

Onde vai o Quê em um sistema Linux?!

Um sistema Linux típico tem uma grande quantidade de arquivos, dos mais variados tipos, organizados em uma estrutura de árvore de diretórios. Para alguém que faz a transição do mundo DOS/Windows, a profusão de diretórios pode intimidar, a ponto de fazer com que a decisão de onde instalar um arquivo de programa se torne um grande enigma. Há também o problema de saber como particionar o disco rígido, HD, antes da instalação: vai tudo em uma grande partição, ou se não, onde vai o quê?!

Este artigo deve responder a essas questões.

Um importante fator ao decidir como particionar o(s) disco(s) é o modo como os sistemas UNIX dão partida, além do conceito relacionado de níveis de execução – *runlevels*¹. Na maior parte do tempo, o sistema estará no *runlevel* 3 (multi-usuário com conectividade à rede e *daemons* de rede ativos), ou no *runlevel* 5 (o mesmo do 3, mais *login* gráfico e servidor X ativo). Quando o sistema dá a partida² (*boot*), ele monta o sistema de arquivos raiz no modo somente leitura e então lê o arquivo */etc/inittab*, que lhe diz para qual nível deve agora mudar (veja a id: entry). Um sistema típico vai agora rodar o script */etc/rc.d/rc.sysinit*, e então rodar os *scripts* de partida para o *runlevel* alvo, *n*, do diretório */etc/rc.d/rcn.d*.

Como parte desse processo, ele remontará o sistema de arquivos raiz em modo leitura e escrita e montará quaisquer sistemas de arquivos adicionais. Onde ele fará isso, varia de distribuição para distribuição, mas a leitura dos *scripts* de partida da sua distribuição lhe revelará tudo. O ponto importante é que antes desse instante somente o sistema de arquivos raiz estará montado – e mesmo ele estará montado em modo somente leitura. Outros sistemas de arquivos – que podem incluir */home*, */usr* e */var* – não estarão acessíveis, e assim não poderão ser usados durante o processo de partida (*boot*). Se algo acontecer de errado durante o *boot*, e você se deparar com um *prompt* do *shell* para reparar o problema, aqueles sistemas de arquivos não estarão acessíveis de imediato.

A maioria dos sistemas suporta um modo de manutenção por usuário único, *runlevel* 1, no qual apenas *logons* de console são admitidos, isto é, outros usuários não podem fazer *logon* via terminais externos ou sessões em rede. Em *runlevel* 1 nenhum sistema de arquivos é montado além do *root*. O modo de manutenção por usuário único é usado quando se realizam tarefas maiores de administração, tais como criar ou remover sistemas de arquivos – algo que seria perigoso se outros usuários estivessem rodando programas no sistema.

Diretórios do Nível Mais Alto.

/bin*, */sbin*, */lib

O diretório */bin* contém os arquivos de programa que são usados pelo sistema durante a partida, além de arquivos de comandos que podem ser solicitados por usuários durante a operação normal do sistema, depois que este está completamente carregado e rodando. Exemplos desses arquivos são: */bin/bash*, *cp*, *grep*, *more*, *pwd*, *touch* e muitos outros.

Ao contrário, */sbin* contém programas que são solicitados pelos *scripts* de partida do sistema, mas não serão normalmente usados por usuários comuns, mas sim pelo usuário *root* – em outras palavras, utilitários para administração do sistema. Exemplos incluem: *modprobe*, *ifconfig*, *syslog*, *shutdown*, *route*, *sysctl* e outros.

No diretório */lib* estão as bibliotecas objeto compartilhadas (análogas às *.dll* do Windows) que são solicitadas durante a partida do sistema. Em particular, os módulos do *kernel* serão encontrados aqui, sob */lib/modules*.

1 - usaremos o termo *runlevel*, uma vez que 'nível de execução' ainda não é de uso comum nos artigos sobre Linux. A maioria dos termos usados aqui em Inglês são assim mantidos para melhor entendimento do texto. (N. do T.)

2 - o termo *partida* substitui aqui os termos *iniciar* e *inicializar* – sendo que este último não é correto em Português.

/usr/bin, /usr/sbin, /usr/lib, usr/share/man, usr/share/doc.³

Os três primeiros diretórios são análogos aos diretórios previamente mencionados, porém seu conteúdo só é solicitado após o sistema estar carregado e rodando normalmente – já que na partida do sistema /usr pode não estar ainda montado.

A documentação do sistema, na forma de *man pages* (páginas de manuais) pode ser encontrada em /usr/share/man, enquanto outras documentações estão em /usr/share/doc.

