... 3.4 - Desempenho do Sistema de Arquivo

- Outra forma de aumentar o desempenho do sistema de arquivos é reduzir a quantidade de movimentos do braço do disco, colocando blocos que estão sendo acessados em seqüência, preferencialmente em um mesmo cilindro.

- ... quando um arquivo é escrito, o sistema de arquivos aloca os blocos uma por vez, a medida do necessário e, neste caso, o desempenho depende se a lista de blocos livres está na memória principal ou no disco;

- ... de qualquer modo com uma lista de blocos livres, agrupamentos de blocos podem ser feitos de modo a manter a trilha do disco armazenada não em blocos, mais em grupos consecutivos de blocos como por exemplo:

  ☞ ... se uma trilha consistir de 64 setores de 512 bytes, o sistema pode usar blocos de 1 KB (2 setores), porém alocando espaço no disco em unidades de 2 blocos (4 setores) tem-se um ganho de busca de um fator de 2;

  ☞ ... isto não é o mesmo que ler um bloco de 2K, posto que na *cache* ainda se usa blocos de 1K com transferência para disco também de 1K.

... 3.4 - Desempenho do Sistema de Arquivo

- Outra variante é fazer uso do posicionamento rotacional - o sistema atenta para colocar blocos consecutivos de um arquivo no mesmo cilindro, mas intercalados.

- ... por exemplo, considere um disco com tempo de rotação de 16,67 ms e 4 ms são necessários para o processo do usuário requerer e acessar um bloco do disco — qual será o espaçamento de cada bloco ?

✯ Agravantes no desempenho dos sistemas que usam *inodes* - mesmo para um pequeno arquivo, 02 acessos são necessários: um para o *inode* e outro para o bloco.

  - caso o *inode* esteja próxima do início do disco, a distância média entre o *inode* e o bloco estará em torno da metade do número de cilindros;
  - melhor alternativa - colocar o *inode* no meio do disco para reduzir a média de busca entre o *inode* e o primeiro bloco de um fator de 2;
  - outra idéia - dividir o disco em grupos de cilindros, cada qual com os seus próprios *inodes*, blocos e lista de blocos livres.

3.4 - Desempenho do Sistema de Arquivo

- ... o desempenho de um sistema de arquivo está diretamente associado ao tempo necessário para acessar as informações e como resultado, muitos sistemas de arquivos têm sido projetados para reduzir o número de acessos ao disco;

- ... a técnica mais comum para reduzir o acesso a disco é a *cache* de *buffer* ou de bloco, ou seja, uma coleção de blocos que pertencem logicamente ao disco, mas que são mantidos na memória por razões de desempenho;

- ... neste processo, blocos precisam ser removidos para dar lugar a outros que serão armazenados e assim como na paginação de memória, todos os algoritmos usuais de paginação podem ser aplicados neste contexto;

- ... se um bloco essencial para a consistência do sistema de arquivo foi modificado na *cache*, é necessário que o mesmo seja escrito no disco imediatamente para reduzir a probabilidade de que falhas danifiquem o sistema de arquivo.

... 3.4 - Desempenho do Sistema de Arquivo

- ... embora estas medidas mantenham a integridade do sistema de arquivo, não é usual manter blocos de dados durante muito tempo na *cache*, assim sistemas de arquivos adotam 02 estratégias:

  ❶ no **unix**, uma chamada de sistema **sync** força com que todos os blocos modificados sejam gravados em disco imediatamente.
  - ... quando o sistema é iniciado, um programa, usualmente chamado *update* é ativado e de 30 em 30 segundos, a atualização da *cache* é estabelecida.

  ❷ no **ms-dos** a solução é gravar todo bloco modificado para o disco tão logo ele tenha sido escrito, ou seja, *cache* de escrita direta;
  - ... embora a *cache* de escrita direta cause muito mais entrada/saída de disco que em *caches* de escrita não direta, esta estratégia foi adotada pelo **ms-dos** como forma de garantir que a remoção de um disco flexível não cause perdas de dados;
  - ... no **unix** é necessário uma chamada **sync** antes da remoção de qualquer meio de armazenamento ou da parada programada do sistema.

... 3.3.3 - Consistência do Sistema de Arquivo

❶ se a contagem no *inode* for maior que o número de registros do diretório, então mesmo se todos os arquivos forem removidos dos diretórios, o contador ainda será diferente de 0 e o *inode* não será liberado;

- ... embora não seja um erro catastrófico, há consumo de espaço em disco com arquivos que não estão em nenhum dos diretórios e portanto cabe atualizar o contador de conexões do *inode*.

❷ se a contagem no *inode* for menor que o número de registros do diretório, este erro pode causar danos catastróficos:

- ... se arquivos que apontam para o *inode* vão sendo removidos, o contador do *inode* pode chegar a zero, momento que o *inode* e seus respectivos blocos são liberados;

- ... a solução é forçar o contador do *inode* para o número real de registros do diretório, obtido pelo utilitário.

