

II - PARTIÇÃO

1 - DEFINIÇÃO

Discos são divididos em uma ou mais partições. Cada partição funciona como se fosse um disco separado, assim, diferentes sistemas operacionais podem ser instalados num mesmo disco sem que ocorra conflito.

A informação sobre como um disco foi particionado é armazenada no 1º setor (1º setor da 1ª trilha do 1º disco). Este setor é chamado registro de boot principal (MBR - Master Boot Record) do disco. O MBR contém um pequeno programa que lê a tabela de partição, verifica qual partição está ativa (como partição de boot) e lê o primeiro setor de boot daquela partição. O setor de boot também contém outro pequeno programa que lê a primeira parte do sistema operacional armazenado naquela partição e então o inicializa.

O esquema original de particionamento para discos de PC permitiam somente 4 partições. Para superar este problema, foram criadas as partições estendidas (extended partitions), permitindo que uma partição primária (primary partition) fosse dividida em sub-partições. A partição primária então subdividida é partição estendida.

O disco é dividido em 3 partições primárias, a segunda das quais é dividida em 2 partições lógicas. Parte do disco não é particionada. O disco como um todo e cada partição primária possuem um setor de boot.

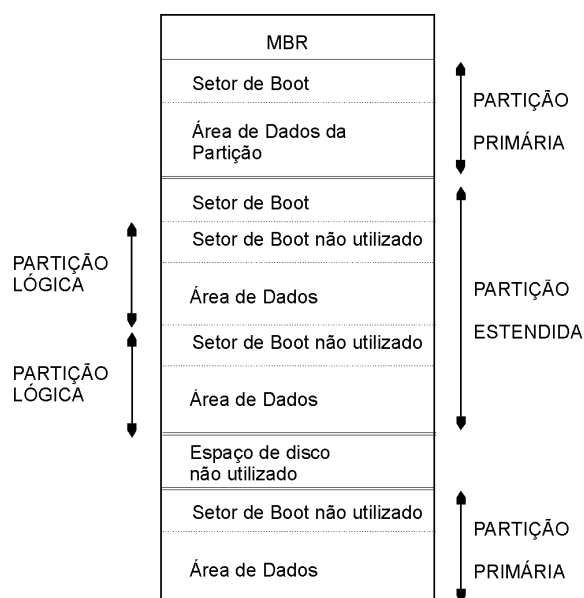


Figura II.1 - Exemplo de Partição de Disco

Existem muitos programas para criar e remover partições. A maioria dos sistemas operacionais possuem o seus próprios programas, geralmente chamados fdisk. Quando estiver usando os discos IDE a partição de boot deve estar completamente dentro dos 1024 cilindros. Isto acontece porque o disco é utilizado pela BIOS durante o boot, e a maioria das versões de BIOS não consegue manipular mais do que 1024 cilindros. Cada partição deve ter um mesmo número de setores. O sistema de arquivos do UNIX utiliza blocos de 1 kbytes de tamanho, ou seja, 2 setores.

Cada partição, incluindo as partições estendidas, possuem o seu próprio arquivo de device. A convenção de nomes para estes arquivos que é um número de partição, é incluído após o nome do disco com a convenção: 1-4 são partições primárias e 5-8 são para as partições lógicas. Por exemplo, /dev/hda (hd - hard drive) refere-se ao primeiro disco como um todo, /dev/hda1 é a primeira partição primária do primeiro disco IDE e /dev/sdb7 é a terceira partição estendida no segundo disco SCSI. Os prefixos mais usados na definição de nomes de discos são:

- fd - floppy disk
- hd - hard drive
- sd - SCSI drive
- st - SCSI tape

Os comandos utilizados para tratar discos e partições não serão cobertos neste curso. Para maiores informações consulte o manual do sistema operacional.

III - SISTEMA DE ARQUIVOS DO UNIX

1 - DEFINIÇÃO

Um sistema de arquivos (filesystem) define os métodos e as estruturas de dados que um sistema operacional usa para manter o caminho de arquivos num disco ou partição, ou seja, é a maneira como os arquivos são organizados no disco.

