

Sistemas Operacionais

Aula 07 - Gerenciamento de Arquivos

Apresentação

Nas aulas anteriores, vimos como são realizadas as transferências de dados entre a CPU, a memória e os diversos dispositivos de E/S. Nesta aula, você vai entender como os dados permanecem salvos, mesmo depois que um computador é desligado.

Sabemos que os programas, para serem executados, precisam ser carregados na memória principal. Entretanto, às vezes, o espaço de memória principal não é suficiente para atender às necessidades de todos os processos. Além disso, discutimos que toda a informação presente na memória de um computador seria perdida quando o computador fosse desligado. Aí você deve estar se perguntando: como fazer, então, para manter e para acessar os dados após o computador ser desligado?

Assim, para atender a essas necessidades, são utilizados discos e outras mídias digitais externas para armazenamento de dados que devem permanecer salvos, mesmo que o processo seja encerrado e até mesmo quando o computador for desligado. O gerenciamento desse “depósito” de informações será o foco desta nossa aula.

Objetivos

- Entender como funciona o armazenamento persistente de dados no computador.
- Identificar as principais chamadas de sistema para manipulação de arquivos.
- Compreender como funciona a hierarquia de diretórios para organização dos arquivos.

Armazenando e Salvando Dados



Vídeo 01 - Armazenamento de Dados

Para entender como os dados são armazenados e salvos, vamos voltar para a analogia que fizemos de um computador com uma mesa de estudos, usada nas aulas anteriores. Nesse caso, sabemos que todo o material de estudo – nossos livros, nossas listas de exercícios – devem permanecer guardados. No caso da mesa representada na Figura 1, esse material está guardado nas gavetas. Fazendo uma analogia com o computador, podemos dizer que essas gavetas correspondem aos discos rígidos de um computador.

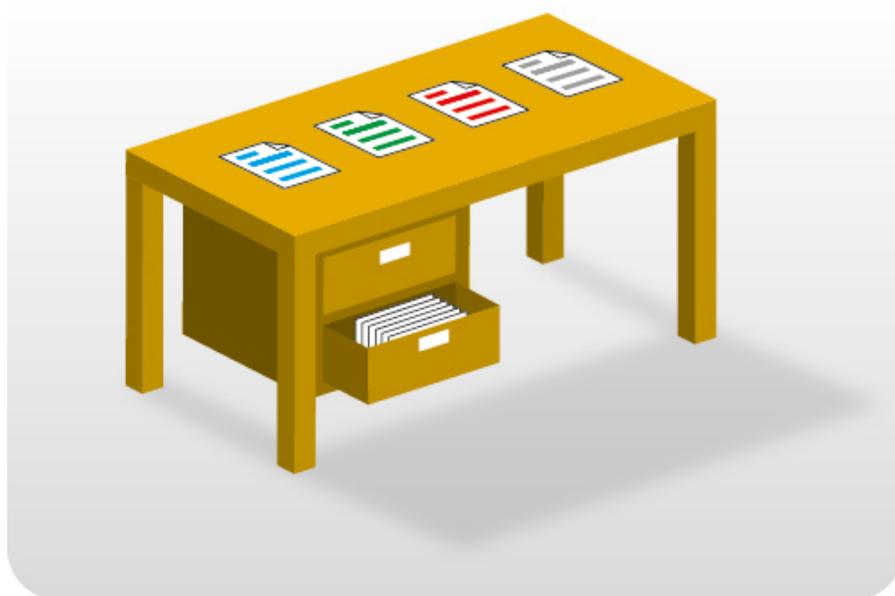


Figura 1 - Mesa de estudos com gavetas para armazenamento

Fonte: Autoria Própria.

As gavetas servem para guardarmos todo o material de estudo, incluindo tudo que foi feito durante nossos estudos anteriores. Além disso, as gavetas servem para que possamos ter a mesa livre para realizar tarefas de disciplinas diferentes sem misturar livros e apostilas de cada

uma. E mais ainda, se nossa mesa de estudos for suficientemente grande para ser compartilhada com outras pessoas, essas pessoas podem compartilhar o mesmo material guardado nas gavetas.

