

 **Bluemix** Desenvolva na nuvem em um clique!

Comece seu trial gratuito

developerWorks Brasil Itens Técnicos Linux Biblioteca técnica

Learn Linux, 101: Criando partições e sistemas de arquivos

Divida e conquiste seu espaço em disco

Aprenda como criar partições em uma unidade de disco e como formatá-las para uso em um sistema Linux® como espaço de troca ou de dados. Use o material deste artigo para se preparar para o exame LPI 101 de certificação de administrador de sistema Linux ou simplesmente para aprender sobre partições e sistemas de arquivos do Linux para seu próprio uso.

Ian Shields trabalha em vários dos projetos Linux para a zona Linux do developerWorks. Ele é um programador senior da IBM em Research Triangle Park, NC. Ele iniciou na IBM em Canberra, Austrália, como um Engenheiro de Sistemas em 1973 e, desde então, trabalhou em sistemas de comunicações e computação disseminada em Montreal, no Canadá, e em RTP na Carolina do Norte, nos Estados Unidos. Ele possui diversas patentes. Sua graduação é em matemática pura e filosofia na Australian National University. Ele possui mestrado em ciências e é doutor em ciências da computação na Universidade do Estado da Carolina do Norte.

30/Jul/2010

Visão geral

Neste artigo, aprenda sobre partições de disco e sistemas de arquivos do Linux. Aprenda a:

Criar uma partição

Usar `mkfs` comandos para definir arquivos de sistema `ext2`, `ext3`, `xfs`, `reiserfs v3` e `vfat`

Criar e gerenciar espaço de troca

Este artigo ajuda na preparação para o Objetivo 104.1 no Tópico 104 do exame 101 do Linux Professional Institute Junior Level Administration (LPIC-1). Esse objetivo tem peso 2.

Pré-requisitos

Para aproveitar os artigos desta série ao máximo, é necessário ter conhecimento básico de Linux e possuir um sistema Linux funcional em que seja possível praticar os comandos aqui abordados.

Algumas vezes, é possível que versões diferentes de determinado programa formatem a saída de modo diferente, assim, pode ser que seus resultados não sejam exatamente iguais às listas e figuras exibidas neste documento.

Também é preciso estar familiarizado com o material em nosso artigo "[Learn Linux 101: Layout do disco](#)"

Sobre esta série

Esta série de artigos destina-se a elucidar as tarefas de administração de sistema Linux. Também é possível usar os materiais nestes artigos como preparação para os exames [Linux Professional Institute Certification nível 1 \(LPIC-1\)](#).

Consulte o nosso [roteiro do developerWorks para o LPIC-1](#) para obter a descrição e o link para cada um dos artigos nesta série. O roteiro está em andamento e reflete os últimos objetivos (abril de 2009) dos exames LPIC-1: na medida em que formos concluindo estes artigos, eles serão adicionados ao roteiro. Enquanto isso, é possível encontrar versões anteriores do mesmo material, que dão suporte aos objetivos do LPIC-1 anteriores a abril de 2009, em nossos [tutoriais de preparação para o exame de certificação LPI](#).

rígido".

Dispositivos de bloco e partições

No artigo "[Learn Linux, 101: Aprenda Linux, 101: Layout do disco rígido](#)," introduziu layouts e unidade de disco rígido e fundamentos do uso do comando `fdisk` para visualizar informações de partição. Você aprendeu sobre partições, incluindo partições *primárias*, *estendidas* e *lógicas*, e também que um sistema de arquivos Linux contém *arquivos* que são dispostos em um disco ou outro *dispositivo de armazenamento de bloco* em *diretórios*. Assim como com muitos outros sistemas, os diretórios em um sistema Linux podem conter outros diretórios chamados de *subdiretórios*. Aquele artigo também discutiu as considerações para guiá-lo ao fazer escolhas sobre particionamento.

Este artigo inicia revendo dispositivos de bloco e partições e depois apresenta o comando `fdisk`, que é usado para criar, modificar ou excluir partições em dispositivos de bloco. Também apresenta as várias formas do comando `mkfs` (`mkfs` significa *make filesystem*); comandos `mkfs` são usados para formatar partições como um tipo específico de sistema de arquivos.

Observação: Além das ferramentas e sistemas de arquivos exigidos para o exame do LPI, você pode encontrar ou precisar de outras ferramentas e sistemas de arquivos. Veja um breve resumo de algumas outras ferramentas disponíveis em [Outras ferramentas e sistemas de arquivos](#).

Dispositivos de bloco

Um *dispositivo de bloco* é uma camada de abstração para qualquer dispositivo de armazenamento que possa ser formatado em *blocos* de tamanho fixo; blocos individuais podem ser acessados independentemente de acesso a outros blocos. Esse acesso normalmente é chamado de *acesso aleatório*.

A camada de abstração de blocos de tamanho fixo acessados de forma aleatória permite que os programas usem esses dispositivos de bloco sem se preocuparem se o dispositivo adjacente é uma unidade de disco rígido, disquete, CD, unidade de estado sólido, unidade de rede ou algum tipo de dispositivo virtual, como um sistema de arquivo em memória.

Exemplos de dispositivos de bloco incluem a primeira unidade de disco rígido IDE no seu sistema (`/dev/sda` ou `/dev/hda`) ou a segunda unidade SCSI, IDE ou USB (`/dev/sdb`). Use o comando `ls -l` para exibir entradas `/dev`. O primeiro caractere em cada linha de saída é **b** de dispositivo de **bloco**, como disquete, unidade de CD, disco rígido IDE ou disco rígido SCSI; e **c** de dispositivo de **caractere**, como um terminal (`tty`) ou dispositivo null. Consulte os exemplos na Listagem 1.