O sistema operacional UNIX usa um filesystem hierárquico para gerenciar e organizar seus arquivos e diretórios. Um arquivo geralmente é um recipiente para dados e diretório é um recipiente para arquivos e/ou outros diretórios. Um diretório contido dentro de outro diretório é também denominado de subdiretório.

Para trabalhar eficientemente com o sistema UNIX, é necessário compreender como o filesystem é organizado e os diferentes tipos de arquivos que podem ser criados. Existem limitações nos nomes de arquivos e diretórios.

O UNIX suporta vários tipos de sistemas de arquivos. Para descobrir quais são os sistemas de arquivos suportados consulte a árvore de origem do kernel no diretório /usr/src/linux/fs. É possível selecionar quais os tipos de sistemas de arquivos serão suportados no momento de construção do kernel (consulte o manual do sistema operacional). Os principais sistemas de arquivos são:

Tabela III.1 - Sistemas de Arquivos

SISTEMA DE ARQUIVOS	NOME DO TIPO	DESCRIÇÃO
Sistema de Arquivos Second Extended	ext2	O sistema de arquivo mais comum do Linux. Residem fisicamente em dispositivos de armazenamento de massa, usualmente drives de disco rígido
Sistema de Arquivos Extended	ext	Substituído pelo ext2
Sistema de Arquivos Minix	minix	O sistema de arquivos Minix original (raramente usado)
Sistema de Arquivos da Rede (NFS)	NFS	Permite o acesso a arquivos remotos na rede NFS - Network File Services
Sistema de Arquivos MS-DOS	msdos	Acessa os arquivos do MS-DOS
Sistema de Arquivos HPFS	HPFS	O sistema de arquivos do OS/2 HPFS - High-performance File System
Sistema de Arquivos SMB	smbfs	Acessa os arquivos a partir de um servidor Windows for Workgroups ou Windows NT
Sistema de Arquivos ISSO 9660	Iso9660	Usado pela maioria dos CD-ROM (<i>Compact Disk-Read Only Memory</i>)

Cada tipo de filesystem tem seus próprios atributos e limites. Por exemplo o MS-DOS restringe os nomes de arquivos a 8 caracteres mais uma extensão com 3 caracteres e deve ser usado apenas para acessar os disquetes no MS-DOS.

O filesystem Second Extended (ext2) que foi desenvolvido basicamente para o Linux, suporta os nomes de arquivos com 256 caracteres, um tamanho máximo do filesystem de 4 terabytes e muitas outras características.

Para acessar qualquer filesystem no UNIX é necessário montá-lo (com o comando mount) em um certo diretório. Exceto o filesystem raiz (root) no disco de sistema, é possível montar e desmontar todos os filesystems nos discos, para e de uma árvore de sistemas existente. Os filesystems montáveis são chamados de sistemas de arquivos auxiliares (auxiliary filesystems).

2 - ESTRUTURA HIERÁRQUICA

Os arquivos são manipulados internamente pelo sistema operacional UNIX. Entretanto, para a visão do usuário, o sistema de arquivos é uma única estrutura hierárquica, onde pode-se ter várias partições, discos e arquivos associados em forma de um árvore.

