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.

3.3.2 - 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.

... 3.3.3 - 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.

... 3.3.3 - 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.