Quando usamos as gavetas para armazenar nosso material de estudo, precisamos nos lembrar de manter sempre todo o material organizado para que todas as pessoas que se utilizam da mesa (inclusive nós mesmos) possam ter acesso fácil à informação armazenada nas gavetas.

Podemos dizer que o sistema operacional gerencia o armazenamento de dados de forma parecida com o exemplo das gavetas na mesa de estudos. Basicamente, é preciso que ele procure atender a três requisitos:

1. Poder armazenar uma grande quantidade de dados.
2. A informação deve permanecer guardada, mesmo com o processo finalizado.
3. Devem existir mecanismos de acesso da informação por mais de um processo.

Para facilitar a organização dos dados nos discos rígidos, o sistema operacional armazena os dados em unidades chamadas “arquivos” (mesmo conceito dos documentos de papel utilizados nas empresas para guardar informações de suas várias atividades). Assim, para permitir o gerenciamento desses dados, o sistema operacional oferece aos processos a capacidade de criar, ler, mover e alterar arquivos.

Arquivo



Vídeo 02 - Estrutura de Arquivos

A parte do sistema operacional responsável pelo gerenciamento (criação, remoção, alteração) dos arquivos é conhecida como sistema de arquivos. É ele o responsável por identificar cada arquivo com um nome único. Por exemplo, todos os seus livros possuem o título na lateral do livro, de modo a facilitar sua identificação e, assim, caso você precise retirar um livro de uma estante ou gaveta, basta procurar pelo nome do livro sem ter que abri-lo (Figura 2).



Figura 2 - Parte lateral dos livros como referência para uma busca rápida

Da forma análoga à manipulação de livros numa estante, em um sistema de gerenciamento de arquivos de um sistema operacional, o nome do arquivo armazenado no disco rígido também é uma informação importante e necessária para sua localização no disco rígido do computador.

Cada sistema de arquivos possui regras específicas para nomear um arquivo; dentre essas regras podemos citar, por exemplo: a quantidade máxima de caracteres que pode ser usada para formar os nomes dos arquivos, e os tipos de caracteres que podem ser usados (geralmente, você não pode usar os seguintes caracteres em um nome de arquivo: \ / . ? : * " | > <).

Tanto no Linux quanto no Windows, os arquivos possuem extensões. As extensões funcionam apenas como uma indicação do tipo de programa que pode manipular aquele arquivo, sendo separada do nome por um ponto '.'. Por exemplo, a extensão '.jpg' indica que o arquivo casa.jpg se

trata de uma imagem e um programa de edição de imagens (GIMP, por exemplo) pode abri-lo, de modo que ele possa ser exibido na tela de seu computador.

Atividade 01

1. Liste dez extensões de arquivos conhecidos e relacione com os programas que podem abrir arquivos com essas extensões.

Estrutura dos Arquivos



Vídeo 03 - Organização e Interpretação de Arquivos

A informação contida em nossos livros e apostilas é representada por um conjunto de letras e palavras impressas nas folhas. Mas, você seria capaz de imaginar como são estruturados os dados nos arquivos do computador? Sabemos que os dados no computador são codificados como sequências de *bits* (unidades binárias com valores 1 ou 0) e que o tráfego entre os diversos componentes é baseado em conjuntos de *bits* cujos tamanhos e valores são definidos como potência de 2 (4, 8, 16, 32, 64 etc.).

Então, para os sistemas operacionais, os arquivos são apenas um conjunto de *bytes* (unidades compostas de 8 *bits*); é de responsabilidade de cada programa converter esses *bytes* em letras, números, sinais de pontuação, ou seja, alguma representação que nós, humanos, possamos entender. Fazendo um paralelo com nossa estante de livros, seria como se a estante de livros só pudesse armazenar livros, apostilas e documentos em uma língua comum, formada apenas por sequências de zeros e uns (*bits*). Caberia, então, a quem fosse ler esse material, interpretar o que significa a sequência de *bits* presentes em cada livro (por exemplo, a sequência de 8 *bits* "00011000" pode representar a letra "A").

