

Todas as aplicações precisam armazenar e recuperar informações. O espaço de endereçamento virtual pode ser pequeno demais para aplicações. Além disso, as informações mantidas na memória são perdidas com o término do processo e múltiplos processos não acessam a mesma informação ao mesmo tempo. A solução para armazenar a informação em discos é criar arquivos. A informação deve ser persistente, ou seja, um arquivo desaparecerá quando for removida explicitamente. O sistema de arquivos (parte do SO) é responsável pelo modo como os dados são estruturados, nomeados, acessados, usados, protegidos e implementados. As regras exatas para nomear um arquivo variam de sistema para sistema. Muitos sistemas permitem nomes com até 255 caracteres permitindo dígitos e caracteres especiais. A extensão do arquivo é separada do nome através de um "." (ponto).

O sistema de arquivos é a parte de um sistema operacional que fica mais visível para o usuário do computador. É por meio dele que se manipulam os arquivos existentes no computador. Os sistemas de arquivos Windows e Linux que ganham popularidade nos últimos anos e vêm sofrendo uma rápida evolução. O Linux manteve o modelo do sistema de arquivos padrão do Unix, que é um arquivo que pode ser qualquer entidade capaz de tratar as entradas e as saídas de um fluxo de dados. O Kernel do Linux lida com todos esses tipos de arquivos ocultando os detalhes de implementação de qualquer tipo de arquivo por de trás de uma camada de software. O sistema de arquivos Windows FAT é aceitável e perfeitamente funcional para a maioria dos usuários domésticos, o qual foi necessário o desenvolvimento de um sistema de arquivos que se adequasse ao funcionamento do Windows NT sendo criado o NTFS que é muito melhor sendo projetada com muitos recursos, inclusive recuperação de dados, segurança, tolerância a falhas.

1. Introdução

Um sistema de arquivos é o método e a estrutura de dados que um sistema operacional utiliza para administrar arquivos em um disco ou partição, ou seja, a forma pela qual os arquivos estão organizados em um disco. A expressão também é utilizada para se referenciar a uma partição ou disco que seja usado para armazenar os arquivos ou outros tipos de sistemas de arquivos.

Alguém pode dizer "eu tenho dois sistemas de arquivos", significando que tem duas partições nas quais armazena arquivos ou aquela pessoa está usando o "sistema de arquivos estendido", exemplificando o tipo do sistema de arquivos.

A diferença entre um disco ou partição e um sistema de arquivos é bastante significativa. Poucos programas (inclusive os programas que criam sistemas de arquivos) operam diretamente em setores não inicializados de um disco ou partição, e caso exista um sistema de arquivos ele será destruído ou danificado seriamente. A maioria dos programas trabalha em um sistema de arquivos e não funcionam em uma partição que não contenha um (ou que contenha um de tipo errado).

Antes de uma partição ou disco ser usado como um sistema de arquivos ele necessita ser inicializado, e a estrutura básica de dados necessita ser gravada no disco.

Este processo é chamado criação de um sistema de arquivos.

O sistema de arquivos FAT

FAT é a sigla para File Allocation Table (ou tabela de alocação de arquivos). O primeiro FAT surgiu em 1977, para funcionar com a primeira versão do DOS. Trata-se de um sistema que funciona através de uma espécie de tabela que contém indicações para onde estão as informações de cada arquivo. Quando um arquivo é salvo num disquete, por exemplo, o FAT divide a área do disco em

pequenos blocos. Assim, um arquivo pode (e ocupa) vários blocos, mas eles não precisam estar numa seqüência. Os blocos de determinados arquivos podem estar em várias posições diferentes. Daí a necessidade de uma tabela para indicar cada bloco.

Com o surgimento de dispositivos de armazenamento com mais capacidade e mais sofisticados, o sistema FAT foi ganhando alterações (identificadas pelos nomes FAT12 e FAT16). Isso foi necessário porque o FAT era limitado a determinada capacidade de armazenamento. Por exemplo, ele só operava com tamanho máximo de 2 GB. Assim, num disco de 5 GB, seria necessário dividi-lo em 3 partições. Fora o fato de que o FAT apresentava problemas com informações acima de 512 MB. Diante de tantos problemas, em 1996, a Microsoft lançou um novo FAT: o FAT32, que é compatível com os Windows 9x/Me/2000 e XP (apesar destes dois últimos terem um sistema de arquivos mais avançado, o NTFS).