Listagem 1. Dispositivos de bloco e de caractere do Linux

```
[ian@echidna ~]$ ls -l /dev/loop1 /dev/null /dev/sd[ab] /dev/sr0 /dev/tty0
brw-rw----. 1 root disk 7, 1 2010-06-14 07:25 /dev/loop1
crw-rw-rw-. 1 root root 1, 3 2010-06-14 07:25 /dev/null
brw-rw----. 1 root disk 8, 0 2010-06-14 07:25 /dev/sda
brw-rw----. 1 root disk 8, 16 2010-06-14 07:25 /dev/sdb
brw-rw----+ 1 root cdrom 11, 0 2010-06-14 07:25 /dev/sr0
crw--w----. 1 root root 4, 0 2010-06-14 07:25 /dev/tty0
```

Partições

Entre em contato com Ian

Ian é um dos nossos autores mais populares e produtivos. Pesquise [todos os artigos do Ian](#) no developerWorks. Confira [o perfil de Ian](#) e entre em contato com ele, com outros autores e leitores no My developerWorks.

Em alguns dispositivos de bloco, como disquetes e discos de CD ou DVD, é comum usar a mídia inteira como um único sistema de arquivos. No entanto, em discos rígidos grandes, e até em chaves de memória USB, é mais comum dividir, ou particionar, o espaço disponível em várias *partições* diferentes.

As partições podem ter tamanhos diferentes, e partições diferentes podem ter nelas diferentes sistemas de arquivos, de modo que um único disco pode ser usado para muitas finalidades, incluindo compartilhá-lo entre múltiplos sistemas operacionais. Por exemplo, uso sistemas de teste com várias distribuições diferentes do Linux e às vezes um sistema Windows®, todos compartilhando um ou dois discos rígidos.

Você lembrará o artigo "[Learn Linux 101: Aprenda Linux, 101: Layout do disco rígido](#)", que discos rígidos têm uma *geometria*, definida em termos de cilindros, cabeçotes e setores. Embora as unidades modernas usem *logical block addressing (LBA)*, que torna a geometria em grande parte irrelevante, a unidade de alocação fundamental para fins de particionamento ainda é, geralmente, o cilindro.

Exibindo informações de partição

As informações de partição são armazenadas em uma *tabela de partições* no disco. A tabela lista informações sobre o início e o fim de cada partição, informações sobre seu *tipo*, e se está marcada como inicializável ou não. Para criar e excluir partições, edite a tabela de partição usando um programa especialmente criado para a tarefa. Para o exame LPI, é preciso conhecer o programa `fdisk`, portanto ele será abordado aqui, embora existam várias outras ferramentas. Mencionaremos algumas delas no final deste artigo.

O comando `fdisk` com a opção `-l` é usado para listar partições. Adicione um nome de dispositivo, como `/dev/sda`, se quiser verificar as partições em uma unidade específica. Observe que as ferramentas de particionamento exigem acesso do root. A Listagem 2 exibe as partições nas unidades de disco rígido primárias de dois de meus sistemas.

Listagem 2. Listando partições com fdisk

```
[root@attic4 ~]# fdisk -l /dev/sda
Disk /dev/sda: 640.1 GB, 640135028736 bytes
255 heads, 63 sectors/track, 77825 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00064a1a

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1                1          127       1020096   83  Linux
/dev/sda2                128         1402      10241437+  82  Linux swap / Solaris
/dev/sda3 *            46340         56538       8192000    83  Linux
/dev/sda4                1403         46339      360956422   5  Extended
/dev/sda5                1403         10420       72437053+  83  Linux
/dev/sda6               10421         19344       71681998+  83  Linux
/dev/sda7               19345         28350       72340663+  83  Linux
/dev/sda8               28351         37354       72324598+  83  Linux
/dev/sda9               37355         46339       72171981   83  Linux

Partition table entries are not in disk order

[root@echidna ~]# fdisk -l /dev/sda
Disk /dev/sda: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000de20f

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1 *                1          9111       73184076   7  HPFS/NTFS
/dev/sda2                9634         9730         779152+  83  Linux
/dev/sda3                9731        116679      859067842+  5  Extended
/dev/sda5                9731         20917         89859546   83  Linux
/dev/sda6               20918         39644        150424596   83  Linux
/dev/sda7               39645         53905        114551451   83  Linux
```

Observações:

1. As informações dos cabeçalhos mostram o tamanho do disco e a geometria. A maioria dos discos grandes que usam LBA possui 255 cabeçotes por cilindro e 63 setores por trilha, em um total de 16.065 setores, ou 8.225.280 bytes por cilindro.
2. Neste segundo exemplo, a primeira partição primária (`/dev/sda1`) é marcada como *inicializável* (ou *ativa*). Isto permite que o registro de inicialização padrão DOS do PC inicialize a partição. Essa sinalização não possui significado para os loaders de boot LILO ou GRUB. O primeiro exemplo usa o GRUB como loader de boot, e o fato de que `/dev/sda3` está marcado como inicializável é, provavelmente, um acidente do histórico de uso desta unidade.
3. As colunas *Start* e *End* mostram o cilindro que inicia e termina cada partição. Eles não devem se sobrepor e geralmente devem ser adjacentes, sem espaço interveniente.
4. A coluna *Blocks* mostra o número de blocos de 1 K (1024 bytes) na partição. Para a maioria dos discos em uso no momento em que este artigo foi escrito, o tamanho do setor é de 512 bytes, portanto o número máximo de blocos na partição é metade do produto do número de cilindros ($\text{Final} + 1 - \text{Inicial}$) e o número de setores por cilindro. Um sinal de rastreio + indica que nem todos os setores na partição estão sendo usados.
5. O campo *Id* indica o uso pretendido da partição. O tipo 82 é uma partição de troca do Linux, e o tipo 83 é uma partição de dados do Linux. Há cerca de 100 diferentes tipos de partição definidos. O segundo disco é compartilhado por vários sistemas operacionais, incluindo Windows/XP, daí a presença de partições Windows NTFS (e possivelmente FAT32).

