**Material para ajudar no entendimento sobre os tópicos:**

**GERÊNCIA DE ARQUIVOS**

**Agradecimento e Observação:**

*Agradeço ao professor Doutor Maziero, por disponibilizar o livro como material de estudo na UTFPR. E ainda, como observação* ***Todo material*** *exposto nos capítulos a seguir é referente aos capítulos de gerência de arquivos e gerência de entrada/saídaretirado do livro do Dr. Carlos A. Maziero. Fazendo apenas a inserção de figuras do livro de Andrew Tanenbaum.*

**Gerência de arquivos**

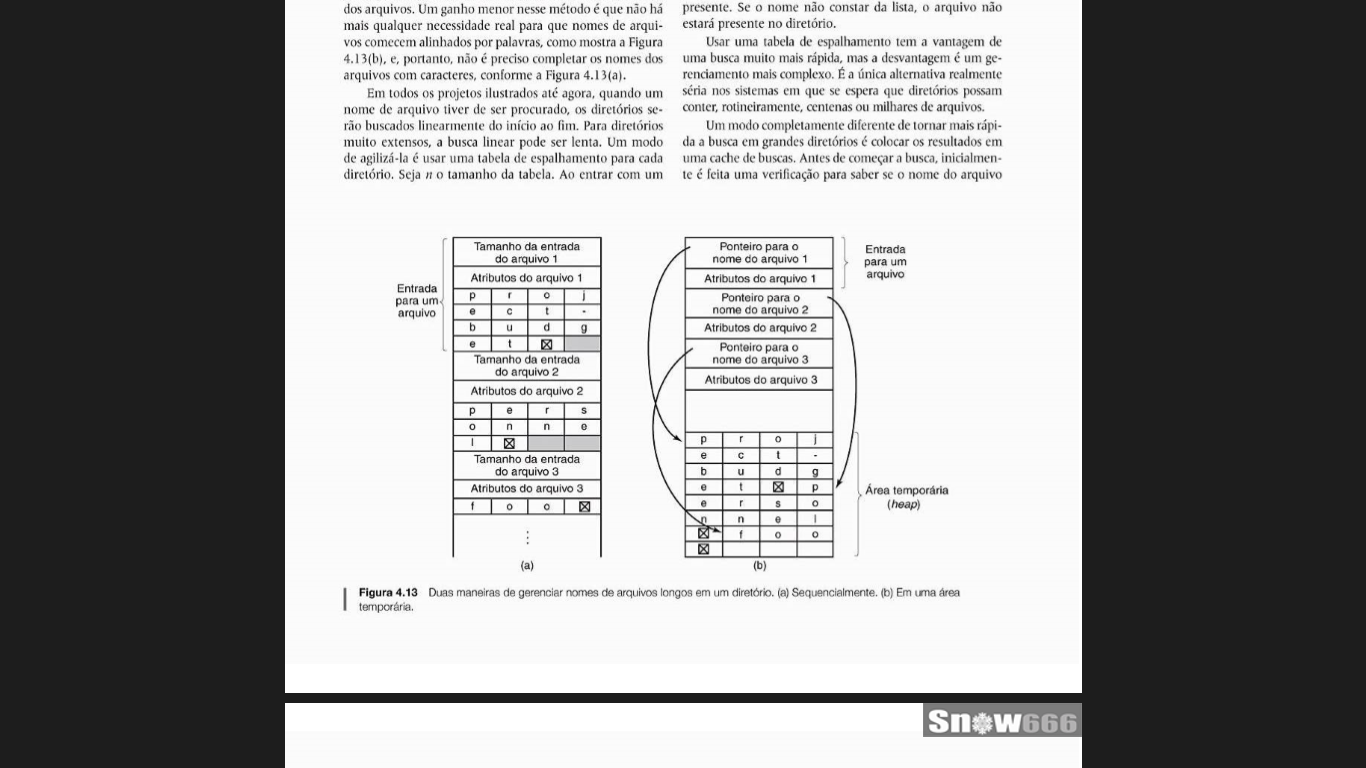
Um sistema operacional tem por ﬁnalidade permitir que os usuários do computador executem aplicações, como editores de texto, jogos, reprodutores de áudio e vídeo, etc. Essas aplicações processam informações como textos, músicas e ﬁlmes, armazenados sob a forma de arquivos em um disco rígido ou outro meio. Este módulo apresenta anoção de arquivo, suas principais características e formas de acesso, a organização de arquivos em diretórios e as técnicas usadas para criar e gerenciar arquivos nos dispositivos de armazenamento.