Funcionamento do sistema FAT

Ao trabalharmos com HDs (e disquetes) é necessário prepará-los, fazendo uma formatação física. Este processo, divide os discos em trilhas (uma espécie de caminho circular) e setores (subdivisões de cada trilha, com geralmente 512 bytes). Um conjunto de trilhas recebe o nome de cilindro. A formatação física já vem de fábrica e pode ser alterada se o usuário quiser dividir o disco em partições. Depois deve-se fazer uma formatação lógica, que nada mais é do que "instalar" o sistema de arquivos no dispositivo de armazenamento.

O sistema de arquivos FAT não trabalha diretamente com cada setor, mas sim com um grupo de setores. Esse grupo é chamado de cluster (ou unidade de alocação). Se por exemplo, um disco com setor de 512 bytes, tiver 5 KB de tamanho, ele terá 10 setores e 5 clusters, se cada cluster ocupar dois setores. Sendo assim, quando o FAT precisar acessar um determinado setor, primeiro ele descobre em qual cluster ele se encontra. É válido citar que tanto o FAT quanto o FAT32 trabalham de acordo com este princípio.

Tamanho de cluster

O sistema FAT exige que cada cluster do disco seja usado somente para um único arquivo, ou seja, num mesmo cluster, não pode haver informações sobre mais de um arquivo. Isso pode até parecer óbvio, mas gera um problema: desperdício. Para mostrar isso, vamos supor que desejamos guardar num disquete um arquivo de 5 KB. Imaginemos que este disquete tenha 8 KB de espaço e dois clusters de 4 KB. Um cluster ocuparia 4 KB do arquivo, enquanto o outro cluster ocuparia apenas 1 KB. Como o cluster só pode trabalhar com um arquivo, haveria desperdício de 3 KB. Vamos imaginar agora que em vez de termos clusters com 4 KB, teremos clusters com 2 KB. Assim, 3 cluster seriam usados, sendo que um ainda apresentaria desperdício de 1 KB. No entanto, sobrou um cluster com 2 KB, que pode ser usado por outro arquivo.

Percebe-se com isso que o tamanho do cluster deve ser o máximo que o FAT consegue manipular. Aliás, a principal diferença entre FAT e FAT 32, é que este último consegue trabalhar com um número maior de clusters.

Diferenças entre FAT e FAT32

O sistema FAT (ou FAT16) consegue trabalhar com 65536 clusters. Esse número é obtido elevando o número 2 a 16 (daí a terminologia FAT16). Mas, na verdade, o sistema FAT16 usa apenas 65525 clusters por disco (ou partição). É importante frisar que o tamanho do cluster deve obedecer também uma potência de 2: 2 KB, 4 KB, 8 KB, 16 KB e 32 KB, ou seja, não é possível ter cluster de 5 KB, 7 KB, etc. O tamanho dos clusters no sistema FAT também é uma potência de 2. O limite máximo de tamanho para uma partição em FAT16 é de 2 GB (correspondente a 2 elevado a 16).

Já no caso do sistema de arquivos FAT32 (seu nome se deve ao mesmo motivo que no FAT32), o tamanho dos clusters é determinado através da relação entre os comandos FDISK e FORMAT, apesar de que é possível determinar o tamanho do cluster do FAT32 também por programas de terceiros, com o Partition Magic (não é possível ter clusters de diferentes tamanhos). O tamanho máximo da partição em FAT32 é de 2 TB. Mas se você fizer a contas notará que 2 elevado a 32 é equivalente a 128 TB. Então porque o FAT32 usa somente 2 TB? Pode parecer confuso, mas o número máximo de clusters no caso do FAT32 não é de 2 elevado a 32. Apesar de seu endereçamento ser de 32 bits, na verdade são usados apenas 28 bits. Com isso, a quantidade máxima de clusters seria 2 elevado a 28, que corresponde a 8 TB. Não está errado, é 8 TB mesmo! Então, qual a razão do FAT32 ter tamanho máximo de espaço de 2 TB? Segundo a Microsoft, o número máximo de setores (setores, não clusters!) que um disco pode ter é de 2 elevado a 32. Como cada setor tem 512 bytes, o tamanho máximo de um disco no FAT32 acaba sendo de 2 TB.

As diferenças entre FAT (ou FAT16) e FAT32 não param por aí. O FAT32 também é mais confiável, além disso este sistema também consegue posicionar o diretório principal em qualquer lugar do disco. Fora o fato de que no sistema FAT, havia uma limitação no número de entradas que podiam ser alocadas no diretório principal (512 arquivos e/ou pastas). Não há essa limitação no FAT32.