/usr/local/bin, /usr/local/sbin, /usr/local/lib, etc.

Os conteúdos dos diretórios /bin, /sbin, /lib e os de seus correspondentes dentro de /usr são de costume aí localizados como parte do processo de instalação do sistema. Em distribuições baseadas nos pacotes RPM, como Red Hat, SuSE e outras, os programas instalados por esses pacotes vão parar ali e sua documentação vai em /usr/share/{man, doc}.

Entretanto, um sistema Linux típico terá também alguns programas instalados a partir de código-fonte em *tarballs*. Neste caso, seus arquivos vão para /usr/local.

Dica: Colocar /usr/local em sua própria partição, evitará sua formatação ao reinstalar o sistema.

/boot.

Ao dar partida, o sistema encontra uma situação de 'ovo ou galinha': o acesso ao *kernel* e arquivos relacionados vai requerer que o arranjo do sistema de arquivos do Linux seja entendido, mas o *kernel* e os *drivers* dos sistemas de arquivos têm que ser carregados daqueles mesmos sistemas de arquivos que a BIOS da máquina não compreende.

A BIOS lê os 512 bytes do MBR (registro de boot principal) do HD e os executa. Se o sistema é Linux, ali estará o carregador de *boot* que pode ser o lilo (*linux loader*) ou GRUB (*Grand Unified Bootloader*). Esse programa então irá carregar uma imagem do *kernel* na RAM e daí o sistema continuará com a seqüência de carregamento.

/etc.

A árvore de diretórios /etc contém os arquivos de configuração para o sistema, tais como *inittab* e *fstab* que são usados no momento da partida do sistema para ajustar o *runlevel* alvo e especificar os pontos de montagem para os sistemas de arquivos adicionais. Por isso, /etc não pode estar em uma partição separada – ao contrário, tem que ser um diretório dentro da partição raiz.

/home.

Os diretórios pessoais dos usuários são normalmente encontrados em /home, com exceção da conta do *root*, cuja pasta /home fica dentro do diretório /root, localizado na partição raiz do sistema. Se a pasta pessoal do usuário *root* estiver no diretório /home, que não está montado quando o sistema dá a partida no modo de manutenção e usuário único, a conta *root* não conseguiria fazer seu *login* corretamente. Frequentemente, /home é uma partição separada, por uma variedade de razões, listadas adiante.

³ - usr é abreviatura de *Unix System Resources*, Recursos do Sistema Unix. Os nomes dos subdiretórios são separados por barras. Ex.: /usr/bin é o subdiretório bin dentro do diretório usr. Não é necessário colocar uma barra ao final de um nome de subdiretório.

/var.

Durante sua operação normal, um sistema típico produz uma série de arquivos que devem ser preservados, mesmo depois do sistema ser reiniciado. Podem ser arquivos de log (registro de atividades), arquivos de filas (*spool file*⁴) de documentos enviados às impressoras, arquivos de filas de email e semelhantes. Estes são, quase sempre, colocados no diretório /var.

Além disso, aplicativos que compartilham dados para usuários diferentes tipicamente colocam seus arquivos em /var.

/tmp.

Esta é a localização de arquivos temporários produzidos por programas – tais como compiladores – enquanto rodam. Em geral, são arquivos temporários de vida curta, úteis para um programa apenas enquanto ele está rodando, e assim, o conteúdo de /tmp não é normalmente preservado ao se reiniciar o sistema. Na verdade, os *scripts* de partida de alguns sistemas limpam o conteúdo de /tmp.

/opt.

Este diretório destina-se aos subsistemas de aplicativos de grande porte de terceiros, tais como Open Office, Lotus Domino, IBM DB2, etc.

Outros Conteúdos.

lost+found.

Em todos⁵ os sistemas de arquivos encontramos um diretório lost+found (perdidos+achados) na raiz do sistema. Quando o fsck (file system check – verificador do sistema de arquivos) é executado em um sistema não-montado, quaisquer dados recuperados serão colocados em arquivos e gravados no diretório lost+found. A existência de um subdiretório lost+found é uma boa indicação de que o diretório onde está contém um ponto de montagem para uma partição.

/dev.

O subdiretório /dev contém arquivos especiais, que apontam para os diversos dispositivos de entrada e saída do sistema. Como o acesso a esses dispositivos é necessário na partida do sistema, ele fica no diretório raiz.

Como particionar o(s) disco(s).

São muitas as razões para formatar um sistema em diferentes partições de disco.