1.Arquivos

Desde os primórdios da computação, percebeu-se a necessidade de armazenar informações para uso posterior, como programas e dados. Hoje, parte importante do uso de um computador consiste em recuperar e apresentar informações previamente armazenadas, como documentos, fotograﬁas, músicas e vídeos. O próprio sistema operacional também precisa manter informações armazenadas para uso posterior, como programas, bibliotecas e conﬁgurações. Geralmente essas informações devem ser armazenadas em um dispositivo não-volátil, que preserve seu conteúdo mesmo quando o computador estiver desligado. Para simpliﬁcar o armazenamento e busca de informações, surgiu o conceito de arquivo, que será discutido a seguir.

1.1 O conceito de arquivo:

Um arquivo é basicamente um conjunto de dados armazenados em um dispositivo físico não-volátil, com um nome ou outra referência que permita sua localização posterior. Do ponto de vista do usuário e das aplicações, o arquivo é a unidade básica de armazenamento de informação em um dispositivo não-volátil, pois para eles não há forma mais simples de armazenamento persistente de dados. Arquivos são extremamente versáteis em conteúdo e capacidade: podem conter desde um texto ASCII com alguns bytes até sequências de vídeo com dezenas de gigabytes, ou mesmo mais. Como um dispositivo de armazenamento pode conter milhões de arquivos, estes são organizados em estruturas hierárquicas denominadas diretórios. A organização física e lógica dos arquivos e diretórios dentro de um dispositivo é denominada sistema de arquivos. Um sistema de arquivos pode ser visto como uma imensa estrutura de dados armazenada de forma persistente em um dispositivo físico. Existe um grande número de sistemas de arquivos, dentre os quais podem ser citados o NTFS (nos sistemas Windows), Ext2/Ext3/Ext4 (Linux), HPFS (MacOS), FFS (Solaris) e FAT (usado em pendrives USB, máquinas fotográﬁcas digitais e leitores MP3).



Duas maneiras de gerenciar nomes de arquivos longos.

A – Sequencialmente B – Em uma área temporariamente

Autor: TANENBAUM, 2009

1.2 Atributos

Conforme apresentado, um arquivo é uma unidade de armazenamento de informações que podem ser dados, código executável, etc. Cada arquivo é caracterizado por um conjunto de atributos, que podem variar de acordo com o sistema de arquivos utilizado. Os atributos mais usuais são:

Nome: string de caracteres que identiﬁca o arquivo para o usuário, como “foto1.jpg”, “relatório.pdf”, “hello.c”, etc.;

Tipo: indicação do formato dos dados contidos no arquivo, como áudio, vídeo, imagem, texto, etc. Muitos sistemas operacionais usam parte do nome do arquivo para identiﬁcar o tipo de seu conteúdo, na forma de uma extensão: “.doc”, “.jpg”, “.mp3”, etc.;

Tamanho: indicação do tamanho do conteúdo do arquivo, em bytes ou registros;

Datas: para ﬁns de gerência, é importante manter as datas mais importantes relacionadas ao arquivo, como suas datas de criação, de último acesso e de última modiﬁcação do conteúdo;

Proprietário: em sistemas multiusuários, cada arquivo tem um proprietário, que deve estar corretamente identiﬁcado;

Permissões de acesso: indicam que usuários têm acesso àquele arquivo e que formas de acesso são permitidas (leitura, escrita, remoção, etc.);

Localização: indicação do dispositivo físico onde o arquivo se encontra e da posição do arquivo dentro do mesmo;

Outros atributos: vários outros atributos podem ser associados a um arquivo. Cada sistema de arquivos normalmente deﬁne seus próprios atributos especíﬁcos, além dos atributos usuais.

Nem sempre os atributos oferecidos por um sistema de arquivos são suﬁcientes para exprimir todas as informações a respeito de um arquivo. Nesse caso, a “solução” encontrada pelos usuários é usar o nome do arquivo para exprimir a informação desejada. Por exemplo, em muitos sistemas a parte ﬁnal do nome do arquivo (sua extensão) é usada para identiﬁcar o formato de seu conteúdo. Outrasituaçãofrequenteé usar parte do nome do arquivo para identiﬁcar diferentes versões do mesmo conteúdo1: relat-v1.txt, relat-v2.txt, etc.

