

AULA 16 - Sistema de Arquivos

Arquivos podem ser vistos como recipientes que contêm dados ou como um grupo de registros correlatos. Os arquivos armazenam informações que serão utilizadas, em geral, por programas aplicativos. Os arquivos são implementados através da criação, para cada arquivo no sistema, de uma estrutura de dados composta por registros. O descritor de arquivo é um registro que mantém informações sobre o arquivo, como posição de início e fim.

As informações típicas (atributos) mantidos pelo sistema operacional são:

1. Nome do arquivo
2. Tamanho (bytes)
3. Data e hora da criação, do último acesso, da última modificação
4. Identificação do usuário que criou o arquivo
5. Listas de controle de acesso
6. Local do disco físico onde o conteúdo do arquivo foi colocado

As operações mais comuns relacionadas à manipulação dos arquivos são:

Create	Delete	Open	Close	Read	Write
Append	Seek	Get attributes	Set attributes	Rename	

Para controlar e organizar os arquivos, os sistemas de arquivos têm, em geral, os *diretórios ou pastas*, que podem ser arquivos em muitos sistemas também. Os diretórios podem ser organizados em um único nível (contendo todos os arquivos) ou em múltiplos níveis (diretórios dentro de diretórios).

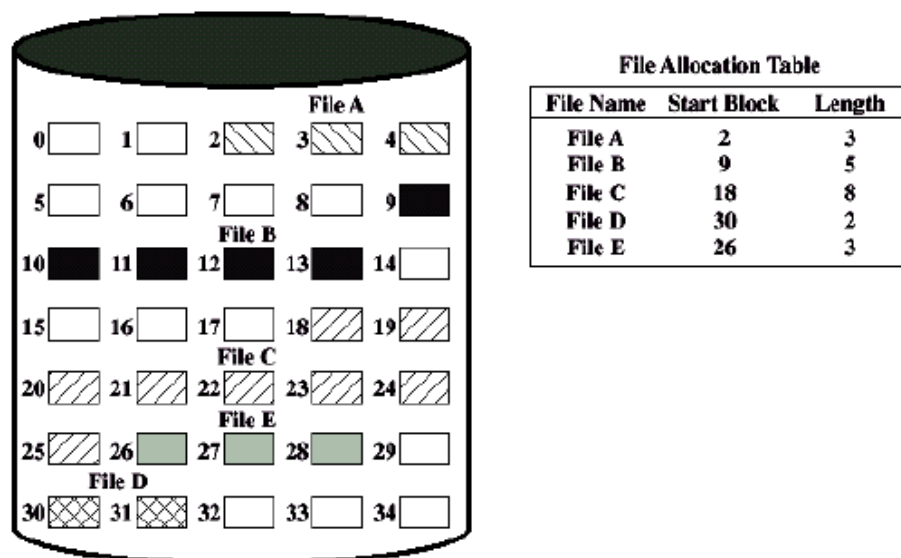
Um dos principais problemas é como alocar espaço em disco para que os arquivos sejam armazenados de forma eficiente e que permita acesso rápido?

Existem alguns métodos que podem ser utilizados, tais como:

1. Alocação contígua
2. Alocação com Lista Ligada
3. Alocação com Lista Usando um Índice

Alocação Contígua

Este é o esquema mais simples de alocação de arquivos, onde cada arquivo é armazenado no disco como um bloco contíguo de dados. Neste esquema, em um disco com blocos de 1 KB, um pequeno arquivo de 20 KB seria armazenado em 20 blocos consecutivos.



W. Stallings. *Operating Systems (4th Ed.)*, Prentice Hall, 2001.

Figura 1. Alocação contígua.