Algo curioso de ser citado, é que o FAT32 pode mudar o tamanho da partição sem perder dados. Apesar desta capacidade, a Microsoft, por alguma razão misteriosa, não implementou esta característica no FAT 32. Hoje em dia, programas particionadores, como o Partition Magic ou então particionadores de disco de distribuições Linux, conseguem redimensionar uma partição FAT32 "inserido" este poder ao sistema de arquivos.

Como o NTFS surgiu

O sistema de arquivos FAT é aceitável e perfeitamente funcional para a maioria dos usuários domésticos. Trata-se um sistema antigo, que mesmo com novas versões, herdou a simplicidade da primeira versão. Veja mais sobre FAT [aqui](#). As limitações do FAT, principalmente quanto à segurança, capacidade e confiança, fizeram do FAT um sistema de arquivos inadequado para uso em servidores e aplicações críticas. A Microsoft, estando ciente disso, decidiu desenvolver um sistema de arquivos que se adequasse aos princípios de funcionamento do Windows NT e lançou o New Technology File System, conhecido pela sigla NTFS. Entre os objetivos da idealização do NTFS estavam o fornecimento de um sistema de arquivos flexível, adaptável, altamente seguro e confiável. Sem dúvida, tais características fizeram do Windows NT um sistema operacional aceitável para as aplicações cujo seu desenvolvimento foi planejado.

A idéia de lançar o Windows NT surgiu em 1990, quando a Microsoft sentiu a necessidade de ter um sistema operacional com as qualidades citadas acima e com funcionalidades típicas de servidor. Nesta época, a Microsoft não tinha nenhum sistema que pudesse se equiparar ao poderoso Unix (tinha somente o MS-DOS e o Windows 3.x). Decidida a ter uma fatia do mercado que pertencia ao Unix, a Microsoft deu início ao desenvolvimento do Windows NT. Esta sigla - NT - significa New Technology. A Microsoft logo percebeu que este novo Windows não teria sucesso se utilizasse o FAT, pelas razões já citadas. Era preciso criar um sistema de arquivos novo.

O NTFS foi desenvolvido e muitos até hoje pensam que ele é um sistema de arquivos inteiramente desenvolvido pela Microsoft, o que não é verdade. Seu projeto foi baseado nas análises das necessidades do novo sistema operacional, mas seus conceitos funcionais foram "herdados" do sistema de arquivos HPFS (High Performance File System).

Um sistema operacional muito conhecido nesta época era o OS/2, um projeto realizado em conjunto entre a Microsoft e a IBM. Ambas as empresas estavam tentando criar um sistema operacional de grande sucesso, cujo apelo principal seria a capacidade gráfica

(lembre-se que naquela época, era muito maior o uso de sistemas operacionais baseados em linha de comando, como o DOS). O OS/2 de fato continha inovações tecnológicas, mas esbarrava nos quesitos suporte e marketing. Fora isso, a IBM e a Microsoft começaram a se desentender e a empresa de Bill Gates decidiu abandonar o projeto e se dedicar ao desenvolvimento do Windows NT. No entanto, a Microsoft acabou levando consigo muitos conceitos funcionais do sistema de arquivos do OS/2, o HPFS. É claro que tais conceitos foram essenciais para a criação do NTFS, o que fez com muitos pensassem que a Microsoft passou um golpe na IBM. No entanto, este artigo não visa discutir isso.

Características do NTFS

O NTFS possui características importantes, que o fez ser considerado um bom sistema de arquivos. Entre essas qualidades estão: confiança, pois permite que o sistema operacional se recupere de problemas sem perder informações, fazendo-o ser tolerante a falhas; segurança, onde é possível ter um controle de acesso preciso e ter aplicações que rodem em rede, fazendo com que seja possível o gerenciamento de usuários, incluindo suas permissões de acesso e escrita de dados; armazenamento, onde é possível trabalhar com uma grande quantidade de dados, permitindo inclusive o uso de arrays RAID; rede, fazendo do sistema plenamente funcional para o trabalho e o fluxo de dados em rede.

Há muitas outras características, que ficam mais ainda visíveis se comparadas ao FAT. A Microsoft vem trabalhando bastante para aperfeiçoar o NTFS, por isso, é de se esperar que novas características sejam implementadas no sistema de arquivos, de acordo com o lançamento de novas versões do Windows.