1.3 Operações

As aplicações e o sistema operacional usam arquivos para armazenar e recuperar dados. O uso dos arquivos é feito através de um conjunto de operações, geralmente implementadas sob a forma de chamadas de sistema e funções de bibliotecas. As operações básicas envolvendo arquivos são:

Criar: a criação de um novo arquivo implica em alocar espaço para ele no dispositivo de armazenamento e deﬁnir seus atributos (nome, localização, proprietário, permissões de acesso, etc.);

Abrir: antes que uma aplicação possa ler ou escrever dados em um arquivo, ela deve solicitar ao sistema operacional a “abertura” desse arquivo. O sistema irá então veriﬁcar se o arquivo existe, veriﬁcar se as permissões associadas ao arquivo permitem aquele acesso, localizar seu conteúdo no dispositivo de armazenamento e criar uma referência para ele na memória da aplicação;

Ler: permite transferir dados presentes no arquivo para uma área de memória da aplicação;

Escrever: permite transferir dados na memória da aplicação para o arquivo no dispositivo físico; os novos dados podem ser adicionados no ﬁnal do arquivo ou sobrescrever dados já existentes;

Mudar atributos: para modiﬁcar outras características do arquivo, como nome, localização, proprietário, permissões, etc.

Fechar: ao concluir o uso do arquivo, a aplicação deve informar ao sistema operacional que o mesmo não é mais necessário, a ﬁm de liberar as estruturas de gerência do arquivo na memória do núcleo;

Remover: para eliminar o arquivo do dispositivo, descartando seus dados e liberando o espaço ocupado por ele.

Além dessas operações básicas, outras operações podem ser deﬁnidas, como truncar, copiar, mover ou renomear arquivos. Todavia, essas operações geralmente podem ser construídas usando as operações básicas.