Apesar dos arquivos serem, basicamente, um conjunto de *bytes* (8 *bits*), eles podem ser de tipos diferentes, de acordo com a sua finalidade. Podemos ter arquivos que servem apenas para armazenar os dados (arquivos com extensão '.dat'), arquivos que armazenam imagens (arquivos com extensão '.bmp'), ou até mesmo arquivos de sistema onde mantemos informações sobre outros arquivos, os quais chamamos de diretórios - ou pastas - e que servem como uma forma de organização das informações.

Existe também um tipo especial de arquivo que serve para termos acesso direto a diferentes dispositivos. Por exemplo, para enviarmos comandos para uma impressora no sistema operacional Linux, basta abrir o arquivo correspondente a esse dispositivo e escrever a informação que será impressa. O sistema de arquivos ficará, então, responsável por enviar as requisições para o driver da impressora para solicitar a impressão do documento que já foi escrito no arquivo correspondente a esse dispositivo.

Interpretando os *bits* Armazenados em um Arquivo

Como os arquivos são um conjunto de bytes, é necessário criar uma forma de interpretar esses bytes em caracteres. Isso é feito relacionando cada valor possível do byte ($8 \text{ bits} = 2^8 = 256$) a um possível caractere. Uma possível forma de interpretar esses bytes é através da tabela ASCII (American Standard Code International Interchange, em português, Código Padrão Americano para o Intercâmbio de Informação), que relaciona um conjunto de 8 bits a um caractere presente no teclado do computador. A Figura 3 mostra uma visão parcial dos elementos definidos na tabela ASCII.

Binário	Glifo	Binário	Glifo	Binário	Glifo
0010 0000		0100 0000	@	0110 0000	`
0010 0001	!	0100 0001	A	0110 0001	a
0010 0010	=	0100 0010	B	0110 0010	b
0010 0011	#	0100 0011	C	0110 0011	c
0010 0100	\$	0100 0100	D	0110 0100	d
0010 0101	%	0100 0101	E	0110 0101	e
0010 0110	&	0100 0110	F	0110 0110	f
0010 0111	'	0100 0111	G	0110 0111	g
0010 1000	(0100 1000	H	0110 1000	h
0010 1001)	0100 1001	I	0110 1001	i
0010 1010	*	0100 1010	J	0110 1010	j
0010 1011	+	0100 1011	K	0110 1011	k
0010 1100	,	0100 1100	L	0110 1100	l
0010 1101	-	0100 1101	M	0110 1101	m
0010 1110	.	0100 1110	N	0110 1110	n
0010 1111	/	0100 1111	O	0110 1111	o

Figura 3 - Visão parcial da tabela ASCII

Sabemos que os programas, para serem executados, precisam ser carregados na memória principal. Entretanto, às vezes, o espaço de memória principal não é suficiente para atender às necessidades de todos os processos. Além disso, discutimos que toda a informação presente na memória de um computador seria perdida quando o computador fosse desligado. Aí você deve estar se perguntando: como fazer, então, para manter e para acessar os dados após o computador ser desligado?

Assim, para atender a essas necessidades, são utilizados discos e outras mídias digitais externas para armazenamento de dados que devem permanecer salvos, mesmo que o processo seja encerrado e até mesmo quando o computador for desligado. O gerenciamento desse “depósito” de informações será o foco desta nossa aula.

Atividade 02

1. Codifique seu nome completo em tabela ASCII. Para isso, você deve procurar na tabela ASCII a representação binária de cada letra do seu nome e dos espaços entre eles.

Acessando os Arquivos



Vídeo 04 - Acesso a Arquivos

Há algum tempo, as informações do computador não eram armazenadas em discos rígidos; elas eram armazenadas em fitas magnéticas (similares às fitas cassetes usadas para escutar música da década de 1980, e nas fitas VHS usadas para armazenar filmes). A forma de acessar a informação nesse tipo de mídia era sequencial, ou seja, para acessar informações no meio da fita, tínhamos que rebobinar a fita até o ponto desejado.

