

## **Princípios de Sistemas Operacionais**

Um sistema operacional é uma camada de software que atua entre o hardware e os diversos aplicativos existentes em um sistema computacional, executando diversas atividades, de modo a garantir a disponibilidade de recursos entre todos os programas em execução.

### **Tipos de Sistemas Operacionais**

#### **Em relação a arquitetura do sistema operacional:**

**Sistema Monolítico:** o kernel consiste em um único processo executando numa memória protegida (espaço do kernel). Ex: Windows, Linux, FreeBSD;

**Sistema em Camadas:** funções do kernel irão executar em camadas distintas de acordo com o nível de privilégio. Ex. Multics;

**Modelo cliente/servidor ou microkernel:** o kernel consiste apenas no essencial (comunicação e gerenciamento de processos) e funções como sistemas de arquivos e gerenciamento de memória são executadas no espaço do usuário como serviço: as aplicações (programas) são os clientes. Ex: GNU Hurd, Mach;

**Monitor de máquinas virtuais:** fornece uma abstração do hardware para vários sistemas operacionais. Ex: VM/370, VMware, Xen.

#### **Quanto a capacidade de processamento, pode usar as seguintes classificações:**

**Monotarefa:** pode-se executar apenas um processo por vez.

Ex: DOS;

**Multitarefa:** além do próprio sistema operacional, vários processos de utilizador (tarefas) estão carregados em memória, sendo que um pode estar ocupando o processador e outros ficam enfileirados, aguardando a sua vez. O compartilhamento de tempo no processador é distribuído de modo que o usuário tenha a impressão de que vários processos estão sendo executados simultaneamente.

Ex: Windows, Linux, FreeBSD.

**Multiprocessamento ou multiprogramação:** o SO pode distribuir as tarefas entre vários processadores. Alguns autores utilizam o termo multiprocessamento como se fosse multitarefa.

Sistemas operacionais para computadores de pequenos e grandes dividem-se em quatro categorias, que se diferenciam pelo tempo de resposta e pela forma de entrada de dados no sistema: em lotes (batch), interativa, em tempo real e sistemas híbridos;

**Os sistemas em lotes (batch):** existem desde o tempo dos primeiros computadores, os quais trabalhavam com cartões perfurados ou fita magnética para a entrada de dados. Os sistemas em lotes de hoje não trabalham mais com cartões perfurados ou com fitas magnéticas, mas as tarefas ainda são processadas em série, sem interação com o usuário;

**Os sistemas interativos:** também conhecidos como sistemas de tempo compartilhado, produzem um tempo de resposta mais rápido do que os sistemas em lotes, mas são mais lentos do que os sistemas em tempo real. Foram criados para atender a necessidade do usuário, que necessitava de tempo de resposta mais rápido na depuração dos programas. O sistema operacional exigiu a criação de programas de tempo compartilhado, permitindo o usuário interagir diretamente com o sistema de computação;

**Os sistemas em tempo real:** os mais rápidos de todos, são utilizados em ambientes onde o tempo é um fator crítico: vôos espaciais, controle de tráfego aéreo etc;

**Os sistemas híbridos:** são uma combinação entre os sistemas em lotes e os sistemas interativos. Executa os programas em lotes em segundo plano. Um sistema híbrido se beneficia do tempo livre entre demandas de processamento para executar programas que não precisam da interferência do operador.

**Base do Sistema Operacional****Base que forma o sistema operacional:**

**Gerenciador de memória:** responsável pela alocação de memória;

**Gerenciador de unidade de processamento:** responsável pela alocação de recursos da CPU;

**Gerenciador de dispositivos:** Sua tarefa é escolher a forma mais adequada para alocação de todos os dispositivos;

**Gerenciador de arquivos:** monitora todos os arquivos no sistema. Faz cumprir as restrições de acesso a cada arquivo.

**Informação adicional**

Módulos do sistema operacional residentes em disco, que são carregados para a memória somente quando são necessários, chamam-se módulos transientes.

**Gerenciador de Memória**

O gerenciador de memória trabalha com tabelas geradas para controlar os programas em execução. Existem 3 ou 4 tabelas, dependendo do tipo de gerenciamento adotado;

A memória é dividida logicamente em tamanhos iguais (páginas de memória) ou diferentes (segmentos de memória), para acomodar os diversos programas residentes em memória.

**Esquemas de gerenciamento de memória:**

- Alocação de memória com paginação;
- Alocação de memória com paginação sob demanda;
- Alocação de memória com segmentação;
- Alocação de memória com segmentação / paginação sob demanda.

**Alocação de Memória Com Paginação**

Divisão da memória em páginas de igual tamanho;

A paginação funciona muito bem quando as páginas, os setores e os quadros de páginas possuem o mesmo tamanho; Não faz uso da memória virtual;

Gerenciador de memória prepara o programa da seguinte forma: determina o número de páginas do programa; localiza quadro de páginas suficientes na memória principal; carrega todas as páginas do programa nesses quadros de memória;

As páginas de um programa, quando forem carregadas para a memória, não precisam ser carregadas em blocos contíguos; Dois programas não podem usar a mesma página de memória ao mesmo tempo, porque causaria a fragmentação interna da memória; Esse método requer que o programa inteiro seja alojado na memória na hora da execução.

