

Sistemas de arquivos

PRÓXIMO: Formatação física

A forma como a controladora do HD vê os dados armazenados nos discos magnéticos pode ser bem diferente da forma como vê o sistema operacional. Enquanto a controladora enxerga as trilhas, setores e cilindros e se esforça para localizá-los nos discos

magnéticos, usando as marcações servo, o sistema operacional enxerga apenas uma longa lista de endereços, chamados de clusters ou blocos. Quando ele precisa de um determinado arquivo, ele não se preocupa em tentar descobrir em qual trilha e setor ele está

armazenado. Ele apenas envia o endereço do bloco que deve ser lido e a controladora se encarrega do restante.

O fato da controladora “esconder” as informações sobre a organização interna dos discos, é o que faz com que os sistemas operacionais sejam compatíveis com todos os HDs do mercado, sem que seja necessário instalar drivers completos para cada um. Quando

acontece de uma versão antiga do Windows, ou de alguma distribuição Linux não detectar seu HD durante a instalação, quase sempre o problema é causado pela falta de drivers para a interface IDE ou para a controladora SATA do chipset da placa-mãe, e não

para o HD em si. A primeira versão do Windows XP, por exemplo, não oferecia suporte nativo à maioria das controladoras SATA, de forma que era preciso fornecer um disquete com drivers durante a instalação.

Originalmente, os discos magnéticos do HD são um terreno inexplorado, uma mata virgem sem qualquer organização. Para que os dados possam ser armazenados e lidos de forma organizada, é necessário que o HD seja previamente formatado.

Em primeiro lugar, temos a formatação física, na qual os discos são divididos em trilhas, setores e cilindros e são gravadas as marcações servo, que permitem que a placa lógica posicione corretamente as cabeças de leitura.

Nos HDs atuais, a formatação física é feita em fábrica, durante a fabricação dos discos. O processo envolve o uso de máquinas especiais e, apenas para garantir, restrições são adicionadas no firmware do drive, para que a placa lógica seja realmente

impedida de fazer qualquer modificação nas áreas reservadas. Graças a isso, é impossível reformatar fisicamente um drive atual, independentemente do software usado.

No caso dos drives “pré-ATA”, como os antigos ST-506 e ST-412, a história era diferente. Eles precisavam ser periodicamente reformatados fisicamente através do setup, pois as mudanças de temperatura, a vibração causada pelo motor de rotação e as

próprias operações de leitura e gravação acabavam por alterar a posição das trilhas na mídia magnética, causando desalinhamento e dificultando a leitura dos dados pela cabeça de leitura. Era necessária, então, uma nova formatação física, para que as

trilhas, setores e cilindros, voltassem as suas posições iniciais.

No caso dos discos atuais, esse processo não é mais necessário, pois as mídias são muito mais confiáveis e a placa controladora pode compensar eventuais desvios rapidamente, simplesmente calibrando o movimento do braço de leitura.

Em seguida, temos a formatação lógica, que adiciona as estruturas utilizadas pelo sistema operacional. Ao contrário da formatação física, ela é feita via software e pode ser refeita quantas vezes você quiser. O único problema é que, ao reformatar o HD, você perde o acesso aos dados armazenados, embora ainda seja possível recuperá-los usando as ferramentas apropriadas, como veremos mais adiante.

Chegamos então ao sistema de arquivos, que pode ser definido como o conjunto de estruturas lógicas que permitem ao sistema operacional organizar e otimizar o acesso ao HD. Conforme cresce a capacidade dos discos e aumenta o volume de arquivos e acessos, esta tarefa torna-se mais e mais complicada, exigindo o uso de sistemas de arquivos cada vez mais complexos e robustos.

Existem diversos sistemas de arquivos diferentes, que vão desde sistemas simples como o FAT16, que utilizamos em cartões de memória, até sistemas como o NTFS, EXT3 e ReiserFS, que incorporam recursos muito mais avançados.

A formatação do HD é feita em duas etapas. A primeira é o particionamento, onde você define em quantas partições o HD será dividido e o tamanho de cada uma. Mesmo que você não pretenda instalar dois sistemas em dual boot, é sempre interessante dividir o HD em duas partições, uma menor, para o sistema operacional, e outra maior, englobando o restante do disco para armazenar seus arquivos. Com isso, você pode reinstalar o sistema quantas vezes precisar, sem o risco de perder junto todos os seus arquivos.

Podemos ter um total de 4 partições primárias ou três partições primárias e mais uma partição estendida, que pode englobar até 255 partições lógicas. É justamente a partição lógica que permite dividir o HD em mais de 4 partições.

Esta limitação das 4 partições primárias é uma limitação que existe desde o primeiro PC, lançado em 1981. Os projetistas que escreveram o BIOS para ele precisavam economizar memória e chegaram à conclusão que 2 bits (4 combinações) para o endereço das partições seriam suficientes, pois na época os HDs mais vendidos tinham apenas 5 MB e só existia um sistema operacional para PCs (o MS-DOS), de forma que era raro alguém precisar criar mais de uma partição. As coisas mudaram “um pouco” de lá pra cá, mas infelizmente a limitação continua até os dias de hoje.

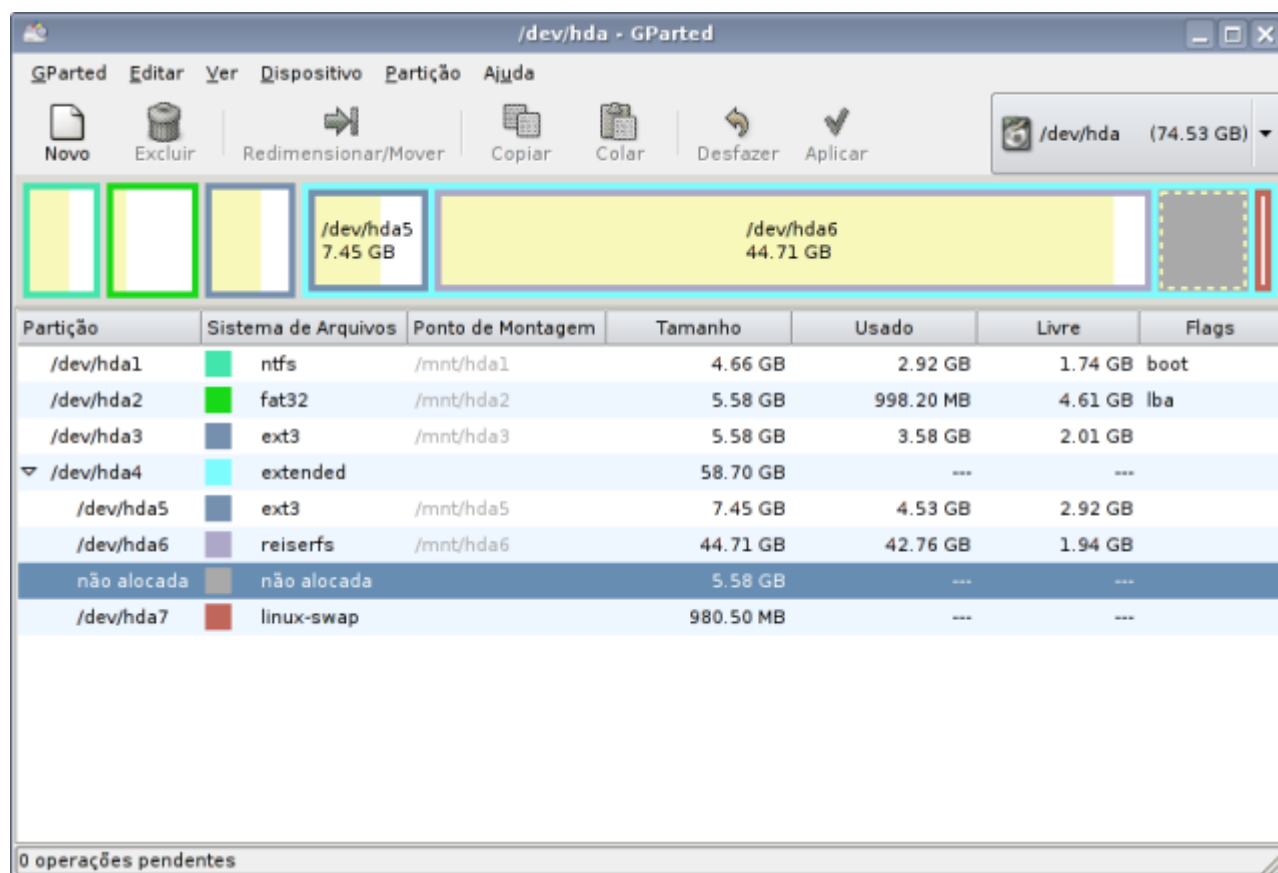
Para amenizar o problema, foi adicionada a possibilidade de criar partições lógicas. Em vez de criar 4 partições primárias e ficar sem endereços para criar novas partições, você cria uma “partição estendida”, que é uma espécie de container, que permite criar mais partições. A partição estendida contém uma área extra de endereçamento, que permite endereçar as 255 partições lógicas. É possível criar até 4 partições estendidas, de forma que (em teoria) é possível dividir o HD em até 1020 partições.