Particionando com fdisk

Você acabou de ver como exibir informações de partição usando o comando `fdisk`. Esse comando também fornece um ambiente com base em menus para editar a tabela de partição para criar ou remover partições.

Avisos

Antes de começar a modificar partições, há coisas importantes a serem lembradas. Você corre o risco de **perder seus dados existentes** se não seguir essas diretrizes.

1. **Faça backup dos dados importantes antes de começar**, assim como em qualquer operação que possa causar perda de dado.
2. **Não altere partições que estão em uso**. Planeje suas ações e execute-as cuidadosamente. Inicializar uma distribuição live a partir de CD, DVD ou USB é uma boa forma de garantir que nenhuma unidade de disco rígido esteja em uso.
3. **Conheça sua ferramenta**. O comando `fdisk` não executa nenhuma modificação no seu disco até que você autorize. Outras ferramentas, incluindo `parted`, poderão executar alterações imediatamente.
4. **Pare se cometer um erro**. Ferramentas de particionamento gravam a tabela de partição. A menos que a ferramenta que estiver usando também inclua a capacidade de mover, redimensionar, formatar ou gravar de outro modo na área de dados do seu disco, seus dados não serão tocados. Se cometer algum erro, pare o mais rápido possível e procure ajuda. Você ainda poderá ser capaz de restaurar suas definições anteriores da tabela de partição e, portanto, recuperar suas partições e dados.

Iniciar fdisk

Para iniciar o `fdisk` em modo interativo, simplesmente forneça o nome de um disco, como `/dev/hda` ou `/dev/sdb`, como parâmetro. O exemplo a seguir inicializa um live DVD do Knoppix. Você precisará de autorização de root e verá uma saída similar à da Listagem 3.

Listagem 3. Iniciando o fdisk interativo

```
knoppix@Microknoppix:~$ su -
root@Microknoppix:~# fdisk /dev/sda

The number of cylinders for this disk is set to 121601.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
 1) software that runs at boot time (e.g., old versions of LILO)
 2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):
```

A maioria dos discos modernos tem mais de 1024 cilindros, portanto você verá, normalmente, o aviso exibido na Listagem 3. Digite `m` para exibir uma lista de comandos de uma letra disponíveis, como mostra a Listagem 4.

Listagem 4. Ajuda no fdisk

```
Command (m for help): m
Command action
 a  toggle a bootable flag
 b  edit bsd disklabel
 c  toggle the dos compatibility flag
 d  delete a partition
 l  list known partition types
 m  print this menu
 n  add a new partition
 o  create a new empty DOS partition table
 p  print the partition table
 q  quit without saving changes
 s  create a new empty Sun disklabel
 t  change a partition's system id
 u  change display/entry units
 v  verify the partition table
 w  write table to disk and exit
 x  extra functionality (experts only)

Command (m for help):
```

Use o comando `p` para exibir a partição existente nesse disco específico; a Listagem 5 mostra a saída.

Listagem 5. Exibindo a tabela de partição existente

```
Command (m for help): p

Disk /dev/sda: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000de20f

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *            1           9111     73184076   7  HPFS/NTFS
/dev/sda2                9634         9730         779152+   83  Linux
/dev/sda3                9731        116679     859067842+   5  Extended
/dev/sda5                9731         20917         89859546   83  Linux
/dev/sda6               20918         39644        150424596   83  Linux
/dev/sda7               39645         53905        114551451   83  Linux

Command (m for help):
```

Este disco em particular possui 1 TB com uma partição do Windows/XP de aproximadamente 80GB. Trata-se de uma partição primária, e é marcada como inicializável, como é típico de um sistema Windows.

Adicionando partições

Agora, usaremos parte do espaço livre para adicionar algumas partições.

1. Criaremos uma partição de troca como `/dev/sda4`. Ela será a partição primária, preenchendo o espaço de 521 cilindros entre o final de `/dev/sda1` e o início de `/dev/sda2`. Nem comece a imaginar que coisas malucas fizeram com que este espaço existisse; criei o espaço intencionalmente para que pudesse

escrever este artigo.

2. Criaremos uma partição lógica de 40 GB como /dev/sda8.
3. Finalmente, criaremos uma pequena partição lógica de 2.000MB para compartilhamento de dados entre os sistemas Linux e Windows. Ela será formatada eventualmente como FAT32 (ou vfat). Ela será /dev/sda9.

Criando nossas partições

Vamos começar usando o comando `n` para criar uma nova partição; consulte a Listagem 6.

Listagem 6. Criando nossa primeira partição

```
Command (m for help): n
Command action
  l   logical (5 or over)
  p   primary partition (1-4)
p
Selected partition 4
First cylinder (9112-121601, default 9112):
Using default value 9112
Last cylinder, +cylinders or +size{K,M,G} (9112-9633, default 9633): +521

Command (m for help): p

Disk /dev/sda: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000de20f

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1          9111     73184076   7   HPFS/NTFS
/dev/sda2                9634         9730       779152+   83  Linux
/dev/sda3                9731       116679     859067842+   5   Extended
/dev/sda4                9112         9633       4192965   83  Linux
/dev/sda5                9731        20917       89859546   83  Linux
/dev/sda6               20918       39644       150424596   83  Linux
/dev/sda7               39645       53905       114551451   83  Linux

Partition table entries are not in disk order

Command (m for help):
```

Usamos o padrão para o primeiro cilindro e especificamos o valor de +521 para o número de cilindros. É possível observar na Listagem 6 que nossa partição possui um tamanho de aproximadamente 4 GB. Como é uma partição primária, ela deverá ser numerada de 1 a 4. É uma boa ideia atribuir números de partição sequencialmente; algumas ferramentas reclamam se isto não for feito e o `fdisk` avisa que nossas entradas na tabela de partições não estão mais em ordem no disco.