**Alocação de Memória Com Paginação Sob Demanda**

O gerenciador de memória trabalha com tabelas para monitorar o uso dos quadros de memória. Essencialmente três tabelas executam essa função: a Tabela de Programas (TP), a Tabela de Mapa de Páginas (TMP) e a Tabela de Mapa de Memória (TMM). Todas residem na parte da memória principal;

Cada programa possui sua própria TMP, contendo informações como número da página, endereço de memória do quadro da página correspondente etc;

A TMM possui uma entrada para cada quadro de página, no qual está a localização e o status (livre/ocupada); A partir desse gerenciamento se começou a trabalhar com memória virtual; A memória continua sendo dividida em tamanhos iguais (páginas).

### **Alocação de Memória Sob Demanda**

**Conjunto de trabalho:** é o conjunto de páginas residentes em memória que podem ser acessadas diretamente sem implicar erro de página. Essa técnica é utilizada para evitar o thrashing (troca excessiva de páginas entre a memória RAM e a virtual).

#### **Conceitos e critérios para substituição de páginas:**

- A política que rege a escolha da página que deve sair da memória, ou seja, o critério de substituição das páginas, é crucial para a eficiência do sistema;
- Duas técnicas mais conhecidas são: “primeiro a entrar, primeiro a sair (FIFO, ou first-in first-out)” e “página usada menos recentemente (UMR)”;
- Nas duas técnicas tem que se observar se não houve alteração dos dados antes de retirar da memória principal;
- O algoritmo FIFO consulta apenas os bits de status e de página modificada antes de fazer as substituições, mas o algoritmo UMR consulta os três bits

### **Alocação de Memória Com Segmentação**

- O conceito de segmentação baseia-se em uma prática comum entre os programadores: a estruturação dos seus programas em módulos (agrupamento lógico de códigos). Segundo esse esquema cada programa é dividido em vários segmentos de tamanhos diferentes;
- Este esquema difere fundamentalmente da paginação, onde um programa é dividido em páginas de igual tamanho que muitas vezes contém instruções de mais de um módulo de programa;
- Uma outra diferença importante é que a memória não é mais dividida em quadros de página, pois o tamanho de cada segmento varia, ou seja, a memória é alocada de maneira dinâmica;
- Esse esquema reduz a segmentação da memória (aproveita melhor a memória).

#### **Monitoramento da memória:**

Quando um programa é compilado ou montado, os segmentos são definidos de acordo com os módulos estruturais do programa. Cada segmento é numerado e uma Tabela de Mapa de Segmentos (TMS) é gerada para cada programa; A TMS inclui os números de segmentos, o tamanho, os direitos de acesso, o status e, quando um segmento é carregado em memória, o endereço de memória.

#### **- O gerenciador de memória monitora os segmentos em memória através de 3 tabelas:**

Tabela de Programas (TP): exibe todos os programas em processamento (uma para todo o sistema);

Tabela de Mapa de Segmentos (TMS): exibe detalhes sobre cada segmento (uma para cada programa);

Tabela de Mapa de Memória (TMM): monitora a alocação de memória principal (uma para todo o sistema).

### **Alocação de Memória Com Segmentação / Paginação Sob Demanda**

É uma combinação entre a segmentação e a paginação sob demanda que alia os benefícios lógicos da segmentação aos benefícios físicos da paginação; Os segmentos são divididos em páginas de igual tamanho, menores do que a maioria dos segmentos e mais facilmente manipuladas.

Assim sendo, muitos dos problemas da segmentação (compactação, fragmentação externa e necessidade de armazenagem secundária) são eliminados, pois as páginas são de tamanho fixo;

A maior desvantagem desse esquema são a sobrecarga gerada pelas tabelas adicionais.

**Monitoramento da memória:**

- Tabela de Programa (TP): exibe todos os programas em processamento (uma para todo o sistema);
- Tabela de Mapa de Segmentos (TMS): exibe detalhes sobre cada segmento (uma para cada programa);
- Tabela de Mapa de Páginas (TMP): exibe detalhes sobre cada página (uma para cada segmento);
- Tabela de Mapa de Memória (TMM): monitora a alocação de quadros de página na memória principal (uma para todo o sistema).

**Sistema de Arquivos NTFS**

**O NTFS (New Technology File System) é um sistema de arquivos que surgiu com o lançamento do Windows NT. Sua confiabilidade e desempenho fizeram com que fosse adotado nos sistemas operacionais posteriores da Microsoft, como Windows XP, Windows Vista, Windows 7 e Windows Server 2008.** Antes, o que é um sistema de arquivos?