Digamos que você queira particionar um HD de 160 GB para instalar Windows e Linux em dual boot, deixando uma partição de 20 GB para o Windows, uma partição de 20 GB para o Linux, uma partição de 1 GB para swap (do Linux) e uma partição maior, englobando os 119 GB restantes para guardar seus arquivos.

Como precisamos de 4 partições no total, seria possível criar diretamente 4 partições primárias, mas neste caso você ficaria sem endereços e perderia a possibilidade de criar novas partições mais tarde, caso resolvesse testar uma outra distribuição, por exemplo.

Ao invés disso, você poderia começar criando a partição de 20 GB do Windows como primária (é sempre recomendável instalar o Windows na primeira partição do HD e em uma partição primária, devido às particularidades do sistema) e em seguida criar uma partição estendida, englobando todo o resto do espaço, criando as demais partições como partições lógicas dentro dela.

Este é um screenshot do Gparted, que mostra um HD dividido em várias partições. Veja que a quarta partição está marcada como “extended”, ou seja, como partição estendida. Ela não armazena dados, nem ocupa um espaço considerável no disco, mas permitiu que fossem criadas as partições de 5 a 7. Veja que existe também um trecho marcado como “não alocada”, ou seja, espaço vago onde é possível criar mais uma partição:



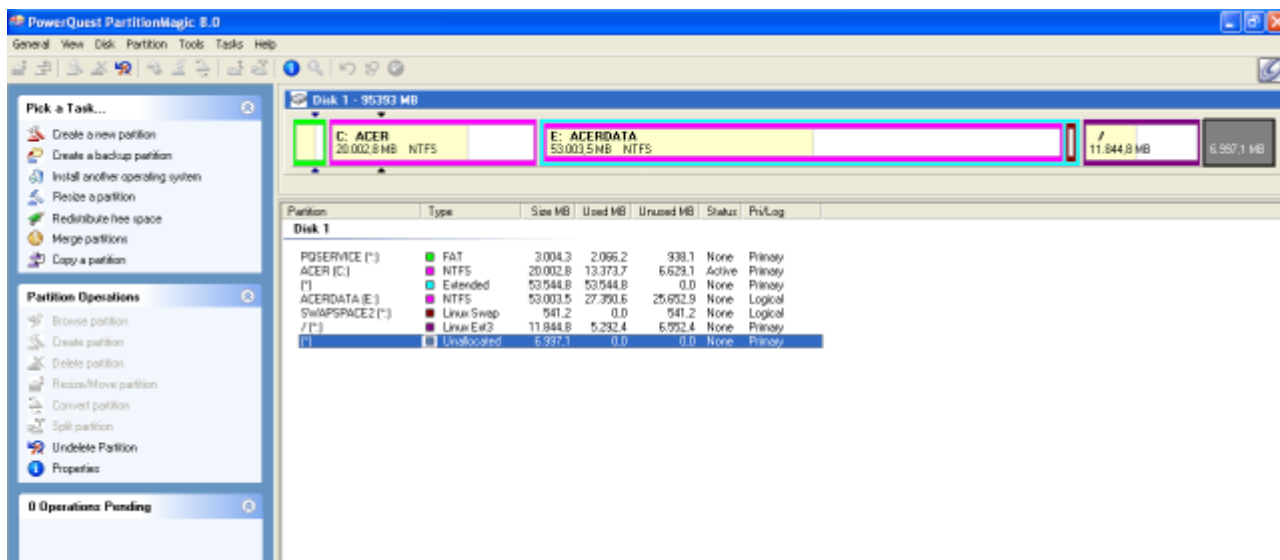
Exemplo de particionamento no Gparted

Do ponto de vista do sistema operacional, cada partição é uma unidade separada, quase como se houvesse dois ou três discos rígidos instalados na máquina. Cada partição possui seu próprio diretório raiz e sua própria FAT. As informações sobre o número de partições, sua localização no disco e o espaço ocupado por cada uma são armazenadas na tabela de partição, que compartilha o primeiro setor do disco com o setor de boot.

Você pode particionar o HD usando o próprio assistente mostrado durante a instalação do Windows XP ou Vista, usando um dos particionadores mostrados durante a instalação de várias distribuições Linux ou através de programas avulsos, como o Partition Magic (no Windows) ou o Gparted (no Linux), que você pode usar dando boot através de uma distribuição live-CD que o traga pré-instalado.

Tanto o PartitionMagic quanto o Gparted são particionadores gráficos fáceis de usar. O espaço disponível é mostrado na forma de uma barra na parte superior da tela, que vai sendo dividida em retângulos menores, conforme você vai criando as partições. A cor de cada partição representa o sistema de arquivos usado e os espaços não particionados do disco aparecem em cinza. Além de criar e deletar partições, os dois programas também oferecem opções adicionais, como redimensionar partições (sem perder os dados), muito útil quando você já tem um sistema operacional instalado e precisa liberar espaço para instalar um segundo sistema em dual boot, por exemplo.

Este é um screenshot do PartitionMagic. Veja que a interface é muito similar à do Gparted, que mostrei a pouco:



PartitionMagic

Em seguida, temos a formatação propriamente dita, onde as estruturas do sistema de arquivos são finalmente gravadas na partição. Na maioria dos casos, o próprio programa de particionamento se encarrega de formatar a partição usando o sistema de arquivos escolhido, mas, em outros, temos dois programas diferentes, como no caso do fdisk e do format, usados no Windows 98.

No mundo Windows, temos apenas três sistemas de arquivos: FAT16, FAT32 e NTFS. O FAT16 é o mais antigo, usado desde os tempos do MS-DOS, enquanto o NTFS é o mais complexo e atual. Apesar disso, temos uma variedade muito grande de sistemas de arquivos diferentes no Linux (e outros sistemas Unix), que incluem o EXT2, EXT3, ReiserFS, XFS, JFS e muitos outros. Para quem usa apenas o Windows, estes sistemas podem parecer exóticos, mas eles são velhos conhecidos de quem trabalha com servidores, já que neles o Linux é o sistema mais popular.

