

S
F

Friday, December 7, 2007

Posted by Chris Tyler in System Recovery Week at 00:01 | Comments (0) | Trackbacks (0)

System Recovery Week: Dealing with Disk Images

This article is part of *System Recovery Week*, examining techniques used to perform maintenance or recovery on a Fedora system in special circumstances.

When performing system recovery, it is sometimes useful to capture the data from a disk drive into an *image file* which can then be manipulated on another system -- this is useful when hardware failure is imminent. You can copy a disk's contents over the network using a command such as:

```
# ssh -C user@remotehost "cat >disk1.img" </dev/sda
```

The `-C` option specifies compression, which may or may not be beneficial, depending on the speed of the network connection and the speed of the local and remote CPUs.

This resulting file (`disk1.img` in this example) is an exact copy of the data on the original disk. The same type of disk image is used for Xen and KVM virtualization.

The `file` command will report this type of file as an x86 boot sector:

```
# file disk1.img
disk1.img: x86 boot sector; partition 2: ID=0x83, starthead 0,
startsector 208845, 1012095 sectors; partition 3: ID=0x83, starthead
0, startsector 1220940, 6972210 sectors
```

You can view the partition table of the disk image using `fdisk -l`. However, to access the individual partitions within the image, you must first use `losetup` to make the image accessible as a block device:

```
# losetup -f -v disk1.img
Loop device is /dev/loop0
```

This contents of the `disk1.img` file are now available through the block device `/dev/loop0`. (The `-f` option specifies that the first available `/dev/loopN` device should be used, and the `-v` enables verbose mode so that you will be told the name of the block device). You can now use `kpartx` to create a block device for each partition within the image:

```
# kpartx -a -v /dev/loop0
add map loop0p1 : 0 208782 linear /dev/loop0 63
add map loop0p2 : 0 1012095 linear /dev/loop0 208845
add map loop0p3 : 0 6972210 linear /dev/loop0 1220940
```

The partition block devices are found in `/dev/mapper` and are named with the original block device name (`loop0`) with the letter "p" and the partition number appended. The `file -s` command will analyze the contents of each partition:

```
# file -s /dev/mapper/loop0p*
/dev/mapper/loop0p1: Linux rev 1.0 ext3 filesystem data
/dev/mapper/loop0p2: Linux/i386 swap file (new style) 1 (4K pages)
size 126510 pages
/dev/mapper/loop0p3: LVM2 (Linux Logical Volume Manager) , UUID:
5zG5aVBny87KnCdsicTz3RQbt3w37db
```

In this example, the first partition contains a filesystem which may be directly mounted (e.g., with `mkdir /mnt/x1 ; mount /dev/mapper/loop0p1 /mnt/x1`).

The second partition contains swap space, which is probably not of any interest. The third partition contains an LVM physical volume (PV), so you will need to use `vgscan` and `vgchange` to gain access to logical volumes contained therein:

Welcome!

The Fedora Daily Package exists to highlight lesser-known [Fedora Linux](#) packages each weekday-- with a special article each Wednesday taking a behind-the-scenes look at some of the configuration options and packaging details that make Fedora tick. For more information, please see the [Fedora Daily Package Welcome](#) posting.

For information on the Fedora package management system and how to install, update, and remove packages, see the postings from [Package Management Week](#) (especially [Using Yum](#)).

To suggest a future Fedora Daily Package, use the [Suggest a Package](#) box below.

[Main Page](#)

Translations

Translations of selected articles:

[fr] [Paquet Fedora du jour](#)

[ru] [Пакет дня](#)

[zh] [Allen Chen's Blog](#)



Books

Books related to Fedora, including [Fedora Linux](#) by Chris

```
# vgscan
Reading all physical volumes. This may take a while...
Found volume group "zephyr" using metadata type lvm2
Found volume group "main" using metadata type lvm2

# vgchange -ay
2 logical volume(s) in volume group "zephyr" now active
4 logical volume(s) in volume group "main" now active

# file -s /dev/mapper/zephyr-*
/dev/mapper/zephyr-home: Linux rev 1.0 ext3 filesystem data (large
files)
/dev/mapper/zephyr-root: Linux rev 1.0 ext3 filesystem data (large
files)
```

Note that in this case the discovered VG name is *zephyr*. Once these steps have been performed, the zephyr LVs can then be mounted in the usual way to provide access to their contents.

Note: If the volume group (VG) name on the disk image conflicts with the VG name on the host conflict, it may be necessary to rename the VG on the host with the `vgrename` command in order to access the VG on the disk image. However, renaming the VG on the host can be a tricky task. For this reason, it's strongly recommended that VGs be given unique names when they are originally created -- naming each system's main VG after the hostname is a good practice.