... 3.3.3 - Consistência do Sistema de Arquivo

- Finalmente, estas operações de verificar blocos e arquivos são freqüentemente integradas por razões de eficiência, ou seja, uma única leitura nos *inodes* é requerida.

✍ ... controles heurísticos podem também ser utilizados, por exemplo:

  - diretórios têm um formato definido, com um número de *inodes* e nomes ASCI;
  - ... se um número *inode* for maior que o número de *inodes* no disco, o diretório encontra-se num estado inconsistente.

... 3.3.3 - Consistência do Sistema de Arquivo

- ... se o sistema for consistente, cada bloco terá o valor 1 na primeira tabela ou na segunda tabela, contudo em caso de falha, pode detectar-se blocos que não ocorrem em nenhuma das tabelas (blocos perdidos);

❶ ... embora blocos perdidos não causem dano real, eles desperdiçam espaço reduzindo a capacidade do disco e, por isso, são acrescentados na lista de blocos livres;

❷ outra possibilidade é a repetição de blocos na lista de blocos livres que pode ser resolvido reconstruindo-se a lista de blocos livres;

❸ o mais grave é a ocorrência do mesmo bloco de dados em 02 ou mais arquivos e em caso de remoção de um arquivo o bloco estará ambigüamente em uso e livre ao mesmo tempo — uma ação apropriada precisa ser tomada:

  - a acão apropriada é alocar um bloco livre, copiar o conteúdo do bloco duplicado nele e inserir a cópia em um dos arquivos, tornando a estrutura do sistema de arquivo, pelo menos, consistente embora um erro seja reportado.

... 3.3.3 - Consistência do Sistema de Arquivo

- falta ainda a consistência no nível de arquivo e, neste caso, é usada uma tabela de contadores por arquivos, não por blocos, como anteriormente;

- ... a verificação começa no diretório raiz e, recursivamente, desce a árvore inspecionando cada diretório no sistema de arquivos e incrementando o contador para o seu respectivo $inode$ de cada arquivo encontrado;

- ... quando toda a lista é percorrida, tem-se uma lista indexada pelo número do $inode$, descrevendo quantos diretórios apontam para aquele $inode$ permitindo ao utilitário comparar os valores com os contadores dos $inodes$;

- ... em um sistema de arquivo consistente, ambos contadores coincidirão.

✘ ... 02 tipos de erros podem ocorrer: o contador do $inode$ ser maior ou menor que o valor obtido da lista do utilitário.

3.3.2 - Backups

- ... mesmo com uma estratégia engenhosa para tratar os blocos defeituosos, é importante proceder *backups* freqüentes, por exemplo utilizando fita magnética.

- Algumas alternativas possíveis são apresentadas:

  ❶ prover o sistema computacional de um segundo disco de igual capacidade com ambos divididos com igual volume de dados e *backups*;
  ❷ prover o sistema computacional com unidades de fita magnética que embora tenham grande capacidade apresentam o inconveniente de consumirem mais tempo se comparados com a proposta anterior;
  ❸ *backup* incremental - copia-se para unidade de fita todos os arquivos a cada semana ou mês e, diariamente, apenas aqueles arquivos que foram modificados desde o último *backup* completo.
  ❸ variação do *backup* incremental - copia-se apenas os arquivos que foram alterados desde o último *backup* o que por outro lado implica manter o horário da última duplicação para cada arquivo em disco.

3.3.3 - Consistência do Sistema de Arquivo

- ... um sistema de arquivo pode se tornar inconsistente desde que ocorra(m) falha(s) durante o processo de atualização de suas informações;

- ... este problema é especialmente crítico se alguns dos blocos que não foram escritos, forem blocos de *inodes*, blocos de diretório ou blocos contendo a lista de blocos livres — nestes casos, fatalmente a consistência estará comprometida;

- para verificar a consistência do sistema de arquivo, muitos sistemas operacionais utilizam programas utilitários desenvolvidos para este fim e que são executados sempre que o sistema é iniciado, particularmente depois de um desligamento abrupto.

✭ No unix, o controle de consistência se dá em 02 níveis: blocos e arquivos.

  - no nível de bloco, o utilitário constrói uma tabela com 02 contadores por bloco, ambos iniciados em 0 sendo que o primeiro rastrea quantas vezes o bloco aparece no arquivo e o segundo a freqüência com que ele aparece na lista de blocos livres.

3.3 - Confiabilidade do Sistema de Arquivo

- Dentre os tópicos envolvendo confiabilidade de um sistema de arquivo, destacamos:

✯ ... discos freqüentemente apresentam blocos defeituosos, causando problemas ou impossibilitando operações de escrita e/ou leitura;

✯ ... mesmo tratando os blocos defeituosos, é importante efetuar *backups* freqüentes;

✯ ... um sistema de arquivo pode se tornar inconsistente desde que ocorra falha durante o processo de atualização de suas informações.