Versões do NTFS

Assim como aconteceu com o FAT, o NTFS também tem versões, que foram lançadas principalmente no surgimento de novos Windows. A cada versão, correções de falhas são feitas, suportes a hardware são implementados e novas características são dadas ao NTFS. A princípio houve o NTFS 1.0 usado no Windows NT 3.1 (por isso, esta versão do NTFS também ficou conhecida por NTFS 3.1).

Com o lançamento do Windows NT 4, o NTFS ganhou a versão 1.1 (ou versão 4). Esta versão também foi usada no Windows NT 3.51. O sucesso do Windows NT foi tão grande que sua versão do NTFS virou referência em sistemas de arquivos.

A Microsoft não ficou parada e lançou a versão conhecida como NTFS 5.0 com o lançamento do Windows 2000, substituto do Windows NT. Apesar da nova versão, o NTFS 4 foi tão difundido que seu suporte a outros sistemas operacionais não acabará tão cedo.

Esta nova versão do NTFS possui novas características importantes, além daquelas herdadas da versão anterior. Essas mudanças foram essenciais para fazer do Windows 2000 um sistema que fosse realmente adequado para substituir o Windows NT. Só para servir de exemplo, o serviço Active Directory é um dos chamativos do Windows 2000 e foi implementado graças a alterações no NTFS. Entre os novos recursos do NTFS 5 estão: Reparse Points, onde arquivos e pastas dentro do sistema de arquivos podem ter ações associadas a eles, de forma que operações particulares a estes arquivos possam ser executadas; novas características de segurança, onde o mecanismo para gerenciamento da segurança e de usuários, principalmente em relação a acesso e arquivos foram melhorados; quotas de discos, onde o administrador do sistema pode determinar o espaço em disco disponível a um usuário ou a um grupo de usuários; diários de alterações, onde volumes podem ser ajustados para rastrear as operações efetuadas nos arquivos e pastas; codificação, onde o sistema permite que arquivos sejam codificados/decodificados automaticamente; suporte a arquivos esparsos, onde é possível armazenar de forma eficiente arquivos esparsos (que são arquivos grandes mas que possuem algumas estruturas vazias, desperdiçando espaço em disco).

Com o lançamento do Windows XP, Windows 2003 Server e futuras versões, o NTFS vai ganhando melhoramentos e novas características, mas certamente a versão 4 ainda será uma referência. Isso deixa claro que o NTFS não deixará de ser usado tão cedo pela Microsoft.

Funcionamento do NTFS

Conforme as características herdadas do HPFS, o NTFS trabalha de uma forma mais eficiente no gerenciamento do espaço de disco. Isso porque as informações são armazenadas em uma base por setor do disco, em vez de utilizar clusters de múltiplos setores (veja mais sobre isso lendo o artigo sobre [FAT](#)). Essa forma de trabalho, traz várias vantagens, como menor necessidade de desfragmentação de disco e maior consistência de dados. Isso porque essa arquitetura de dados por base em setor permite manter os dados próximos, ou seja, não espalhados pelo disco. Até o gerenciamento de grandes quantidades de dados é beneficiado por esta característica, já que como acontecia com o FAT, trabalhar com clusters por setor, fazia do sistema de arquivos dependente de um número pré-determinado de setores.

Funcionamento padrão dos sistemas de arquivos em Linux

Os sistemas de arquivos são criados em partições do disco, de forma que seja possível armazenar programas e dados em formato de arquivos e diretórios (pastas). O Linux, assim como praticamente todos os sistemas operacionais baseados em Unix, usa um sistema de arquivos que possuem uma hierarquia, composta de arquivos e diretórios, que podem conter outros diretórios ou arquivos.

Os arquivos/diretórios (sistemas baseados em Unix tratam os diretórios como arquivos especiais) em um sistema de arquivos para Linux são disponibilizados (ou montados) para manipulação através do comando "mount", geralmente acionado no processo de startup (inicialização), que ocorre quando o computador é ligado e começa-se a carregar o sistema operacional. O Linux consegue trabalhar com vários sistemas de arquivos num mesmo disco (situação comum à usuários que possuem Windows e Linux em suas máquinas, por exemplo) e para "enxergá-los", armazena a lista de sistemas de arquivos disponíveis no arquivo `/etc/fstab` (repare que `/etc/` indica um caminho de diretório). No entanto, há uma lista de sistemas de arquivos que estão efetivamente em uso, disponível no arquivo `/etc/mstab`, também conhecido como "tabela mount". Esta lista é atualizada no processo de startup, para indicar ao sistema operacional quais sistemas de arquivos ele poderá acessar.