Atualmente, como utilizamos discos, os dados podem ser acessados fora da ordem (assim como num CD, podemos ir direto para a música 10 sem precisar passar por todas as outras; em um HD podemos acessar os arquivos armazenados em qualquer ordem). Portanto, o acesso à informação se tornou mais rápido. Os sistemas operacionais modernos se utilizam apenas dessa forma de acesso e os arquivos que podem ter seus dados acessados em qualquer ordem são chamados de arquivos de acesso aleatório.

Uma comparação interessante que podemos fazer com relação a essa evolução é comparar os antigos Papiros – rolos de papel utilizados na antiguidade para registrar os documentos – com os livros que podemos selecionar um determinado capítulo saltando as suas páginas (Figura 4).



Figura 4 - Papiro (acesso sequencial) x Livro (acesso aleatório)
Fonte: Autoria Própria.

Atributos de um Arquivo

Na capa de um livro ou documento, normalmente, temos informações básicas sobre ele. Dados como autor, edição, título, editora são disponibilizados na capa com o objetivo de auxiliar nas nossas consultas.

O sistema operacional também guarda informações sobre os arquivos que vão além do seu nome e dos dados armazenados: data e hora de criação, do último acesso, e da última alteração, bem como o tamanho do arquivo e códigos de segurança para restrição de acesso são exemplos de atributos que são armazenados pelo sistema operacional para cada arquivo criado (Figura 5).

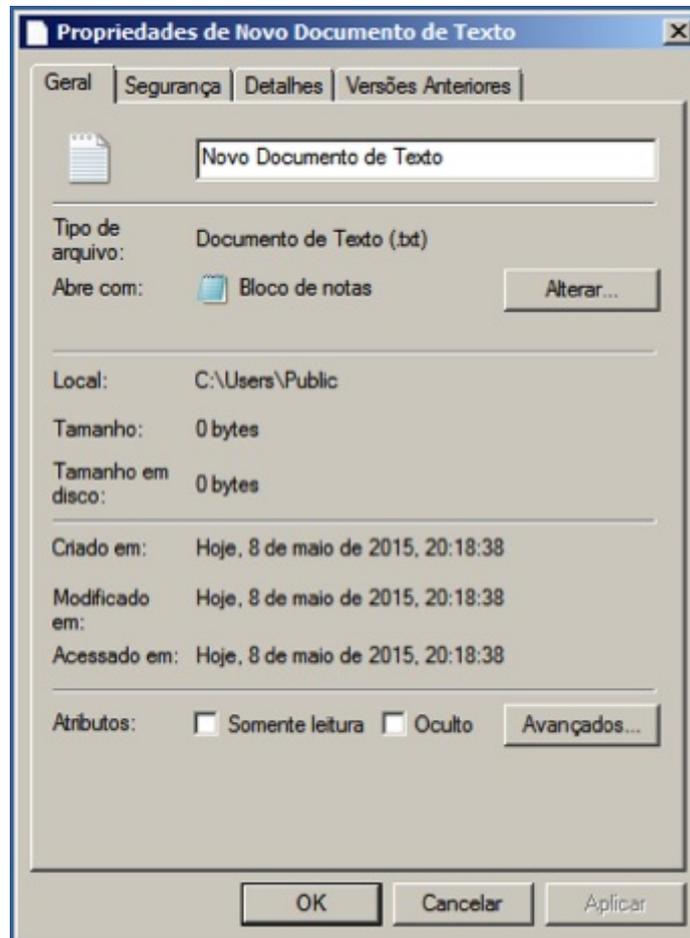


Figura 5 - Informações de alguns atributos do arquivo no Windows

Operações com Arquivos



Vídeo 05 - Operações para Manipulação de Arquivos

Os sistemas de arquivos possuem conjuntos de operações diferentes, mas eles disponibilizam, basicamente, os seguintes recursos de acesso e manipulação de arquivos:

Operação	Descrição
Criar	O arquivo é criado sem dados e alguns atributos são preenchidos
Apagar	Para excluir um arquivo, uma solicitação explícita deve ser enviada ao sistema operacional.
Abrir	Os atributos e localização do arquivo no disco têm que ser enviado à memória para que o processo possa acessar os dados
Fechar	Como arquivos abertos ocupam boa parte da memória reservada aos processos, é essencial uma chamada de sistemas que libere o espaço de memória de um arquivo que não será mais utilizado.
Ler	Os dados são lidos de acordo com uma posição predefinida e de uma quantidade de <i>bytes</i> .
Escrever	Os dados serão copiados da memória para uma posição predefinida; caso seja no fim do arquivo, seu tamanho será aumentado, caso não, os dados serão substituídos.
Acrescentar	Variação da escrita que a restringe apenas para o final do arquivo, sem sobrescrever dados já existentes.
Posicionar	Define uma posição específica do arquivo para leitura e escrita. Basicamente, informa a partir de que byte o dado será lido ou escrito no arquivo.
Ler atributos	Para obter as informações presentes nos atributos dos arquivos, essa chamada é bastante útil para diversas situações, por exemplo, quando queremos ordenar nossos arquivos pelo seu tamanho.
Alterar atributos	Alguns arquivos podem ser alterados. Um exemplo simples é alterar as permissões de acesso ao arquivo.

Quadro 1 - Operações de manipulação de arquivos

Sabemos que os programas, para serem executados, precisam ser carregados na memória principal. Entretanto, às vezes, o espaço de memória principal não é suficiente para atender às necessidades de todos os processos. Além disso, discutimos que toda a informação presente na memória de um computador seria perdida quando o computador fosse desligado. Aí você deve estar se perguntando: como fazer, então, para manter e para acessar os dados após o computador ser desligado?

Assim, para atender a essas necessidades, são utilizados discos e outras mídias digitais externas para armazenamento de dados que devem permanecer salvos, mesmo que o processo seja encerrado e até mesmo quando o computador for desligado. O gerenciamento desse “depósito” de informações será o foco desta nossa aula.

Atividade 03

1. Pesquise como as operações de manipulação de arquivos podem ser realizadas em um sistema operacional real, por exemplo, no Linux ou Windows. Um site que contém comandos básicos do Linux é o seguinte:

<<http://www.infowester.com/comandoslinux.php>>

Nessa página, você pode encontrar informações interessantes para essa tarefa.

Diretórios



Vídeo 06 - Níveis de Diretórios

Mesmo tendo uma estante, e algumas gavetas para guardar nosso material de estudo, isso não garante que conseguiremos encontrar facilmente nosso material quando desejarmos. Para isso, precisamos de certa organização. Diferentes tipos de livros, apostilas e documentos devem ser organizados de tal forma que eu possa encontrá-los facilmente quando eu precisar deles.

Por exemplo, seria interessante que todos os questionários estivessem organizados em pastas que indicassem a qual disciplina pertencem, assim, caso você queira revisar as questões para uma prova, você saberia exatamente onde encontrar. Veja na Figura 6 um exemplo de organização de documentos em pastas.



Figura 6 - Organizando documentos em pastas

Fonte: <<http://www.kaiserkraft.pt/estantes/estantes-para-pastas-de-arquivo/estante-de-aparafusar-para-pastas-de-arquivo-galvanizada/p/M1172/>> Acesso em: 7 out.

2011.

Para organizar os arquivos, o sistema operacional utiliza uma técnica parecida: os diretórios. Esses funcionam como pastas, armazenando informações sobre um conjunto de arquivos e facilitando a localização dos mesmos.

Localizando Diretórios e Arquivos



Vídeo 07 - Diretórios

Como os arquivos nos sistemas operacionais estão organizados em diretórios, é fundamental que exista uma forma de localizá-los. Para isso, não basta termos o seu nome, precisamos também do nome do diretório onde ele se encontra; e mais ainda, caso esse diretório esteja também inserido em outros, devemos saber a sequência de diretórios até chegar ao arquivo. A Figura 7 mostra um exemplo de árvore de diretórios e arquivos.