- Com base no exposto, descrevemos nas próximas seções cada um dos tópicos anteriormente mencionados e que sem o devido tratamento podem causar sérios danos a um Sistema de Arquivo.

3.3.1 - Blocos Defeituosos

- ... discos freqüentemente presentam blocos defeituosos (*bad blocks*) e para este problema 02 soluções podem ser empregadas:

❶ **solução em hardware** - consiste em dedicar um setor no disco para uma lista de blocos defeituosos, assim quando o controlador de disco é iniciado, este lê a lista de blocos defeituosos e escolhe blocos sobressalentes para substituí-los.

- ... são feitas então indireções dos blocos defeituosos para os blocos sobressalentes permitindo que operações envolvendo um bloco defeituoso tenham efeito em seu respectivo bloco sobressalente.

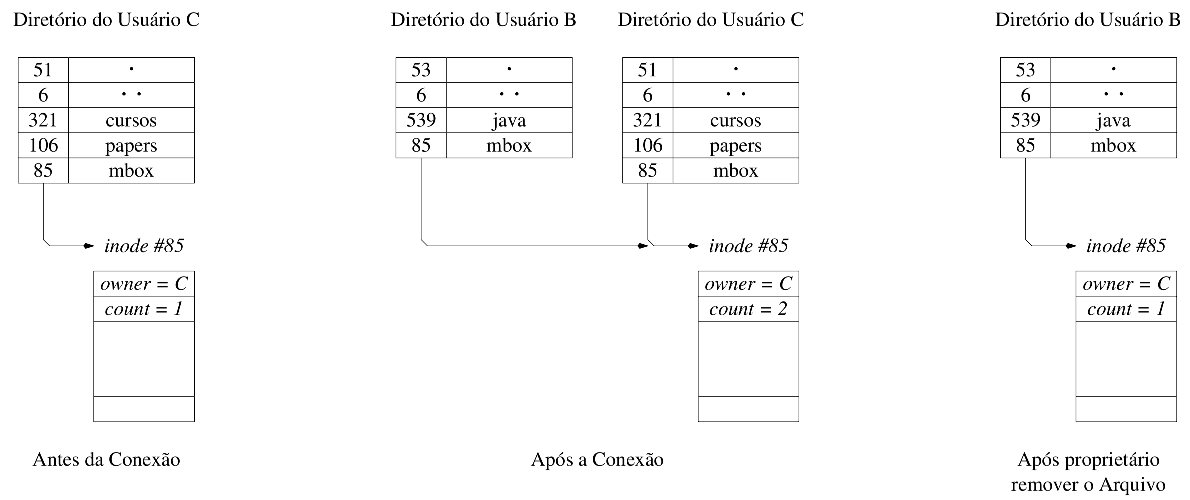
❷ **solução por software** - requer que o usuário informe ou que o sistema de arquivo detecte os blocos defeituosos e assim o sistema operacional não os incorpora na lista de blocos livres do disco o que permite eliminar sua ocorrência futura.

- ... estes blocos são armazenados num arquivo criado quando da construção da lista de blocos livres do dispositivo de armazenamento.

... 3.2.4 - Arquivos Compartilhados

✘ conexões simbólica, no entanto, introduzem um *overhead* extra na manipulação de arquivos, pois vários *inodes* devem ser lidos no disco.

✘ ... adicionalmente, um *inode* extra é necessário para cada conexão simbólica, juntamente com um bloco extra para armazenar o caminho.



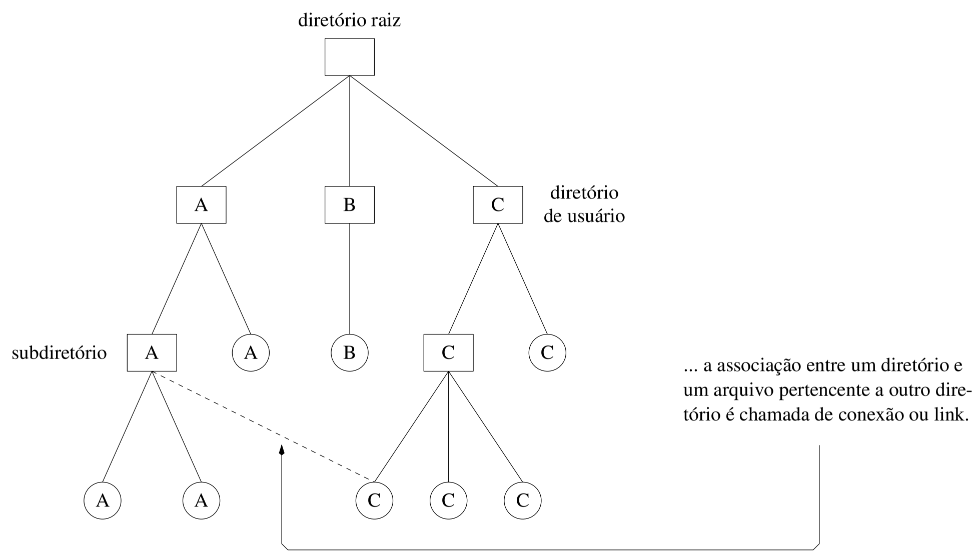
... 3.2.4 - Arquivos Compartilhados

- ... por fim, existe um outro problema introduzido pelas conexões simbólicas ou não:

✘ ... quando conexões são permitidas, uma pesquisa numa árvore de diretórios pode encontrar o mesmo arquivo várias vezes, o que pode ser um grande problema, por exemplo para aplicativos que efetuam *backups*.