Vamos começar estudando as estruturas do sistema FAT. Por ser o sistema mais antigo ele é também o mais simples e mais fácil de entender.

O sistema FAT16 é uma espécie de “pau pra toda obra”, já que é compatível com praticamente todos os sistemas operacionais e também dispositivos como câmeras, palmtops, celulares e mp3players. Ele é o sistema de arquivos usado por padrão nos cartões SD e também nos pendrives de até 2 GB. Só recentemente os cartões passaram a utilizar FAT32, com a introdução do padrão SDHC.

No sistema FAT, o HD é dividido em clusters, que são a menor parcela do HD vista pelo sistema operacional. Cada cluster possui um endereço único, que permite ao sistema localizar os arquivos armazenados. Um grande arquivo pode ser dividido em vários clusters, mas não é possível que dois arquivos pequenos sejam gravados dentro do mesmo cluster. Cada cluster pode ser composto por de 1 a 64 setores (ou seja, de 512 bytes a 32 KB), de acordo com o tamanho da partição.

A principal limitação é que, como o nome sugere, o FAT16 usa endereços de 16 bits para endereçar os clusters dentro da partição, permitindo um máximo de 65536 clusters, que não podem ser maiores que 32 KB. Isso resulta num limite de 2 GB para as partições criadas.

No caso de HDs (e também pendrives ou cartões) maiores que 2 GB, é possível criar várias partições de 2 GB cada uma, até utilizar todo o espaço disponível. Esta pode ser uma solução no caso de dispositivos com 4 ou 5 GB, por

exemplo, mas, naturalmente, não é uma opção realística no caso de um HD de 60 GB, por exemplo, onde seria necessário criar 30 partições!

Numa partição de 2 GB, cada cluster possui 32 KB, o que acaba resultando num grande desperdício de espaço ao gravar uma grande quantidade de arquivos pequenos. Imagine que gravássemos 10.000 arquivos de texto, cada um com apenas 300 bytes. Como um cluster não pode conter mais do que um arquivo, cada arquivo iria ocupar um cluster inteiro, ou seja, 32 kbytes. No total, os 10.000 arquivos ocupariam um total de 10.000 clusters, ou seja, um total de 320 MB!

O tamanho dos clusters em partições FAT16 varia de acordo com o tamanho da partição. Quanto maior o tamanho da partição, maior o tamanho dos clusters:

Tamanho da Partição	Tamanho dos Clusters usando FAT16
Entre 1 e 2 GB	32 KB
Menor que 1 GB	16 KB
Menor que 512 Mb	8 KB
Menor que 256 Mb	4 KB

Como em toda regra, existe uma exceção. O Windows NT permitia criar partições FAT de até 4 GB usando clusters de 64 KB, mas este foi um recurso pouco usado, devido ao desperdício de espaço.

A versão original do Windows 95 suportava apenas o FAT16, obrigando quem possuía HDs maiores que 2 GB a dividi-los em duas ou mais partições e lidar com o desperdício de espaço causado pelos clusters de 32 KB.

A solução foi a criação do sistema FAT32, que foi incorporado no Windows 95 OSR/2 e continuou sendo usado nas versões seguintes.

A principal evolução foi o uso de endereços de 32 bits para o endereçamento dos clusters, o que possibilita a criação de partições muito maiores, de até 2 terabytes. Isso foi possível por que o Windows 95 era um sistema de 32 bits, ao contrário do MS-DOS e do Windows 3.1, que eram sistemas de 16 bits.

A princípio, o uso de endereços de 32 bits permitiriam o uso de clusters de 4 KB mesmo em partições muito grandes mas, por questões de desempenho, ficou estabelecido que por default os clusters de 4 KB seriam usados apenas em partições de até 8 GB.

Acima disto, o tamanho dos clusters varia de acordo com o tamanho da partição:

Tamanho da partição	Tamanho do cluster
Menor do que 8 GB	4 KB
De 8 GB a 16 GB	8 KB
De 16 GB a 32 GB	16 KB
Maior do que 32 GB	32 KB

Usando clusters de 4 KB, os 10.000 arquivos do exemplo anterior ocupariam apenas 40 MB, uma economia considerável. De fato, ao converter uma partição FAT16 para FAT32 é normal conseguir de 10 a 20% de redução no espaço ocupado, devido à redução do espaço desperdiçado.

O Windows 98 inclui um conversor, que permite converter partições FAT16 para FAT32 sem perda de dados. Atualmente ele não possui muita utilidade, já que o FAT16 é raramente usado fora dos pendrives e cartões de memória mas, de qualquer forma, caso precise dele, o ícone está disponível no Iniciar > Ferramentas de Sistema.

A grande limitação do sistema FAT32 está relacionada ao tamanho máximo dos arquivos. Mesmo usando uma grande partição, não é possível armazenar arquivos com mais de **4 GB**, o que é um grande problema para quem trabalha com arquivos grandes, como vídeos em formato RAW (sem compressão). Não é possível sequer armazenar um ISO de DVD, já que a cópia ou transferência será sempre abortada depois de transferidos os primeiros 4 GB.

Não existe qualquer sinal de que futuras versões do sistema de arquivos derrubarão esta limitação, já que a Microsoft vem recomendando o uso do NTFS desde a primeira versão do Windows XP, de forma que a melhor opção, para quem usa Windows, é seguir a recomendação e migrar para ele.

Outra limitação é que o particionador usado durante a instalação do Windows XP se recusa a formatar partições FAT32 maiores do que 32 GB. Este é um limite do software e não do sistema de arquivos em si. A solução para criar partições FAT maiores é utilizar o PartitionMagic, Gparted ou outro particionador para criar a partição e em seguida apenas instalar o sistema na partição criada.

Uma curiosidade é que, antes do FAT16, 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.

Todos os vários sistemas de arquivos são constituídos de um conjunto de estruturas lógicas, que permitem ao sistema operacional organizar os dados gravados e acessá-los com a maior velocidade e confiabilidade possíveis.

Tudo começa com o **setor de boot**, que é lido pelo BIOS da placa-mãe no início do boot, logo após a contagem de memória e outros procedimentos executados durante o POST.

O setor de boot, também chamado de MBR ou trilha zero, contém dois componentes essenciais. O primeiro é um bootstrap, o software responsável por iniciar o carregamento do sistema operacional. Tipicamente, é utilizado um gerenciador de boot, como o NTLDR (usado pelo Windows XP) ou o Grub (usado pela maior parte das distribuições Linux). A função do gerenciador de boot é mostrar uma lista com os sistemas operacionais instalados no início do boot e carregar o sistema escolhido.

O bootstrap ocupa os primeiros 446 bytes do MBR. Os 66 bytes restantes são usados para armazenar a **tabela de partições**, que guarda informações sobre onde cada partição começa e termina. Alguns vírus, além de acidentes em geral, podem danificar os dados armazenados na tabela de partição, fazendo com que pareça que o HD foi formatado. Mas, na maioria dos casos, os dados continuam lá, intactos, e podem ser recuperados.

Depois que o disco rígido foi formatado e dividido em clusters, mais alguns setores são reservados para guardar a **FAT** ("file allocation table" ou "tabela de alocação de arquivos"). A função da FAT é servir como um índice, armazenando informações sobre cada cluster do disco. Através da FAT, o sistema sabe se uma determinada área do disco está ocupada ou livre e pode localizar qualquer arquivo armazenado.