Principais Vantagens:

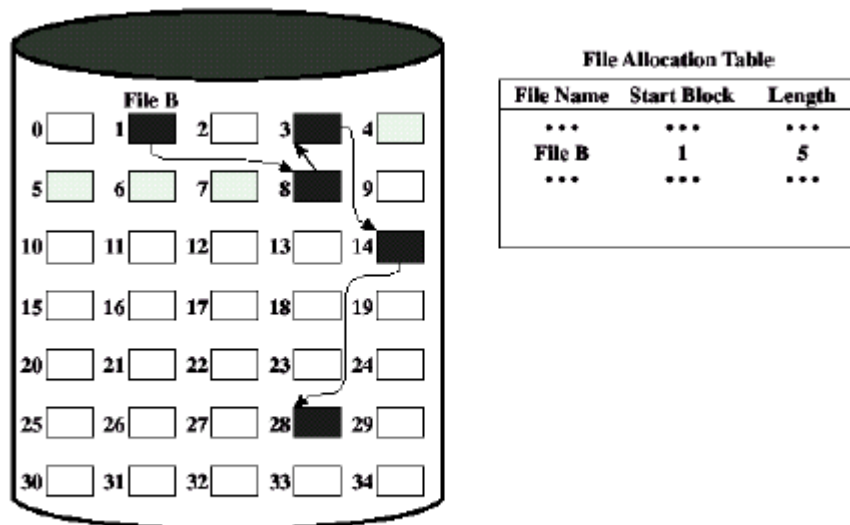
1. simples implementação: controle de onde está cada arquivo no disco é feito por 1 único número (endereço em disco do 1º bloco).
2. performance: todo o bloco (arquivo) pode ser lido do disco de uma única vez. É necessário o tempo de somente um *seek*.

Problemas:

1. a estratégia só pode ser usada se o tamanho Máx. do arquivo for conhecido no momento de sua criação (devido a necessidade existente em saber o tamanho total do arquivo ou quantidade de blocos que ele ocupa).
2. fragmentação do disco: perde-se muito espaço útil com este esquema de alocação. Ao remover um arquivo a área ocupada pelo mesmo é liberada ocasionando lacunas por todo o disco. Necessidade de compactação (custo alto).

Alocação com Lista Ligada

Nesta estratégia de alocação, usamos uma lista ligada para indicar os espaços ocupados em disco pelo arquivo. Assim, não é mais necessário que o arquivo seja armazenado em posições contíguas do disco. A primeira palavra de cada bloco é usada com um ponteiro para o próximo bloco e o restante do bloco é usado para armazenar as informações (dados) do arquivo.



W. Stallings. *Operating Systems (4th Ed.)*, Prentice Hall, 2001.

Figura 2. Alocação com lista ligada.

Principais Vantagens:

1. não se perde espaço por fragmentação externa.
2. qualquer bloco pode ser utilizado, permitindo que os arquivos cresçam indefinidamente enquanto houver espaço no disco.

3. a entrada do diretório só precisa armazenar o endereço do 1º bloco do arquivo (em cada bloco existirá um ponteiro para o próximo bloco do arquivo).

Problemas:

1. o acesso randômico é lento pois existe a necessidade de percorrer a lista.
2. a implementação deste método de alocação é mais complicada.

Alocação com lista ligada usando uma Tabela na Memória

Para eliminarmos os problemas apresentados por último, podemos dispor de uma tabela em memória armazenando os ponteiros para cada bloco do arquivo. Esta tabela recebe o nome de **FAT (File Allocation Table)** - esquema adotado pelo DOS. Com este esquema o acesso aleatório fica muito mais fácil pois todo o esquema de ponteiros agora fica armazenado em memória, o que é muito mais rápido. O principal problema é o gasto com memória para manter a tabela com as informações. Para um disco de 40 GB e blocos de 1 KB, a tabela precisará de 40 milhões de entradas. Considerando que cada entrada tem no mínimo 3 bytes a tabela ocupará um espaço total de 120 MB.