3.2.4 - Arquivos Compartilhados

- não raro, usuários compartilham arquivos e assim é conveniente que um mesmo arquivo pertença simultaneamente a diferentes diretórios.



... 3.2.4 - Arquivos Compartilhados

- ... para o compartilhamento faz-se necessário mecanismos eficientes capazes de manter a consistência da informação sendo compartilhada:

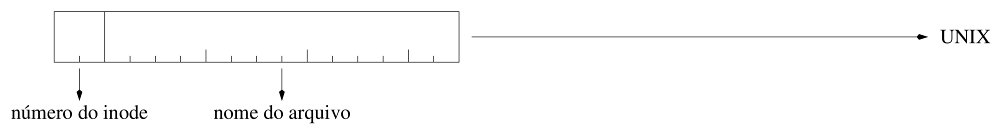
❶ no unix os diretórios apontam para o *inode* do arquivo;
❷ **conexão simbólica** - outra possibilidade é criar um arquivo que contém o caminho do arquivo conectado e assim o sist. oper. poderá identificar o caminho para o arquivo compartilhado.

- cada um desses métodos apresenta suas desvantagens:

✘ no primeiro caso, como remover um arquivo dado que a criação de uma conexão não muda o proprietário do arquivo, mas incrementa um contador no *inode* informando quantos diretórios apontam para o arquivo ?!
✔ já no segundo caso, este problema não ocorre pois somente o diretório proprietário guarda o ponteiro para o *inode* enquanto que diretórios que estabeleceram conexão guardam somente o caminho para o arquivo.

... 3.2.3 - Estrutura de Diretório

- já a estrutura de diretório usada no UNIX é extremamente simples: cada registro contém exatamente a estrutura do arquivo e seu número de *inode*.

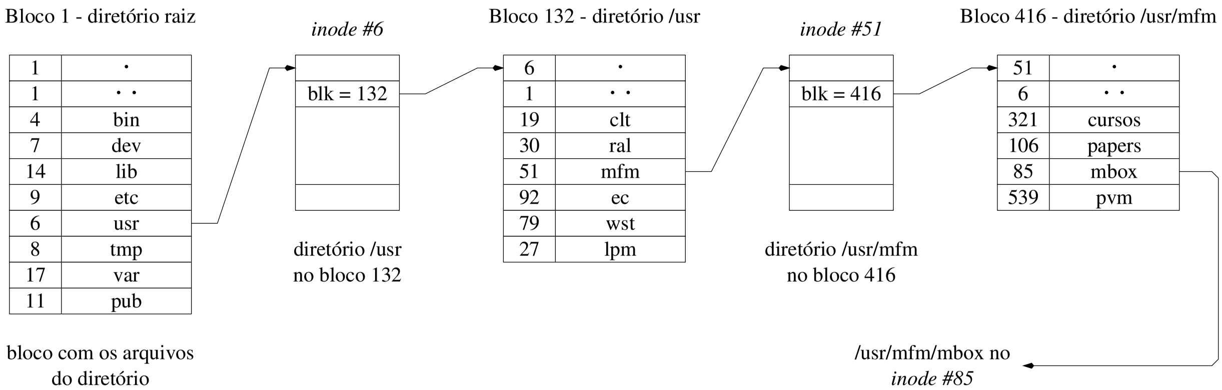


- ... assim, todas as demais informações como tipo, tamanho, marcas de tempo, propriedade e blocos do disco estão contidas no *inode*;

- ... todos os diretórios no UNIX são arquivos e podem conter um número arbitrário destes registros, mas todo diretório necessariamente tem o *inode* para o diretório corrente **.** e o *inode* para o diretório pai **..**

... 3.2.3 - Estrutura de Diretório

- considere o exemplo para achar: **/usr/mfm/mailbox**

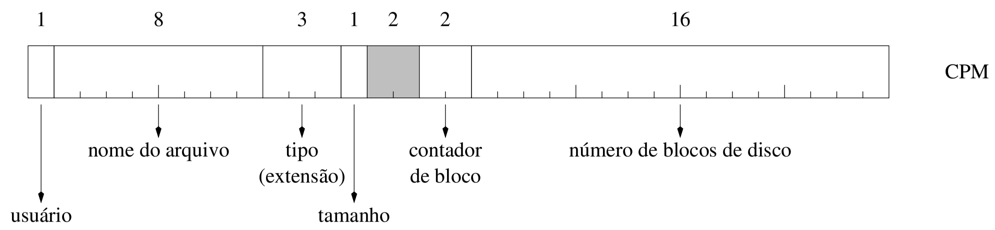


- ... nomes de caminhos relativos são pesquisados de forma idêntica, apenas partindo do diretório de trabalho ou diretório corrente **.** em vez de partir do diretório raiz;

- ... nenhum mecanismo especial é necessário para manipular estes nomes.