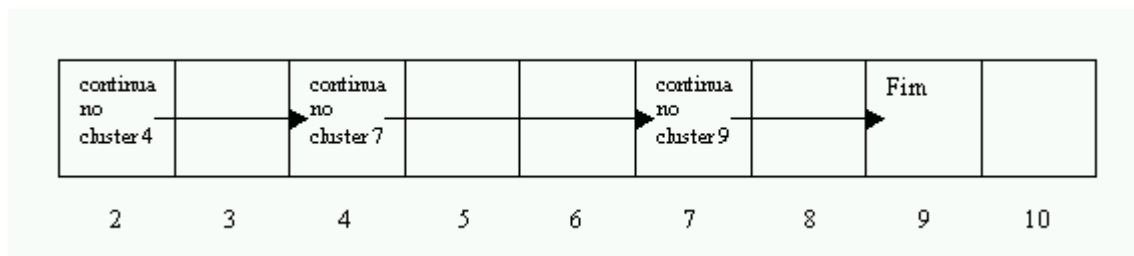
Cada vez que um novo arquivo é gravado ou apagado, o sistema operacional altera a FAT, mantendo-a sempre atualizada. A FAT é tão importante que, além da tabela principal, é armazenada também uma cópia de segurança, que é usada sempre que a tabela principal é danificada de alguma maneira.

Todos os demais sistemas de arquivos utilizam algum tipo de índice, similar à FAT. Quando o HD é reformatado, este índice é apagado e substituído por uma tabela em branco. Apesar disso, os arquivos continuam gravados nas mesmas posições, embora inacessíveis. Enquanto eles não forem realmente sobrescritos por outros, é possível recuperá-los usando um programa de recuperação de dados, como veremos em detalhes mais adiante.

Em seguida, temos o **diretório raiz**. Se fôssemos comparar um disco rígido, formatado em FAT16 ou FAT32 com um livro, as páginas seriam os clusters, a FAT serviria como as legendas e numeração das páginas, enquanto o diretório raiz seria o índice, com o nome de cada capítulo e a página onde ele começa.

O diretório raiz ocupa mais alguns setores no disco, logo após os setores ocupados pela FAT. Cada arquivo ou diretório do disco rígido possui uma entrada no diretório raiz, com o nome do arquivo, a extensão, a data de quando foi criado ou quando foi feita a última modificação, o tamanho em bytes e o número do cluster onde o arquivo começa.

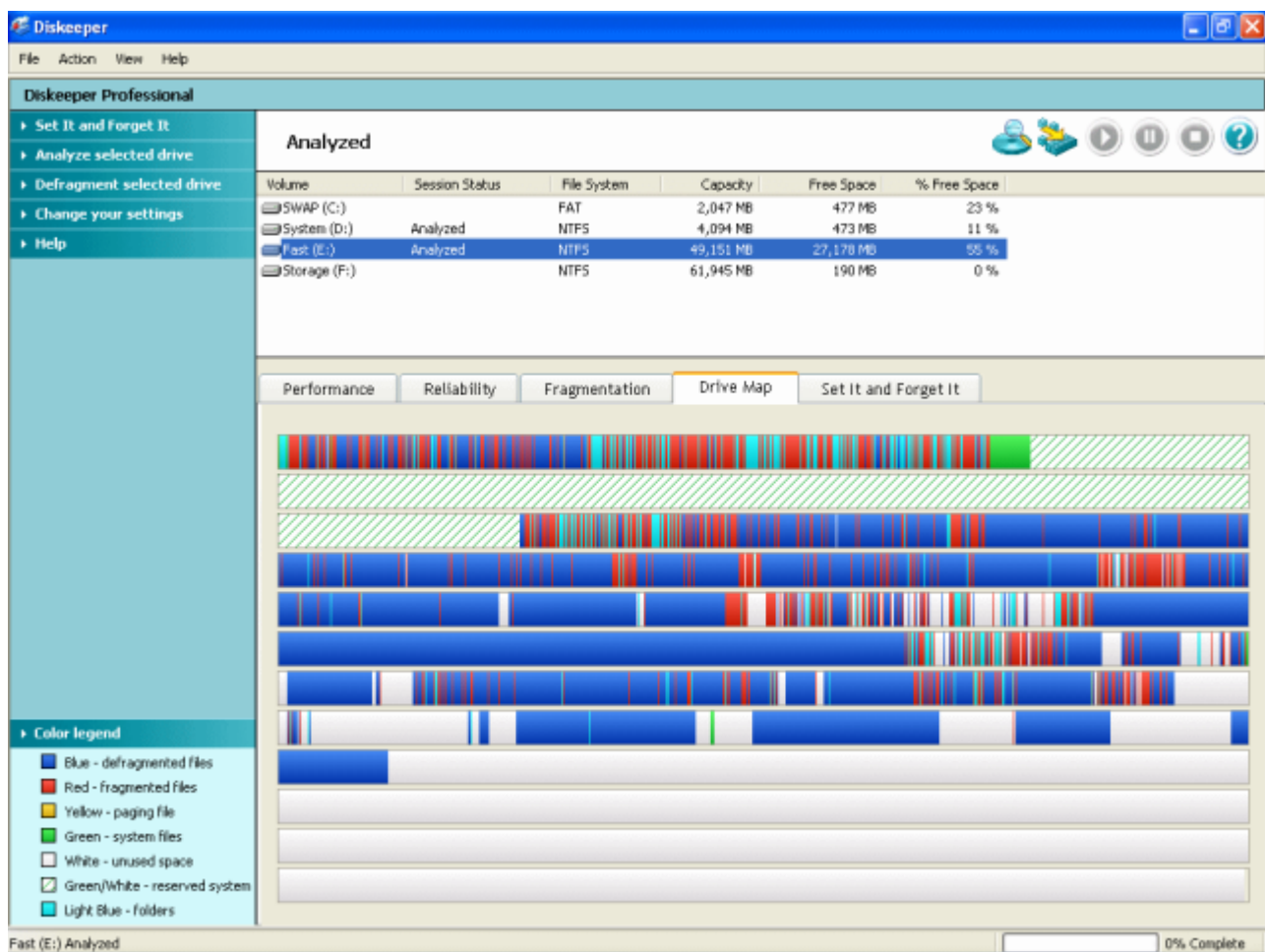
Um arquivo pequeno pode ser armazenado em um único cluster, enquanto um arquivo grande é "quebrado" e armazenado ocupando vários clusters. Nesse caso, haverá no final de cada cluster uma marcação, indicando o próximo cluster ocupado pelo arquivo. No último cluster ocupado, temos um código que marca o fim do arquivo.



Quando um arquivo é deletado, simplesmente é removida sua entrada no diretório raiz, fazendo com que os clusters ocupados por ele pareçam vagos para o sistema operacional. Ao gravar um novo arquivo no disco, o sistema simplesmente procura o primeiro setor livre, continuando a gravá-lo nos setores livres seguintes, mesmo que estejam muito distantes uns dos outros. Surge então o problema da **fragmentação**, que reduz consideravelmente a velocidade de acesso, já que dados espalhados significam mais movimentos da cabeça de leitura.

Ao contrário de outros sistemas de arquivos mais modernos, o sistema FAT (tanto o FAT16 quanto o FAT32) não possui nenhum mecanismo que impeça, ou pelo menos diminua a fragmentação, daí a necessidade de rodar o

defrag ou outro programa desfragmentador periodicamente. A função deles é mover os arquivos, de forma que eles fiquem gravados em clusters seqüenciais.



Desfragmentação do HD (no Windows), usando o Diskeeper

Uma curiosidade é que a fragmentação é um problema apenas nos HDs, já que eles trabalham com tempos de acesso muito altos. Nos cartões de memória, o tempo de acesso é comparativamente muito baixo, de forma que a fragmentação possui um impacto muito pequeno sobre a performance.

