Sistemas de Arquivos

LICENCIATURA EM COMPUTAÇÃO

Sistemas de Arquivos

- É a parte do Sistema Operacional responsável pelo **gerenciamento dos arquivos** (estrutura, identificação, acesso, utilização, proteção e implementação). Ou seja, é um conjunto de **tipos abstratos de dados** que são implementados para o armazenamento, a organização hierárquica, a manipulação, navegação, acesso e recuperação de dados.

- Entenda que um arquivo é um recipiente no qual os dados são armazenados, tendo ele um significado para o sistema ou usuário, e estes podem ser programas executáveis, texto, figura, etc.

Sistemas de Arquivos

- Todos os arquivos possuem um nome o qual o usuário faz referência a ele. Além do **nome**, cada arquivo possui uma série de outros **atributos** que são mantidos pelo sistema operacional como o **tipo de conteúdo, tamanho, data e hora do último acesso, data e hora da última alteração, lista de usuários que podem acessar o arquivo**, etc.
- A forma como os dados são dispostos dentro de um arquivo determina sua **estrutura interna**.
- Cada tipo de arquivo possui uma estrutura interna apropriada para a sua finalidade. Por exemplo, arquivos de texto são organizados em linha ou parágrafos.

Sistemas de Arquivos

- Discos Rígidos:
  - Os discos rígidos são compostos por vários discos internos, onde cada um deles é dividido em círculos concêntricos chamados de cilindros ou **trilhas**, e nestas trilhas temos uma certa quantidade de **setores**. Cada setor possui, normalmente, **512 bytes** de informações.
  - Para descobrir a **capacidade total de um HD**, basta **multiplicar o tamanho do setor pela quantidade total de setores que ele tem**. Vale lembrar que 1KB é representado por 1024 bytes, e não 1000 bytes como muitos pensam.

FAT16

- O significado da palavra FAT é **Tabela de Alocação de Arquivos** (*File Allocation Table*) que seria um **mapa de utilização do disco**. Graças a isto, o SO saberá onde determinado arquivo está.
- Normalmente, é reconhecido por todos os Sistemas Operacionais, também é utilizado em cartões de memória de estado sólido, e não trabalham com setores, mas sim **comunidades de clusters que são conjuntos de setores.**
  - Usado em vários SOs, na maior parte do DOS, incluindo o DR-DOS, OpenDOS, FreeDOS, MS-DOS, Microsoft Windows (até e incluindo o Windows Me). FAT é usado também para flash drives e cartões de memória removíveis.
- Uma característica marcante é a capacidade de **nomear** os arquivos somente com **8 caracteres + 3 para extensão**. Caso seja excedido o valor de caracteres, os caracteres excedidos (do nome do ficheiro) desaparecerão e no lugar deles aparecerá ~1 ou ~2 (se já existir um outro arquivo com os 8 primeiros caracteres iguais).

FAT16

- Existe um inconveniente que quando arquivos são apagados e novos arquivos são escritos no suporte, as suas partes tendem a dispersar-se, **fragmentando-se** por todo o espaço disponível, **tornando a leitura e a escrita um processo lento.** Para isso, precisamos **desfragmentar** o disco para um melhor desempenho na sua função de leitura e gravação.
- Outro problema é que o FAT16 utiliza **16 bits para endereçamento**, permitindo assim armazenar no máximo 65536 clusters (2^{16}).
  - Geralmente, cada *cluster* pode ter um dos seguintes tamanhos: 2 KB, 4 KB, 8 KB, 16 KB e, por fim, 32 KB.
  - Por exemplo, se tivermos um arquivo com 50 KB, é possível guardá-lo em dois clusters de 32 KB cada – DESPERDICIO (um arquivo por *cluster*).
  - Não reconhece mais que 2 GB por ser de 16 bits, utilizando clusters com no máximo 32 KB (65536 x 32 = 2.097.152 KB, que corresponde a 2 GB.).
  - Caso haja um disco com mais de 2GB, será necessário particioná-lo em pedaços máximos de 2GB.

