

1. Sistemas de Arquivos

1.1. Arquivos

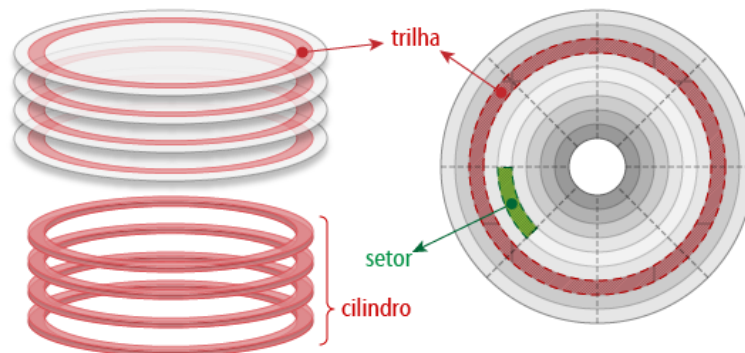
Os computadores podem armazenar informações fisicamente em diferentes formas: discos e fitas magnéticas, cartões de memória, discos óticos, entre outros. Cada um destes dispositivos possui organização e características próprias. Mas o sistema operacional deve abstrair as propriedades físicas destes dispositivos e fornecer uma visão lógica e transparente para o usuário: o arquivo.

Um arquivo é um conjunto de informações relacionadas e definidas pelo seu criador. **Geralmente, arquivos representam programas (fontes ou objetos) ou dados, que podem ser numéricos, alfabéticos ou alfanuméricos. Ele pode ser composto por uma sequência de bits, bytes, linhas ou registros,** mas seu significado é definido pelo seu criador e usuário.

Arquivos possuem nomes e algumas propriedades (também chamadas de atributos), tais como: **tipo, data de criação, nome do proprietário, tamanho etc., que são armazenadas em seu Descritor de Arquivo.** As rotinas de Entrada e Saída implementam operações sobre arquivos, tais como: criação, deleção, leitura, gravação, obtenção do tamanho e truncamento.

1.2. O Disco Rígido

Os Discos Rígidos possuem um formato circular, onde suas duas superfícies são cobertas por um material magnético. **Durante a formatação, cada superfície é dividida em trilhas, e cada trilha é dividida em setores (Também chamadas de Blocos do Disco), onde são armazenadas as informações.** As informações são lidas ou escritas através de uma cabeça de leitura/gravação.



Um setor é a menor unidade de informação que pode ser lida ou escrita no disco. Todas as operações de entrada e saída no disco, são feitas em uma certa quantidade de setores. O usuário acessa o disco através das rotinas de E/S fornecidas pelo SO. Estas rotinas permitem ao usuário realizar operações sobre os discos, sem se preocupar com detalhes tais como: como o disco está formatado, qual a trilha ou setor que o arquivo será gravado, etc. Estas rotinas também são as responsáveis pelo controle de permissões do usuário.

Conforme mencionado anteriormente, os sistemas de arquivos são armazenados em discos. A maioria dos discos são divididos em uma ou mais partições, com sistemas de arquivos independentes para cada partição. O Setor 0 do disco é chamado de **MBR** (*master boot record* – registro mestre de inicialização) e é usado para inicializar o computador. O fim do MBR contém a tabela de partição, que indica os endereços iniciais e finais de cada partição. Uma das partições na tabela é marcada como ativa. Quando o computador é inicializado, a BIOS lê e executa o MBR. A primeira coisa que o programa do MBR faz é localizar a partição ativa, ler seu primeiro bloco, chamado de **Bloco de Inicialização**, e executá-lo. O programa no bloco de inicialização carrega o sistema operacional contido naquela partição.

1.3. O que é um Sistema de Arquivos?

Um Sistema de Arquivos, também conhecido com *FileSystem*, funciona basicamente como um intermediário entre o disco rígido e os arquivos armazenados no computador. O Disco Rígido tem a função de armazenar todos os dados dos documentos do seu PC, porém, ele por si só não sabe como organiza-los em seu espaço. Nesse ponto o Sistema de Arquivos entra em ação, pois é ele quem dita as regras de organização no disco rígido. Como curiosidade, temos de um lado os sistemas operacionais Windows/DOS que utilizam Sistemas de Arquivos FAT, FAT32, NTFS, e de outro lado, Sistemas Operacionais como o Linux que utilizam Sistemas de Arquivos totalmente diferentes, como por exemplo EXT2, EXT3 e ReiserFS. FAT32 possui uma organização totalmente diferente do Sistema NTFS.

Ok! Mas então, como o Sistema Operacional armazena arquivos no disco?