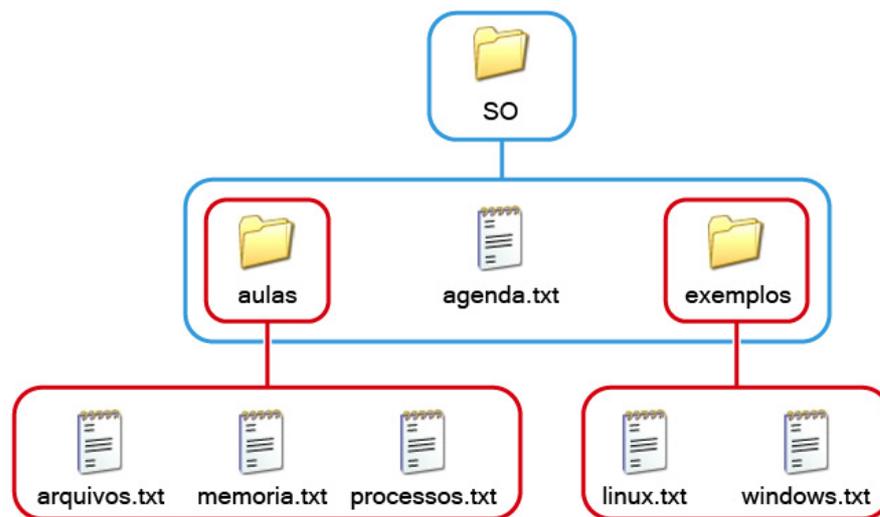


Figura 7 - Árvore de diretórios e arquivos

Fonte: Autoria Própria.

Dependendo do sistema de arquivos adotado pelo sistema operacional, a estrutura da árvore de diretórios e até sua representação podem mudar. Por exemplo, no Windows, o sistema de arquivos utilizado atualmente é o NTFS (*New Technology File System*, em português, Sistema de Arquivos de Nova Tecnologia) que diferencia as unidades de armazenamento (discos, drivers de CD, pendrives, etc.) por letras (C:\, D:\, E:\, etc.) e, a partir de cada unidade especificada, temos uma árvore de diretórios associada.

Já o sistema de arquivos EXT3 (Third Extended file system, em português, Terceira Versão do Sistema EXT), presente em muitas distribuições do sistema operacional Linux, não representa as unidades de armazenamento através de letras, apenas existe um diretório padrão conhecido como diretório-raiz representado pela barra '/'. Caso seja necessário acessar uma unidade diferente, deve ser feita uma associação com um diretório, procedimento esse conhecido como montagem de unidade. A Figura 8 apresenta a árvore de diretórios no Windows e no Linux.

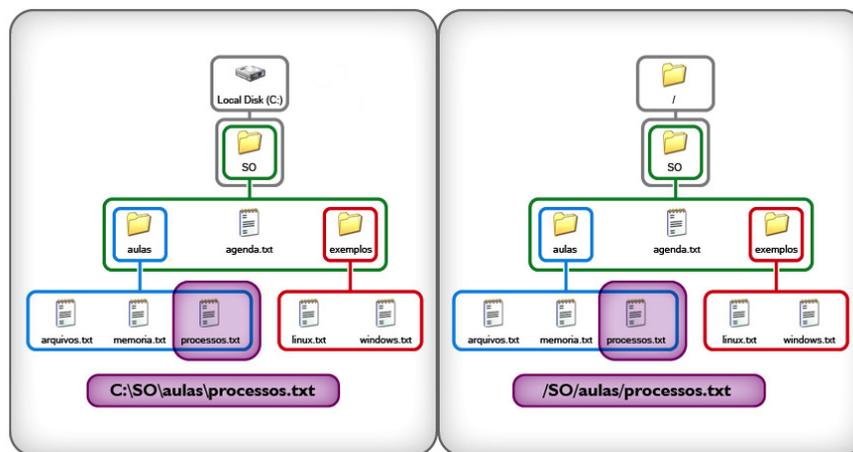


Figura 8 - Formas diferentes de acesso para um arquivo nos diferentes sistemas
Fonte: Autoria Própria.