Não é possível gravar dados em um HD ou em qualquer outro dispositivo de armazenamento de forma a manter as informações acessíveis e organizadas sem um sistema de arquivos (file system) - essencialmente, um tipo de estrutura que indica como os arquivos devem ser gravados e lidos pelo sistema operacional do computador.

É o sistema de arquivos que determina como as informações podem ser guardadas, acessadas, copiadas, alteradas, nomeadas e até apagadas. Ou seja, resumindo, toda e qualquer manipulação de dados em um dispositivo de armazenamento necessita de um sistema de arquivos para que estas ações sejam possíveis. Sem um sistema de arquivos, os dados armazenados seriam apenas um conjunto de bits sem utilidade.

Há vários sistemas de arquivos disponíveis, para os mais diversos sistemas operacionais e para as mais variadas finalidades.

**Como O NTFS Surgiu?**

O NTFS é um sistema de arquivos amplamente utilizado nos sistemas operacionais da Microsoft. Sua primeira aparição foi no Windows NT, sistema operacional para uso em servidores cuja primeira versão foi lançada em 1993. No entanto, a história do NTFS começa muito antes disso.

Até aquela época, a Microsoft não possuía nenhum sistema operacional capaz de fazer frente ao Unix e suas variações em aplicações de servidores. Seus principais produtos eram o MS-DOS e a linha Windows 3.x, essencialmente, sistemas operacionais para uso doméstico ou em escritório. Era preciso criar algo novo, capaz de disputar mercado com as soluções baseadas em Unix. Foi aí que surgiu o Windows NT.

De nada adianta um sistema operacional novo se o seu sistema de arquivos for limitado. Na época, a Microsoft tinha em mãos o sistema de arquivos FAT. Este funcionava razoavelmente bem em aplicações domésticas, mas não serviria aos propósitos do novo projeto por uma série de restrições, entre elas, baixa tolerância a falhas, inviabilidade de uso de permissões de arquivos e limitações para o trabalho com grande volume de dados.

Para superar esses e outros problemas, a Microsoft decidiu utilizar o NTFS. Porém, ao contrário do que muita gente pensa, a empresa não desenvolveu esse sistema de arquivos sozinha. Ela utilizou como base o HPFS (High Performance File System), sistema de arquivos que tinha a IBM por trás.

No início da década de 1980, ambas as companhias fecharam um acordo para o desenvolvimento do OS/2, um sistema operacional até então moderno, que se destacaria por sua capacidade gráfica (naquela época, era muito comum o uso de sistemas operacionais baseados em linha de comando).

O problema é que, logo, Microsoft e IBM passaram a divergir em relação a diversos pontos. Como consequência, desfizeram a parceria. A IBM continuou tocando o projeto do OS/2, enquanto que a

Microsoft foi cuidar de seus interesses, mais precisamente, do projeto que resultou no Windows NT. No entanto, a companhia não abandonou a parceria de mãos vazias: levou vários conceitos do HPFS - o sistema de arquivos do OS/2 - relacionados à segurança, confiabilidade e desempenho para posteriormente implementá-los no NTFS.

Sabe-se também que o NTFS tem alguma relação com o Files-11, sistema de arquivos do sistema operacional VMS, que passou às mãos da Compaq em 1998, empresa que posteriormente foi adquirida pela HP. Quando os trabalhos no VMS estavam em andamento, parte de sua equipe se transferiu para Microsoft, com destaque para o engenheiro de software Dave Cutler, um dos nomes por trás do NTFS e do próprio Windows NT.

### **Principais Características do NTFS**

Os conceitos aplicados ao NTFS fizeram com que o Windows NT e versões posteriores do sistema fossem bem recebidos pelo mercado. Uma dessas características 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 que registra todas as operações efetuadas e entra em ação ao identificar nele os pontos problemáticos. Ainda neste aspecto, o NTFS também suporta redundância de dados, isto é, replicação, como o que é feito por sistemas RAID, por exemplo.

Outra característica marcante do NTFS é o seu esquema de permissões de acesso. O Unix sempre foi considerado um sistema operacional seguro por trabalhar com o princípio de que todos os arquivos precisam ter variados níveis de permissões de uso para os usuários. O NTFS também é capaz de permitir que o usuário defina quem pode e como acessar pastas ou arquivos.

O NTFS também é bastante eficiente no trabalho com arquivos grandes e unidades de discos volumosos, especialmente quando comparado ao sistema de arquivos FAT. Você vai entender o porquê no tópico a seguir.

### **Lidando Com Arquivos**

Em um disco rígido, a área de armazenamento é dividida em trilhas. Cada trilha é subdividida em setores (saiba mais neste artigo sobre HDs), cada um com 512 bytes, geralmente. FAT e NTFS trabalham com conjuntos de setores, onde cada um é conhecido com cluster (ou unidade de alocação). O FAT16, por exemplo, pode ter, comumente, clusters de 2 KB, 4 KB, 8 KB, 16 KB e 32 KB.