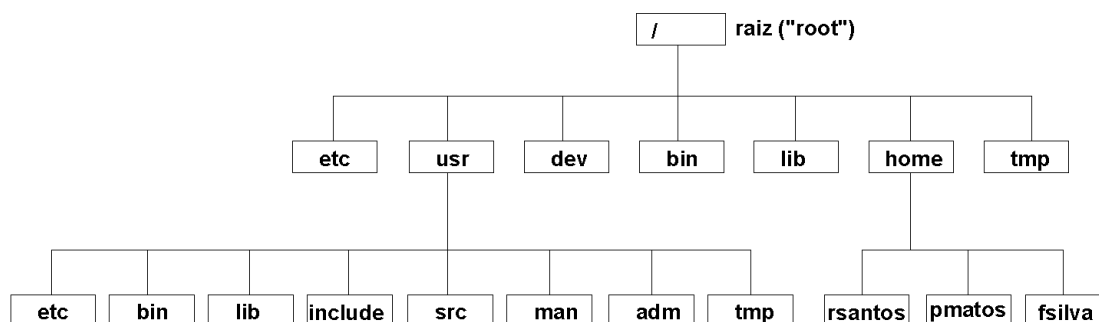


Figura III.1 - Árvore de Diretório no UNIX

No sistema de arquivo do UNIX, arquivos e diretórios são agrupados por função. Usuários podem controlar seus próprios arquivos e diretórios enquanto que o administrador do sistema controla e gerencia o sistema de arquivos e diretórios. Assim fica mais fácil organizar as centenas de arquivos e programas que normalmente são instalados em uma máquina rodando o UNIX. O topo da hierarquia é denominado diretório raiz (root) e é indicado com uma barra (/).

O sistema UNIX também fornece comandos que permitem que você crie novos diretórios quando sua organização precisar de mudanças, como mover ou copiar arquivos de um diretório para outro, o que pode ser feito de forma extremamente fácil. A estrutura de diretório pode variar de acordo com a implementação do sistema, mas a seguir descrevemos aqueles que são encontrados na maioria dos sistemas UNIX:

Tabela III.2 - Diretórios

DIRETÓRIO	FINALIDADE
/etc	Contém os comandos do administrador de sistemas e muitos dos arquivos de configuração do sistema, arquivos solicitados durante o bootup ou shutdown.
/dev	Contém os arquivos especiais (devices files) para todos os tipos de dispositivos de hardware. Estes arquivos funcionam como gateways para o periférico, para desta forma, os dados não serem diretamente armazenados nos periféricos.
/bin	Contém os comandos necessários para a operação básica do UNIX e para a manipulação de arquivos. Geralmente estes comandos são acessíveis por todos usuários.
/lib	Contém as bibliotecas compartilhadas. Estes arquivos geralmente possuem um nome como /lib/libc.so.4.3.2 que neste exemplo indica a versão 4.3.2 da biblioteca padrão da linguagem C.
/home	Contém um subdiretório para cada usuário do sistema UNIX, ou seja, para cada conta. Cada usuário tem controle total sobre o conteúdo de seu próprio diretório, inclusive com a possibilidade de criar outros subdiretórios. O nome do diretório /home pode variar, dependendo de cada sistema operacional, mas os principais são: /users e /usr/homes.
/tmp	Contém os arquivos temporários gerados durante a execução de certas aplicações. Estes arquivos podem ser eliminados sempre que assim o desejar.
/usr/etc	Este diretório é similar ao diretório /usr. Mas quando existir, basicamente irá conter arquivos de configuração para programas instalados em /usr/bin. Em alguns sistemas este diretório é apenas um link para o diretório /usr, que neste caso conterão a mesma coisa.
/usr/bin	Contém os comandos suplementares do UNIX. É possível executar o sistema operacional sem a maioria destes comandos.
/usr/lib	Contém as bibliotecas estáticas (geralmente possuem um nome como libc.a), arquivos de configuração e dados para programas individuais, programas auxiliares que somente são invocados por outros programas. Os arquivos deste diretório somente são acessados por programas residentes em /usr/bin.
/usr/include	Contém os arquivos de cabeçalhos (*.h) para a linguagem de programação C.
/usr/src	Contém os códigos fonte do kernel.
/usr/man	Contém todos os arquivos associados às páginas dos manuais on-line.
/usr/adm	Contém arquivos de log administrativos gerados por vários programas do sistema e demons.
/usr/tmp	Possui a mesma finalidade do diretório /tmp. A diferença reside no fato deste diretório poder ser montados em outras partições (discos)

2.1 - Formato do Sistema de Arquivos

Como visto anteriormente um disco pode ter várias divisões, chamadas de partições. Cada partição pode conter um sistema de arquivos.