Continuando, além do nome, cada arquivo recebe também uma **extensão** de até três caracteres, como “EXE”, “DOC”, etc. Através da extensão, o sistema operacional sabe que um determinado arquivo deve ser executado ou aberto usando o Word, por exemplo.

A extensão não tem nenhuma influencia sobre o arquivo, apenas determina como ele será visto pelo sistema operacional. Se você abrir o Notepad, escrever um texto qualquer e salvá-lo como “carta.exe”, por exemplo, conseguirá abrir e editar este arquivo sem problemas, mas se você chamar o arquivo clicando duas vezes sobre ele dentro do Windows Explorer, o sistema operacional verá a extensão “EXE” e tentará executar o arquivo, ao invés de tentar abri-lo usando o Notepad, como faria caso o arquivo tivesse a extensão “TXT”.

Inicialmente, as extensões de arquivo eram utilizadas apenas no Windows, enquanto o Linux e os sistemas Unix em geral se baseavam no conteúdo do arquivo. Mas, recentemente isso vem mudando, com o KDE e o Gnome utilizando associações de arquivos também baseadas nas extensões.

Depois da extensão, existe mais um byte reservado para o atributo do arquivo, que pode ser “somente leitura”, “oculto”, “sistema”, “volume label”, “diretório” ou “arquivo”. O atributo permite instruir o sistema operacional e demais aplicativos sobre como lidar com o arquivo. Os atributos suportados mudam de acordo com o sistema de arquivos. Como veremos adiante, o NTFS suporta um conjunto de atributos que não existem no sistema FAT. O mesmo pode ser dito em relação aos sistemas de arquivo usados no Linux.

O atributo **“somente leitura”** indica que o arquivo não deve ser modificado ou deletado. Se você tentar deletar ou modificar um arquivo somente leitura pelo DOS, receberá a mensagem **“Access Denied”**. Tentando apagar o arquivo através do Windows Explorer você receberá um aviso explicando que o arquivo é somente para leitura, perguntando se você tem certeza que deseja deletá-lo. O atributo **“sistema”** possui uma função parecida, indicando apenas que, além de ser oculto, o arquivo é utilizado pelo sistema operacional.

Por padrão, o Windows XP não mostra arquivos marcados com o atributo **“sistema”**, nem os arquivos ocultos. É necessário configurar o Windows Explorer para exibi-los, marcando a opção **“Mostrar todos os arquivos”** no menu Exibir/Opções. Outra opção é usar o comando **“DIR /AH”** através da linha de comando.

Para nomear um disquete ou a uma partição de um disco rígido, usamos o atributo **“volume label”**. O nome dado é armazenado em uma área reservada do diretório raiz.

De todos os atributos, o mais importante é o atributo de **“diretório”**, pois ele permite a existência de subpastas. As pastas, mesmo quando vazias, são vistas pelo sistema operacional como arquivos. Dentro desse arquivo ficam armazenadas informações sobre o nome da pasta, atributos como somente leitura, oculto, etc., a posição da pasta na árvore de diretórios (C:\Windows\System, por exemplo) e informações sobre quais arquivos ou subpastas estão guardados dentro dela, assim como a localização destes arquivos no disco.

Como o diretório raiz ocupa (no sistema FAT) um espaço equivalente a apenas 16 KB no disco rígido (32 setores), podemos ter apenas 512 entradas de arquivos ou diretórios. Cada sub-pasta funciona mais ou menos como um novo diretório raiz, permitindo que tenhamos mais arquivos no disco. Como uma pasta (que nada mais é do que um arquivo, marcado com o atributo especial) pode ocupar o espaço que quiser, não temos a limitação de 512 arquivos, como no diretório raiz.

Qualquer arquivo com o atributo **“diretório”**, passa a ser visto pelo sistema operacional como uma pasta, mas a tentativa de transformar um arquivo qualquer em pasta não daria certo pois, apesar de, em essência, as pastas também serem arquivos, elas possuem um formato específico.

Uma curiosidade sobre as subpastas é que elas só passaram a ser suportadas a partir da versão 2.0 do DOS. Os usuários do DOS 1.0 tiveram que conviver durante algum tempo com um sistema que permitia armazenar arquivos apenas no diretório raiz, com a conseqüente limitação de 512 arquivos no HD.

Finalizando, o atributo **“arquivo”** é usado com relação a arquivos que raramente são modificados, ou que são cópias de backup de outros arquivos. Muitos programas de backup verificam este atributo quando fazem um backup incremental (quando são salvos apenas os arquivos que foram alterados desde o último backup). Nesse caso, o programa de backup retira o atributo após salvar o arquivo. Ao ser alterado por algum outro programa, o arquivo novamente recebe o atributo, permitindo ao programa de backup saber quais arquivos foram modificados.

Para alterar os atributos de um arquivo através do Windows Explorer, basta clicar sobre ele com o botão direito do mouse e abrir a janela de propriedades. Também é possível alterá-los via linha de comando, usando o comando ATTRIB.

Concluindo, temos o **VFAT** (Virtual File Allocation Table), uma extensão incluída a partir do Windows 95 e suportado também no Linux e outros sistemas. Inicialmente, o sistema FAT possuía uma grave limitação quanto ao tamanho dos nomes de arquivos, que não podiam ter mais que 11 caracteres, sendo 8 para o nome do arquivo e mais 3 para a extensão, como em **“formular.doc”**.

O limite de apenas 8 caracteres era um grande inconveniente para os usuários do MS-DOS. O **“Boletim da 8a reunião anual de diretoria”**, por exemplo, teria de ser gravado na forma de algo como **“8reandir.doc”**.

Através do VFAT, arquivos com nomes longos são gravados no diretório raiz respeitando o formato 8.3 (oito letras e uma extensão de até 3 caracteres), sendo o nome verdadeiro armazenado numa área reservada. Se tivéssemos dois

arquivos, chamados de “Reunião anual de 1998” e “Reunião anual de 1999”, por exemplo, teríamos gravados no diretório raiz “Reunia~1” e “Reunia~2”. Se o disco fosse lido a partir do DOS, o sistema leria apenas este nome simplificado. Lendo o disco através do Windows, é possível acessar as áreas ocultas do VFAT e ver os nomes completos dos arquivos. Isso permitiu que a Microsoft derrubasse a limitação, sem quebrar a compatibilidade com os softwares antigos.

O NTFS é um sistema de arquivos mais antigo do que muitos acreditam. Ele começou a ser desenvolvido no início da década de 1990, quando o projeto do Windows NT dava os seus primeiros passos.