Observe também que foi atribuído à nossa partição um tipo de 83, para uma partição de dados do Linux. Pense nisso como um indicador para o sistema operacional do uso pretendido da partição. O uso eventual deve corresponder a isso, mas neste ponto nem temos ainda a partição formatada, muito menos dados nela. Criaremos nossas outras partições primeiro e, a seguir, veremos como mudar o tipo de partição.

Pode-se notar que, quando inserimos o subcomando `n` para criar uma nova partição, as únicas escolhas eram 'l' para lógica e 'p' para primária. Você verá somente as opções para os tipos possíveis remanescentes de partições. Você veria 'e' para estendido se a unidade já não tivesse uma partição estendida. Note também que nossa partição estendida (/dev/sda3) é do tipo 5.

Agora vamos definir a partição principal do Linux com 40 GB e a partição FAT32 com 2000 MB. Desta vez, iremos simplesmente especificar tamanhos de +40G e +2000M, indicando 40GB e 2.000MB, respectivamente. Deixamos que o `fdisk` calcule o número de cilindros para nós. Os resultados são exibidos na Listagem 7.

Listagem 7. Criando nossas partições de dados

```

Command (m for help): n
First cylinder (53906-116679, default 53906):
Using default value 53906
Last cylinder, +cylinders or +size{K,M,G} (53906-116679, default 116679): +40G

Command (m for help): n
First cylinder (59129-116679, default 59129):
Using default value 59129
Last cylinder, +cylinders or +size{K,M,G} (59129-116679, default 116679): +2000M

Command (m for help): p

Disk /dev/sda: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000de20f

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1         9111     73184076   7   HPFS/NTFS
/dev/sda2             9634         9730       779152+   83   Linux
/dev/sda3             9731     116679    859067842+   5   Extended
/dev/sda4             9112         9633       4192965   83   Linux
/dev/sda5             9731         20917     89859546   83   Linux
/dev/sda6            20918         39644    150424596   83   Linux
/dev/sda7            39645         53905    114551451   83   Linux
/dev/sda8            53906         59128     41953716   83   Linux
/dev/sda9            59129         59384     2056288+   83   Linux

Partition table entries are not in disk order

Command (m for help):

```

Alterando o tipo de partição

Antes de sairmos do `fdisk`, alteraremos os tipos de partição para as partições de troca e `vfat`. Fazemos isto usando o subcomando `t` para definir o tipo de partição. Definimos `/dev/sda4` como tipo 82 (swap do Linux) e `/dev/sda9` como tipo 9 (FAT32). Se quiser ver a lista completa de tipos suportados, insira `L`, como mostra a Listagem 8.

Listagem 8. Alterando tipos de partição

```

Command (m for help): t
Partition number (1-9): 4
Hex code (type L to list codes): 82
Changed system type of partition 4 to 82 (Linux swap / Solaris)

Command (m for help): t
Partition number (1-9): 9
Hex code (type L to list codes): b
Changed system type of partition 9 to b (W95 FAT32)

Command (m for help):

```

Salvando sua partição de dados

Até o momento, só fizemos uma edição em memória de uma tabela de partição. Poderíamos usar o comando `q` para sair sem salvar as alterações. Se alguma coisa não estiver como desejamos, poderemos usar o comando `d` para excluir uma ou mais partições, para que possamos redefini-las. Se nossa configuração estiver a contento, usaremos o comando `v` para verificar a configuração e, a seguir, o comando `w` para gravar a nova tabela de partição e sair. Consulte a Listagem 9. Se executar `fdisk -l` novamente, verá que o Linux agora reconhece as novas partições. Diferentemente de outros sistemas operacional, não é sempre necessário reinicializar para ver as alterações. Uma reinicialização pode ser necessária se, por exemplo, `/dev/hda3` se tornou `/dev/hda2` porque o `/dev/hda2` original foi excluído. Se uma reinicialização for necessária, o `fdisk` deverá dizer a você para fazer isso.

Listagem 9. Salvando a tabela de partição

```

Command (m for help): v
999521580 unallocated 512-byte sectors

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
Syncing disks.

```

```

root@Microknoppix:~# fdisk -l /dev/sda

Disk /dev/sda: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000de20f

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           9111    73184076   7   HPFS/NTFS
/dev/sda2                9634         9730     779152+   83   Linux
/dev/sda3                9731        116679    859067842+   5   Extended
/dev/sda4                9112         9633     4192965   82   Linux swap / Solaris
/dev/sda5                9731        20917     89859546   83   Linux
/dev/sda6               20918        39644    150424596   83   Linux
/dev/sda7               39645        53905    114551451   83   Linux
/dev/sda8               53906        59128     41953716   83   Linux
/dev/sda9               59129        59384     2056288+   b   W95 FAT32

Partition table entries are not in disk order

```

Mais sobre fdisk

Você pode ter notado que não alteramos a sinalização de reinicializável em nenhuma partição. Como o nosso disco se encontra agora, ele ainda possui o Windows Master Boot Record (MBR) e, portanto, reinicializará a primeira partição primária que estiver marcada como reinicializável (a partição NTFS em nosso exemplo).