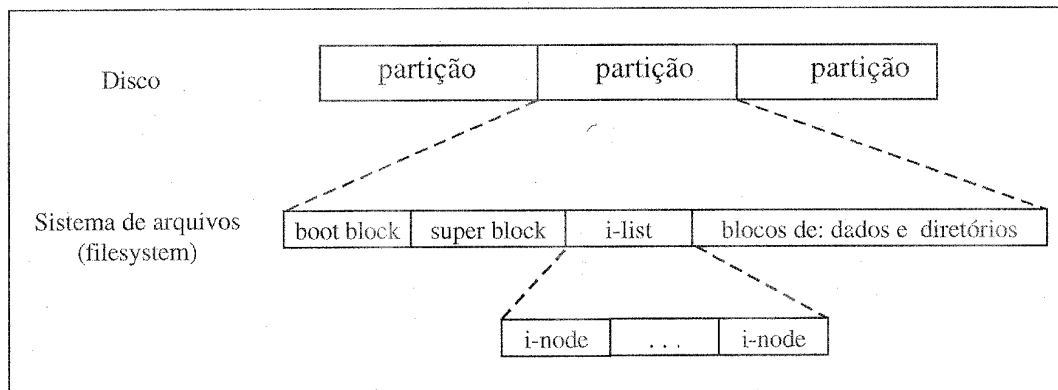


Figura III.2 - Representação Interna do Sistema de Arquivos

Para cada sistema de arquivos tem-se um setor de boot (boot block), que contém o código encarregado de, entre outras coisas, realizar a carga do código do kernel para a memória e iniciar sua execução. Posterior ao setor de boot, encontra-se o superblock, onde são armazenadas várias informações a respeito de seu sistema de arquivos. As informações mais importantes armazenadas no superblock são:

- o tamanho do sistema de arquivos;
- o número de blocos livres no sistema de arquivos;
- uma lista de blocos livres disponíveis no sistema de arquivos;
- o tamanho da lista de *i-nodes*, ou seja, o número de *i-nodes* para o sistema de arquivos;
- o número de *i-nodes* livres no sistema de arquivos;
- uma lista de *i-nodes* livres no sistema de arquivos;
- outras informações.

Devido à importância do superblock, o UNIX sempre faz uma cópia do superblock na memória RAM. O superblock é atualizado no disco quando o comando sync é executado. Por questões de segurança existe uma cópia do superblock em cada grupo de cilindros, para que caso seja perdido, danificado ou corrompido o mesmo possa ser recuperado a partir de um grupo de cilindros.

Em seguida tem-se a lista de i-nodes do sistema (i-list), e posteriormente a região onde são armazenados os conteúdos dos arquivos e diretórios.

2.2 - I-nodes

I-node é uma estrutura que contém várias informações a respeito do arquivo e a identificação do(s) bloco(s) de disco que contém os dados do arquivo. Alguns campos de informação que estão presentes em um i-node são:

- o identificador do “dono” do arquivo (composto por um UID e um GID);
- o tipo de arquivo;
- permissões de acesso ao arquivo;
- data e hora da última modificação nos dados do arquivo;
- data e hora do último acesso ao arquivo;
- data e hora da última modificação do *i-node*;
- número de *links* do arquivo;
- tamanho do arquivo em bytes;
- endereços do disco ou ponteiros de endereço no disco onde os dados do arquivo estão armazenados;
- outras informações.