FAT16

- Cálculo da capacidade máxima de endereçamento com o FAT16:
  - Considerando clusters de 32 KB de tamanho (64 setores).
  - $(2^{16})*(32KB)=(2^{6})*(2^{10})*([2^{5}]*[2^{10}])=2*(2^{10})*(2^{10})*(2^{10})=2GB$

- Cálculo considerando clusters de 1KB de tamanho (2 setores) e 12bits para endereçamento (FAT12):

- ??????????????????

FAT32

- Utiliza **28 bits para endereçamento** **(deveria chamar-se FAT-28, mas as potências de dois soam bem melhor)**. Suporta partições de até 2TB, tamanho de arquivos de 4 GB e o **nome dos arquivos passou de 8 para 256 caracteres e superou o antigo limite de 3 caracteres para a extensão, embora este padrão ainda seja largamente utilizado.**

- Com o FAT32, o **desperdício** em disco foi sensivelmente **reduzido**. O FAT16, seu antecessor, utilizava clusters de até 32KB enquanto o FAT32 pode utilizar clusters de 4KB. Se um arquivo ocupa 4KB de espaço, tanto no FAT16 como no FAT32 a ocupação será de 1 cluster, porém, no caso do FAT16 os 28KB restantes serão alocados, apesar de ficarem fisicamente vazios.

FAT 32

- Tem a desvantagem de ser **6% mais lenta** que FAT16 e a **incompatibilidade** com SOs antigos.

- Não possui **recursos de segurança** como o NTFS.

- Utiliza uma **cópia *backup* da tabela de alocação** como sistema de segurança para corrupções de arquivos. Este procedimento é ineficiente, pois uma **queda de energia** durante uma operação que modifique os metadados pode tornar a partição inacessível ou corromper severamente diversos arquivos.

NTFS

- O NTFS (*New Technology File System*) é um sistema de arquivos que surgiu com o lançamento do **Windows NT**, e passou a ser bem aceito e utilizado nas outras versões do Windows posteriormente.
- Uma de suas vantagens diz respeito ao quesito **“recuperação”**: em caso de falhas, como o desligamento repentino do computador, o NTFS é capaz de **reverter os dados à condição anterior** ao incidente. Isso é possível, em parte, porque, durante o processo de *boot*, o sistema operacional consulta um **arquivo de log** (*JOURNALING*) que registra todas as operações efetuadas e entra em ação ao identificar nele os pontos problemáticos.

NTFS

- Ainda neste aspecto, o NTFS também suporta **redundância de dados**, isto é, replicação, como o que é feito por sistemas **RAID**, por exemplo.
- Possui a **MFT** (*Master File Table)* é uma **tabela que registra atributos de cada arquivo armazenado.** Esses atributos consistem em uma **série de informações**, entre elas: nome, data da última modificação, permissões e, principalmente, localização na unidade de armazenamento.
- Como necessita guardar várias informações de praticamente todos os arquivos no disco, o NTFS **reserva um espaço para a MFT** - Zona MFT - geralmente de 12,5% do tamanho da partição. Cada arquivo pode necessitar de pelo menos 1KB para o registro de seus atributos na MFT, daí a necessidade de um espaço considerável para este.

Características do NTFS

- Neste modelo, temos o **tamanho limite do arquivo** de acordo com o tamanho do **volume**;
- O *maxsize* de um arquivo no Fat16 é 2GB, no Fat32 é de 2TB e no NTFS é o disponível;
- Os nomes dos VOLUMES podem ter 32 caracteres;
- Utiliza 64 bits para endereçamento;
- Clusters de 512 bytes a 64KB, o que permite partições de até 256 TB;
- Suporte a **quotas de disco (que permitem aos administradores controlar a quantidade de dados que cada usuário pode armazenar em um volume do sistema de arquivos NTFS);**
- Suporte a **criptografia** (*Encrypting File System* - EFS), indexação e compactação;

Características do NTFS

- O **tamanho dos clusters** é definido com base na capacidade de armazenamento do dispositivo durante a instalação do SO ou durante a formatação de uma partição.
- É mais **seguro** que o FAT; Permite política de segurança e gerenciamento;
- **Menos fragmentação**; Recuperação de erros mais fácil;
- Casos seja usado em mídias, podem se corromper mais facilmente;
- É um pouco **mais lenta que o FAT32** devido as diretivas de segurança que o FAT32 não tem e precisam ser acessados durante leitura e gravação de dados;
- Utiliza a MFT (*Master File Table*) para registrar a utilização de cada *cluster* de um disco;

