocos / (8 * bytes_por_bloco))$$

- *tam_mapa_bits_blocos*: Tamanho total do mapa de bits em blocos.

No cálculo do tamanho total do mapa de bits em blocos, estamos dividindo o número total de blocos pelo tamanho de um byte multiplicado pelo número de bytes por bloco. Fazemos isso para converter o tamanho total do mapa de bits de bytes para blocos, levando em conta que um byte tem 8 bits e precisamos dividir pelo número de bytes por bloco para obter o número de blocos necessários para armazenar o mapa de bits.

Posteriormente, caso necessário, para obter o **tamanho total do mapa de bits em bytes** basta seguir a seguinte fórmula:

$$tam_mapa_bits_bytes = tam_mapa_bits_blocos * bytes_por_bloco$$

- *tam_mapa_bits_bytes*: Tamanho total do mapa de bits em bytes.

Após essas operações, temos que inicializar o mapa de bits, sendo assim, atribuímos a todas as posições dele o valor '0'. Depois, precisamos marcar com '1' os blocos pré-alocados para o Boot Record, Mapa de Bits e Diretório Raiz. Isso pode ser feito de *n* maneiras, vai da decisão de implementação.

Pode acontecer do Mapa de Bits extrapolar o tamanho total do disco e mapear endereços que 'saem' dele. Dessa maneira, deve-se ser feita uma atribuição de valores '1' para essas posições inválidas, para que nunca sejam utilizadas pelo sistema de arquivos.

7. Listagem de arquivos

Para listar os arquivos armazenados no diretório raiz, basta fazer um laço de repetição percorrendo ele e ir printando as respectivas informações desses arquivos:

nome arquivo
tamanho do arquivo
tipo do arquivo

Nesse momento não é necessário processar os arquivos, somente mostrar essas informações básicas mesmo.

8. Cópia de um arquivo do sistema de arquivos para o disco rígido

Para essa função as escolhas de projeto ficam a cargo dos programadores em busca de uma maior liberdade. Todavia, deve-se garantir que o arquivo seja corretamente copiado do sistema de arquivos para o disco rígido.

9. Cópia de um arquivo do disco rígido para o sistema de arquivos

Para implementar essa funcionalidade, basta seguir o tópico 5 'Inserção de um arquivo'. Caso apareça algumas dificuldades, pode-se fazer escolhas de implementação adicionais. Todavia, como explicado no tópico acima, também deve-se garantir que o arquivo seja corretamente copiado do disco rígido para o sistema de arquivos.

10. Guia de Programação extra

Localização do Boot Record:

Bloco 0

Localização do início do Mapa de Bits:

$mapa_de_bits = num_de_blocos_reservados * bytes_por_bloco$

- *mapa_de_bits*: Início do mapa de bits.
- *num_de_blocos_reservados*: Número de blocos reservados (geralmente 1 para o Boot Record).
- *bytes_por_bloco*: Número de bytes por bloco.

Localização do início do Diretório Raiz:

$dir_raiz = mapa_de_bits + tam_mapa_bits_bytes$

- *dir_raiz* = Início do diretório raiz.

Localização do início da Área de Dados:

$$area_de_dados = dir_raiz + (32 * num_entradas_dir_raiz)$$

- *area_de_dados*: Início da área de dados.

- 32: Tamanho de cada entrada no diretório raiz.

- *num_entradas_dir_raiz*: Número de entradas no diretório raiz (fixado em 512).