Aqui há um possível problema: cada arquivo gravado utiliza tantos clusters quanto forem necessários para cobrir o seu tamanho. Se, por exemplo, tivermos um arquivo com 50 KB, é possível guardá-lo em dois clusters de 32 KB cada.

Você deve ter percebido então que, neste caso, um cluster ficou com espaço sobrando. Esta área pode ser destinada a outro arquivo, correto? Errado! Acontece que cada cluster só pode ser utilizado por um único arquivo. Se sobrar espaço, este permanecerá vazio. Esse é um dos problemas do sistema FAT.

Há ainda outra limitação: o FAT16 trabalha com discos ou partições com até 2 GB. Essa situação só melhora com o FAT32, que pode trabalhar com até 2 TB (terabytes).

O NTFS, por sua vez, não pode contar com esse tipo de limitação. Por isso, utiliza 64 bits para endereços de dados, contra 16 do FAT16 e 32 do FAT32. Essa característica, aliada ao tamanho dos clusters, determina o volume máximo de dados com que cada partição NTFS pode trabalhar. Com o uso de clusters de 64 KB, esse limite pode chegar a 256 TB.

Por padrão, o tamanho dos clusters é definido automaticamente com base na capacidade de armazenamento do dispositivo durante o processo de instalação do sistema operacional ou de formatação de uma partição - indo de 512 bytes a 64 KB -, podendo também ser definido pelo usuário com procedimentos específicos.

### Tolerância a Falhas

Para a preservação dos dados, o NTFS utiliza um esquema de journaling, isto é, o arquivo de log mencionado anteriormente. De maneira resumida, seu funcionamento ocorre da seguinte forma: o log registra toda as ações que acontecem no sistema operacional em relação aos arquivos. Quando um documento é criado, um espaço em disco é alocado para ele, suas permissões são definidas e assim por diante.

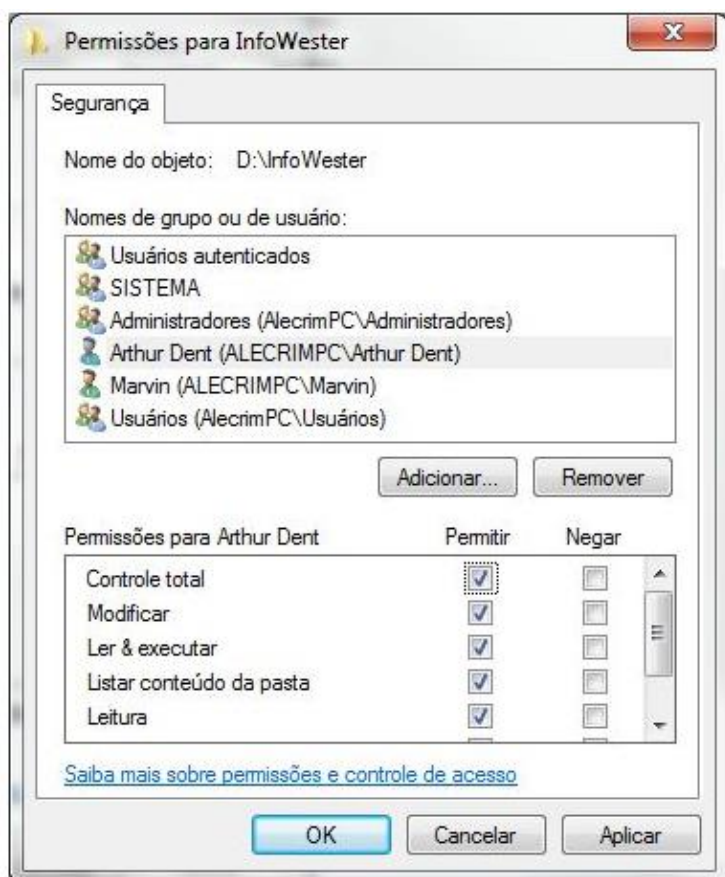
A questão é que se, por exemplo, o computador ficar repentinamente sem energia, o espaço definido para o arquivo pode ser alocado, mas não utilizado. Quando o sistema operacional é reativado, consulta o arquivo de log para saber quais procedimentos não foram executados por completo e executa a ação correspondente para corrigir o problema.

Para manter a integridade do sistema, basicamente, três passos são executados: verificação do log para checar quais clusters devem ser corrigidos; nova execução das transações marcadas como completas no final do log; reversão de procedimentos que não puderam ser concluídos.

Perceba que, com isso, o NTFS pode não conseguir recuperar os últimos dados gravados antes da interrupção, mas garante o pleno funcionamento do sistema operacional eliminando erros que podem comprometer o desempenho ou causar problemas ainda maiores.

### Permissões

O NTFS possibilita o uso de permissões no sistema operacional, ou seja, é possível definir como usuários - ou grupos de usuários - podem acessar determinados arquivos ou determinadas pastas. Por exemplo, você pode permitir ao usuário Arthur Dent ter controle total da pasta InfoWester, mas só permitir ao usuário Marvin ler e executar o referido conteúdo, sem poder alterá-lo.