Nem o LILO nem o GRUB usam sinalização de reinicializável. Se algum deles estiver instalado no MBR, então ele pode reinicializar a partição Windows/XP. Também é possível instalar LILO ou GRUB em sua partição /boot (/dev/hda2) e marcar essa partição como reinicializável e remover a sinalização de reinicializável de /dev/hda1. Deixar o MBR original pode ser útil se a máquina for mais tarde reconfigurada para ser uma máquina com Windows somente.

Também é possível usar `fdisk` para corrigir a ordem das partições na tabela, se desejar. Isto normalmente alterará os números das partições, portanto poderá ser necessário restaurar seu sistema em um sistema em funcionamento. Para fazer esta alteração, use o subcomando `f` para alternar para o modo expert `e`, a seguir, o subcomando `f` para corrigir a ordem das partições, conforme mostra a Listagem 10. Se quiser somente ver como ficaria a nova ordem das partições sem alterá-la, é possível usar o subcomando `q` para sair, como fizemos neste exemplo, em vez de gravar a tabela de partições atualizada em disco.

Listagem 10. Corrigindo a ordem da tabela de partições.

```

Command (m for help): p

Disk /dev/sda: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000de20f

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           9111    73184076   7   HPFS/NTFS
/dev/sda2                9634         9730     779152+   83   Linux
/dev/sda3                9731        116679    859067842+   5   Extended
/dev/sda4                9112         9633     4192965   82   Linux swap / Solaris
/dev/sda5                9731        20917     89859546   83   Linux
/dev/sda6               20918        39644    150424596   83   Linux
/dev/sda7               39645        53905    114551451   83   Linux
/dev/sda8               53906        59128     41953716   83   Linux
/dev/sda9               59129        59384     2056288+   b   W95 FAT32

Partition table entries are not in disk order

Command (m for help): x

Expert command (m for help): f
Done.

Expert command (m for help): r

Command (m for help): p

Disk /dev/sda: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000de20f

```



```

Device Boot      Start          End      Blocks      Id System
/dev/sda1 *        1            9111      73184076    7  HPFS/NTFS
/dev/sda2          9112         9633      4192965    82  Linux swap / Solaris
/dev/sda3          9634         9730        779152+   83  Linux
/dev/sda4          9731       116679     859067842+  5  Extended
/dev/sda5          9731       20917      89859546    83  Linux
/dev/sda6         20918       39644     150424596    83  Linux
/dev/sda7         39645       53905     114551451    83  Linux
/dev/sda8         53906       59128     41953716    83  Linux
/dev/sda9         59129       59384      2056288+   b  W95 FAT32
Command (m for help): q

```

Você viu uma forma de adicionar partições a uma estação de trabalho Linux. Outras opções são abordadas no artigo "[Aprenda Linux, 101: Encontrar e localizar arquivos de sistema.](#)"

Tipos de sistema de arquivos

O Linux suporta vários sistemas de arquivos diferentes. Ele possui pontos fortes e fracos e seu próprio conjunto de características de desempenho. Um atributo importante de um sistema de arquivos é o *journaling*, que permite uma recuperação mais rápida depois de um crash do sistema. Geralmente, um sistema de arquivos de journaling é preferido em relação a um arquivo que não seja de journaling, se você tiver a opção. Talvez você queira também considerar se seu sistema de arquivos escolhido suporta *Security Enhanced Linux* (ou SELinux). A seguir, um breve resumo dos tipos que você precisa conhecer para o exame LPI. Consulte [Recursos](#) para obter informações adicionais sobre segundo plano.

O sistema de arquivos ext2

O sistema de arquivos ext2 (também conhecido como o *segundo sistema de arquivos estendido*) foi desenvolvido para abordar defeitos no sistema de arquivos Minix usado em versões anteriores do Linux. Ele foi usado intensivamente no Linux por muitos anos. Não há journaling no ext2, que foi substituído em grande parte pelo ext3.

O sistema de arquivos ext3

O sistema de arquivos ext3 adiciona capacidade de journaling a um sistema de arquivos ext2 padrão e é, portanto, um crescimento evolutivo de um sistema de arquivos muito estável. Ele oferece desempenho razoável sob a maioria das condições e ainda está sendo aprimorado. Como ele adiciona journaling no topo do sistema de arquivos ext2 comprovado, é possível converter um sistema de arquivos ext2 em ext3 e até mesmo converter de volta se necessário.

O sistema de arquivos ReiserFS

O ReiserFS é um sistema de arquivos baseado em árvore B que possui um bom desempenho geral, especialmente para um grande número de arquivos pequenos. O ReiserFS também escala bem e possui journaling. Ele não está mais em desenvolvimento ativo, não suporta SELinux e foi amplamente substituído pelo Reiser4.

O sistema de arquivos XFS

O XFS é um sistema de arquivos sem journaling. Ele vem com recursos robustos e é otimizado para escalabilidade. O XFS oculta com rigor dados em trânsito na RAM, de modo que uma fonte de alimentação ininterrupta é recomendada se você usa XFS.

O sistema de arquivos de troca

O espaço de troca deve ser formatado para uso como espaço de troca, mas geralmente não é considerado um sistema de arquivos.

O sistema de arquivos vfat

Esse sistema de arquivos (também conhecido como *FAT32*) não é gravado e perde muitos recursos necessários para uma implementação completa de um sistema de arquivos do Linux. Ele é útil para trocar dados entre os sistemas Windows e Linux já que pode ser lido pelos dois. **Não** use esse sistema de arquivos para Linux, a não ser para compartilhar dados entre o Windows e o Linux. Se você descompactar o arquivo zip ou tar de um archive do Linux em um disco vfat, você perderá permissões, como executar permissão, e perderá qualquer link simbólico que tenha sido armazenado no archive.