Para cada sistema de arquivos montado no startup, um bit no cabeçalho do sistema de arquivos é zerado para indicar que o sistema de arquivos está em uso a partir daquele momento e que as estruturas de dados usadas para o alocação e organização de arquivos/diretórios podem sofrer mudanças (atualizações).

Quando o usuário decide desligar o computador e usa comandos para encerrar o Linux, os sistemas de arquivos são desmontados, fazendo com que o bit citado acima seja modificado para indicar que o sistema de arquivos está consistente, ou seja, não pode mais sofrer mudanças.

Erros e corrompimento do sistema de arquivos

Os sistemas de arquivos para Linux já sofreram muitos tipos de tratamentos e reescrita de código para eliminar o corrompimento causado por aplicações ou pelo próprio kernel. No entanto, eliminar o corrompimento de dados em arquivos causados, por exemplo, pela queda de energia ou pelo desligamento incorreto por parte do usuário, sempre foi uma tarefa praticamente impossível. Quando o sistema é desligado incorretamente o bit do cabeçalho do sistema de arquivos não é ajustado. A solução foi fazer com que, no próximo

processo de carregamento do Linux, seja verificado se o cabeçalho está com o bit de cabeçalho setado para indicar que o sistema de arquivos está consistente e não manipulável. Caso não esteja, a ferramenta "fsck" verifica o sistema na busca de erros.

Sistema de arquivos Ext3

Existem vários sistemas de arquivos disponíveis com a tecnologia Journaling, como o XFS, desenvolvido originalmente pela Silicon Graphics e posteriormente disponibilizado com código aberto, o ReiserFS, desenvolvido especialmente para Linux, JFS, desenvolvido originalmente pela IBM mas também liberado com código aberto, e o mais conhecido deles: o ext3, desenvolvido pelo Dr. Stephen Tweedie juntamente com outros colaboradores, na Red Hat, e que veremos agora.

O sistema de arquivos ext3 é basicamente o sistema de arquivos ext2 com recursos de Journaling. Talvez, essa seja a razão de seu uso amplo: ele é totalmente compatível com ext2 (que foi um sistema de arquivos muito usado), o que nenhum outro sistema de arquivos baseado em Journaling é.

O ext3 passou a ser efetivamente suportado pelo kernel do Linux a partir da versão 2.4. Conseqüentemente, todas as distribuições Linux lançadas com esse kernel ou superior, tem suporte padrão para ext3.

No ext3, o código de Journaling usa uma camada chamada "Journaling Block Device" (JBD). A JBD foi criada com o propósito de implementar Journal em qualquer tipo de dispositivo com base em blocos de dados. Por exemplo, o código ext3 informa e "pede autorização" à JBD para efetuar as mudanças, antes de modificar/adicionar qualquer dado no disco. Sendo assim, é o JBD que verdadeiramente "gerencia" o Journal. O fato mais interessante disso é que, a JBD funciona como uma entidade independente, permitindo que não só o ext3 a use, mas também outros sistemas de arquivos.

A JBD utiliza um método diferente de outros Journalings para recuperação de informações. Ao invés de armazenar as informações em bytes que depois devem ser implementados, a JBD grava os próprios blocos modificados do sistema de arquivos. Assim, o ext3 também armazena "réplicas" completas dos blocos modificados em memória para rastrear as operações que ficaram pendentes. A desvantagem desta forma de trabalho é que o Journal acaba sendo maior. No entanto, o ext3 não precisa lidar com a complexidade dos Journalings que trabalham gravando bytes.

Sistema de arquivos Ext2

O sistema de arquivos mais antigo dos atualmente em uso. o ext2 é sucessor do ext (extended). É muito robusto, rápido e confiável. Mas não possui o recurso de journal.

O Journal é uma área especial da partição onde são gravados um tipo de log de transação, este log pode auxiliar o sistema a recuperar a integridade do disco em caso de um desligamento inesperado do sistema. Infelizmente o ext2 não possui este importante recurso.. e é a primeira partição a dar problema quando o sistema não é deligado de modo adequado.

O ext2 ainda é utilizado em alguns casos pois seu sistema de quota de disco é perfeito. o do ext3 não é confiável, o reiserfs nem dispõe deste recurso, o xfs possui um quota só dele que funciona muito bem.

Sistema de arquivos ReiserFS

Este sistema de arquivos é muito confiável quanto aos quesitos confiabilidade e rapidez. Possui um excelente journal que raramente apresenta algum tipo de problema. O que este sistema de arquivos peca é justamente não dispor de sistema quota de disco.