4 - *spool file*: Qualquer arquivo ao qual dados são incorporados para esperar o próximo estágio de processamento. Especialmente usados em circunstâncias onde a colocação de dados em uma fila é feita em função de diferenças de velocidade de processamento entre dois dispositivos ou programas. Por exemplo: quando você envia emails no Unix, eles são copiados para um arquivo de fila e esperam a atenção de um daemon de transporte.

5 - Por acaso, estou traduzindo este texto em Open Office Writer, rodando no Ubuntu 7.10 e não encontrei /lost+found na árvore de diretórios do Ubuntu. (N. do T.)

Para evitar sobrescrever quando se faz upgrade ou reinstalação.

Os programas ou *scripts* de instalação do Linux oferecem a chance de particionar o(s) disco(s) e especificar os pontos de montagem para as várias partições. Você vai notar, ao instalar sobre uma versão pré-existente, que o instalador do Linux mostra a opção 'Format:' e coloca como *default* 'Yes' para algumas partições e 'No' para outras. Em geral a raiz (/, incluindo /etc, /root, /bin, e outras) e /usr serão reformatadas; outras partições são deixadas intactas.

Em particular, se /home é uma partição separada, ela provavelmente não será reformatada por *default*, de modo que os arquivos de usuários serão preservados. Sempre tenha o cuidado de verificar quais partições serão reformatadas pelo instalador do Linux.

Subdiretórios que você gostaria de colocar em partições separadas – para evitar perdas de conteúdo durante upgrades, incluem /home, /usr/local e /usr/src.

Desempenho.

Em uma máquina com um sistema pesado – tendo múltiplos usuários, ou múltiplos servidores de rede, ao realizar várias tarefas ao mesmo tempo – um único *driver* terá sua cabeça de leitura-escrita saltando constantemente entre arquivos de programas, arquivos de dados e a partição *swap*. Tenha em mente que o uso de múltiplas partições ou sistemas de arquivos tende a tornar esta situação ainda pior se arquivos de dados e de programas usados em conjunto estão em partições separadas. Entretanto, esse efeito pode ser diminuído com planejamento do leiaute das partições de modo a manter próximas as partições mais freqüentemente usadas.

Em sistemas pesados, um meio de melhorar o desempenho é usar mais de um HD. Com discos separados para *swap*, /var, /tmp e /home as cabeças de leitura e escrita de cada disco permanecerão mais tempo sobre os arquivos mais usados por longos períodos, minimizando os movimentos e maximizando o desempenho. Além disso, em particular com *drivers* SCSI, ciclos de leitura e escrita podem ser feitos de maneira superposta, aumentando mais ainda o desempenho.

Por outro lado, tenha cuidado para não colocar diretórios vitais em discos diferentes. Quanto mais discos houver, e se houver em cada um deles um diretório que seja solicitado durante a partida do sistema, maior a probabilidade de você ter que fazer manutenções em baixo nível.

Backup.

O comando *dump* tem como finalidade realizar backups completos e incrementais de um sistema de arquivos. Sua operação é parcialmente controlada pela coluna *fs_freq* do arquivo /etc/fstab, que indica quais os sistemas de arquivos precisam ter seus backups atualizados, junto com o arquivo /etc/dumpdates, que grava as datas em que os vários níveis de *dump* foram feitos nos diferentes sistemas de arquivos.

O comando *dump* só opera em sistemas de arquivos completos e não pode operar sobre subdiretórios. Portanto, árvores de diretórios que tenham de ser tratadas individualmente são candidatas à criação de seus próprios sistemas de arquivos. A candidata mais óbvia para este caso é /home, mas /var deveria também ser considerada.

Sistemas de Arquivos com Diferentes Formatos.

O sistema de arquivos padrão para Linux é o ext3⁶, sistema de terceira extensão. Entretanto, muitas distribuições provêm suporte para outros formatos, tais como ReiserFS, xfs, jfs, e é claro, ext2 – embora já obsoleto. Cada um tem suas combinações de benefícios e custos, e o administrador sábio vai usá-los de modo judicioso. Por exemplo, ReiserFS usa B+trees – árvores B+, para seus diretórios o que o torna uma boa escolha para diretórios que possam conter milhares de arquivos em um único subdiretório, tal como em um servidor de notícias.

Quotas.

O *kernel* do Linux suporta a imposição de quotas de disco por usuário. Quotas devem ser estabelecidas por sistemas de arquivos – ou partições. Uma vez que a partição /home é obviamente a maior candidata a ter esse sistema de quotas, essa é uma outra razão para que ela ocupe uma partição em separado.

Gerenciamento de Espaço.