Cada conta de usuário criada no sistema (ou grupo) recebe um código único chamado Security Identifier (SID). Assim, se um usuário for eliminado e, posteriormente, outro for criado com o mesmo nome, será necessário reaplicar as permissões, pois o SID deste será diferente, apesar da denominação igual.



### Master File Table (MFT)

FAT é a sigla para File Allocation Table e recebe este nome porque trabalha com uma tabela que, basicamente, indica onde estão os dados de cada arquivo. O NTFS, porém, utiliza uma estrutura chamada Master File Table (MFT), que tem praticamente a mesma finalidade do FAT, mas funciona de maneira diferente.

O MFT 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 (conceito explicado no tópico anterior) 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 o MFT - Zona MFT -, geralmente de 12,5% do tamanho da partição. Cada arquivo pode necessitar de pelo menos 1 KB para o registro de seus atributos no MFT, daí a necessidade de um espaço considerável para este.

### Outros Recursos do NTFS

O NTFS é dotado de vários recursos de natureza complementar ou definitiva que enriquecem suas características de segurança, desempenho e confiabilidade. A seguir, algumas delas.

### Encrypting File System (EFS)

Este é um recurso que surgiu com o Windows 2000 e, tal como o nome sugere, é um reforço de segurança, pois permite a proteção de dados por criptografia com o uso do esquema de chaves públicas (saiba mais sobre isso clicando no link anterior). A principal vantagem é que o dono dos arquivos protegidos pode determinar quais usuários podem acessá-los. Esse conteúdo é criptografado quando o usuário o fecha, mas passa a estar imediatamente pronto para uso quando aberto.

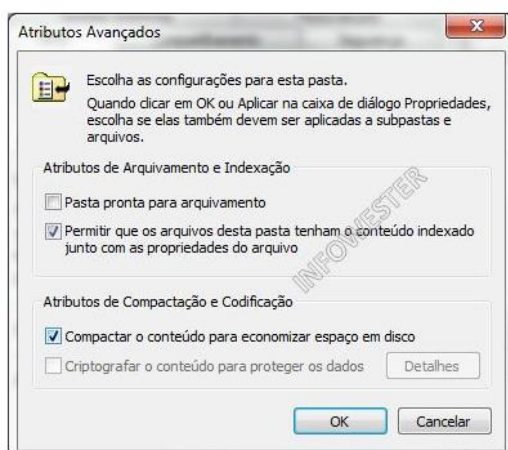
Note que o EFS não está plenamente disponível em todas as versões do Windows compatíveis com NTFS, como o Windows 7 Home Basic, por exemplo. Além disso, vale frisar também que não é possível utilizar criptografia em conteúdo compactado (tópico a seguir).

### Compressão de Dados Outro

Outro recurso interessante do NTFS é a sua capacidade de lidar com compressão de dados para economizar espaço em disco. Essa compactação consiste, basicamente, em aproveitar estruturas repetidas de arquivos para reduzir seu tamanho.

O FAT também conta com essa capacidade, mas de maneira limitada, uma vez que é necessário compactar a partição como um todo. No NTFS, é possível a compressão somente de pastas ou de determinados arquivos.

Para compactar uma pasta no Windows 7, por exemplo, basta clicar nela com o botão direito do mouse e selecionar Propriedades. Na janela que surgir, basta clicar no botão Avançados da aba Geral e marcar a opção compactar o conteúdo para economizar espaço em disco.



É claro que também é possível comprimir uma partição inteira: clique com o botão direito do mouse na unidade, selecione Propriedades e, na aba Geral, marque a opção compactar este disco para economizar espaço.

Vale frisar que o usuário não precisa se preocupar em descompactar os arquivos quando precisar trabalhar com eles. Ao acessá-los, o próprio sistema operacional se encarrega disso. Além disso, é recomendável utilizar esse recurso com moderação e apenas em caso de necessidade para evitar possíveis problemas de desempenho.

### **Quotas de Disco**

Assim como o EFS, esta é uma funcionalidade mais recente e que não está disponível em todas as versões do Windows compatíveis com NTFS. Sua função, basicamente, é a de permitir ao administrador do sistema definir quanto espaço em disco cada usuário pode utilizar. Dessa forma, pode-se evitar problemas de desempenho ou o esgotamento dos recursos de armazenamento de um servidor, por exemplo.

### **Versões do NTFS**

Você já sabe que o NTFS não é, necessariamente, um sistema de arquivos novo e que, apesar disso, é utilizado com sucesso pela Microsoft até os dias de hoje. Então, nada mais natural ele sofrer atualizações para se adaptar às necessidades das versões mais atuais do Windows.