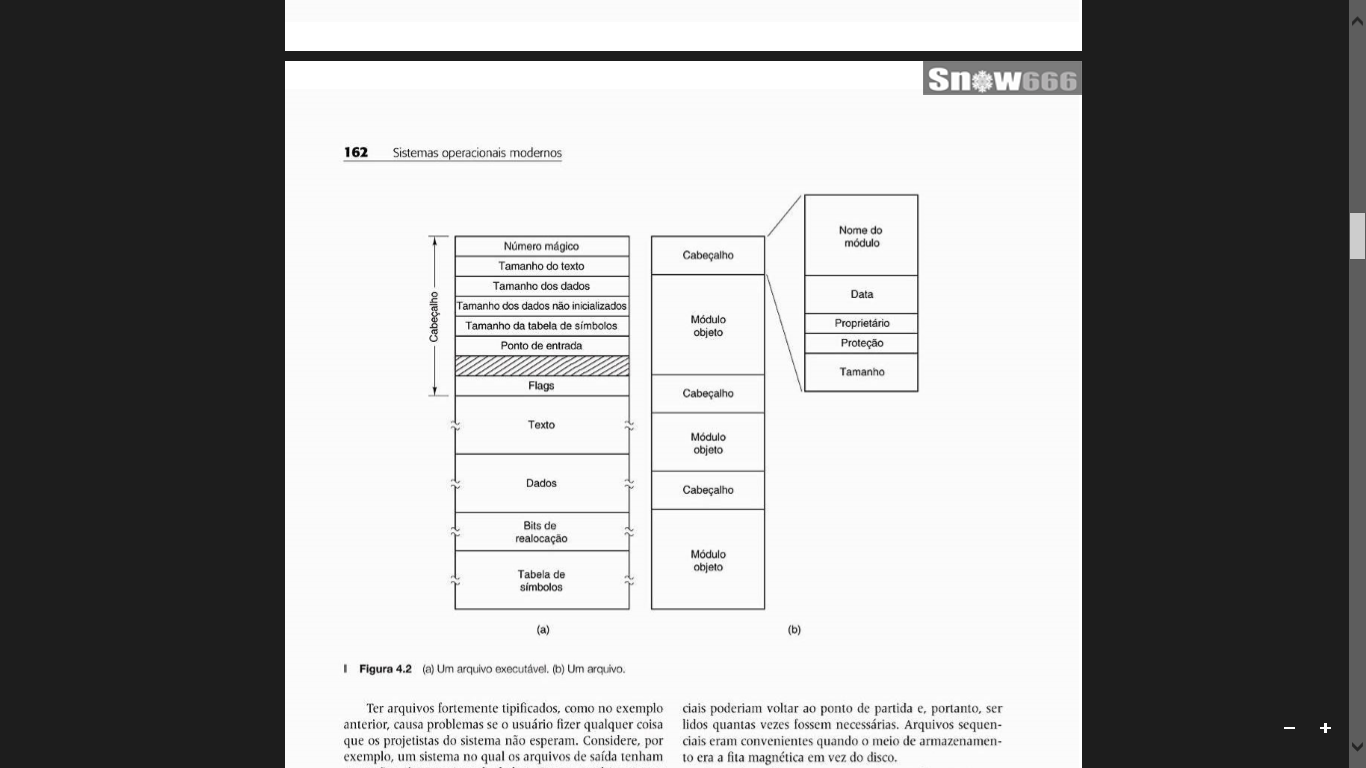
1.4 Formatos

Em sua forma mais simples, um arquivo contém basicamente uma sequência de bytes, que pode estar estruturada de diversas formas para representar diferentes tipos de informação. O formato ou estrutura interna de um arquivo pode ser deﬁnido – e reconhecido – pelo núcleo do sistema operacional ou somente pelas aplicações. O núcleo do sistema geralmente reconhece apenas alguns poucos formatos de arquivos, como binários executáveis e bibliotecas. Os demais formatos de arquivos são vistos pelo núcleo apenas como sequências de bytes sem um signiﬁcadoespecíﬁco, cabendo às aplicações interpretá-los. Os arquivos de dados convencionais são estruturados pelas aplicações para armazenar os mais diversos tipos de informações, como imagens, sons e documentos. Uma aplicação pode deﬁnir um formato próprio de armazenamento ou seguir formatos padronizados.

Arquivos de registros: Alguns núcleos de sistemas operacionais oferecem arquivos com estruturas internas que vão além da simples sequência de bytes.

Arquivos de texto: Um tipo de arquivo de uso muito frequente é o arquivo de texto puro (ou plaintext). Esse tipo de arquivo é muito usado para armazenar informações textuais simples, como códigos-fonte de programas, arquivos de conﬁguração, páginas HTML, dados em XML, etc. Um arquivo de texto é formado por linhas de caracteres ASCII de tamanho variável, separadas por caracteres de controle. Nos sistemas UNIX, as linhas são separadas por um caractere New Line (ASCII 10 ou “\n”). Já nos sistemas DOS/Windows, as linhas de um arquivo de texto são separadas por dois caracteres: o caractere CarriageReturn (ASCII 13 ou “\r”) seguido do caractere New Line.

Arquivos executáveis: Um arquivo executável é dividido internamente em várias seções, para conter código, tabelas de símbolos (variáveis e funções), listas de dependências (bibliotecas necessárias) e outras informações de conﬁguração. A organização interna de um arquivo executável ou biblioteca depende do sistema operacional para o qual foi deﬁnido.



A – Um arquivo Executável B – Um arquivo

Fonte: TANENBAUM, 2009

2 Uso de arquivos

Arquivos são usados por processos para ler e escrever dados de forma não-volátil. Para usar arquivos, um processo tem à sua disposição uma interface de acesso, que depende da linguagem utilizada e do sistema operacional subjacente. Essa interface normalmente é composta por uma representação lógica de cada arquivo usado pelo processo (uma referência ao arquivo) e por um conjunto de funções (ou métodos) para realizar operações sobre esses arquivos. Através dessa interface, os processos podem localizar arquivos no disco, ler e modiﬁcar seu conteúdo, entre outras operações. Na sequência desta seção serão discutidos aspectos relativos ao uso de arquivos, como a abertura do arquivo, as formas de acesso aos seus dados, o controle de acesso e problemas associados ao compartilhamento de arquivos entre vários processos.