O sistema de arquivos ext3 é maduro e usado como o sistema de arquivos padrão em uma série de distribuições. O sistema de arquivos ReiserFS foi usado por muitos anos como o padrão em algumas distribuições, incluindo o SUSE, mas é menos usado hoje em dia.

Criando arquivos de sistema

O Linux usa o comando `mkfs` para criar sistemas de arquivo e o comando `mkswappara` para fazer espaço de troca. O comando `mkfs` é, na verdade, um front-end para diversos comandos específicos de sistema de arquivos, como `mkfs.ext3` para ext3 e `mkfs.reiserfs` para ReiserFS.

Que suporte de sistema de arquivos já se encontra instalado em seu sistema? Use o comando `ls /sbin/mk*` para descobrir. Um exemplo é mostrado na Listagem 11.

Listagem 11. Comandos de criação de sistema de arquivos

```
[ian@echidna ~]$ ls
/sbin/mk*
/sbin/mkdosfs      /sbin/mkfs.ext2      /sbin/mkfs.ntfs
/sbin/mke2fs       /sbin/mkfs.ext3      /sbin/mkfs.vfat
/sbin/mkfs         /sbin/mkfs.ext4      /sbin/mkfs.xfs
/sbin/mkfs.btrfs   /sbin/mkfs.ext4dev   /sbin/mkhomedir_helper
/sbin/mkfs.cramfs  /sbin/mkfs.msdos     /sbin/mkswap
```

Você observará várias formas de alguns comandos. Por exemplo, normalmente você verá que `mke2fs`, `mkfs.ext2` e `mkfs.ext3` são idênticos, da mesma forma que `mkreiserfs` e `mkfs.reiserfs`. Sistemas de arquivos que podem ser necessários para inicializar o sistema usarão vários arquivos idênticos com nomes diferentes. Sistemas de arquivos que não podem ser usados para o sistema de arquivos / no Linux, como vfat ou msdos, poderão usar links simbólicos.

Existem algumas opções comuns para todos os comandos `mkfs`. As opções que são específicas do tipo de sistema de arquivos que está sendo criado são passadas para o comando de criação apropriado, com base no tipo de sistema de arquivos especificado na opção `-type`. Nossos exemplos usam `mkfs -type`, mas é possível usar as outras formas diretamente com o mesmo efeito. Por exemplo, é possível usar `mkfs -type ext2`, `mk2fs` ou `mkfs.ext2`. Para as páginas de manuais de um sistema de arquivos específico, use o comando `mkfs` apropriado como nome, por exemplo, `man mkfs.ext3`. Muitos dos valores exibidos nos exemplos de saída abaixo podem ser controlados por opções ao `mkfs`.

Agora que criamos todas as nossas partições, reinicializaremos o sistema Fedora 12 e formataremos os sistemas de arquivos usando essa opção, em vez do live Knoppix DVD mais lento. É claro que é possível continuar a usar o sistema Knoppix, se desejar. Lembre-se de que é necessário ter propriedade de administrador para criar sistemas de arquivos.

Criando um sistema de arquivos ext3

Formataremos a partição `/dev/sda8` como ext3 usando o comando `mkfs`, como mostra a Listagem 12.

Listagem 12. Criando um sistema de arquivos ext3

```
[root@echidna ~]# mkfs -t ext3 /dev/sda8
mke2fs 1.41.9 (22-Aug-2009)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
2624496 inodes, 10488429 blocks
524421 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
321 block groups
32768 blocks per group, 32768 fragments per group
8176 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 20 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

Note que um diário é criado com ext3. Se desejar adicionar um journal a um sistema ext2 existente, use o comando `tune2fs` com a opção `-j`.

Uma opção útil para os sistemas de arquivo ext2 e ext3 é opção `-L` com um nome, que atribui um rótulo à partição. Ele pode ser usado em vez do nome do dispositivo ao montar sistemas de arquivo; ele fornece alguns níveis de isolamento contra alterações que podem ter de serem refletidos em vários arquivos de controle. Para exibir ou definir um rótulo para um sistema de arquivos existente ext2 ou ext3, use o comando `e2label`. Os rótulos são limitados a um tamanho máximo de 16 caracteres.

Um desenvolvimento mais recente é usar um *Universally Unique Identifier*, ou *UUID*, em vez de um rótulo. Um UUID é um identificador de 128 bits normalmente exibido como 32 dígitos hexadecimais e quadro hífen. A maioria dos sistemas de arquivos do Linux gera um UUID automaticamente quando o sistema de arquivos é formatado. Use o comando `blkid` (que não precisa de propriedade de administrador) conforme mostrado na Listagem 13 para ver o UUID para a partição que acabamos de formatar. UUIDs têm maior probabilidade de serem únicos do que rótulos e são especialmente úteis para dispositivos conectados a quente, como unidades USB.

Listagem 13. Exibindo um UUID usando blkid

```
[ian@echidna ~]$ blkid /dev/sda8
/dev/sda8: UUID="87040def-920e-4525-9c81-c585ddc46384" SEC_TYPE="ext2" TYPE="ext3"
```

Criando um sistema de arquivos XFS

Agora, reformataremos a partição que acabamos de formatar como ext3 usando o sistema de arquivos XFS. Nosso sistema Fedora 12 usa SELinux (Security Enhanced Linux), portanto deveremos especificar inodes maiores do que o padrão de 256 usando o parâmetro `-i`. O valor recomendado é 512. Note que o formatador XFS o notifica se encontrar um sistema de arquivos reconhecido já existente na partição. Note também que o UUID foi reatribuído pelo formato XFS.

Listagem 14. Criando um sistema de arquivos XFS