1.4. Alocação de Espaço em Disco

A questão mais importante na implementação do armazenamento de arquivos talvez seja a manutenção do controle de quais blocos de discos estão relacionados a quais arquivos. Diversos métodos são utilizados em diferentes Sistemas Operacionais. A seguir, são apresentados alguns dos métodos mais utilizados.

1.4.1. Alocação contígua

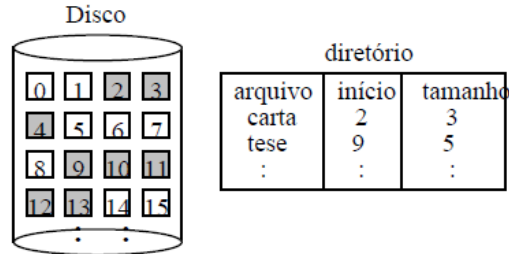
A alocação contígua é o sistema mais simples de alocação, onde o intuito é armazenar cada arquivo em blocos contínuos de disco. Sendo assim, este método procura identificar uma sequência contínua de blocos que apresente um tamanho suficiente para armazenar todas as informações relativas ao arquivo. Ou seja, caso um arquivo precise de N blocos para o seu armazenamento, o sistema operacional procurará identificar uma sequência contínua de N blocos livres do disco.

Assim, em um disco com blocos de 1KB, um arquivo com 50 KB seria alocado em 50 blocos consecutivos. Já com blocos de 2KB, o arquivo seria alocado em 25 blocos consecutivos.

Na técnica de alocação contígua, o espaço só será alocado se for contínuo, caso contrário, o arquivo não poderá ser armazenado ou estendido. Se por acaso existir mais

de uma possibilidade de armazenamento para o arquivo em questão, poderá ser utilizado uma técnica para selecionar uma opção. Podem ser utilizadas as técnicas Worst-Fit, Best-Fit e First-Fit.

A alocação contígua de um arquivo é definida pelo endereço do primeiro bloco e seu comprimento, e dessa forma é colocado no diretório, como mostra a figura abaixo.

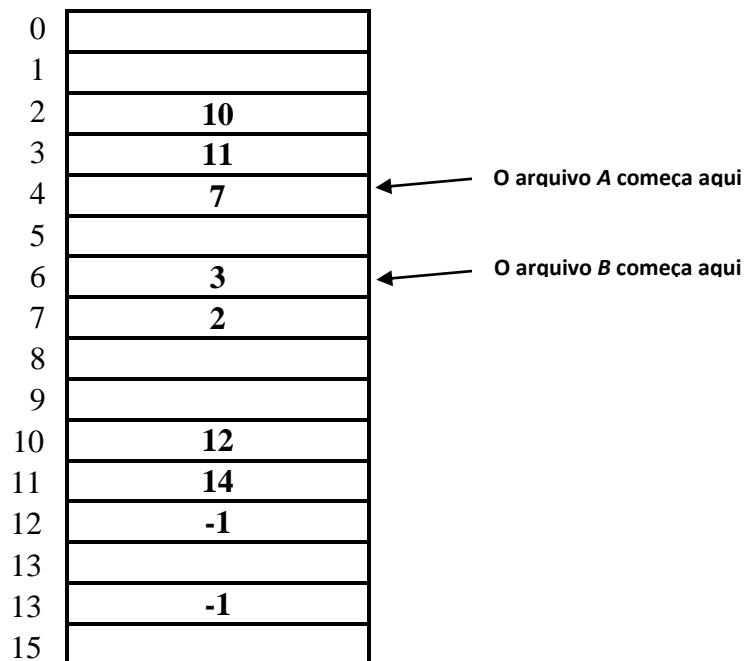


O método de alocação contígua apresenta dois principais problemas:

- Encontra o espaço contíguo suficiente para o arquivo. Isto pode causar fragmentação externa, isto é, existe espaço no disco mas não é contíguo.
- Determinar quanto de espaço uma arquivo que está sendo criado irá precisar. Deve-se portanto estimar uma quantidade de blocos. Se for pouco, o arquivo não poderá crescer, e se for muito, o disco poderá estar sendo subutilizado caso o arquivo for pequeno.

1.4.2. Alocação por lista encadeada utilizando uma tabela na memória

As desvantagens da alocação por lista encadeada podem ser eliminadas colocando-se as palavras do ponteiro de cada bloco em uma tabela na memória. A figura abaixo mostra como seria a tabela para o exemplo apresentado na figura do método de lista encadeada.



