



LABORATÓRIO DE SISTEMAS OPERACIONAIS

PROF^a. M.Sc. JULIANA HOFFMANN QUINONEZ BENACCHIO

Sistema Operacional

Conteúdo retirado do livro



Sistemas Operacionais

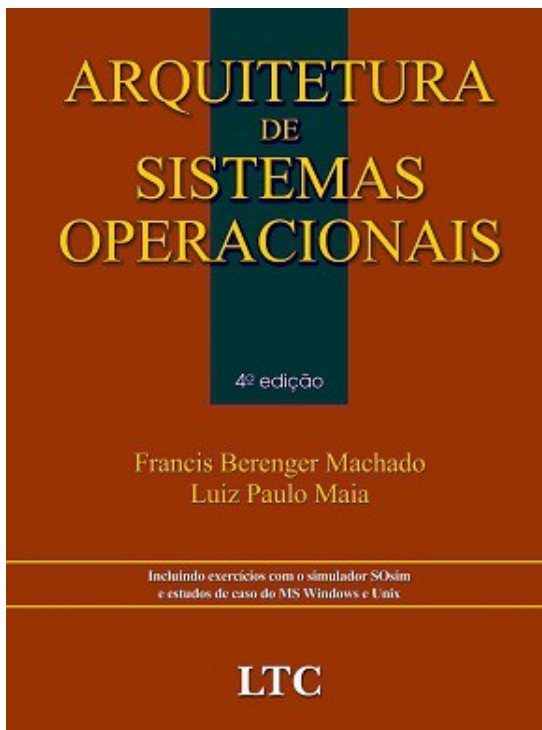
Marcos Aurélio Pchek Laureano

Diogo Roberto Olsen

Editora LT

Sistema Operacional

Conteúdo retirado do livro



Arquitetura de Sistemas Operacionais
Francis Berenger Machado
Luiz Paulo Maia
4^a. edição
Editora LTC



- Responsável pelo acesso e integridade dos dados residentes na memória secundária
- Ao gerenciá-los, fornece serviços de:
 - Abertura e leitura de um arquivo;
 - Gravação e fechamento de um arquivo;
 - Leitura ou modificação dos metadados de arquivos (nome, tamanho, propriedade e etc.)

- Um sistema de arquivos é a maneira pela qual os dados são organizados, armazenados e controlados em qualquer mídia além de ser a forma como o sistema operacional representa essas informações.
- Ele é criado durante a formatação de uma partição.

Sistema de Arquivos

- É uma estrutura de armazenamento e manipulação de dados
- Todo Sistema Operacional possui sistemas de arquivos
- Sistemas de arquivos UNIX-Like são hierárquicos

- Principais sistemas de arquivos
 - MS-Windows
 - FAT – *File Allocation Table*
 - NTFS – *Network File System*
 - UNIX- Like
 - EXT – *Extended File System*
 - ReiserFS
 - MINIX

- **FAT12** – Usado em disquetes no DOS. Não possui suporte a permissões nem *journaling*
- **FAT16** – Usado no DOS e oferece suporte até discos de 2GB. Não possui suporte a permissões e *journaling*
- **FAT32** – Também usado no DOS e oferece suporte a discos de até 2TB. Não possui suporte a permissões e *journaling*

Sistema de Arquivos

- **NTFS** – Formato nativo de discos de sistemas operacionais Windows XP e superiores. Possui suporte a permissões de acesso e compactação nativa. Suporta até 16 Exabytes
- **MINIX** – nativo do MINIX, primeiro sistema de arquivos do Linux. Substituído em 1992
- **EXT** – Upgrade do MINIX com suporte a discos maiores e maior performance

- **EXT2** – Segunda versão do **EXT**. Usado em partições Linux Nativas para o armazenamento de arquivos. É identificado pelo código 83. Suporta até 8TB e dá suporte a arquivos de até 2 terabytes e nomes de arquivos com 255 caracteres

- **EXT3** – Terceira versão do **EXT**. Este sistema de arquivos possui melhorias em relação ao **EXT2**, como o recurso de *journaling* e suporte a arquivos de até 16GB. Suporta até 16TB de disco. Também é identificado pelo tipo 83 e é compatível com o **EXT2** em estrutura
- **Reiserfs** – criado por Hans Reiser. Tamanho de arquivo 8TB e HD de 16TB também implementa *journaling*

- **Journaling** – um setor do *filesystem* onde é feita a reportagem (*journaling*) de todas as ações feitas no hd antes de se escrever diretamente no *filesystem*. Assim, em casos de sinistros, como um desligamento inadequado ou uma queda de energia, basta que o *filesystem* consulte a seção de *journaling* e restaure tudo o que foi perdido sem a necessidade de uma checagem completa.

- *inode* (ou *i-node*) – estrutura de dados que contém informações sobre um arquivo ou diretório
 - UID e GID
 - Tipo e permissões do arquivo
 - Data e hora de criação
 - Tamanho do arquivo
 - Localização do arquivo

- Quando criamos um sistema de arquivo no Linux, cada dispositivo tem um número finito de *inodes* que será diretamente proporcional ao número de arquivos que este dispositivo poderá acomodar.