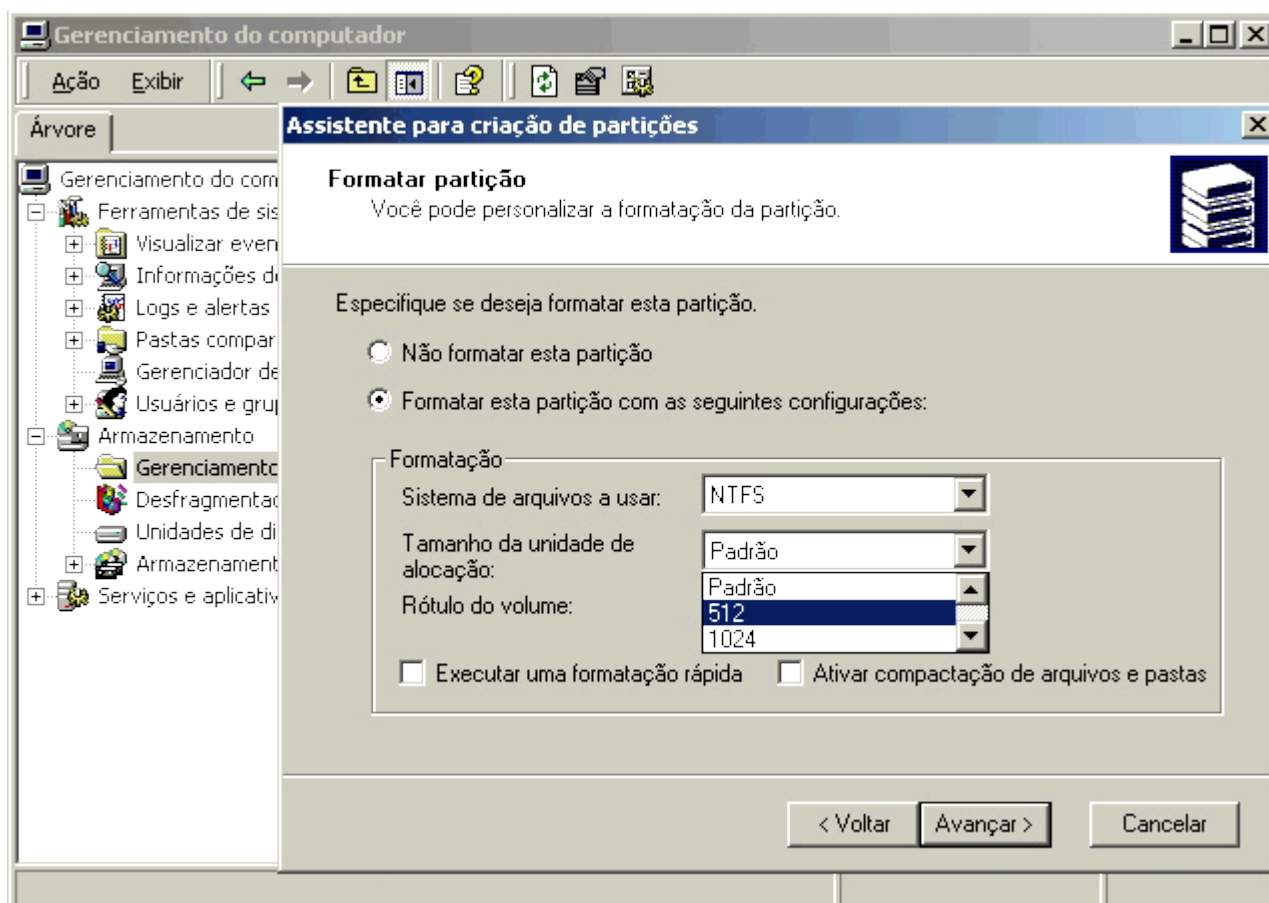
Já que o grande problema do sistema FAT16 era o fato de serem usados apenas 16 bits para o endereçamento de cada cluster, permitindo apenas 65 mil clusters por partição, o NTFS incorporou desde o início a capacidade para endereçar os clusters usando endereços de 64 bits. A única limitação agora passa a ser o tamanho dos setores do HD. Como cada setor possui 512 bytes, o tamanho de cada cluster usando NTFS também poderá ser de 512 bytes, independentemente do tamanho da partição.

É sem dúvida um grande avanço sobre os clusters de 32 KB e as partições de até 2 GB do sistema FAT16. Mas, existe um pequeno problema em endereçar partições muito grandes usando clusters de 512 bytes: o desempenho. Com um número muito grande de clusters, o processamento necessário para encontrar os dados desejados passa a ser muito grande, diminuindo a performance.

Assim como na FAT 32, ficou estabelecido que o tamanho mínimo de clusters seria usado por default apenas em partições de um certo tamanho:

Tamanho da partição	Tamanho do cluster
até 512 MB	512 bytes
até 1 GB	1 KB
até 2 GB	2 KB
acima de 2 GB	4 KB

Apesar do default ser usar clusters de 4 KB em qualquer partição maior do que 2 GB, você pode criar partições com clusters do tamanho que desejar, através do assistente para criação de partições do Windows 2000/XP, que pode ser encontrado em Painel de controle > Ferramentas Administrativas > Gerenciamento do computador > Armazenamento > Gerenciamento de disco. Do lado direito da tela será mostrado um mapa dos HDs instalados na máquina, basta clicar com o botão direito sobre uma área de espaço livre e em seguida em “criar partição”:



Continuando, mais uma vantagem do sistema NTFS é que os nomes de arquivos e pastas utilizam caracteres em Unicode, em vez de ACSII. O ASCII é o sistema onde cada caracter ocupa 1 byte de dados, mas são permitidas apenas letras, números e alguns caracteres especiais. No Unicode, cada caracter ocupa dois bytes, o que permite 65 mil combinações, o suficiente para armazenar caracteres de diversos idiomas. Isso permite que usuários do Japão, China, Taiwan e outros países que não utilizam o alfabeto ocidental, possam criar arquivos usando caracteres do seu próprio idioma, sem a necessidade de instalar drivers e programas adicionais.

Outro ponto importante onde o NTFS é superior ao sistema FAT é na tolerância a falhas. No sistema FAT, sempre que o sistema trava ou é desligado enquanto estão sendo atualizados arquivos e diretórios no HD, existe uma possibilidade muito grande do sistema tornar-se inconsistente, com arquivos interligados, agrupamentos perdidos e outros problemas. Surge, então, a necessidade de rodar o scandisk depois de cada desligamento incorreto.

No NTFS, o sistema mantém um log de todas as operações realizadas. Com isto, mesmo que o micro seja desligado bem no meio da atualização de um arquivo, o sistema poderá, durante o próximo boot, examinar este log e descobrir exatamente em que ponto a atualização parou, tendo a chance de automaticamente corrigir o problema. Além de reduzir a perda de tempo, a possibilidade de perda de dados é muito menor.

Se você chegou a usar o Windows 95/98/ME, deve lembrar-se da “tela de boas vindas” do scandisk, que era executado após cada desligamento incorreto:

Microsoft ScanDisk

ScanDisk is now checking the following areas of drive C:

- √ Media descriptor
- √ File allocation tables
- √ Directory structure
- » File system
- Free space
- Surface scan

◀ Pause ▶

< More Info >

< Exit >

3% complete



O famigerado teste do scandisk

Clusters contendo setores defeituosos também são marcados automaticamente, conforme são detectados, sem a necessidade de usar o scandisk ou qualquer outro utilitário. Nesse caso, a marcação é feita na tabela de endereçamento da partição, de forma que a lista de setores defeituosos é perdida ao reparticionar o HD.

Antigamente, os HDs eram menos confiáveis e o aparecimento de setores defeituosos, um fenômeno muito mais comum, de forma que a maioria dos programas de formatação realizavam um teste de superfície durante a formatação da partição (como no caso do format usado no Windows 95/98, onde formatar uma partição podia demorar mais de uma hora). Atualmente, a maioria dos programas realiza uma formatação rápida, presumindo que o HD não possua setores defeituosos.

Existiram diversas versões do NTFS, que acompanharam a evolução do Windows NT. A partir do Windows 2000, foi introduzido o NTFS 5, que trouxe diversos aperfeiçoamentos, incluindo o suporte ao Active Directory.

Outro recurso interessante é a possibilidade de encriptar os dados gravados, de forma a impedir que sejam acessados por pessoas não autorizadas, mesmo caso o HD seja removido e instalado em outro micro. Este recurso de encriptação é interessante, por exemplo, para profissionais de campo, que levam dados secretos em seus laptops. É possível tanto criptografar o disco inteiro quanto pastas ou arquivos individuais.