Sistema de arquivos Xfs

Um dos mais recente sistema de arquivos disponível para Linux. é um código que veio da Silicom Gráficos, rápido, robusto, com suporte melhorado a arquivos muito grandes e diretórios com muitos arquivos, tem um excelente journal que funciona muito bem.. e também possui um sistema de quota que é próprio dele mas que tem um funcionamento bem adequado.

Estrutura de diretórios do Linux

O sistema de arquivos raiz (root) do Linux

Diretório raiz /

O diretório raiz geralmente não contém nenhum arquivo, exceto, em algumas distribuições, pela imagem de inicialização (boot) do sistema chamada /vmlinuz. Todos os outros arquivos estão em subdiretórios do raiz: /bin Comandos necessários durante a inicialização do sistema que podem ser utilizados pelos usuários (provavelmente após a ativação). /sbin Similar ao /bin, porém os comandos não são destinados aos usuários comuns, apesar de poderem ser utilizados por estes se necessário. /etc Arquivos de configuração específico da máquina. /lib Bibliotecas compartilhadas necessárias aos programas no sistema de arquivos raiz. /lib/modules Módulo dinâmico e carregáveis pelo kernel, especialmente aqueles necessários para inicializar o sistema em caso de acidentes (por exemplo, programas de controle de sistemas de arquivos de redes). /dev Arquivos de dispositivos. /tmp Arquivos temporários. Programas que são executados após a ativação do sistema devem usar o /var/tmp e não o /tmp, uma vez que provavelmente encontrarão mais espaço disponível neste sistema de arquivos. /boot Arquivos utilizados pelo gerenciador de inicialização do sistema, p. ex., LILO. Imagens do kernel não normalmente mantidas aqui ao invés do no diretório raiz. Caso existam inúmeras imagens, o diretório pode facilmente crescer demasiadamente, sendo aconselháveis mantê-lo em um sistema de arquivos à parte. Outra razão é estar seguro de que as imagens do kernel estejam nos primeiros 1024 cilindros de um disco IDE. /mnt Ponto para montagens temporárias realizadas pelo administrador do sistema. Programas e usualmente não prevêm a montagem automática no /mnt. Este deve estar dividido em subdiretórios, como por exemplo, /mnt/dosa sendo uma unidade de disquetes utilizando um sistema de arquivos MS-DOS, e /mnt/exta pode ser a mesma unidade, porém com usam extensão ext2. /proc, /usr, /var, /home Pontos de montagem para outros sistemas de arquivos.

Diretório /etc

O diretório /etc contém inúmeros arquivos. Alguns deles estão descritos resumidamente a seguir. Para outras descrições é necessário determinar a quais programas eles pertencem e ler as páginas de manual daqueles programas. Para sistemas que utilizam a tecnologia RPM, basta executar “rpm -qf arquivo” que o sistema lhe dirá a qual pacote o arquivo pertence. Muitos arquivos de configuração de programas de rede estão no etc. /etc/rc ou /etc/rc.d ou /etc/rc?.d São scripts (arquivos de lote) ou os diretórios de scripts que são executados no início do sistema ou na mudança de nível do sistema. /etc/passwd A base de dados dos usuários, cujos campos definem nome do usuário, nome real, diretório pessoal, senha criptografada e outras informações específicas de cada usuário. /etc/fdprm A tabela de parâmetros de disquetes. Descreve os diferentes formatos disponíveis. É utilizada pelo programa setfdprm. /etc/fstab Lista os sistemas de arquivos montados automaticamente, durante a inicialização do sistema, pelo comando mount -a (no script de inicialização /etc/rc.d/rc.sysinit ou similar). No Linux, contém ainda informações sobre as áreas de troca (swap) usadas automaticamente pelo comando swapon -a. /etc/group Similar ao /etc/passwd, porém descreve grupos ao invés de usuários. /etc/inittab Arquivo de configuração do init. /etc/issue Saída do programa getty antes do prompt de acesso ao sistema. Normalmente contém uma breve ou mensagem de boas-vindas. O conteúdo fica a critério do administrador do sistema. /etc/motd Escreve uma “mensagem do dia” automaticamente após um acesso bem sucedido. Conteúdos ficam a critério do administrador do sistema. Normalmente são usadas para enviar informações para os usuários, como por exemplo, avisos de desligamentos planejados. /etc/mstab Lista dos sistemas