Sistema de Arquivos

- O sistema de arquivos Linux é semelhante a uma árvore de cabeça para baixo. No topo da hierarquia do Linux existe um diretório raiz nomeado de root e identificado com o sinal `"/`.
- A estrutura do sistema de arquivo Linux é definida por um padrão de mercado chamado FHS (Filesystem Hierarchy Standard). As distribuições não são obrigadas a seguir este padrão, mas elas entendem a importância da localização dos arquivos e diretórios padronizados.

- Embora todos os arquivos do sistema operacional Linux, estejam conectados em uma árvore de diretórios que começa no diretório raiz, os arquivos por si mesmos residem em dispositivos de armazenamento diferentes como: Discos flexíveis, discos rígidos, cd-roms, etc.

- Os arquivos Linux residentes em um dispositivo de armazenamento qualquer são organizados em um sistema de arquivos.
- A árvore de diretórios do Linux pode ter vários sistemas de arquivos diferentes, localizados em dispositivos de armazenamento diferentes.

- Um sistema de arquivos residente em um dispositivo de armazenamento qualquer tem seus arquivos organizados em sua própria árvore de diretórios, que permanece separada da árvore de diretórios principal, até que seja conectada a ela.

- Por exemplo, um drive de discos flexíveis com arquivos Linux tem sua própria árvore de diretórios e é necessário conectar essa sub-árvore de diretórios à árvore principal residente na partição do disco rígido.
- Até que isto seja feito, não será possível acessar os arquivos presentes no disco rígido.

- Estrutura hierárquica com funções definidas:
 - `/` (raiz - todos os arquivos e diretórios do sistema)
 - `/home` (diretórios dos usuários)
 - `/root` (home do super usuário)
 - `/boot` (arquivos de inicialização)
 - `/bin` (executáveis e comandos mais comuns)
 - `/etc` (executáveis essenciais ao sistema, arquivos de configuração e administração)

Sistema de Arquivos

- `/dev` (referência aos dispositivos)
- `/usr` (programas do usuários, comandos, bibliotecas, aplicativos)
- `/lib` (biblioteca básica do sistema)
- `/tmp` (arquivos temporários)
- `/mnt` (ponto de montagem temporária ex.: CD-ROM, pen- drive, etc.)
- `/proc` (sistema de arquivos do kernel)
- `/sbin` (programas usados pelo root para administração e controle do funcionamento do sistema)

- Diretórios que são obrigatórios estarem localizados na partição “/”:
 - `/bin`
 - `/sbin`
 - `/lib`
 - `/etc`
 - `/dev`
 - `/proc`

- Diretórios recomendados como ponto de montagem (normalmente se recomenda os diretórios que normalmente crescem dinamicamente, para não lotar a partição “/”:
 - `/boot`
 - `/home`
 - `/var`
 - `/tmp`
 - `/usr/local`

- MBR (*Master Boot Record*), ou setor de partições, é um setor de *boot* de 512 bytes o qual é o primeiro setor de um disco particionado.
- Funções:
 - Abrigar a tabela de partições primária;
 - Inicializar sistemas operacionais.

- Um ***Bootloader*** é um programa responsável por gerenciar a inicialização de um sistema.
- Ele permite escolher uma entre diversas opções disponíveis e a carrega para a memória RAM do computador.
- Os *bootloaders* modernos trabalham em diversos estágios distintos.

- ***Bootloaders :***
 - LILO
 - GRUB / GRUB2
 - NTLDR

- **Partições**: São divisões existentes no disco rígido que marcam onde começa onde termina um sistema de arquivos. As partições nos permitem usar mais de um sistema operacional no mesmo computador (como o GNU/Linux, Windows e DOS), ou dividir o disco rígido em uma ou mais partes para ser usado por um único sistema operacional ou até mesmo por diferentes arquiteturas (32 e 64 bits).

Discos e Partições

- A partição do tipo *Linux Native* (Tipo 83) é a usada para armazenar arquivos no GNU/Linux, tanto ext2, ext3, ext4, reiserfs, xfs, etc.
- A partição do tipo *Linux Swap* (Tipo 82) é usada como memória virtual.

Discos e Partições

- Uma partição de disco não interfere em outras partições existentes, por este motivo é possível usar o Windows, Linux e qualquer outro sistema operacional no mesmo disco.
- No Linux, os dispositivos existentes no computador (como discos rígidos, disquetes, monitor, portas de impressora, modem, etc.) são identificados por um arquivo referente a esses dispositivos no diretório `/dev`.

Discos e Partições

- Depois de criada e formatada, a partição será identificada como um dispositivo no diretório `/dev` e deverá ser montada para permitir seu uso no sistema.