3.2.3 - Estrutura de Diretório

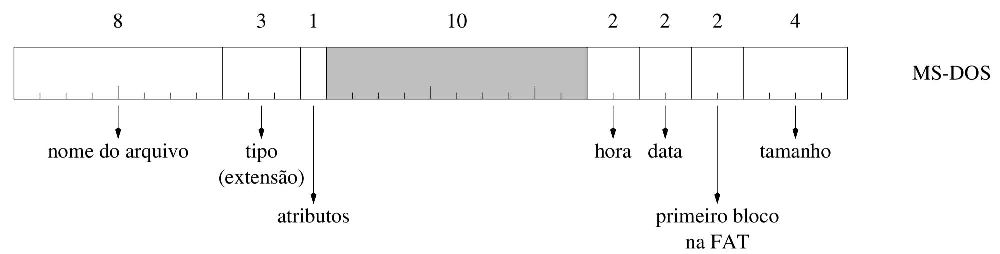
- Um arquivo para ser manipulado precisa ser aberto e, nesse sentido, o sist. oper. usa o nome de caminho fornecido pelo usuário para localizar os blocos no disco.
- ... mapeando-se nomes de caminhos em *inodes* ou equivalentes, introduz-se o tópico de como sistemas de diretórios são organizados.



- **campo do usuário** - armazena a qual usuário pertence o arquivo;
- **campo tamanho** - necessário para especificar um arquivo com mais de 16 blocos;
- **contador de blocos** - informa quais dos 16 blocos de disco estão em uso;

... 3.2.3 - Estrutura de Diretório

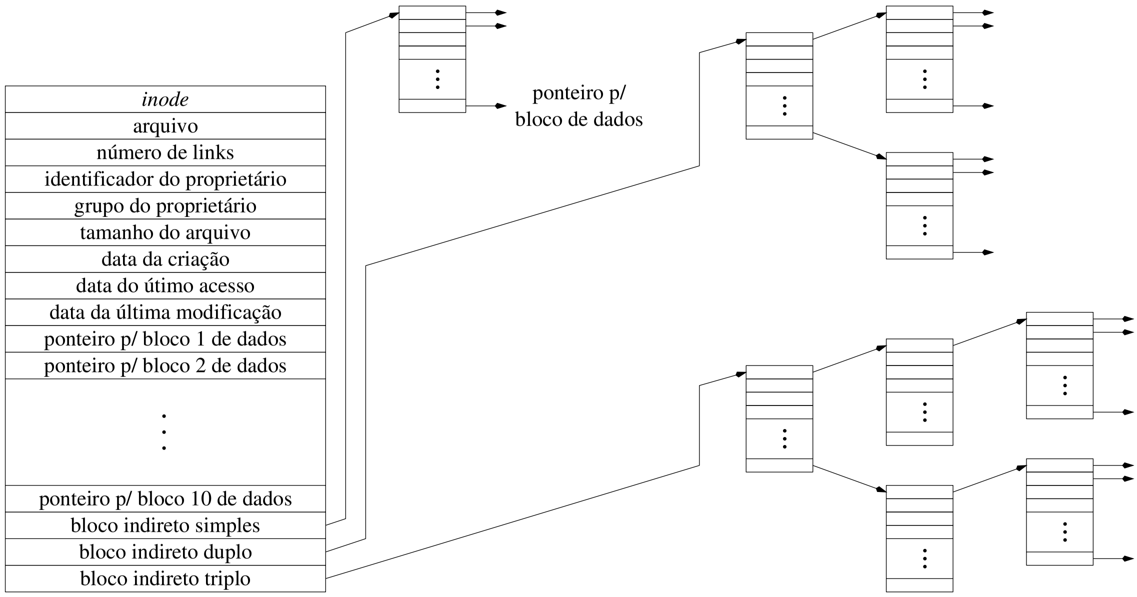
- ... no MS-DOS, o registro de diretório tem 32 bytes divididos entre: nome do arquivo, número do primeiro bloco, dentre outros itens.



- ... o número do primeiro bloco pode ser usado como um índice dentro da FAT para achar o segundo bloco, e assim todos os demais blocos podem ser encontrados;
- a exceção é para o diretório raiz, o qual é de tamanho fixo, por exemplo, 112 registros para um disco flexível de 360KB;
- ... os diretórios no MS-DOS são tratados como arquivos e podem conter um número arbitrário desses registros.