A versão mais conhecida do Windows NT, o Windows NT 4, fazia uso do NTFS 1.2. Aparentemente, houve versões antes desta: a 1.0, que foi utilizada no Windows NT 3.1, e a 1.1, aplicado ao Windows NT 3.5. O NTFS 1.2 passou a ser chamado de NTFS 4 em alusão à existência do número em questão no nome "Windows NT 4", embora esta versão do NTFS também tenha sido utilizada no Windows NT 3.51.

O Windows NT ficou um bom tempo no mercado, mas depois foi substituído pelo Windows 2000, que trouxe também o NTFS 5, versão dotada de vários novos recursos, entre eles: Reparse Points, onde arquivos e pastas dentro do sistema de arquivos podem ter ações associadas a eles, de forma que operações particulares a estes arquivos possam ser executadas; quotas de discos; encriptação (EFS); suporte a dados esparsos, onde é possível armazenar de forma eficiente arquivos grandes mas que possuem estruturas vazias.

Revisões do NTFS 5 foram lançadas para os sistemas operacionais Windows XP e Serve 2003. O NTFS 6 e suas variações surgiram para as versões Vista, 7 e Server 2008 do Windows.

Novas versões podem surgir com o lançamento de outras edições do Windows.

Até quando o NTFS será utilizado? Para o lançamento do Windows Vista, a Microsoft estava trabalhando em um novo sistema de arquivos chamado WinFS, cuja principal característica seria sua capacidade de trabalhar como um banco de dados relacional, permitindo ao usuário localizar facilmente os arquivos que procura.

Uma série de problemas fez com que o WinFS não fosse implementado no Windows Vista, permitindo ao NTFS manter o seu "reinado". Há, inclusive, rumores de que o projeto WinFS tenha sido cancelado.

A Microsoft continua trabalhando no desenvolvimento de novas versões de seus sistemas operacionais, mas pelo menos até a publicação deste artigo, não havia nenhuma informação sobre um novo sistema de arquivos, indicando que o NTFS terá ainda uma longa vida pela frente.

### **Sistemas de Arquivos FAT16 E FAT32**

**FAT16 e FAT 32** são nomes de sistemas de arquivos (file systems) utilizados por padrão em versões antigas do sistema operacional Windows (como o Windows 98, por exemplo), da Microsoft. Neste artigo, você conhecerá as principais características de cada um deles, assim como as diferenças entre ambos



## O Que é Um Sistema de Arquivos?

Não é possível gravar dados em um HD ou em qualquer outro dispositivo de armazenamento de forma a manter as informações acessíveis e organizadas sem um sistema de arquivos - essencialmente um tipo de estrutura que indica como os arquivos devem ser gravados e lidos pelo sistema operacional do computador.

É o sistema de arquivos que determina como as informações podem ser guardadas, acessadas, copiadas, alteradas, nomeadas e até apagadas. Ou seja, resumindo, toda e qualquer manipulação de dados em um dispositivo de armazenamento necessita de um sistema de arquivos para que estas ações sejam possíveis. Em resumo, sem um sistema de arquivos, os dados armazenados seriam apenas um conjunto de bits sem utilidade.

Há vários sistemas de arquivos disponíveis, para os mais diversos sistemas operacionais e para as mais variadas finalidades. Por exemplo, sistemas de arquivos utilizados em aplicações críticas - servidores de internet, por exemplo - costumam ter mais recursos para segurança ou para armazenamento de grandes volumes de dados.

## Sistema de Arquivos FAT

FAT é a sigla para File Allocation Table (traduzindo: Tabela de Alocação de Arquivos). A primeira versão do FAT surgiu em 1977, para trabalhar com o sistema operacional MS-DOS, mas foi padrão até o Windows 95.

Trata-se de um sistema de arquivos que funciona com base em uma espécie de tabela que indica onde estão os dados de cada arquivo. Esse esquema é necessário porque o espaço destinado ao armazenamento é dividido em blocos, e cada arquivo gravado pode ocupar vários destes, mas não necessariamente de maneira sequencial: os blocos podem estar em várias posições diferentes. Assim, a tabela acaba atuando como um "guia" para localizá-los.

Com o surgimento de dispositivos de armazenamento mais sofisticados e com maior capacidade, o sistema FAT foi ganhando revisões, identificadas pelos nomes FAT12 e FAT16, sendo o primeiro quase um desconhecido e o último padrão dos sistemas operacionais da Microsoft por muito tempo.

As versões surgem com o intuito de eliminar determinadas limitações do sistema de arquivos anterior. O próprio FAT16, por exemplo, passou por isso: esta versão só trabalha com, no máximo, 2 GB, assim, para aplicá-lo em um disco de 5 GB, seria necessário dividi-lo em 3 partições (2 GB + 2 GB + 1 GB, por exemplo) para ser possível o aproveitamento de toda a capacidade da unidade.

Diante deste e de outros problemas, a Microsoft lançou, em 1996, o FAT32, que se tornou o sistema de arquivos do Windows 95 (versão OSR 2) e do Windows 98, sendo também compatível com versões lançadas posteriormente, como Windows 2000 e Windows XP, embora estes tenham um sistema de arquivos mais avançado, o NTFS.