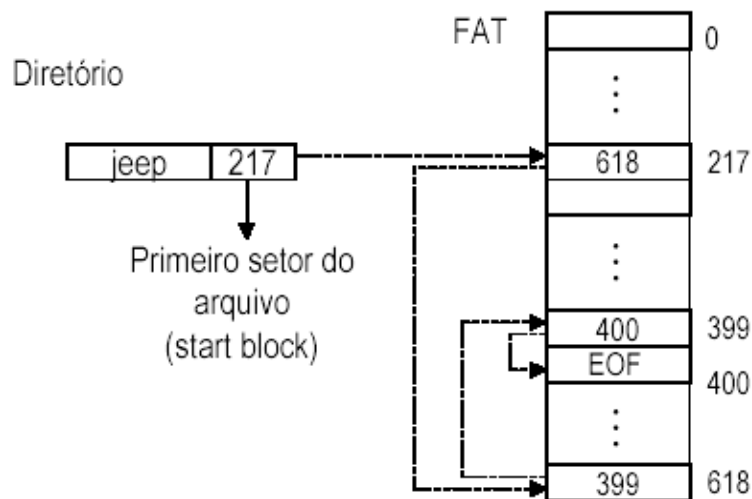


Figura 3. Tabela de alocação de arquivos (FAT).

Alocação com Lista Usando um Índice

Busca resolver o problema de “ponteiros” esparramados pelo disco que a alocação encadeada provoca. Para isso, mantém, por arquivo, um índice de blocos que o compõe. Este o índice é mantido em um bloco do disco. O diretório possui um ponteiro para o bloco onde está o índice associado a um determinado arquivo.

Este esquema elimina as desvantagens existentes na alocação com lista ligada: retira os ponteiros de cada um dos blocos e os coloca em uma tabela ou índice na memória. Apesar de o acesso ser randômico também, sua implementação é bem mais simples.

A tabela é armazenada na memória principal e pode ser seguida sem a necessidade de acessar o disco.

Desvantagem: a tabela também deve estar na memória o tempo todo, o que implica em utilização de espaço de memória.

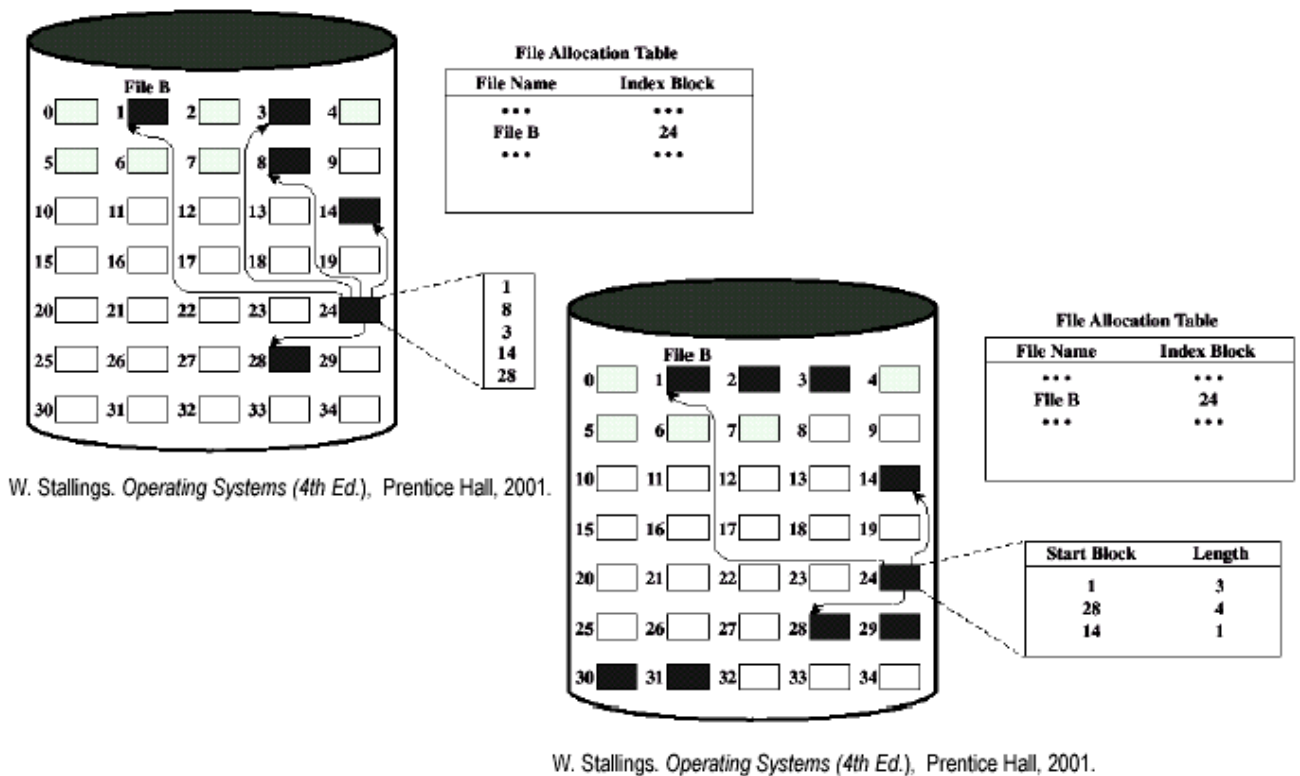
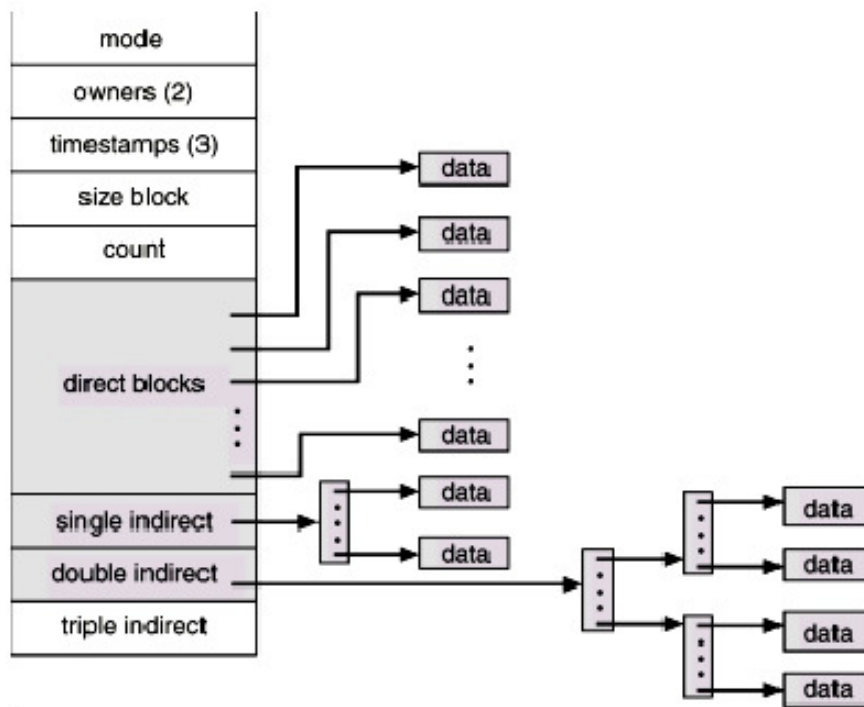