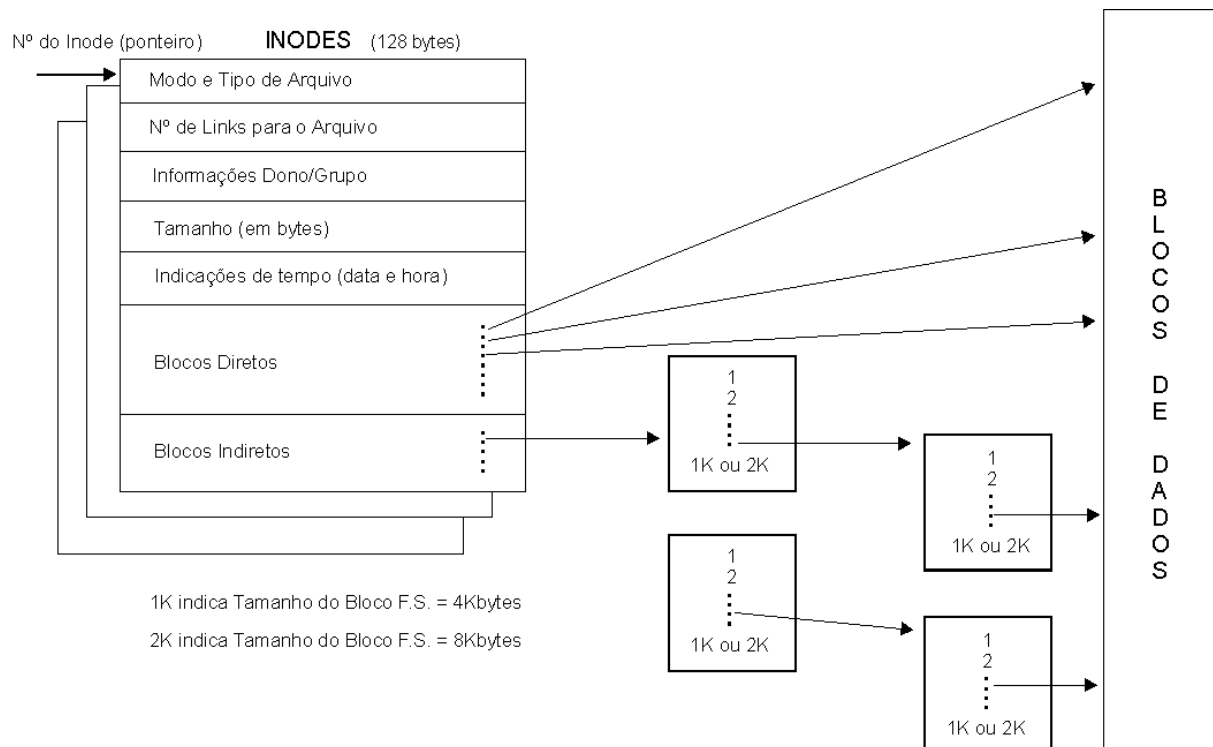


Figura III.3 - Representação de um Inode

Os i-nodes apontam para blocos de dados no disco, os quais não precisam estar necessariamente localizados fisicamente um após o outro. O i-node é o único elemento que sabe onde os dados do arquivo estão.

2.3 - Entrada de Diretório

A entrada de diretório de um arquivo é composta de 4 partes: O número do i-node, o comprimento desta entrada de diretório, o comprimento do nome de arquivo e o nome de arquivo. A entrada de diretório “mapeia” um nome de arquivo particular para um número de i-node. O número de i-node “mapeia” para um i-node.

Um diretório comporta-se como um arquivo no filesystem, exceto que um usuário não pode escrever dados no diretório. A função do diretório é localizar todos os arquivos em um nível definido na árvore do filesystem.

Para executar, o diretório contém uma entrada para cada arquivo que deve localizar. A entrada contém o nome do arquivo e seu número de i-node.

No exemplo da figura 8, na entrada de diretório existe um arquivo chamado produtos que possui o i-number #173, dentre outras informações. O i-number aponta para um determinado i-node de #173, na lista de i-nodes (i-lists), que por sua vez aponta para os blocos de dados que armazenam o conteúdo do arquivo “produtos” no diretório.

Uma entrada de diretórios pode ter um i-number referenciando um i-node que aponta para um outro diretório. Nesse caso, tem-se o conceito de sub-diretórios.