Também é possível compactar pastas e arquivos individuais, economizando espaço em disco. No Windows 95/98 era possível compactar partições usando o drvspace, mas só era possível compactar partições inteiras, o que normalmente acaba não sendo um bom negócio, pois diminuía bastante a velocidade do micro e aumentava a possibilidade de perda de dados. Naturalmente, a compactação também é diferente da feita por programas como o Winzip, já que os arquivos e pastas continuam acessíveis exatamente da mesma forma, com o sistema fazendo a compactação e descompactação do arquivo de forma transparente.

Com a possibilidade de compactar pastas individuais, você pode comprimir apenas as pastas contendo um grande volume de arquivos que suportam um bom nível de compressão, deixando de lado pastas com fotos, músicas e arquivos de vídeo, arquivos que já estão comprimidos. Para compactar uma pasta, acesse o menu “Propriedades”. Na seção “avançadas”, marque a opção “Compactar arquivos para economizar espaço”.

A compactação de arquivos exige uma carga adicional de processamento, já que o sistema tem o trabalho de descompactar os arquivos antes de acessá-los. Antigamente, usar compactação reduzia muito o desempenho do sistema, já que os processadores eram mais lentos. Num micro atual, a redução é muito menos significativa e, em muitos casos, o uso da compactação pode até mesmo melhorar o desempenho, já que arquivos compactados ocupam menos espaço e, conseqüentemente, são lidos mais rapidamente pela cabeça de leitura.

Assim como no sistema FAT, no NTFS são incluídas várias estruturas lógicas no HD. Apesar da idéia ser basicamente a mesma, estas estruturas são bem diferentes no NTFS.

Em primeiro lugar, temos a **MFT** (Master File Table), que substitui a FAT, armazenando as localizações de todos os arquivos e diretórios, incluindo os arquivos referentes ao próprio sistema de arquivos. Mas, a forma como este mapeamento é feito difere um pouco do sistema FAT.

Cada entrada de arquivo ou diretório no MFT possui 2 KB, nos quais são armazenados o nome do arquivo e seus atributos. Em geral, sobra uma pequena área, usada para armazenar dados, com cerca de 1500 bytes (pode ser maior ou menor, dependendo do espaço ocupado pelo nome e pelos atributos do arquivo) que é usada para guardar o início do arquivo.

Caso o arquivo seja muito pequeno, ele poderá ser armazenado diretamente na entrada no MFT. Caso contrário, serão armazenados apenas os números dos clusters ocupados pelo arquivo.

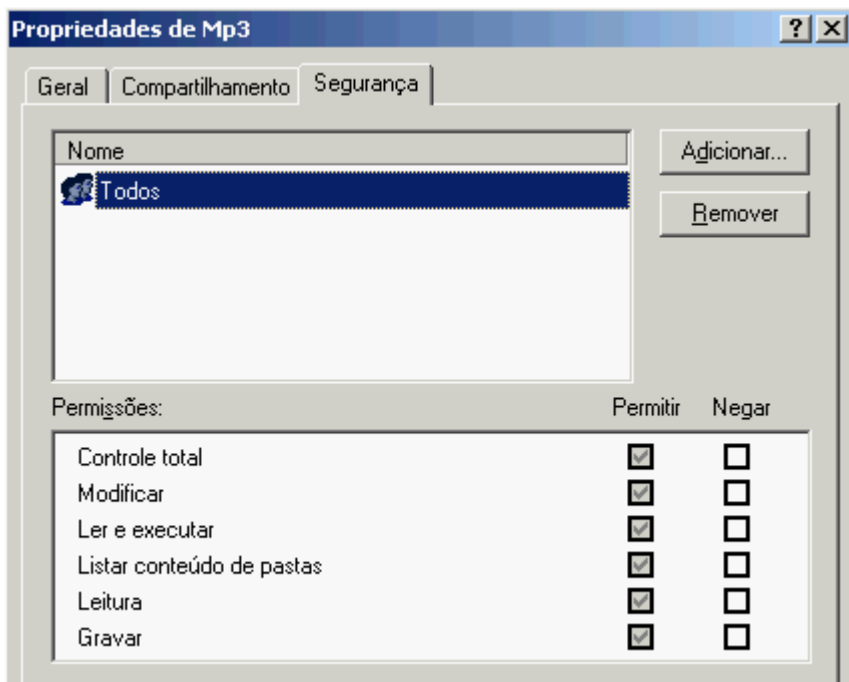
Em alguns casos, não é possível armazenar nem mesmo os atributos do arquivo no MFT. Nesse caso, os atributos serão gravados em clusters vagos do HD, e a MFT conterá apenas entradas que apontem para eles. Pode parecer estranho que um arquivo possa ter mais de 2 KB só de atributos, mas no NTFS os atributos do arquivo vão muito além dos atributos de arquivo, diretório, oculto, etc. que existem no sistema FAT.

Os atributos do arquivo incluem seu nome, versão, nome MS-DOS (o nome simplificado com 8 caracteres e extensão), mas, principalmente incluem as permissões do arquivo, quais usuários do sistema poderão acessá-lo ou não e, ainda, um espaço reservado para auditoria, que permite armazenar informações sobre quais operações, relacionadas ao arquivo, devem ser gravadas para que seja possível realizar uma auditoria posteriormente, caso necessário.

Em seguida, temos a questão das **permissões de acesso**, uma parte importante da configuração de um servidor, ou de qualquer máquina que vá ser utilizada por diversos usuários.

Para configurar as permissões de acesso, abra a guia “Segurança” dentro das propriedades do arquivo. As configurações valem tanto para acessos locais quanto acessos através da rede. O Windows aplica as permissões de acesso de acordo com o usuário logado na máquina.

Por default, todos têm acesso total à pasta. Você verá no campo “nome” o grupo “todos” e todas as permissões marcadas como “permitir”. O grupo “todos” significa todos os usuários do sistema. Se você quiser um melhor controle, pode deletar o grupo “todos” e adicionar um a um os usuários que terão acesso à pasta.



Depois de fazer isso, clique sobre o usuário para configurar as suas permissões, que aparecem na parte de baixo da janela. Você pode até mesmo configurar as permissões de modo que nem você mesmo possa acessar a pasta :). Nesse caso, você receberá uma mensagem de acesso negado até voltar e reconfigurar as permissões.

A configuração das permissões pode ser a parte mais importante da implantação de uma rede baseada no Windows 2000, XP ou mesmo do antigo NT, ao mesmo tempo em que pode ser de longe a mais trabalhosa, dependendo do número de usuários e restrições que tiver de configurar.

É possível, também, estabelecer quotas de disco, que limitam a quantidade de espaço que determinados usuários podem utilizar. Este recurso é muito utilizado em servidores web e em servidores de arquivo, mas pode ser útil também em situações mais corriqueiras, como quando você precisa limitar quanto espaço seu irmão menor pode usar, evitando que ele entupa o HD de downloads. 😊

O NTFS inclui também recursos que reduzem de forma significativa a fragmentação do sistema de arquivos. Quando existe um volume considerável de espaço em disco, o sistema reserva até 15% do espaço da partição para armazenar o MFT e as entradas referentes aos atributos dos arquivos, de forma que todas as informações possam ser gravadas em setores contínuos. Os próprios arquivos são salvos de forma inteligente, com o sistema dando preferência a áreas onde ele pode ser gravado sequencialmente, sem fragmentação.