```
[root@echidna ~]# mkfs -t xfs -i size=512 /dev/sda8
mkfs.xfs: /dev/sda8 appears to contain an existing filesystem (xfs).
mkfs.xfs: Use the -f option to force overwrite.
[root@echidna ~]# mkfs -t xfs -f -i size=512 /dev/sda8
meta-data=/dev/sda8          isize=512    agcount=4, agsize=2622108 blks
                         =               sectsz=512   attr=2
data      =                  bsize=4096  blocks=10488429, imaxpct=25
                         =               sunit=0    swidth=0 blks
naming    =version 2        bsize=4096  ascii-ci=0
log       =internal log    bsize=4096  blocks=5121, version=2
                         =               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none            extsz=4096  blocks=0, rtextents=0
[root@echidna ~]# blkid /dev/sda8
/dev/sda8: UUID="1b6798f2-f07f-4d5e-af06-2470df37ddb3" TYPE="xfs"
```

É possível rotular um sistema XFS usando a opção `-L` com um nome. É possível usar o comando `xfs_admin` com a opção `-L` para adicionar um rótulo a um sistema de arquivos XFS existente. Use a opção `-l` de `xfs_admin` para exibir um rótulo. Diferentemente do `ext2`, `ext3` e `ReiserFS`, os rótulos XFS são limitados a um máximo de 12 caracteres.

Criando um sistema de arquivos ReiserFS

Você cria um sistema de arquivos ReiserFS usando o comando `mkfs` com a opção `-t reiserfs` ou o comando `mkreiserfs`. O ReiserFS não suporta o SELinux e está sendo substituído pelo Resier4

É possível rotular um sistema ReiserFS usando `-l` (ou a opção `--label` com um nome). É possível usar o comando `reiserfstune` para adicionar um rótulo ou exibir o rótulo em um sistema de arquivos ReiserFS existente. Os rótulos são limitados a um tamanho máximo de 16 caracteres.

Pode ser preciso instalar o pacote ReiserFS em seu sistema para usá-lo, pois ele pode não estar incluído em uma instalação padrão. Consulte as man pages ou as páginas de informações para mais detalhes.

Criando um sistema de arquivos vfat

Agora, criaremos o sistema de arquivos FAT32 (`vfat`) em `/dev/sda9`.

Listagem 15. Criando um sistema de arquivos vfat

```
[root@echidna ~]# mkfs -t vfat /dev/sda9
mkfs.vfat 3.0.9 (31 Jan 2010)
[root@echidna ~]# blkid /dev/sda9
/dev/sda9: LABEL="" UUID="CF72-99A8" TYPE="vfat"
```

UUIDs para sistemas de arquivos `vfat` são menores do que UUIDs normais e, portanto, menos prováveis de serem únicos. Se, em vez disso, quiser um rótulo, use o comando `dosfslabel`. Rótulos para partições DOS são limitados ao tamanho de 11 caracteres.

Criando o espaço de troca

Agora, criaremos o espaço de troca na partição `/dev/sda4` usando o comando `mkswap`, como mostra a Listagem 16.

Listagem 16. Criando espaço de troca