Figura 4. Alocação com lista usando um índice.

Inode

A cada arquivo associa-se uma pequena tabela, denominada Inode, que lista os atributos e os endereços em discos dos blocos do arquivo. Os primeiros endereços de disco são armazenados no próprio Inode. Se o arquivo for pequeno, toda a informação é encontrada diretamente no Inode. O conteúdo do arquivo só é transferido do disco para a memória quando o arquivo for aberto (esquema utilizado pelo UNIX).



W. Stallings. *Operating Systems (4th Ed.)*, Prentice Hall, 2001.

Figura 5. Inode.

Exercícios

1. O que é um arquivo?
2. Como os arquivos podem ser organizados?
3. Diferencie os métodos de acesso: seqüencial, direto e indexado.
4. O que é alocação contígua de blocos e quais benefícios a desfragmentação pode proporcionar quando esta técnica é utilizada?
5. Quais os tipos de proteção de acesso a arquivos existentes e quais suas principais vantagens?
6. Em alguns sistemas operacionais é possível mapear parte de um arquivo na memória. Quais restrições devem ser impostas a esses sistemas? Como é implementado esse mapeamento parcial?
7. Apresente as diferenças entre os sistemas FAT32 e NTFS.

Bibliografia

- SILBERSCHATZ, A. et al. **Sistemas Operacionais: conceitos e aplicações**. Campus, 2001.
- STALLINGS, W. **Operating Systems**. Prentice Hall, 4th Ed, 2001.