When you are done using the disk image, reverse the procedure to take the disk image out of use:

```
# vgchange -an zephyr
0 logical volume(s) in volume group "zephyr" now active
# kpartx -d /dev/loop0
# losetup -d /dev/loop0
```

Thursday, December 6, 2007

Posted by Chris Tyler in System Recovery Week at 00:01 | Comments (0) | Trackbacks (0)

System Recovery Week: Recovering RAID Devices

This article is part of *System Recovery Week*, examining techniques used to perform maintenance or recovery on a Fedora system in special circumstances.

When you boot from a Fedora installation disc and enter rescue mode, you have the option of having the filesystems from your Fedora installation mounted at `/mnt/sysimage`. If you do not select this option, or if it fails, RAID devices will not be configured for use (as is also the case with LVM, as [discussed yesterday](#)).

On a Fedora system, RAID arrays are managed by the `mdadm` utility. This program expects the RAID configuration to be available at `/etc/mdadm.conf`. In order to use `mdadm` without this configuration file, it is necessary to create a dummy configuration file. This is a multi-step process:

1. Create a dummy `mdadm` configuration file containing only the partitions to be scanned for possible RAID elements:

```
sh-3.2# echo "DEVICE /dev/[hs]d?[0-9]" >/tmp/mdadm.conf
```

2. Use `mdadm`'s scanning capability to identify any RAID arrays and array members and append that information to the dummy configuration file:

```
sh-3.2# mdadm --examine --scan --config=/tmp/mdadm.conf >>/tmp/mdadm.conf
```

3. Start the detected arrays:

Tyler (this site's editor):

- [Amazon.com](#)
- [Amazon Canada](#)

Suggest a Package

Package name (Fedora 8 repository only):












Comments:

Your name:

E-mail address:

Suggest package

Categories

-  [Administrivia](#)
-  [Artsy Tuesday](#)
-  [Focus Weeks](#)
-  [Fedora 7 Week](#)
-  [Package Management](#)
- Week
-  [System Recovery](#)
- Week
-  [Friday Fun](#)
-  [GUI Thursday](#)
-  [Productive Monday](#)
-  [Wednesday Why](#)
-  [Weekly Video Summary](#)

Go!

All categories

Chris Tyler's Blog

TechTalk Links

Sunday, November 11, 2007

Remembrance Day

Sunday, November 11, 2007

Atomic Time in Your Pocket

Saturday, November 3, 2007

```
sh-3.2# mdadm --assemble --scan --config=/tmp/mdadm.conf
mdadm: /dev/md0 has been started with 2 drives (out of 3).
```

4. You can now perform any RAID recovery tasks that are needed -- adding (or re-adding) elements to arrays, for example. To view the status of your arrays, `cat /proc/mdstat`.

If you have layered LVM on top of RAID, manually scan for and activate your volume groups as [discussed yesterday](#).

Wednesday, December 5, 2007

Posted by Chris Tyler in System Recovery Week at 00:01 | Comments (5) | Trackbacks (0)

System Recovery Week: Using LVM In Rescue Mode

This article is part of *System Recovery Week*, examining techniques used to perform maintenance or recovery on a Fedora system in special circumstances.

Logical Volume Management (LVM) is a powerful storage system layer which abstracts the logical view of storage from the actual physical layout. It is automatically configured in the default Fedora storage configuration. LVM enables you to grow and shrink volumes, add storage from new devices to existing volumes, and migrate volumes between storage devices, all without taking the system offline. In fact, the [system-config-lvm](#) package provides a convenient way to perform all of these operations using a graphical interface.

However, you can't (yet) shrink a filesystem while it is mounted. This isn't a limitation of LVM itself, but of most filesystems, including ext3. To shrink a partition that is always mounted when the system is running -- such as the root filesystem -- it's necessary to boot from another medium, which is where rescue mode comes in (see [yesterday's article](#)). It may also be necessary to use rescue mode to recover from some particularly nasty corruption or misconfiguration issues.

To do LVM and filesystem-resizing work within the Fedora rescue mode, boot the rescue mode from disc as usual, but select **Skip** when the system offers to mount your hard-disk partitions. Unfortunately, this will mean that your volume groups will not be detected or activated, so you will need to do that manually.

The commands normally used for logical volume management such as `vgdisplay`, `pvcreate`, and `lvreduce` are actually symbolic links to a single executable named `lvm`. These symbolic links are not available in rescue mode, so you must explicitly use the `lvm` command followed by the operation you wish to perform: if you wish to do a `vgscan`, for example, enter the command `lvm vgscan`.