Ficar sem espaço em disco é algo que acontece a todos nós em algum momento. Neste caso, uma solução fácil é colocar um segundo (ou terceiro, ou...) HD, e mudar o sistema de arquivos sem espaço para esse novo disco.

Segurança.

Algumas vezes, a segurança pode ser incrementada colocando-se um subdiretório em um sistema de arquivos próprio. Dois exemplos: primeiro - /usr pode ser montado usando a opção *ro* (*read-only*). Isso dificulta um pouco mais para um invasor gravar arquivos ou mexer no diretório. Segundo: se você permitir *uploads* em FTP anônimo – o que não é uma boa idéia, mas supomos que você tenha lá suas razões para isso, então pelo menos coloque /var/ftp/pub/incoming em um sistema de arquivos separado, para evitar que um invasor encha um outro sistema de arquivos carregando o conteúdo de /dev/zero nele.

Pesando todos esses fatores, pode-se chegar a vários arranjos diferentes. Eu, em particular, resolvo o problema com um arranjo simples nos meus *notebooks*: uma partição /home separada para evitar perda de trabalhos quando faço *upgrade*, com tudo mais na partição raiz. Entretanto, meu *desktop* e servidor têm arranjos mais complexos, com vários discos rígidos, partições /usr, /var, /tmp e /opt separadas. E, é claro, para os usuários mais avançado, existem as técnicas avançadas, tais como RAID⁷ e LVM⁸. Mas isso fica para outro artigo.

Runlevel	Descrição
0	Desliga o sistema – pára todos os <i>daemons</i> , faz <i>logout</i> de todos os usuários, desmonta os sistemas de arquivos e desliga a máquina.
1	Modo de manutenção com usuário único.
2	Modo multi-usuário, com suporte a rede mas nenhum <i>daemon</i> carregado.
3	Modo multi-usuário, com todo o suporte a rede ativado.
5	Modo multi-usuário, com todo o suporte a rede ativado e <i>login</i> gráfico com o servidor X iniciado.
6	Reinicia.

Tabela 1: Runlevels.

6 - quando o artigo foi escrito originalmente, ainda era o ext2. (N. do T.)

7 - RAID = Redundant Array of Independent Disks. Arranjo Redundante de Discos Independentes.

8 - LVM = Logical Volume Management. Gerenciamento de Volumes Lógicos.

Movendo Sistemas de Arquivos.

Algumas vezes o crescimento de um sistema de arquivos – particularmente /home – significa que ele deve ser mudado para uma nova localização. Em outras palavras, colocar um novo disco e realocar o sistema de arquivos para seu novo lugar, onde terá espaço para crescer.

Abaixo, o procedimento para adicionar um novo disco, com uma partição única que será o novo sistema de arquivos /home (assumindo que o fdisk já foi usado para para particioná-lo).

Como raiz (*root*):

```
# mkdir /mnt/newhome
# mkfs -t ext2 /dev/hdb1
# mount /dev/hdb1 /mnt/newhome
# (cd /home && tar cf - .) | (cd /mnt/newhome && tar xpf -)
```

então:

```
# cd /
# mv /home /home.old
# mkdir /home
# umount /mnt/newhome
# mount /dev/hdb1 /home
```

Uma vez que o novo diretório /home tenha sido verificado, você pode seguramente rodar:

```
# cd /home.old
# rm -rf *
# cd ..
# rmdir /home.old
# rmdir /mnt/newhome
```

para limpar. Finalmente, você pode editar o arquivo /etc/fstab e adicionar a linha para montar o novo sistema de arquivos durante a partida do sistema.

```
/dev/hdb1 /home ext3 defaults 1 2
```

deve resolver.

Removendo arquivos antigos.

Ao longo do tempo, arquivos antigos e desnecessários acumulam-se no sistema, tais como:

- Arquivos de backup (terminados em ~ ou .bak);
- Arquivos temporários tais como editores de swap (terminados em .swp);
- Arquivos de núcleo do *dump* (se *ulimit* é non-zero).

Use o comando *find* para encontrar e remover esses arquivos. Por exemplo:

```
find / -type f -atime +7 \( -name '*.~' -o -name '*~' -o -name '*.bak' \) -exec
rm -f {} \;
```

Copyright © 1987–2005 Les Bell and Associates Pty Ltd. All rights reserved.

Nota: Este texto, originalmente em inglês, foi traduzido do site de Les Bell e Associados:

<http://www.lesbell.com.au/Home.nsf/b8ec57204f60dfcb4a2568c60014ed0f/87cd60bdf18a2a75ca256ca-f001594b4?OpenDocument>

Todos os direitos sobre este texto pertencem aos seus proprietários...