2.1 Abertura de um arquivo

Para poder ler ou escrever dados em um arquivo, cada aplicação precisa antes “abrí-lo”. A abertura de um arquivo consiste basicamente em preparar as estruturas de memória necessárias para acessar os dados do arquivo em questão. Assim, para abrir um arquivo, o núcleo do sistema operacional deve realizar as seguintes operações:

1. Localizar o arquivo no dispositivo físico, usando seu nome e caminho de acesso;

2. Veriﬁcarseaaplicaçãotempermissãoparausaraquelearquivodaformadesejada (leitura e/ou escrita);

3. Criar uma estrutura na memória do núcleo para representar o arquivo aberto;

4. Inserir uma referência a essa estrutura na lista de arquivos abertos mantida pelo sistema, para ﬁns de gerência;

5. Devolver à aplicação uma referência a essa estrutura, para ser usada nos acessos subsequentes ao arquivo recém-aberto.

Concluída a abertura do arquivo, o processo solicitante recebe do núcleo uma referência para o arquivo recém-aberto, que deve ser informada pelo processo em suas operações subsequentes envolvendo aquele arquivo. Assim que o processo tiver terminado de usar um arquivo, ele deve solicitar ao núcleo o fechamento do arquivo, que implica em concluir as operações de escrita eventualmente pendentes e remover da memória do núcleo as estruturas de gerência criadas durante sua abertura. Normalmente, os arquivos abertos são automaticamente fechados quando do encerramento do processo, mas pode ser necessário fechá-los antes disso, caso seja um processo com vida longa, como um daemon servidor de páginas Web, ou que abra muitos arquivos, como um compilador. As referências a arquivos abertos usadas pelas aplicações dependem da linguagem de programação utilizada para construí-las.

2.2 Controle de acesso

Como arquivos são entidades que sobrevivem à existência do processo que as criou, é importante deﬁnir claramente o proprietário de cada arquivo e que operações ele e outros usuários do sistema podem efetuar sobre o mesmo. A forma mais usual de controle de acesso a arquivos consiste em associar os seguintes atributos a cada arquivo e diretório do sistema de arquivos:

Proprietário: identiﬁca o usuário dono do arquivo, geralmente aquele que o criou; muitos sistemas permitem deﬁnir também um grupo proprietário do arquivo, ou seja, um grupo de usuários com acesso diferenciado sobre o mesmo;

Permissões de acesso: deﬁne que operações cada usuário do sistema pode efetuar sobre o arquivo.

Existem muitas formas de se deﬁnir permissões de acesso a recursos em um sistema computacional; no caso de arquivos, a mais difundida emprega listas de controle de acesso (ACL - Access ControlLists) associadas a cada arquivo. Uma lista de controle de acesso é basicamente uma lista indicando que usuários estão autorizados a acessar o arquivo, e como cada um pode acessá-lo.