de arquivos montados. Inicialmente configurado por scripts e posteriormente atualizado pelo comando mount. Usado quando uma listagem dos sistemas de arquivos é necessária, como, por exemplo, pelo comando `df. /etc/shadow/` Arquivo de senhas em sistemas onde o shadow esteja instalado. Esta opção move senhas criptografadas do `/etc/passwd` para o arquivo `/etc/shadow` o qual somente pode ser lido pelo superusuário. O shadow dá uma maior segurança às senhas dos usuários. `/etc/login.defs` Arquivo de configuração do programa login. `/etc/termcap` O arquivo de configuração de terminal. Descreve as seqüências de escape para os diversos tipos de terminais. Os programas ao invés de escreverem diretamente uma seqüência que funcione em determinado tipo de terminal devem buscar a seqüência correta no `/etc/termcap`. Desta maneira os programas funcionam na maioria dos terminais. `/etc/printcap` Similar ao `/etc/termcap`, mas direcionado à impressora e com sintaxe diferente. `/etc/profile`, `/etc/bashrc` Arquivos executados pelo Bourne Shell e Bash, no momento do login do usuário. Estes arquivos permitem ao administrador manipular o ambiente de trabalho de todos os usuários do sistema. Veja as páginas de manual dos interpretadores de comandos para detalhes específicos. `/etc/securetty` Identifica terminais seguros, ou seja, aquele nos quais é permitido que o root acesse o sistema. Normalmente somente quando os consoles virtuais são listados, tornando muito mais difícil obter os privilégios de superusuário do sistema através da rede ou de uma conexão via modem. `/etc/shells/` Lista as shells válidas. O comando `chsh` permite que os usuários mudem sua shell de login, porém, somente para aquelas listadas neste arquivo. O servidor de transferência de arquivos, `ftpd`, checará se a shell do usuário está listada no `/etc/shells` e não aceitará acessos, a menos que ela esteja presente neste arquivo.

Diretório /dev

O diretório `/dev` contém arquivos especiais (drivers) de controle de todos os dispositivos. Estes arquivos utilizam uma denominação especial descrita na lista de dispositivos. Os controladores são criados durante a instalação, e posteriormente através do script `/dev/MAKEDEV.local`, é um script escrito pelo administrador para criar somente os dispositivos locais que não façam parte do padrão do MAKEDEV.

Diretório /usr

O diretório de arquivos `/usr` é grande, uma vez que a maioria dos programas estão ali instalados. Os arquivos `/usr` são normalmente instalados pela sua distribuição do Linux. Programas locais são instalados no `/usr/local`. Isso torna a atualização do sistema possível para uma nova versão da distribuição ou mesmo para uma distribuição completamente nova, sem que seja necessário instalar todos os programas novamente. Alguns dos subdiretórios do `/usr` estão listados a seguir (os mais importantes); `/usr/X11r6` Arquivos do X Window. Para simplificar o desenvolvimento e a instalação do X, seus arquivos não foram integrados ao restante do sistema. Há uma árvore de diretórios sob `/usr/X11R6` similar ao `/usr`. `/usr/X386` Similar ao `/usr/X11R6`, mas para o X11 Release 5. `/usr/bin` Praticamente todos os comandos de usuários. Alguns outros podem ser encontrados em `/bin` ou `/usr/local/bin`. `/usr/sbin` Comandos de administração do sistema que não necessitem estar no sistema de arquivos raiz, como servidores. `/usr/man`, `/usr/info`, `/usr/doc` Página de manual, documentos de GUN/Info e diversos outros documentos, respectivamente. `/usr/include` Arquivos header para a linguagem de programação C. Este diretório sob o `/usr/lib`, porém por tradição tem sido mantido neste local. `/usr/lib` Arquivos estáticos de dados para programas e subsistemas, incluindo alguns arquivos de configuração globais. O “lib” vem de library (biblioteca); originalmente as bibliotecas de programação eram armazenadas neste subdiretório. `/usr/local` O local para programas instalados localmente e outros arquivos. `/usr/share/magic` O arquivo de configuração para o comando `file`. Contém as descrições de vários formatos de arquivos nas quais o `file` pode descobrir o tipo do arquivo. Veja as páginas de manual do `magic` e do `file` para maiores informações.