EXT2

- **EXT (antecessor do EXT2)** permitia a criação de partições de até 2GB e suportava nomes de arquivos com até 255 caracteres. Foi um grande avanço, mas o sistema ainda estava muito longe de ser perfeito. O **desempenho era baixo** e ele era tão sujeito a **fragmentação** de arquivos quanto o sistema FAT. Além disso, logo começaram a surgir HDs com mais de 2GB, de forma que em 1993 surgiu a primeira grande atualização, na forma do EXT2.
- O EXT2 trouxe suporte a partições de até 32TB, manteve o suporte a nomes de arquivos com até 255 caracteres, além de diversos outros.

EXT2

- O maior problema do EXT2 é que ele **não inclui nenhum sistema de tolerância a falhas.**
- Sempre que o sistema é desligado incorretamente, é necessário utilizar o fsck, um utilitário similar ao candisk do Windows, que verifica todos os blocos do sistema de arquivos, procurando por inconsistências entre as estruturas e descrições e os dados efetivamente armazenados e, depois, repara o sistema de arquivos.
- O teste do fsck demora bastante (bem mais que os candisk) e o tempo cresce proporcionalmente de acordo com o tamanho da partição.
- Este problema foi **corrigido** com o **EXT3**.

EXT2

- Quando é realizada uma operação de escrita em um arquivo, o Ext2 tenta, sempre que possível, **alocar blocos de dados** no mesmo grupo que contém o nó-i. Esse comportamento **reduz o movimento da(s) cabeça(s) de leitura–gravação** da unidade de disco.
- Em um sistema de arquivos ocorrem **dois tipos de fragmentação:**
  - **Fragmentação interna (ou de espaço)** é causada pelo fato do tamanho do arquivo geralmente não ser múltiplo do tamanho do bloco (portanto o último bloco terá um espaço não utilizado), a consequência é a perda de espaço;
  - A **fragmentação externa (ou de arquivo)** decorre da impossibilidade do sistema determinar, a priori, qual o tamanho do arquivo (p.ex., arquivos de texto e de logs são muito modificados, e o seu tamanho pode aumentar ou diminuir), assim, um arquivo pode alocar **blocos não contíguos,** prejudicando o desempenho.

EXT2

- Para diminuir o impacto do primeiro tipo (fragmentação interna), existem duas estratégias básicas. A primeira, mais simples, é determinar, na formatação, **o menor tamanho de bloco** possível.
- Um tamanho de bloco pequeno, como 1024 bytes, diminui a fragmentação e perda de espaço, mas em **contrapartida** gera um impacto negativo no desempenho, pois acarreta o **gerenciamento de uma maior quantidade de blocos.**
- O tamanho de bloco padrão para volumes grandes é de 4096 bytes.

EXT2

- A segunda estratégia é **alocar a parte final de um arquivo**, menor que o tamanho de um bloco, **juntamente com pedaços de outros arquivos.**
- O Reiserfs (*File System Default* das distros Suse e Open Suse) chama esse método de *tail packing*;
- Para diminuir o impacto da fragmentação externa, o Ext2 **pré-aloca (reserva) até oito blocos quando um arquivo é aberto** para gravação. Esses blocos reservados, quando possível, são adjacentes ao último bloco utilizado pelo arquivo.
- Um mapa de bits de blocos é usado para mostrar quais os blocos que, dentro do grupo, estão livres ou alocados.

EXT3

- O Ext3 (*Third Extended File System*) é um sistema de arquivos desenvolvido por Stephen C. Tweedie para o Linux, que **acrescenta** alguns recursos ao Ext2, dos quais o mais visível é o ***journaling***, que consiste em um **registro** (*log* ou *journal*) **de transações** cuja finalidade é **recuperar o sistema** em caso de desligamento não programado.
- O EXT3 (assim como o EXT2) utiliza **endereços de 32 bits** e blocos (análogos aos *clusters* usados no sistema FAT) de até 8KB.
- Tanto o tamanho máximo da partição, quanto o tamanho máximo dos arquivos são determinados pelo tamanho dos blocos, que pode ser escolhido durante a formatação.