... 3.2.2 - Armazenamento de Arquivos

- ... os itens chaves são os 10 números de blocos de disco e os 3 blocos indiretos.



... 3.2.2 - Armazenamento de Arquivos

- ... quando o tamanho de um arquivo supera 10 blocos, um bloco livre é adquirido e o ponteiro indireto simples passa a apontar para ele;

- ... esse bloco é usado para armazenar os ponteiros dos blocos de disco.

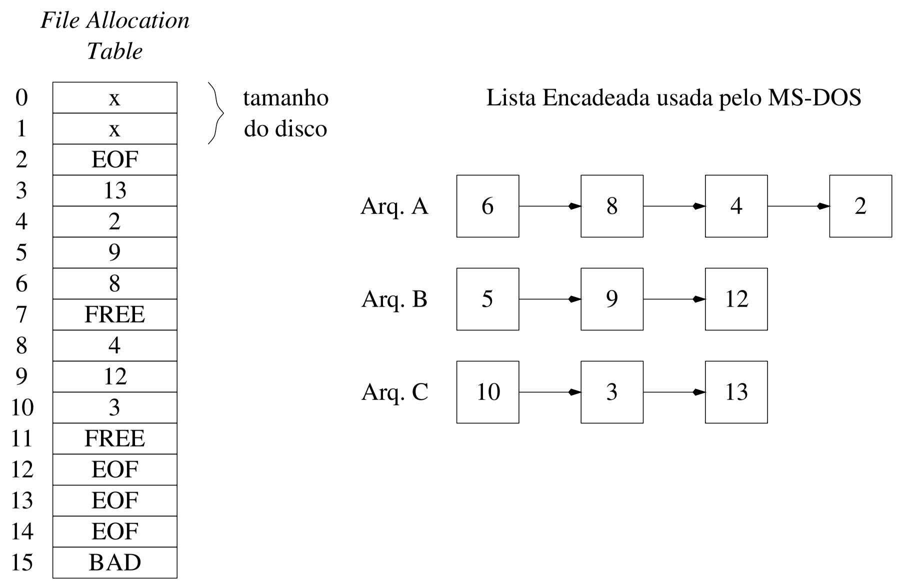
- **exemplo** - com um bloco de disco de 1024 bytes e endereços de disco de 32 bits, o bloco indireto simples pode acessar 256 endereços de disco, ou seja, este esquema é suficiente para arquivos até 266 blocos (10 no $inode$ e 256 no bloco ind. simples).

- ... de modo similar, os ponteiros de bloco indireto duplo e triplo possibilitam arquivos ainda maiores:
  - **bloco indireto duplo** - para arquivos até $266 + 256^2 = 65802$ blocos.
  - **bloco indireto triplo** - para arquivos até $266 + 256^3 = 16$ Gigabytes

✯ ... blocos indiretos são somentes usados quando necessário e considerando blocos de 1 KB, arquivos com menos de 10 KB não necessitam de blocos indiretos.

... 3.2.2 - Armazenamento de Arquivos

- ... o registro de diretório fornece para cada arquivo o endereço inicial na FAT e, na seqüência, cada unidade da FAT contém o número do próximo bloco do arquivo.



... 3.2.2 - Armazenamento de Arquivos

- Este esquema vai se tornando ineficiente a medida que a capacidade do disco aumenta — dado que o número de acessos pode aumentar consideravelmente.

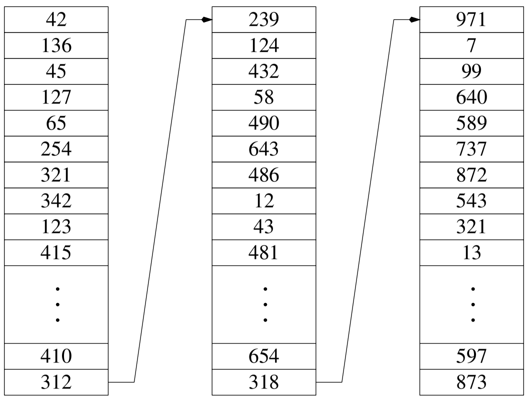
- **essência do problema da FAT** - os ponteiros para todos os arquivos do disco estão misturados de forma arbritrária na mesma tabela;
  - ... isto significa que toda a FAT deve estar presente na memória, mesmo quando um único arquivo estiver aberto.

- **proposta** - um método mais eficaz, seria manter listas dos blocos para diferentes arquivos em lugares diferentes (p. ex., UNIX)
  - ... associado a cada arquivo, tem-se uma pequena tabela no disco, chamada $inode$, que contém informações sobre o arquivo tais como tamanho e proteção.

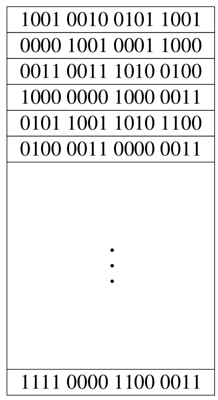
... 3.2.1 - Gerenciamento de Espaço em Disco

- ... uma vez escolhido o tamanho do bloco, a próxima questão é como manter o rastreamento de blocos livres no disco !?