Therefore, to discover and activate all volume groups, you must execute these commands:

```
sh-3.2# lvm vgscan
Reading all physical volumes. This may take a while...
Found volume group "VolGroup00" using metadata type lvm2
sh-3.2# lvm vgchange -ay
2 logical volume(s) in volume group "VolGroup00" now active
```

You can display information about the logical volumes using `lvm lvs` (or `lvm lvdisplay` for a more verbose display):

```
sh-3.2# lvm lvs
LV VG Attr LSize Origin Snap% Move Log Copy%
LogVol100 VolGroup00 -wi-a- 28.66G
LogVol101 VolGroup00 -wi-a- 992.00M
```

Note that the LV and VG names are not very descriptive -- which is why it's a good idea to override the default names during system installation. In this case, we know the approximate size of the LV containing the root filesystem, so we can determine that it is `LogVol100` in `VolGroup00`.

Armed with this information, you can now check (`fscck`) the filesystem, shrink the filesystem within that logical volume, and then shrink the LV. Because of the potential for rounding errors, it's best to shrink the filesystem to a size slightly smaller than the new LV size, resize the LV, and then grow the filesystem to fully fill the LV:

```
sh-3.2# e2fscck -f /dev/VolGroup00/LogVol100
```

TechTalk Links

Sunday, October 28, 2007

FSOSS 2007 Begins

Thursday, October 25, 2007

Quicksearch

Syndicate This Blog

-  RSS 0.91 feed
-  RSS 1.0 feed
-  RSS 2.0 feed
-  ATOM 0.3 feed
-  ATOM 1.0 feed
-  RSS 2.0 Comments

Blog Administration

[Open login screen](#)

Powered by



License



Original material in the *Fedora Daily Package* is licensed under a [Creative Commons Attribution-Share Alike 2.5 Canada License](#).

```

e2fsck 1.40.2 (12-Jul-2007)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking refernce counts
Pass 5: Checking group summary information
/dev/VolGroup00/LogVol100: 122967/7514560 files (0.9% non-contiguous),
1113272/7512064 blocks

sh-3.2# resize2fs /dev/VolGroup00/LogVol100 19G
resize2fs 1.40.2 (12-Jul-2007)
Resizing the filesystem on /dev/VolGroup00/LogVol100 to 4980736 (4k)
blocks.
The filesystem on /dev/VolGroup00/LogVol100 is now 4980736 blocks
long.

sh-3.2# lvm lvresize VolGroup00/LogVol100 --size 20G
WARNING: Reducing active logical volume to 20.00 GB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce LogVol100? [y/n]: y
Reducing logical volume LogVol100 to 20.00 GB
Logical volume LogVol100 successfully resized

sh-3.2# resize2fs /dev/VolGroup00/LogVol100
resize2fs 1.40.2 (12-Jul-2007)
Resizing the filesystem on /dev/VolGroup00/LogVol100 to 5242880 (4k)
blocks.
The filesystem on /dev/VolGroup00/LogVol100 is now 5242880 blocks
long.

```

Note that the operations following the first `resize2fs` do not need to be performed in rescue mode -- you can reboot the system at that point and continue after the system has booted normally. Earlier versions of the Fedora rescue mode may not set up the symbolic link `/dev/VolumeGroup/LogicalVolume`, so you may need to refer to `/dev/mapper/VolumeGroup-LogicalVolume` (e.g., use `/dev/mapper/VolGroup00-LogVol100` in place of `/dev/VolGroup00/LogVol100`).

Tuesday, December 4, 2007

Posted by Chris Tyler in System Recovery Week at 00:01 | Comments (0) | Trackbacks (0)

System Recovery Week: Rescue Mode and Reinstalling Grub

This article is part of *System Recovery Week*, examining techniques used to perform maintenance or recovery on a Fedora system in special circumstances.

When a system is too damaged to permit booting from the hard disk drive, it's necessary to boot from another medium. The Fedora installation discs support a "Rescue mode" in which the system is booted from the CD and the hard disk partitions are optionally mounted for access.

To access this mode, boot from your Fedora install media and select "Rescue installed system" from the boot menu using the arrow keys and Enter or by pressing the **R** key (if you need to edit the boot options first -- to disable ACPI, for example -- navigate to the Rescue option with the arrow keys and press Tab).



The kernel will boot from CD and the system will prompt you to select a keyboard style and language from scrollable lists of options. You will then be given the opportunity to enable the network interfaces on the system, either by entering the IP information or by using DHCP.