Sistemas de Arquivos



Vídeo 08 - Sistemas de Arquivos

Diferentes sistemas operacionais usam diferentes sistemas de arquivos. Dependendo da capacidade dos discos e do volume de arquivos e acessos, as ações de manipulação de arquivos tornam-se mais complexas, exigindo constantes evoluções nos sistemas de arquivos. Existem diversos sistemas de arquivos, que vão desde sistemas simples, como o FAT16 até sistemas como o NTFS, EXT3 e ReiserFS, que incorporam recursos muito mais avançados.

No ambiente 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 atual. Para o Linux (e outros sistemas Unix), temos uma variedade muito grande de sistemas de arquivos diferentes, que incluem o EXT2, EXT3, ReiserFS, XFS, JFS e muitos outros que podem parecer estranhos para usuários do Windows, mas não para os usuários do Linux.

Apesar das diferenças existentes nos sistemas de arquivos, alguns conceitos básicos sempre estarão presentes. Por exemplo, para determinar o caminho para um dado arquivo, basta informar todos os diretórios que são necessários percorrer, desde a base da árvore até o arquivo; então, como vimos na Figura 8, o caminho do arquivo processos.txt é:

- **no Windows:** C:\SO\aulas\processos.txt
- **no Linux:** /SO/aulas/processos.txt

Esse nome corresponde ao percurso completo, desde a raiz do sistema de arquivos, até o diretório onde se encontra o arquivo. Esse percurso é denominado **caminho absoluto**.

Além do conceito de caminho absoluto, existe também o **caminho relativo**. Quando estamos em um dado diretório (também chamado de diretório atual, ou diretório de trabalho), podemos acessar um arquivo como se o diretório atual fosse o diretório-raiz. No exemplo anterior, se o diretório de trabalho fosse C:\SO\aulas\, no Windows, ou /SO/aulas, no Linux, poderíamos acessar o mesmo arquivo processos.txt de duas formas:

Caminho	Windows	Linux
Diretório Atual	C:\SO\	/SO/
Absoluto	C:\SO\aulas\processos.txt	/SO/aulas/processos.txt
Relativo	aulas\processos.txt	aulas/processos.txt

Quadro 2 - Formas de como acessar o arquivo processos.txt

Em sistemas operacionais diferentes, o símbolo que separa os caminhos pode ser diferente. Por exemplo, no caso do sistema operacional Linux, o símbolo utilizado é o '/' e no caso do Windows é o '\'. Além do símbolo que delimita os nomes de diretórios e arquivos, os símbolos ponto '.' e ponto-ponto '..' são utilizados para identificar, respectivamente, o diretório atual e o diretório-pai, assim, podemos percorrer pela árvore de diretórios com maior flexibilidade.

Na prática, os caminhos relativos são utilizados para agilizar o acesso aos arquivos. Por exemplo, se no LINUX utilizarmos o comando cp (cópia) para copiar o arquivo memoria.txt para o diretório 'exemplos', de acordo com a Figura 8, poderíamos digitar os caminhos da origem e do destino da seguinte maneira, se o nosso diretório de trabalho fosse /SO/exemplos/:

- cp /SO/aulas/memoria.txt /SO/exemplos/
- cp ../aulas/memoria.txt /SO/exemplos/
- cp ../aulas/memoria.txt

Operações com Diretórios

Um diretório é um tipo de arquivo especial, pois pode armazenar outros arquivos dentro dele. Existe um conjunto específico de operações para a manipulação de diretórios, são elas:

Operação	Descrição
Criar	Um diretório vazio é criado.
Apagar	Um diretório é removido. Apenas diretórios vazios podem ser excluídos. Ou seja, se desejar remover um diretório, você deve antes apagar os arquivos e diretórios que ele contém.
Abrir	Os diretórios precisam ser abertos para serem lidas as suas informações, como a lista de seus arquivos e outros diretórios
Fechar	Da mesma forma que um arquivo, o diretório aberto deve ser fechado para liberar a memória alocada.
Renomear	Os diretórios também podem ser renomeados normalmente.