## Entendendo os Sistemas de Arquivos FAT

Em um disco rígido, a área de armazenamento é dividida em trilhas. Cada trilha é subdividida em setores (saiba mais neste artigo sobre HDs), cada um com 512 bytes, geralmente. Desse modo, é de se presumir que os sistemas de arquivos FAT trabalhem diretamente com esses setores. Mas não é bem assim.

Na verdade, o FAT trabalha com grupos de setores, onde cada um recebe a denominação **cluster** (ou unidade de alocação). No caso do FAT16, cada cluster pode ter, comumente, um dos seguintes tamanhos: 2 KB, 4 KB, 8 KB, 16 KB e, por fim, 32 KB. A definição desse tamanho é uniforme, ou seja, não pode haver tamanhos variados de clusters em uma mesma unidade de armazenamento.

Cada arquivo gravado utiliza tantos clusters quanto forem necessários para cobrir o seu tamanho. Se, por exemplo, tivermos um arquivo com 50 KB, é possível guardá-lo em dois clusters de 32 KB cada. Você deve ter percebido então que, neste caso, um cluster ficou com espaço sobrando. Esta área pode ser destinada a outro arquivo, correto? Errado! Acontece que cada cluster só pode ser utilizado por um único arquivo. Se sobrar espaço, este permanecerá vazio. Esse é um dos problemas do sistema FAT: desperdício.

Normalmente, o tamanho dos clusters é definido no procedimento de instalação do sistema operacional, na etapa de formatação da unidade de armazenamento.

### **Diferenças Entre FAT16 E FAT32**

O FAT16 utiliza 16 bits para endereçamento dos dados (daí o número 16 na sigla), o que, na prática, significa que o sistema de arquivos pode trabalhar com até 65536 clusters, no máximo. Para chegar a este número, basta fazer 2 elevados a 16 (65536).

Se temos então até 65536 clusters e cada um pode ter até 32 KB de tamanho, significa que o sistema FAT16 é capaz de trabalhar com discos ou partições com até 2 GB:  $65536 \times 32 = 2.097.152$  KB, que corresponde a 2 GB.

O sistema de arquivos FAT32 consegue solucionar esse problema por utilizar 32 bits no endereçamento de dados (novamente, aqui você pode perceber o porquê do número na sigla). No FAT16, quanto maior o espaço em disco (considerando o limite de até 2 GB, é claro), maior o tamanho do cluster. Com o FAT32, é possível usar clusters menores - geralmente de 4 KB - mesmo com a unidade oferecendo maior capacidade de armazenamento. Desta forma, o desperdício acaba sendo menor.

O limite do FAT32 é de 2 TB (terabytes). Perceba, no entanto, que se você fizer o cálculo anterior considerando 32 em vez de 16 (2 elevados a 32) e, posteriormente, multiplicar o resultado pelo tamanho máximo do cluster (também 32), o valor obtido será de 128 TB. Então, qual o motivo do limite de 2 TB?

Na verdade, cada endereçamento tem tamanho de 32 bits, mas, no FAT32, o número máximo de clusters é calculado considerando apenas 28 bits, fazendo a conta ser 2 elevados a 28, que é igual a 268.435.456, ou seja, pouco mais de 268 milhões de clusters. Multiplicando esse número por 32, teremos então 8 TB.

Ok, novamente a conta não fechou, afinal, 8 TB para 2 TB é uma diferença muito grande. A explicação está no fato de que a Microsoft limitou o FAT32 a ter 2 elevados a 32 como quantidade máxima de setores, não de clusters (se fosse diferente, poderia haver problemas com a inicialização do sistema operacional devido a limitações na área de boot). Como cada setor, geralmente, possui 512 bytes (ou 0,5 kilobyte), a conta seria 2 elevados a 32 ( $4.294.967.296$ ) multiplicado por 0,5, que é igual 2.147.483.648 KB ou 2 TB.

### **Fragmentação**

Para os usuários do Windows 95/98, uma recomendação era frequente: utilizar um aplicativo de desfragmentação de disco regularmente. Isso tem um bom motivo: toda vez que um arquivo é apagado, seus clusters ficam disponíveis para nova utilização. Acontece que o sistema operacional sempre ocupa os primeiros clusters livres e, se houver áreas ocupadas no caminho, continuará utilizando os clusters livres subsequentes.

O resultado é que, com o passar do tempo, há fragmentos de dados por todo o disco. Isso torna o acesso aos arquivos mais lento, uma vez que o sistema precisa procurar "pedaço por pedaço". A desfragmentação consegue amenizar este problema porque reorganiza os arquivos em clusters sequenciais, deixando-os acessíveis mais rapidamente.