The system will then present a dialog stating that the rescue environment is about to find and mount the filesystems from your hard disk Fedora installation, and asks if you wish to continue. This is a critical question: if your filesystems are intact and you wish to access the data that is in them, you can select **Continue**, the default option. If you are concerned about the state of your filesystems and want to ensure that they will not be altered, but still want to access them, select **Read-Only**. If your filesystems are damaged, you have multiple Fedora installations, or you wish to perform an operation such as reducing the size of the root filesystem, choose **Skip**. After some additional messages, you will be presented with a root shell prompt.

If you have elected to continue with read/write mounting of your filesystems, all of the files from your Fedora installation should be available under `/mnt/sysimage` -- so the normal `/etc/passwd` file will be available at `/mnt/sysimage/etc/passwd`.

Although regular Fedora commands and utilities are available in rescue mode, most of them will not work because of the altered paths. You can work around this issue by temporarily changing the root directory using the `chroot` command:

```
chroot /mnt/sysimage
```

However, you need to be aware that files within the mounted Fedora filesystems will not have been updated during the rescue mode boot process, including `/etc/mtab` and `/var/log/messages`. You can compensate for this by some degree by getting the information from other places (such as `dmesg` for kernel messages and `/proc/mounts` for mount information).

If you have been forced to use rescue mode because your system's Grub bootloader code has become damaged or has been overwritten by another bootloader, you can reinstall the Grub bootloader in rescue mode:

1. Start the Grub shell with the `grub` command:

```
# grub
Probing devices to guess BIOS drives. This may take a long time.

GNU GRUB version 0.97 (640K lower / 3072K upper memory)

[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename.]
grub>
```

2. Use the `find` command to locate the partition containing the boot files by searching for `/grub/grub.conf` (or `/boot/grub/grub.conf` if that fails). Grub will report the partition using its own syntax:

```
grub> find /grub/grub.conf
(hd0,0)
```

3. Use the `root` command to configure the partition from which the boot files are to be loaded (use the partition ID from step 2):

```
grub> root (hd0,0)
Filesystem type is ext2fs, partition type 0x83
```

4. The partition ID from step 2 can be converted to a drive ID by removing the comma and partition number -- for example, the partition `(hd0,0)` is on the drive `(hd0)`. Use the `setup` command with this drive ID to install the Grub bootloader code:

```
grub> setup (hd0)
Checking if "/boot/grub/stage1" exists... no
Checking if "/grub/stage1" exists... yes
Checking if "/grub/stage2" exists... yes
Checking if "/grub/e2fs_stage1_5" exists... yes
Running "embed /grub/e2fs_stage1_5 (hd0)"... 16 sectors are embedded.
```

```
succeeded
Running "install /grub/stage1 (hd0) (hd0)1+16 p (hd0,0) /grub/stage2
/grub/grub.conf"... succeeded
Done.
```

5. Exit the Grub shell with **quit**:

```
grub> quit
#
```

You can also use rescue mode to set the root password, create alternate superuser accounts, or change or remove a boot password. Whether these are important recovery operations or a type of attack depends only on the context in which they are performed. You can slow down such an attack by configuring the system BIOS to boot only from the hard disk and installing a BIOS password, but that can be reset using a motherboard jumper in most cases. The moral of the story: if you don't have physical security, you don't have system security.

When you are finished using rescue mode, type `exit` or press Ctrl-D twice. The system will then reboot.

Monday, December 3, 2007

Posted by Chris Tyler in System Recovery Week at 00:02 | Comments (0) | Trackbacks (0)

System Recovery Week: Single-user mode

This article is part of *System Recovery Week*, examining techniques used to perform maintenance or recovery on a Fedora system in special circumstances.



```
( Minimal BIOS-like line editing is supported. For the first word, TAB
 lists possible command completions. Anywhere else TAB lists the possible
 completions of a device/filename. ESC at any time cancels. ENTER
 at any time accepts your changes.)

grub append> ro root=/dev/blkGroup00-LynXb100 rhgb quiet s
```

Figure 1 - Appending single-user mode ("s") to the system boot options.

There are times when a Fedora system will not boot normally, due to the state of the filesystem, the absence of startup files, or incorrect configuration. Most users will never encounter these circumstances, but it's important to know what to do if they arise.

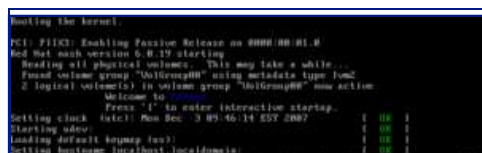
The simplest recovery mode available is "single user" or "maintenance mode". This is a special runlevel which will take you directly into a superuser (root) shell prompt without executing most of the normal system startup scripts. Because this mode boots from the normal filesystems, it will only work when the kernel, initrd (initial ramdisk), and basic filesystem are intact; however, it does not require a valid `/etc/inittab`, `/etc/passwd`, or `/etc/shadow` file or a working set of boot scripts,

so it can be used in many situations where a normal boot would fail.