- 02 métodos são largamente usados: **lista ligada de blocos** e **mapa de bits**.



Blocos Livres armazenados em um Lista Ligada



Mapa de Bits

3.2.2 - Armazenamento de Arquivos

- Se o arquivo consistir de uma seqüência de blocos, uma maneira de acessar todos os blocos do arquivo é armazenar os blocos de um arquivo em uma lista ligada.
  - ✯ ... cada bloco de disco de 1024 bytes contém 1022 bytes de dados e 2 bytes para o próximo elemento da lista;
  - ✘ número de bytes de dados em um elemento da lista não é potência de 2;‘
  - ✘ acesso aleatório é de difícil implementação por causa da busca exaustiva.

- No entanto, a idéia de representar um arquivo como uma lista encadeada, pode ser ainda explorada se mantermos os ponteiros em memória;

- ... este esquema é utilizado pelo Sistema de Arquivos do Sistema MS-DOS.
  - ... associada a cada disco, existe uma Tabela de Alocação de Arquivos (*File Allocation Tabla - FAT*) que contém um registro para cada bloco do disco.

3.2 - Projeto do Sistema de Arquivo

- ... até agora, apresentamos o sistema de arquivos do ponto de vista do usuário, ou seja, como os arquivos são identificados, quais operações são permitidas, como os diretórios são organizados, etc.

- ... projetistas estão interessados em como o espaço de disco é gerenciado, como os arquivos são armazenados e como manipulá-los de forma eficiente e confiável.

✰ **gerenciamento de espaço em disco** - arquivos são normalmente armazenados em disco, sendo assim o gerenciamento desse espaço é de interesse do projetista.

- ... 02 estratégias são possíveis para armazenar um arquivo com n bytes:

❶ n bytes consecutivos no disco são alocados;
❷ divisão do arquivo em um nro. de blocos não necessariamente contíguos.

- ... as mesmas estratégias se aplicam no sistema de gerenciamento de memória, ou seja, de um lado a segmentação pura e de outro a paginação.

... 3.2.1 - Gerenciamento de Espaço em Disco

☞ !? porquê razão, normalmente todos os sistemas de arquivos armazenam os arquivos em blocos de tamanho fixo, adjacentes ou não ?!

- ... armazenar um arquivo como uma seqüência contígua de bytes apresenta um problema óbvio que é o crescimento do arquivo, uma ocorrência muito comum.

☞ !? então a outra questão é definir qual o tamanho do bloco a ser usado !?

- ... considerando a forma pela qual os discos são organizados, são possíveis candidatos para alocação os setores, as trilhas e os cilindros.

- ... uma boa escolha deve considerar a relação custo-benefício em se ter arquivos menores com muitos blocos, ou arquivos maiores com um menor número de blocos.

- **compromisso usual** - escolher blocos entre 512, 1024 e 2048 bytes.

... 3.1.2 - Diretórios

✰ ... em um sistema de arquivos, na identificação referência a um arquivo 02 modos são comumente empregados:

- **caminho absoluto** - cada arquivo é identificado pela seqüência de diretórios desde o diretório raiz até o arquivo;
- **caminho relativo** - cada arquivo é identificado a partir do diretório corrente ou diretório de trabalho que tenha sido designado pelo usuário.

- ... em UNIX todos os caminhos começando com / são caminhos absolutos e os outros são relativos ao diretório de trabalho do usuário;

- ... assim, se o corrente diretório é "*/home/iacopo*" então o arquivo cujo caminho absoluto é "*/home/iacopo/mailbox*" pode ser referenciado apenas por "*mailbox*".

... 3.1.2 - Diretórios

... 3.1.1 - Fundamentos Básicos

✫ ... outra característica importante é que a maior parte dos sistemas operacionais suportam vários tipos de arquivos, como por exemplo o UNIX:

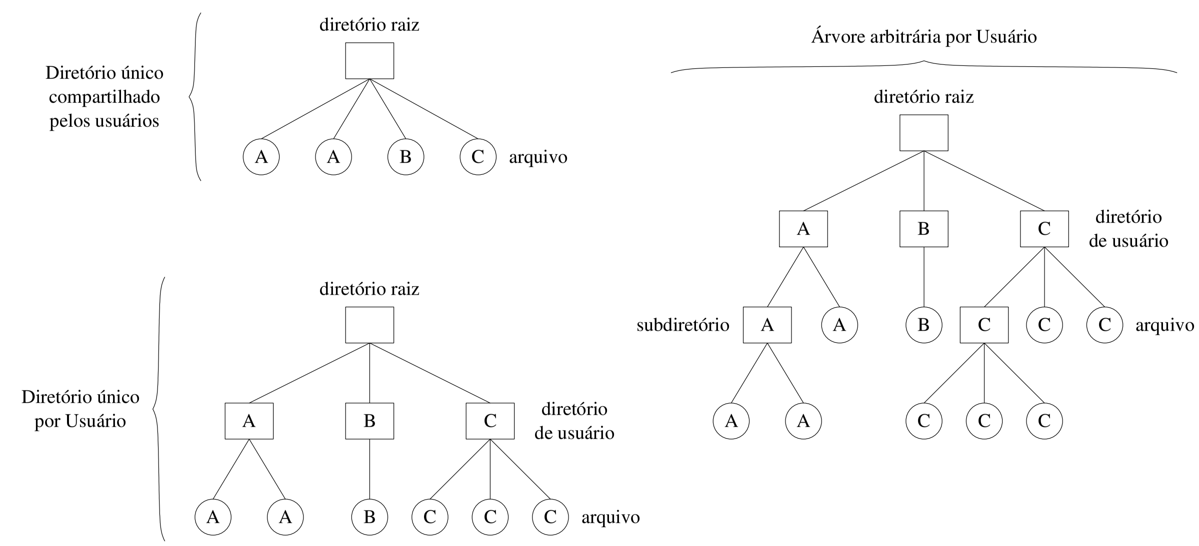
- **arquivos regulares** - contém dados e programas do usuário;
- **diretórios** - identificam arquivos através de nomes simbólicos;
- **arquivos especiais** - são usados para especificar periféricos como terminais, impressoras, unidades de fita, *cdrom*, *zipdrive*, etc.

- arguivos regulares, por sua vez, podem ser subdivididos em diferentes categorias: *file.obj* é um arq. objeto e *file.lib* é um biblioteca de arquivos.

- ... as extensões podem ser simples convenções, neste caso a extensão não é utilizada pelo sistema operacional, ou seguirem regras rígidas, por exemplo, um sistema não executaria um arquivo a menos que sua extensão fosse *.bin*.

3.1.2 - Diretórios

✫ **diretório** - providos para organizar arquivos, os diretórios em muitos casos são também arquivos que contém tipicamente um número de registros, um por arquivo.



3.1 - Interface do Sistema de Arquivos

- o sistema de arquivos pode ser considerado, por assim dizer, a **vitrina** de um sistema operacional e assim a conveniência e facilidade de seu uso é fortemente determinada pela interface, estrutura e confiabilidade de seu sistema de arquivos.

- ... via chamada de sistema, programas aplicativos utilizam o sistema de arquivos para criar, ler, gravar e remover arquivos;

- ... usuários utilizam interativamente o sistema de arquivos, agora via *shell*, para listar, alterar propriedades e remover arquivos.

- ... um ponto de vista interessante é como o sistema de arquivos se apresenta para o usuário, ou seja, o que constitui um sistema de arquivos, como os arquivos são identificados, protegidos e que operações sobre eles são permitidas !?

3.1.1 - Fundamentos Básicos

- **proposta comum** - permitir aos usuários definir objetos chamados arquivos, que podem armazenar programas, dados, ou qualquer outra informação

- ... para tanto, o sistema operacional provê operações via chamadas de sistema para criar, destruir, ler, atualizar e proteger os arquivos;

- ... além disso, todos os sistemas operacionais visam uma independência dos dispositivos de armazenamento, permitindo acessar um arquivo sem especificar em qual dispositivo o mesmo se encontra fisicamente armazenado;

✰ ... alguns sistemas operacionais possuem uma maior independência dos dispositivos de armazenamento que outros, como por exemplo:

  - no UNIX um sist. de arquivo pode ser montado em qualquerdispositivo de armazenamento, permitindo que qualquer arquivo seja acessado pelo *path name*;
  - no MS-DOS, por outro lado, o usuário deve especificar em qual dispositivo o arquivo se encontra, exceto quando o disp. é *default* e for omitido.

Cap. 3 - Sistema de Arquivo

3.1 - Interface do Sistema de Arquivo

- fundamentos básicos; diretórios.

3.2 - Projeto do Sistema de Arquivo

- gerenciamento de espaço em disco; armazenamento de arquivo; estrutura de diretório; arquivos compartilhados.

3.3 - Confiabilidade do Sistema de Arquivo

- blocos defeituosos; *backups*; consistência do sistema de arquivo.

3.4 - Desempenho do Sistema de Arquivo

- aspectos de desempenho.

Cap. 3 - Sistema de Arquivo - Referências Bibliográficas

✯ Eleri CARDOZO; Maurício MAGALHÃES; Luís F. FAINA; “Sistemas Operacionais”, Dep. de Eng. de Computação e Automação Industrial, Fac. de Engenharia Elétrica e de Computação, UNICAMP, 1996.

- Cap. 3 - Sistema de Arquivo

✯ A.S. TANEMBAUM; “Modern Operating Systems”,
Prentice-Hall International Editions, 1984, ISBN: 0-13-637331-3

- Cap. 5 - Sistema de Arquivo