```
[root@echidna ~]# mkswap /dev/sda4
Setting up swapspace version 1, size = 4192960 KiB
no label, UUID=8f5a3a05-73ef-4c78-bc56-0e9b1bcc7fdb
```

Note que versões recentes do `mkswap` mostram o UUID gerado.

Diferentemente de sistemas de arquivos comuns, as partições de troca não são montadas. Em vez disso, elas são habilitadas usando o comando `swapon`. Os scripts de inicialização do seu sistema Linux cuidarão de habilitar automaticamente suas partições de espaço.

Outras ferramentas e sistemas de arquivo

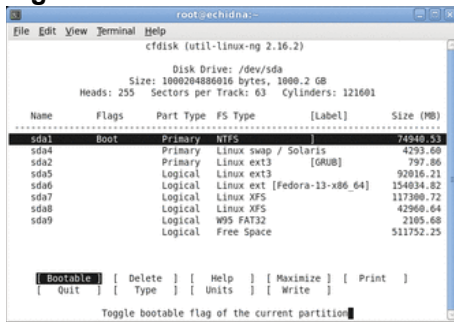
As ferramentas e sistemas de arquivo a seguir não fazem parte dos objetivos do LPI para o exame. Esta breve visão geral aborda algumas das ferramentas e sistemas de arquivos que você poderá encontrar.

Ferramentas de particionamento

Muitas distribuições do Linux incluem os comandos `cfdisk` e `sfdisk`. O comando `cfdisk` fornece uma

interface mais gráfica que o fdisk, usando as funções da biblioteca de ncurses, como mostrado na Figura 1. O comando `sfdisk` é destinado ao uso de programadores e podem ser colocados na forma de script. Só faça isso se souber o que está fazendo.

Figura 1. Usando cfdisk

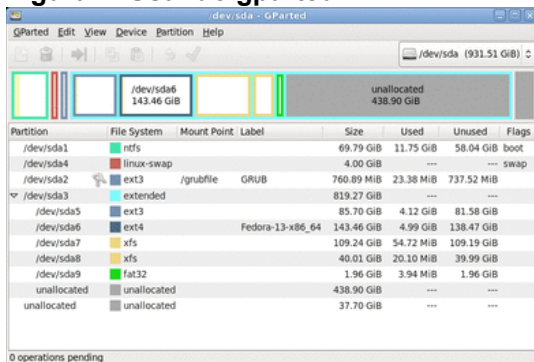


Outra ferramenta popular para trabalhar com a tabela de partição é `parted`, que pode redimensionar e formatar muitos tipos de partição, assim como criá-las e destruí-las. Embora `parted` não possa redimensionar partições NTFS, `ntfsresize` pode. A ferramenta `qtparted` usa o kit de ferramentas Qt para fornecer uma interface gráfica. Ele inclui as funções `parted`, assim como as funções `ntfsresize`.

A ferramenta `gparted` é uma outra ferramenta de particionamento gráfico, destinada ao desktop do GNOME. Ela usa a biblioteca GTK+GUI e é mostrada na Figura 2. (Consulte [Recursos](#) para obter links para `qtparted` e `gparted`.)

É possível ter que instalar os pacotes acima para usá-los, pois podem não ser parte de sua instalação padrão.

Figura 2. Usando gparted



Muitas distribuições permitem particionar o disco, às vezes encolhendo uma partição existente de Windows NTFS ou FAT32, como parte do processo de instalação. Consulte a documentação de instalação da sua distribuição.

Gerente lógico de volume

O gerente lógico de volume (ou LVM) para Linux permite combinar múltiplos dispositivos físicos de armazenamento em um único *grupo de volume*. Por exemplo, você pode adicionar uma partição a um grupo de volume existente, em vez de ter de cavar um espaço adjacente grande o suficiente para o sistema de arquivos desejado.

RAID

RAID (Redundant Array of Independent Disks) é uma tecnologia que fornece um armazenamento confiável de dados usando discos de baixo custo, que são bem menos caros que os discos encontrados em sistemas high-end. Há diversos tipos diferentes de RAID, e a RAID pode ser implementada em hardware ou software. O Linux suporta RAID de hardware e software.

Mais sistemas de arquivos

Provavelmente, você encontrará sistemas de arquivos além desses discutidos acima.

O *Journaled File System (JFS)* da IBM, atualmente usado em servidores corporativos da IBM, é destinado a ambientes de servidor de alto rendimento. Ele está disponível para Linux e está incluído em diversas distribuições. Para criar sistemas de arquivos JFS, use o comando `mkfs.jfs`.

O *btrfs* (sistema de arquivos de árvore B) foi inicialmente desenvolvido pela Oracle e está disponível sob a licença GPL. É um novo sistema de arquivos com cópia durante gravação para o Linux que objetiva implementar recursos avançados, ao mesmo tempo em que tem foco em tolerância a falhas, reparo e fácil administração. Ele poderá não ser instalado por padrão, portanto poderá ser necessário instalar um pacote, como o `btrfs-progs`, para ativar o suporte a `btrfs`.

Há outras sistemas de arquivos também, como o sistema de arquivos `cramfs` frequentemente usado em dispositivos embarcados.

Recursos

Aprender

Use o [Roteiro do developerWorks para LPIC-1](#) e localize os artigos do developerWorks que se destinam à preparação para a certificação LPIC-1, conforme os objetivos de abril de 2009.

No site [Programa LPIC](#), localize os objetivos detalhados, as listas de tarefas e as questões de amostra para os três níveis de certificação de administração de sistema Linux do Linux Professional Institute. Em particular, consulte os objetivos de abril de 2009 para o [exame LPI 101](#) e o [exame LPI 102](#). Sempre visite o site do Programa LPIC para obter os últimos objetivos.

Estude toda a [série de preparação para o exame LPI](#) no developerWorks a fim de conhecer os fundamentos do Linux e se preparar para a certificação de administrador do sistema com base nos objetivos para o exame LPI traçados antes de abril de 2009.

Visite a home do [Filesystem Hierarchy Standard \(FHS\)](#).

Saiba mais sobre o [Linux Standard Base \(LSB\)](#), um projeto do Free Standards Group (FSG) para desenvolver um ambiente operacional binário padrão.

Aprenda sobre o [sistema de arquivos btrfs](#) (sistema de arquivos árvore B).

Aprenda sobre o [sistema de arquivos Reiser4](#), o sucessor do ReiserFS.

O [Linux Documentation Project](#) apresenta uma variedade de documentos úteis, especialmente suas questões práticas.

Na [zona Linux do developerWorks](#), encontre centenas de [artigos e tutoriais](#),



Guias de capacitação

Se capacite através de diversos recursos de treinamento.



Programa IBM Champion

O programa reconhece contribuidores que estão ajudando a construir um Planeta Mais Inteligente.



Programa Global de Empreendedorismo da IBM

Faça parte do programa que busca por empreendedores que ajudam a modificar a maneira como o mundo funciona.

bem como downloads, fóruns de discussão e muitos outros recursos para desenvolvedores e administradores Linux.

Mantenha-se atualizado com os [eventos técnicos e webcasts do developerWorks](#) focados em uma variedade de produtos IBM e assuntos da indústria de TI.

Participe de uma [orientação gratuita do developerWorks Live!](#) para atualizar-se rapidamente sobre produtos e ferramentas da IBM, assim como tendências da indústria de TI.

Assista às [demos on demand do developerWorks](#) que abrangem da instalação do produto e demos de configuração para iniciantes à funcionalidade avançada para desenvolvedores experientes.

Siga o [developerWorks no Twitter](#), ou inscreva-se em um feed dos [tweets do Linux no developerWorks](#).

Obter produtos e tecnologias

[Avalie os produtos da IBM](#) da forma que melhor lhe convém: Faça o download de uma versão de teste de produto, experimente um produto on-line, use um produto em um ambiente de nuvem ou passe algumas horas no [SOA Sandbox](#) aprendendo como implementar Arquitetura Orientada a Serviço de forma eficiente.

Discutir

[Participar do fórum de discussão.](#)

Envolve-se na [comunidade My developerWorks](#). Entre em contato com outros usuários do developerWorks enquanto explora os blogs, fóruns, grupos e wikis dos desenvolvedores.