3 Sistemas de arquivos

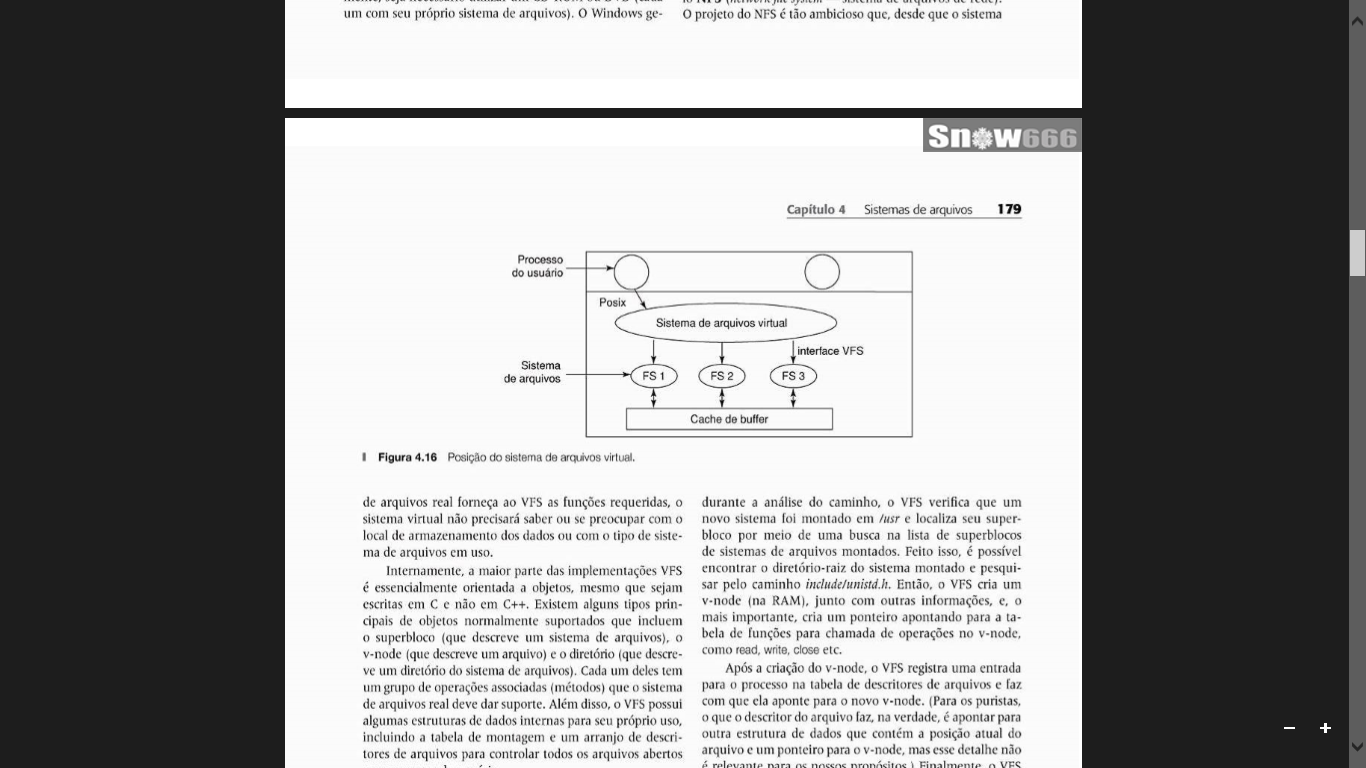
Vários problemas importantes devem ser resolvidos na construção de um sistema de arquivos, que vão do acesso de baixo nível aos dispositivos físicos de armazenamento à implementação da interface de acesso a arquivos para os programadores. Na implementação de um sistema de arquivos, considera-se que cada arquivo possui dados e meta-dados. Os dados de um arquivo são o seu conteúdo em si (uma música, uma fotograﬁa, um documento ou uma planilha); por outro lado, os meta-dados do arquivo são seus atributos (nome, datas, permissões de acesso, etc.) e todas as informações de controle necessárias para localizar e manter seu conteúdo no disco.

3.1 Arquitetura geral

Os principais elementos que constituem a gerência de arquivos estão organizados em camadas. No nível mais baixo dessa arquitetura estão os dispositivos de armazenamento, como discos rígidos ou bancos de memória ﬂash, responsáveis pelo armazenamento dos dados e meta-dados dos arquivos. Esses dispositivos são acessados através de controladores, que são circuitos eletrônicos dedicados ao controle e interface dos dispositivos. A interface entre controladores e dispositivos de armazenamento segue padrões como SATA, ATAPI, SCSI, USB e outros.

Os controladores de dispositivos são conﬁgurados e acessados pelo núcleo do sistema operacional através de drivers de dispositivos, que são componentes de software capazes de interagir com os controladores. Os drivers usam portas de entrada/saída, interrupções e canais de acesso direto à memória (DMA) para interagir com os controladores e realizar as operações de controle e de entrada/saída de dados. Como cada controlador deﬁne sua própria interface, também possui um driver especíﬁco. Os drivers ocultam essas interfaces e fornecem às camadas superiores do núcleo uma interface padronizada para acesso aos dispositivos de armazenamento. Desta forma, os detalhes tecnológicos e particularidades de cada dispositivo são isolados, tornando o restante do sistema operacional independente da tecnologia subjacente. Acima dos drivers está a camada de gerência de blocos, que gerencia o ﬂuxo de blocos de dados entre a memória e os dispositivos de armazenamento. É importante lembrar que os discos são dispositivos orientados a blocos, ou seja, as operações de leitura e escrita de dados são sempre feitas com blocos de dados, e nunca com bytes individuais. As funções mais importantes desta camada são efetuar o mapeamento de blocos lógicos nos blocos físicos do dispositivo, oferecer às camadas superiores a abstração de cada dispositivo físico como sendo um imenso vetor de blocos lógicos, independente de sua conﬁguração real, e também efetuar o caching/buﬀering de blocos. A seguir está a camada de alocação de arquivos, que tem como função principal alocar os arquivos sobre os blocos lógicos oferecidos pela gerência de blocos. Cada arquivo é visto como uma sequência de blocos lógicos que deve ser armazenada nos blocos dos dispositivos de forma eﬁciente, robusta e ﬂexível. Acima da alocação de arquivos está o sistema de arquivos virtual (VFS-VirtualFile System), que provê uma interface de acesso a arquivos independente dos dispositivos físicos e das estratégias de alocação de arquivos empregadas pelas camadas inferiores. O sistema de arquivos virtual normalmente gerencia as permissões associadas aos arquivos e as travas de acesso compartilhado, além de construir as abstrações de diretórios e atalhos.