Quadro 3 - Algumas operações para a manipulação de diretórios

Associando Arquivos

O sistema de arquivos de alguns sistemas operacionais pode associar um mesmo arquivo a vários diretórios através de *links*. Ou seja, um arquivo pode ser acessado a partir de diferentes diretórios, como se ele estivesse localizado neles. Em alguns casos, quando desejamos apagar o correspondente arquivo, só conseguimos, de fato, quando forem apagadas todas as ligações para o mesmo.

No sistema Linux, podemos "*linkar*" arquivos através do comando *ln*, que funciona de maneira similar aos atalhos presentes na área de trabalho do Windows. Os atalhos do Windows funcionam apenas como *links* simbólicos, ou seja, caso o arquivo seja apagado do seu diretório original, ele será removido e o atalho não servirá mais.

Atividade 04

1. Faça uma pesquisa identificando os comandos reais utilizados em sistemas operacionais que realizam as mesmas funções das chamadas do sistema de arquivos apresentados para manipulação de diretórios. Você pode usar como fonte de pesquisa o mesmo site indicado para a atividade anterior, ou qualquer outro site que liste os comandos básicos para manipulação de diretórios em sistemas operacionais como o Linux ou Windows.

Leitura Complementar

O *link* <<http://www.asciitable.com/>> possui a descrição completa da tabela ASCII. Lá você verifica como cada um dos caracteres do teclado é relacionado a um código binário específico, servindo de apoio na interação homem-computador.

Arquivos em ASCII têm a vantagem de poderem ser lidos por qualquer programa de edição de texto, como um simples bloco de notas, e diferentes aplicações podem manipular os dados desses arquivos, facilitando a comunicação entre processos.

Para os arquivos em código binário, não adianta visualizá-los ou imprimi-los, pois veremos apenas um monte de símbolos incompreensíveis. Esses tipos de arquivos podem estar numa codificação específica baseada nos programas que os manipulam ou podem ser arquivos executáveis. Nesse caso, eles possuem um formato específico para que o sistema operacional os reconheça como programas a serem executados.

Resumo

Ao longo da aula você viu como o sistema operacional torna mais simples a tarefa de leitura e escrita dos dados no computador. Vimos que os dados do computador são armazenados na forma de arquivos e que

existe um tipo de arquivo especial que pode armazenar outros arquivos dentro deles, os chamados diretórios. Você também aprendeu um pouco mais sobre os possíveis comandos para manipular arquivos e diretórios em diferentes sistemas operacionais.

Autoavaliação

1. Descreva as operações para manipulação de arquivos e diretórios presentes nos sistema de arquivos dos sistemas operacionais que você conhece.
2. Indique algumas diferenças básicas entre o sistema de arquivos utilizado pelo Windows e o utilizado pelo Linux.
3. O que determina qual será o caminho relativo de um arquivo? Mostre um exemplo, apresentando uma árvore de diretórios e os caminhos absoluto e relativo para um dos arquivos.

Referências

ALECRIM, Emerson. **Sistema de arquivos NTFS**. Disponível em: <<http://www.infowester.com/ntfs.php>>. Acesso em: 7 out. 2011.

SILVA, Roberto Rodrigues da. **Sistemas de arquivos suportado pelo Linux**. Disponível em: <<http://www.vivaolinux.com.br/artigo/Linux-Sistema-de-arquivos/>>. Acesso em: 7 out. 2011.

TANEMBAUM, Andrew S.; WOODHULL, Albert S. **Sistemas operacionais: projeto e implementação**. Trad. Edson Furmankiewicz. 2. ed. Porto Alegre: Bookman, 2000.

TERMOS técnicos sobre sistema de arquivos. Disponível em: <<http://www.guiadohardware.net/termos/sistema-de-arquivos>>. Acesso em: 7 out. 2011.