Diretório /var

O sistema de arquivos `/var` contém dados que são alterados quando o sistema está sendo executado. É específico de cada sistema, ou seja, não compartilhado através da rede com outros equipamentos. `/var/catman` Um cache para páginas de manual que são formatadas. Fontes para as páginas são armazenadas em `/usr/man/man*`, sendo que algumas podem vir préformadas, estando armazenadas em

`/usr/man/cat*`. Outras páginas necessitam ser formatadas quando acessadas pela primeira vez. A versão formatada fica armazenada em `/var/catman`, fazendo com que outro usuário que acesse a página, tenha esta pronta e disponível. O diretório `/var/catman` é freqüentemente esvaziado, da mesma forma que os arquivos temporários. `/var/lib` Arquivos que mudam enquanto o sistema está ativo. `/var/lock` Arquivos de bloqueio. Muitos programas seguem uma convenção de se criar arquivos no `/var/lock` para indicar que eles estão utilizando um dispositivo em particular ou um determinado arquivo de bloqueio e não tentarão utilizar o mesmo dispositivo ou arquivo.

Diretório /var/log

Arquivos de históricos de vários programas, especialmente o login (`/var/log/wtmp`, o qual registra todas as entradas e saídas do sistema) e o syslog (`/var/log/messages`, o qual contém as mensagens do kernel e programas do sistema). Os arquivos no `/var/log` podem crescer indefinidamente com frequência e podem requerer limpezas periódicas. `/var/run` Arquivos que contêm informações sobre o sistema e que são válidas até a próxima inicialização. Por exemplo, o `/var/run/utmp` contém informações sobre os usuários atualmente conectados. `/var/tmp` Diretórios para mail, news, filas de impressão e outros trabalhos em fila. Cada fila tem o seu próprio subdiretório sob o `/var/spool`, por ex. as mensagens de correio eletrônico estão armazenadas em `/var/spool/mail`. `/var/tmp` Arquivos temporários que sejam muito grandes ou que necessitem existir por um período maior que o definido para o `/tmp`. Note que o administrador do sistema pode não permitir arquivos muito antigos também no `/var/tmp`.

Diretório /proc

O `/proc` é um sistema de arquivos ilusório. Na verdade ele não existe em um disco rígido. É criado em memória pelo kernel, sendo usado para disponibilizar informações sobre o sistema (originalmente sobre os processos, daí a origem de seu nome). O `/proc` é descrito mais detalhadamente na página de manual `proc`. `/proc/1` Um diretório com as informações 1 (init). Cada processo tem um diretório embaixo do `/proc`, cujo nome equivale ao seu número de identificação. `/proc/cpuinfo` Informações sobre o processador, tais como tipo, fabricante, modelo, e performance. `/proc/devices` Lista dos controladores de dispositivos configurados no kernel atualmente em execução. `/proc/dma` Mostra quais canais DMA estão sendo utilizados no momento. `/proc/filesystems` Sistemas de arquivos configurados no kernel. `/proc/interrupts` Mostra quais interrupções estão em uso e quantas vezes foram chamadas. `/proc/ioports` Quais portas de entrada e saída estão em uso no momento. `/proc/kcore` Imagem da memória física do sistema. Tem exatamente o mesmo tamanho da memória física, mas não ocupa toda aquela área. Ela é gerada em tempo de execução, através do acesso dos programas. Lembre-se: a menos que você copie esta imagem para outro lugar, nada abaixo do `/proc` ocupa espaço em disco. `/proc/ksyms` Mensagem de saída do kernel. São direcionadas também para o syslog. `/proc/loadavg` A "carga média" do sistema, contendo três indicadores de quanto trabalho o sistema está executando no momento. `/proc/meminfo` Informações sobre o uso de memória, tanto física como de swap. `/proc/modules` Descreve quais módulos estão carregados no momento. `/proc/net` Informações sobre a situação dos protocolos de rede. `/proc/self` Um link simbólico para o diretório dos processos do programa que está olhando para o `/proc` no momento. Quando dois processos acessam o `/proc`, estes recebem diferentes links. Isto é feito para facilitar a vida dos programas quando o seu diretório de processos. `/proc/stat` Diversas estatísticas sobre o sistema, como o número de paginações desde seu início, E/S, processos executados, etc... `/proc/uptime` O tempo que o sistema está ativo. `/proc/version` Versão do kernel do sistema. Enquanto os esses arquivos são de fácil leitura, por estarem em formato texto, não quer dizer que eles sejam facilmente lidos. Há muitos comandos que fazem mais que somente ler os arquivos acima e formatar o seu conteúdo para um melhor entendimento. Por exemplo, o programa `free` lê o conteúdo do `/proc/meminfo` e converte as informações dadas em bytes, para kilobytes, com algumas pequenas informações adicionais.