Outra responsabilidade importante desta camada é manter informações sobre cada arquivo aberto pelos processos, como a posição da última operação no arquivo, o modo de abertura usado e o número de processos que estão usando o arquivo. A interface de acesso ao sistema de arquivos virtual é oferecida aos processos através de um conjunto de chamadas de sistema. Finalmente,asbibliotecasdeentrada/saídausamaschamadasdesistemaoferecidas pelo sistema operacional para construir funções padronizadas de acesso a arquivos para cada linguagem de programação.



Posição do sistema de arquivo virtual.

Autor: TANENBAUM, 2009.

3.2 Blocos físicos e lógicos

Um dos aspectos mais importantes dos sistemas de arquivos é a forma como o conteúdo dos arquivos é disposto dentro do disco rígido ou outro dispositivo de armazenamento secundário. Um disco rígido pode ser visto como um conjunto de blocos de tamanho ﬁxo (geralmentede512bytes). Osblocosdo disco rígido são normalmente denominados blocos físicos. Como esses blocos são pequenos, o número de blocos físicos em um disco rígido pode ser:

Um disco rígido de 250 GBytes contém mais de 500 milhões de blocos físicos! Para simpliﬁcar a gerência dessa quantidade de blocos físicos e melhorar o desempenho das operações de leitura/escrita, os sistemas operacionais costumam trabalhar com blocos lógicos ou clusters, que são grupos de 2n blocos físicos consecutivos. Blocos lógicos com 4K,8K,16K e 32K bytes são frequentemente usados. A maior parte das operações e estruturas de dados deﬁnidas nos discos pelos sistemas operacionais são baseadas em blocos lógicos, que também deﬁnem a unidade mínima de alocação de arquivos e diretórios: cada arquivo ou diretório ocupa um ou mais blocos lógicos para seu armazenamento. O número de blocos físicos em cada bloco lógico de uma partição é deﬁnido pelo sistema operacional ao formatar a partição, em função de vários parâmetros, como o tamanho da partição, o sistema de arquivos usado e o tamanho das páginas de memória RAM. Blocos lógicos muito pequenos implicam em ter mais blocos a gerenciar e menos bytes transferidos em cada operação de leitura/escrita, o que tem impacto negativo sobre o desempenho do sistema. Por outro lado, blocos lógicos muito grandes podem levar à fragmentaçãointerna: um arquivo com 200 bytes armazenado em um sistema de arquivos com blocos lógicos de 32.768 bytes (32K) ocupará um bloco lógico, do qual 32.568 bytes serão desperdiçados, pois ﬁcarão alocados ao arquivo sem serem usados. A fragmentação interna diminui o espaço útil do disco rígido, por isso deve ser evitada. Uma forma de evitá-la é escolher um tamanho de bloco lógico adequado ao tamanho médio dos arquivos a armazenar no disco, ao formatá-lo. Além disso, alguns sistemas de arquivos (como o UFS do Solaris e o ReiserFS do Linux) permitem a alocação de partes de blocos lógicos, através de técnicas denominadas fragmentos de blocos ou alocação de sub-blocos.

**Exercícios**

1 - Quais operações posso acrescentar ao arquivo gerado;

2 - Qual o processo de criar um arquivo (Passo a Passo)?

3 - Quando está com o arquivo em uso, o que ocorre dentro da memória?

4 - Explique o procedimento para criar uma cópia do arquivo, detalhando a partir da memória.

**Referência Bibliográfica:**

MAZIERO, Carlos A. **Sistemas Operacionais: Conceitos e Mecanismos**, 2013. Editora DAINF UTFPR. 1ª Edição

TANENBAUM, Andrew S. **Sistemas Operacionais Modernos**, 2009. Editora Pearson. São Paulo. 3ª Edição