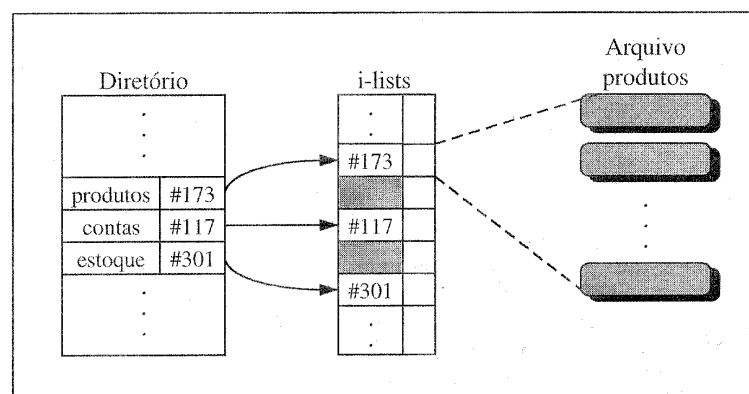


Figura III.4 - Exemplo de Entrada de Diretórios

3 - CAMINHOS

A maioria dos comandos do sistema UNIX operam sobre arquivos e/ou diretórios. Um caminho (path) representa os diretórios e subdiretórios que são passados para alcançar um arquivo ou diretório especificado.

O diretório inicial é chamado diretório raiz, representado por uma barra (/). Um path pode ser especificado de duas formas:

- absoluto;
- relativo.

Paths absolutos são aqueles que começam com uma barra (/). Paths relativos, ao contrário, sempre tomam o diretório atual como referência. Por exemplo:

- | | |
|--------------------|--|
| • /dev | absoluto |
| • /home/joao/texto | absoluto |
| • .. | relativo (diretório imediatamente acima) |
| • . | relativo (diretório corrente) |
| • usr/lib | relativo |
| • ../home/joao | relativo |

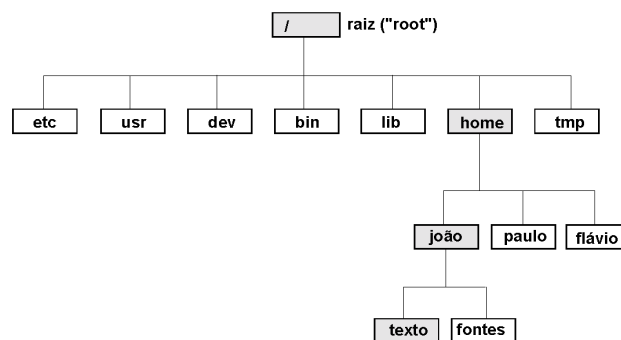


Figura III.5 - Caminho no UNIX

Quando um diretório é criado, duas entradas são automaticamente criadas, chamadas ponto (.) e ponto-ponto (..). Geralmente são utilizadas para designar caminhos relativos.

A entrada ponto (.) representa sua posição no diretório corrente. Se você está no diretório /home/joão então temos:

- . representa o diretório corrente /home/joão
- ./texto representa /home/joão/texto

A entrada ponto-ponto (..) representa o diretório imediatamente acima do seu diretório corrente, também chamado de diretório-pai. Se você está no diretório /home/joão então temos:

- .. representa o diretório / (raiz)
- ../.. também representa o diretório / (raiz)
- ../bin representa o diretório /bin

4 - DIRETÓRIO DE LOGIN

Quando um usuário é incluído no sistema receberá um username e opcionalmente uma password, e um diretório será criado para o seu próprio uso. O diretório é criado sob o diretório /home (ou /users) e tem o mesmo nome da identificação do usuário no login. A partir deste diretório, o usuário tem acesso livre para criar quantos arquivos e diretórios quiser.

Quando o usuário entrar no sistema, o UNIX automaticamente o colocará sob este diretório, ou seja, este será o diretório corrente logo após o login. Este diretório é denominado de diretório login ou diretório HOME. A figura abaixo mostra o diretório de login do usuário joão.

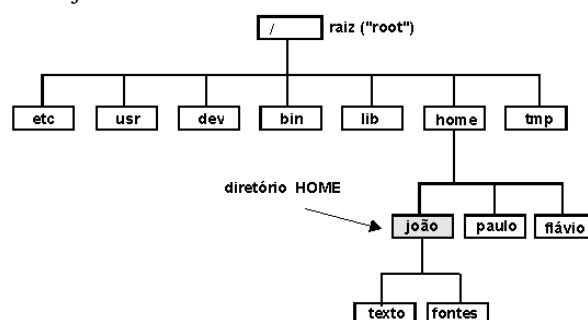


Figura III.6 - Diretório HOME