O arquivo *A* utiliza os blocos 4,7, 2, 10 e 12 e o arquivo *B* utiliza os blocos 6,3,11 e 14, ambos nesta ordem. Seguindo a estrutura da figura acima, partindo do bloco 4, é possível seguir o encadeamento até o final do arquivo. O mesmo pode ser feita a partir do bloco 6. Ambos os encadeamentos têm uma marca de término (neste caso, -1) que corresponde a um número inválido de bloco. Essa tabela na memória principal é chamada de FAT (file allocation table) ou tabela de alocação de arquivos.

A principal desvantagem deste método é que, para funcionar, toda a tabela deve estar na memória o tempo todo. Para um disco de 200 GB e blocos de 1KB, a tabela precisará de 200 milhões de entradas, uma para cada um dos 200 milhões de blocos de disco. Cada entrada tem no mínimo 3 bytes. Para aumentar a velocidade de consulta elas deveriam ter 4 bytes. Portanto, a tabela ocupará 600 MB ou 800 MB de memória principal o tempo todo, o que não é muito prático. Sendo assim, a idéia de organização da FAT claramente não engloba discos grandes.

Ok! E como o Sistema Operacional faz a leitura dos arquivos no disco?

1.5. Escalonamento de Disco

Para que o sistema operacional atenda uma solicitação de acesso ao disco por uma aplicação, ele gasta um certo tempo. Este tempo é equivalente a somatória do tempos gastos em 3 etapas:

- **Tempo de Seek:** tempo gasto para locomover a cabeça de leitura/gravação da trilha atual para a trilha desejada;
- **Tempo de Latência:** tempo gasto para esperar que o disco rotacione até que o bloco desejado esteja sob a cabeça de leitura/gravação.
- **Tempo de Transferência:** tempo gasto para transferir os dados do bloco do disco para a memória principal.

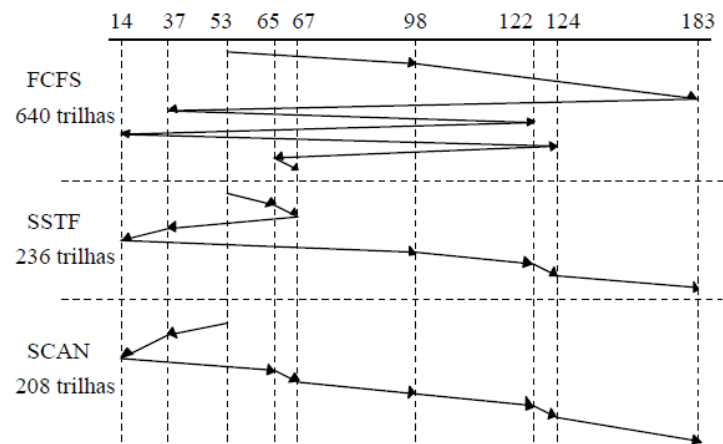
Em um sistema multiprogramado, várias solicitações de acesso ao disco são feitas pelas diferentes aplicações, de forma que o sistema operacional deve ser hábil em escalonar qual solicitação atender, visando obter o menor tempo médio nos acessos ao disco, para obter o melhor desempenho do sistema.

As principais políticas de escalonamento do disco são: *First-Come-First-Served*, *Shortest-Seek-Time-First* e *SCAN*.

- **FCFS:** As solicitações são atendidas por ordem de chegada (FIFO)
- **SSTF:** A solicitação cuja trilha estiver mais próxima da trilha atual da cabeça de leitura/gravação do disco é atendida primeiro.
- **SCAN:** O sistema atende as solicitações que estiverem em um determinado sentido (da trilha 0 em direção a última trilha ou vice-versa), e quando não houver mais solicitação naquele sentido, o sistema inverte o sentido e repete o processo. Desta forma, a cabeça de leitura/gravação executa uma varredura de um lado para o outro do disco.

Para exemplificarmos as três técnicas, vamos supor que existe uma fila de requisição de acesso ao disco, onde as trilhas que devam ser acessadas estão na seguinte sequência 98, 183, 37, 122, 14, 124, 65 e 67, de forma que a primeira solicitação que foi feita é a trilha 98 e a última é a 67. Supomos que a cabeça de leitura/gravação do disco

esteja na posição 53. O gráfico abaixo mostra o caminho percorrido utilizando-se as 3 técnicas.



Para fazermos um comparativo, podemos calcular a quantidade de trilhas percorridas em cada sistema:

$$\text{Trilhas Percorridas (FCFS)} = 45+85+146+85+108+110+59+2=640$$

$$\text{Trilhas Percorridas (SSTF)} = 12+2+30+23+84+24+2+59=236$$

$$\text{Trilhas Percorridas (SCAN)} = 16+23+51+2+31+24+2+59=208$$