Discos e Partições

- A identificação de discos rígidos no Linux é feita da seguinte forma:

```
/dev/sda1
|  |  |
|  |  | _Número que identifica o número da partição no disco rígido.
|  |  |
|  |  | _Letra que identifica o disco rígido (a=primeiro, b=segundo, etc...).
|  |  |
|  |  | _Sigla que identifica o tipo do disco rígido (sd=SATA/SCSI, hd=IDE, xt=xt ).
|
|_Diretório onde são armazenados os dispositivos existentes no sistema.
```

- Em um computador, os dados são guardados na forma de arquivos.
- O arquivo é um agrupamento de bytes que seguem uma regra estrutural, e que contém informações (dados) sobre uma área específica.

- Geralmente o tipo de informação encontrada dentro de um arquivo pode ser prevista observando-se os últimos caracteres do seu nome, após o ponto (por exemplo, txt para arquivos de texto sem formatação, mp3 para músicas).
- Esse conjunto de caracteres é chamado de extensão do arquivo.

- É o modo como o sistema organiza os diferentes arquivos contidos em um disco.
- Utilizando um método relativamente simples, o diretório é implementado como um arquivo estruturado, cujo conteúdo é uma relação de entradas na forma de uma estrutura de dados.
- Cada estrutura contém entradas associadas aos arquivos onde estão informações, como localização física, nome, organização e demais atributos.

- Os tipos de entradas normalmente consideradas nessa relação são arquivos normais, diretórios, atalhos e entradas associadas a arquivos especiais.
- Cada entrada contém ao menos o nome do arquivo (ou do diretório), seu tipo e a localização física dele na partição.

- O Linux utiliza algumas entradas padronizadas.
- A entrada “.” (ponto), que representa o próprio diretório
- E a entrada “..” (ponto-ponto), que representa seu diretório pai (o diretório imediatamente acima dele na hierarquia de diretórios)
- No caso do diretório raiz, ambas as entradas apontam para ele próprio.

Diretórios especiais “. ” e “. . ”

- Toda vez que um diretório é criado, sempre são criadas duas entradas nele.
- Uma entrada com o nome de “. ”, referencia-se ao próprio diretório criado e a outra entrada, com o nome de “. . ”, uma referencia ao diretório anterior, ou diretório pai, na estrutura do sistema de arquivos.

Diretórios especiais “. ” e “. . ”

- Estes dois arquivos podem ser usados para compor qualquer caminho, relativo ou absoluto, dentro dos comandos do Linux, e visam facilitar a digitação de comandos.
- Normalmente, eles não aparecem na relação de arquivos, pois o sistema esconde todos os arquivos que começam com um ponto em seu nome.

Diretórios especiais “. ” e “. . ”

- Para que se listem estas duas entradas, devemos usar a opção **-a** no comando **ls**.

```
#ls -la
```

Caminho de Arquivos

- O sistema de arquivos é apresentado como uma única hierarquia unificada que se inicia no diretório / e prossegue abaixo até um número arbitrário de subdiretórios.
- Sempre que precisamos localizar um arquivo para qualquer operação (ler, gravar, remover, criar, etc.) o sistema operacional deve conhecer em que ponto do sistema de arquivos ele se encontra.

Caminho de Arquivos

- Isto é feito por meio da especificação de um caminho antes do nome do arquivo.
- Esse caminho, chamado de **path**, pode ser indicado de duas maneiras:
 - Absoluto
 - Relativo

- O **caminho absoluto** sempre começa com uma barra /. Esse caminho dá a localização do arquivo desde o diretório-raiz do sistema.
- A procura de um arquivo por meio de um **caminho relativo** começa no próprio diretório atual da sessão.

Caminho de Arquivos

- Considere a árvore de diretórios abaixo.

```
\
|-- bin
|-- boot
|   |-- boot
|   |   --grub
|-- etc
|   |-- X11
|   |-- acpi
|-- home
|   |-- marcos
|       |-- Documentos
|       |-- Downloads
|       |-- Imagens
|-- lib
|-- mnt
```

Caminho de Arquivos

- Suponha que você está posicionado no diretório **Documentos** e deseja acessar o diretório **Downloads**. Utilizando caminho relativo, você usaria o seguinte comando:

```
cd ../Downloads
```

Utilizando o caminho absoluto (ou seja, informar o caminho completo a partir da raiz).

```
cd /home/aluno/Downloads
```

Caminho de Arquivos

- Ambos os comandos fazem a mesma coisa, só que no primeiro caso foi digitado menos palavras.
- Agora considere que você está no diretório **Downloads** e deseje acessar o diretório **etc**.
- Utilizando caminho relativo:
`cd ../../../etc`
- Utilizando caminho absoluto:
`cd /etc`

Sistema de cores dos Arquivos

- Ao serem exibidos pelo comando `ls`, os arquivos podem exibir uma cor diferente apontando o tipo de cada arquivo.

Cor	Tipo
Azul escuro	Diretório
Cinza	Arquivos comuns
Branco	Arquivo texto
Azul ciano	Link
Verde	Arquivos binários
Amarelo	Dispositivos de E/S
Magenta	Arquivo compactado