Apesar disso, com o passar dos meses, é normal que um certo nível de fragmentação ocorra, de forma que um desfragmentador vem a calhar. Tanto o Windows 2000 e XP quanto o Vista incluem um desfragmentador capaz de lidar com partições NTFS. Apesar de demorado, vale a pena usá-lo de vez em quando.

Com relação ao desempenho, existe uma certa polêmica, já que, por ser mais complexo, o NTFS é realmente mais lento que o sistema FAT em micros muito antigos, ou quando são manipuladas pequenas quantidades de arquivos. Esse é um dos motivos dele ser utilizado apenas em micros PCs e não em câmeras e celulares, por exemplo, onde o processamento necessário seria proibitivo.

Apesar disso, o NTFS é capaz de lidar eficientemente com volumes muito maiores de arquivos, de forma que a balança pende para o outro lado em cenários mais complexos, como na bagunça de arquivos, e-mails, fotos, músicas e arquivos temporários que é o HD de um PC atual.

Por exemplo, o NTFS utiliza o conceito de balanced trees (árvores balanceadas ou B+ trees), onde as informações relacionadas a cada diretório são gravadas próximas umas das outras, ao invés de ficarem em uma tabela central, como no sistema FAT. A cabeça de leitura precisa percorrer uma distância maior para acessar as informações

relacionadas a cada diretório, mas, em compensação, perde menos tempo lendo informações referentes aos arquivos dentro deste diretório, o que acaba compensando a perda inicial e até revertendo em ganho, que se torna mais e mais expressivo conforme cresce o volume de arquivos e pastas armazenados.

Concluindo, temos o **LFS** (Log File Service), que é o principal responsável pela tolerância a falhas do sistema NTFS. Tolerância a falhas, neste caso, significa não perder dados ou estruturas do sistema de arquivos quando o sistema travar, ou houver qualquer outro imprevisto, ou que, pelo menos, o estrago seja o menor possível.

Para isso, o sistema mantém um log com todas as alterações feitas no sistema de arquivos. Ao gravar um arquivo qualquer, por exemplo, será primeiro gravada uma entrada no log, com os detalhes sobre a operação, qual arquivo está sendo gravado, em que parte do disco, etc. ao terminar a gravação é gravada uma outra entrada, um OK confirmando que tudo deu certo. Caso o sistema seja desligado incorretamente durante a gravação, é possível verificar no próximo boot o que estava sendo feito e fazer as correções necessárias. Periodicamente, o sistema verifica todas as entradas do Log e caso esteja tudo em ordem, deleta o antigo log, para evitar que o arquivo ocupe muito espaço em disco.

O EXT3 é atualmente o sistema de arquivos mais utilizado no mundo Linux. Usado por padrão pela grande maioria das distribuições.

Tudo começou com o sistema EXT (Extended File System), introduzido em 1992. Nos estágios primários de desenvolvimento, o Linux utilizava um sistema de arquivos bem mais antigo, o MinixFS (o Minix é um sistema Unix, que Linux Torvalds usou como base nos estágios primários do desenvolvimento do Linux). Entretanto, o MinixFS possuía pesadas limitações, mesmo para a época. Os endereços dos blocos de dados tinham apenas 16 bits, o que permitia criar partições de no máximo 64 MB. Além disso, o sistema não permitia nomes de arquivos com mais de 14 caracteres.

Não é de se estranhar que, em pouco tempo o Linux ganhou seu sistema de arquivos próprio, o “Extended File System”, ou simplesmente EXT, que ficou pronto em abril de 92 a tempo de ser incluído no Kernel 0.96c.

Nesta primeira encarnação, o EXT permitia a criação de partições de até 2 GB 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 à fragmentação de arquivos quanto o sistema FAT. Além disso, logo começaram a surgir HDs com mais de 2 GB, de forma que em 1993 surgiu a primeira grande atualização, na forma do EXT2.

O EXT2 trouxe suporte a partições de até 32 TB, manteve o suporte a nomes de arquivos com até 255 caracteres, além de diversos outros recursos.

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 scandisk do Windows, que verifica todos os blocos do sistema de arquivos, procurando por inconsistências entre as estruturas e descrições e os dados efetivamente armazenados.

O teste do fsck demora bastante (bem mais que o scandisk) e o tempo cresce proporcionalmente de acordo com o tamanho da partição. Em um HD atual, o teste pode, literalmente, demorar horas.

Este problema foi corrigido com o EXT3, que foi introduzido em 1999. A principal característica do EXT3 é o uso do recurso de journaling, onde o sistema de arquivos mantém um journal (diário) das alterações realizadas, um recurso similar ao LFS usado no NTFS.

Este “diário” armazena uma lista das alterações realizadas, permitindo que o sistema de arquivos seja reparado de forma muito rápida após o desligamento incorreto. O fsck continua sendo usado, mas agora ele joga de acordo com as novas regras, realizando o teste longo apenas quando realmente necessário.

O EXT3 possui três modos de operação:

No modo **ordered** (o default), o journal é atualizado no final de cada operação. Isto faz com que exista uma pequena perda de desempenho, já que a cabeça de leitura do HD precisa realizar duas operações de gravação, uma no arquivo que foi alterada e outra no journal (que também é um arquivo, embora especialmente formatado) ao invés de apenas uma.

No modo **writeback** o journal armazena apenas informações referentes à estrutura do sistema de arquivos (metadata) e não em relação aos arquivos propriamente ditos, e é gravado de forma mais ocasional, aproveitando os momentos de inatividade. Este modo é o mais rápido, mas em compensação oferece uma segurança muito menor contra perda e corrompimento de arquivos causados pelos desligamentos incorretos.

Finalmente, temos o modo **journal**, que é o mais seguro, porém mais lento. Nele, o journal armazena não apenas informações sobre as alterações, mas também uma cópia de segurança de todos os arquivos modificados, que ainda não foram gravados no disco. A cada alteração, o sistema grava uma cópia do arquivo (no journal), atualiza as informações referentes à estrutura do sistema de arquivos, grava o arquivo e atualiza novamente o journal, marcando a operação como concluída. Como disse, isso garante uma segurança muito grande contra perda de dados, mas em compensação reduz o desempenho drasticamente. Justamente por causa disso, este é o modo menos usado.

Para usar o modo writeback ou o modo journal, você deve adicionar a opção “data=writeback”, ou “data=journal” nas opções referentes à partição, dentro do arquivo “/etc/fstab”.

Desta forma, ao invés de usar “/dev/hda5 /mnt/hda5 ext3 defaults 0 2”, por exemplo, você usaria “/dev/hda5 /mnt/hda5 ext3 data=writeback 0 2”

O EXT3 (assim como o EXT2) utiliza endereços de 32 bits e blocos (análogos aos clusters usados no sistema FAT) de até 8 KB. 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:

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

Uma observação é que, em versões antigas do Kernel, o limite para o tamanho máximo de arquivos no EXT2 já foi de 2 GB e em seguida de 16 GB, mas ambas as limitações caíram a partir do Kernel 2.6, chegando à tabela atual.

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 2 KB) ou “`mkfs.ext3 -b 4096 /dev/hda1`” (para blocos de 4 KB).

Assim como no caso do NTFS, usar clusters maiores resulta em mais espaço desperdiçado (sobretudo ao guardar uma grande quantidade de arquivos pequenos) mas, além do aumento no tamanho máximo dos arquivos e partições, resulta em um pequeno ganho de desempenho, já que reduz o processamento e o número de atualizações na estrutura do sistema de arquivos ao alterar os dados gravados.

Embora o limite de 32 TB para as partições EXT3 não seja um problema hoje em dia, ele tende a se tornar um obstáculo conforme os HDs cresçam 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 2 TB 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.