To enter single user mode, interrupt the normal grub boot display (which usually shows a message counting down the seconds until Fedora is booted, or which may show a menu of available operating systems if you have altered the default grub boot configuration) by pressing the spacebar. If you have a boot password, press **P** and enter the password now.

Press the **A** key to append boot options to the default kernel, and type a space and the letter **s** to indicate that you want to enter single-user mode, as shown in Figure 1. Press Enter to continue booting.

The system will boot and then go directly to a root shell prompt (Figure 2). You can perform any normal administrative functions at this prompt - but since the normal system startup has not taken place, you will not be able to use networking,



```
Booting the kernel...
PCI: Fixup: Enabling Function Release on 0000:00:01.0
Red Hat mach version 0.8.19 starting
Reading all physical volumes. This may take a while...
Found volume group "fedora" using metadata type lvm2
  2 logical volume(s) in volume group "fedora" now active
Welcome to
Press 'I' to enter interactive startup.
Setting clock (utc): Mon Dec 3 03:46:14 EST 2007      [  0% ]
Starting udev:                                       [  0% ]
Loading default keyboard map:                         [  0% ]
Setting hostname localhost.localdomain:               [  0% ]
```

printing, or other services. If your init scripts are intact, you can start specific services, such as network or cups, using the `service` command: `service nameOfService start`

Operations commonly performed in single user mode include:

- Selecting a new root password: `passwd`
- Replacing or repairing the `/etc/inittab`, `/etc/passwd`, or `/etc/shadow` files by copying or editing the files
- Checking a filesystem which will not start up cleanly during normal boot, using a command such as this (Caution! the command as written here will proceed with all repair operations without asking further questions. This will likely result in a clean filesystem which can be mounted but may in rare cases result in some data loss): `fsck -f -y /dev/filesystemDevice`

When you are finished using single user mode, exiting the shell with the `exit` command or Ctrl-D will start a normal system boot. It's usually a better idea to perform a full reboot, using the `reboot` command.

Note that single user mode presents an extreme security risk: any person who has physical access to your system can use single user mode to gain root (unrestricted) access to your system. A boot password will make it slightly more difficult to execute this type of attack. If you did not create a boot password at installation, you can add one at any time:

1. Use the `grub-md5-crypt` command to generate an encrypted version of your selected password:

```
# grub-md5-crypt
Password: hello
Retype password: hello
$1$gNc9G$BppzXI37ogNVc2aJ8tjSe0
```

2. Enter the encrypted password into the top of your Grub configuration file, `/boot/grub/grub.conf`:

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes
to this file
# NOTICE: You have a /boot partition. This means that
# all kernel and initrd paths are relative to /boot/, eg.
# root (hd0,0)
# kernel /vmlinuz-version ro root=/dev/concord3/f8root
# initrd /initrd-version.img
#boot=/dev/md0
password --md5 $1$gNc9G$BppzXI37ogNVc2aJ8tjSe0
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora (2.6.23.1-49.fc8)
root (hd0,0)
kernel /vmlinuz-2.6.23.1-49.fc8 ro root=/dev/concord3/f8root
rhgb quiet
initrd /initrd-2.6.23.1-49.fc8.img
title Fedora (2.6.23.1-42.fc8)
root (hd0,0)
kernel /vmlinuz-2.6.23.1-42.fc8 ro root=/dev/concord3/f8root
rhgb quiet
initrd /initrd-2.6.23.1-42.fc8.img
```

However, a user with physical access to your machine can circumvent the boot password by booting from another device, as we will see later this week.

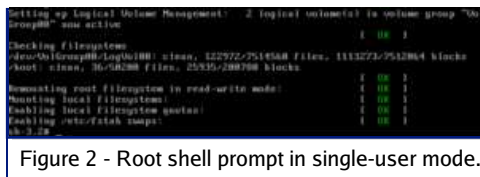


Figure 2 - Root shell prompt in single-user mode.

Focus Week: System Recovery Week

This week is a special *Focus Week* where we'll be examining the techniques used by experienced system administrators to recover a system or perform special maintenance in unusual circumstances. This will include the use of single-user mode, rescue mode (booted from optical disc), LVM and RAID recovery, and dealing with disk images from other computers.

(Page 1 of 1, totaling 6 entries)

Design [Garvin Hicking](#), Icons [Tango Project](#)