EXT3

Tamanho dos blocos	Tamanho máximo da partição	Tamanho máximo dos arquivos
1 KB	2 TB	16 GB
2 KB	8 TB	256 GB
4 KB	16 TB	2 TB
8 KB	32 TB	2 TB

EXT3

- Há três níveis de *journaling* disponíveis na implementação do Ext3:
  - ***Journal*:** os metadados e os dados (conteúdo) dos arquivos são escritos no *journal* antes de serem de fato escritos no sistema de arquivos principal. Isso aumenta a **confiabilidade** do sistema com uma **perda de desempenho**, devido a **necessidade de todos os dados serem escritos no disco duas vezes**.
  - ***Writeback*:** os metadados são escritos no *journal*, mas não o conteúdo dos arquivos. Essa opção permite um melhor desempenho em relação ao modo *journal*, porém introduz o risco de escrita fora de ordem onde, por exemplo, arquivos que são apensados durante um *crash* (colisão) podem ter adicionados a eles trechos de lixo na próxima montagem.
  - ***Ordered*:** é como o *writeback*, mas força que a escrita do conteúdo dos arquivos seja feita após a marcação de seus metadados como escritos no *journal*. Esse é considerado um meio-termo aceitável entre confiabilidade e performance, sendo, portanto, o nível padrão/*default*.

EXT3

- Embora o seu **desempenho** (velocidade) seja **menos atrativo** que o de outros sistemas de arquivos (como Reiser FS e XFS), ele tem a importante vantagem de permitir que seja feita a atualização direta a partir de um sistema com ext2, sem a necessidade de realizar um *backup* e restaurar posteriormente os dados, bem como o menor consumo de processamento.
- Enquanto em alguns contextos a falta de funções de sistemas de arquivos “modernos”, como alocação dinâmica de *inodes* e estruturas de dados em árvore, poderia ser considerada uma desvantagem, em termos de “recuperabilidade” isso dá ao ext3 uma significante vantagem sobre sistemas de arquivos que possuem-nas.
- Os **metadados** do sistema de arquivos **estão todos em locais fixos** e bem conhecidos, e há certa redundância inerente à estrutura de dados, que permite que sistemas ext2 e ext3 sejam recuperáveis no caso de uma corrupção de dados significante, em que sistemas de arquivos em árvore não seriam recuperáveis.

EXT3

- Por padrão, o tamanho do bloco é determinado automaticamente, de acordo com o tamanho da partição, mas é possível forçar o valor desejado usando o parâmetro **“**-b” do comando mkfs.ext3 (usado para formatar as partições EXT3 no Linux), como em “mkfs.ext3 –b 2048 / dev / hda1” (cria blocos de 2KB) “mkfs.ext3 –b 4096 / dev / hda1” (para blocos de 4KB).

EXT3 e EXT4

- Embora o limite de 32TB para as partições EXT3 não seja um problema hoje em dia, ele tende a se tornar um obstáculo conforme os HDs crescerem em capacidade, assim como os limites anteriores.
- Para evitar isso, o **EXT4**, legítimo sucessor do EXT3, incorporou o uso de **endereços de 48 bits**, o que permite endereçar um volume virtualmente ilimitado de blocos. Só para referência, o EXT4 permite criar **partições de até 1024 petabytes**).
- O limite de 2TB para os arquivos também foi removido, abrindo espaço para o armazenamento de bases de dados gigantes e outros tipos de arquivos que eventualmente venham a superar esta marca. Embora existam diversos outros sistemas de arquivos para o Linux, como o ReiserFS, XFS, JFS e assim por diante, o **EXT3 continua sendo o sistema de arquivos mais utilizado**, já que ele atende bem à maioria e é muito bem testado e por isso bastante estável. A tendência é que o EXT3 seja lentamente substituído pelo EXT4 e os demais sistemas continuem entrincheirados em seus respectivos nichos.

Ler!!!

- **Capítulo 4 do Livro Texto.**

Atividade

- **Atividade no AVA.**