### **VFAT**

VFAT é a sigla para Virtual File Allocation Table. Trata-se de uma espécie de extensão para FAT16 introduzido no Windows 95 para que este sistema operacional possa suportar arquivos com mais de 11 caracteres no nome. É que, por padrão, o FAT16 limita o tamanho dos nomes para 8 caracteres mais 3 destinados à extensão, por exemplo, daniella.txt. Caso nomes maiores sejam utilizados, estes aparecem de forma abreviada no sistema.

Com o uso do VFAT, as características do FAT16 são mantidas e, ao mesmo tempo, nomes maiores podem ser utilizados sem qualquer dificuldade. Isso porque o VFAT faz o sistema enxergar o nome abreviado, mas guarda o nome original em uma área separada.

O FAT32 trabalha com VFAT por padrão.

## **FAT12**

Antes do sistema de arquivos FAT16, que bem conhecemos, existiu o FAT12, um sistema ainda mais primitivo, utilizado em disquetes e também nas primeiras versões do MS-DOS. Nele, são usados endereços de apenas 12 bits para endereçar os clusters, permitindo um total de 4096 clusters de até 4 KB, o que permitia partições de até 16 MB.

Em 1981, quando o IBM PC foi lançado, 16 MB parecia ser uma capacidade satisfatória, já que naquela época os discos rígidos tinham apenas 5 ou 10 MB. Claro que, em se tratando de informática, por maior que seja um limite, ele jamais será suficiente por muito tempo. Um excelente exemplo é a célebre frase "Por que alguém iria precisar de mais de 640 KB de memória RAM?" dita por Bill Gates em uma entrevista, no início da década de 80. Logo começaram a ser usados discos de 40, 80 ou 120 MB, obrigando a Microsoft a criar a FAT 16, e incluí-la na versão 4.0 do MS-DOS.

Apesar de obsoleto, o FAT12 ainda continua vivo até os dias de hoje, fazendo companhia para outro fantasma da informática: os disquetes. Por ser mais simples, o FAT12 é o sistema padrão para a formatação dos disquetes de 1.44, onde são usados clusters de apenas 512 bytes.

## **EXT2**

Ext2 foi projetado e implementado para corrigir as deficiências do Ext e prover um sistema que respeitasse a semântica UNIX. A influência do UNIX pode ser vista, por exemplo, na utilização de grupos de blocos, que são análogos aos grupos de cilindros utilizados pelo FFS.

O bloco, que consiste num conjunto de setores (cada setor tem 512 bytes), é a menor unidade de alocação para o Ext2. O tamanho pode ser de 1024, 2048 ou 4096 bytes e é definido na formatação.

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:

- (i) a 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;
- (ii) 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.

Para diminuir o impacto do primeiro tipo, existem duas estratégias básicas. A primeira, mais simples, é determinar, na formatação, o menor tamanho de bloco possível. O Ext2 permite tamanhos de blocos de 1024, 2048 e 4096 bytes.

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.

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 chama esse método de tail packing; o UFS usa fragmentos, que são submúltiplos do tamanho do bloco. Apesar do Ext2 possuir, no superbloco, a previsão para uso de fragmentos, esse método não foi implementado.

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. [CARD, TS'O e TWEEDIE, 1994]

### 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.

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 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.

Embora o seu desempenho (velocidade) seja menos atrativo que o de outros sistemas de arquivos (como ReiserFS 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 significativa vantagem sobre sistemas de arquivos que as possuem.

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 significativa, em que sistemas de arquivos em árvore não seriam recuperáveis.

### Introdução ao Sistema de Arquivos Reiserfs

#### Principais Características

O sistema de arquivos ReiserFS teve sua primeira aparição no ano de 2001 pelas mãos de Hans Reiser (daí o nome do padrão), que também montou uma equipe de nome NAMESYS para gerenciar os trabalhos do projeto. Desde então, o ReiserFS vem sendo cada vez mais utilizado, principalmente por estar disponível como padrão em muitas das distribuições Linux, fazendo frente ao sistema de arquivos ext3.

A boa aceitação do ReiserFS é devida ao seu conjunto de características, que o tornam um sistema de arquivos seguro, eficiente, rápido e confiável. Entre seus principais recursos, tem-se:

- Journaling, um recurso que ajuda a manter a integridade dos dados em caso de erros no sistema causados por desligamento incorreto ou determinadas falhas de hardware, por exemplo. O journaling é uma das características mais importantes do ReiserFS, motivo pelo qual é explicado com mais detalhes adiante;
- Suporte a arquivos com mais de 2 GB (limitação existente em alguns filesystems);
- Organização dos objetos do sistema de arquivos em uma estrutura de dados chamada B+Trees (árvores B+). Nesse esquema, os dados são fixados em posições organizadas por divisões denominadas folhas. Por sua vez, as folhas são organizadas por nós ou ponteiros chamados de sub-árvores, que estão ligados a um nó raiz (ver ilustração abaixo para entender melhor). Esse processo organizacional exige algoritmos mais complexos, porém apresenta performance superior na gravação e no

