Using JFS

Presented by developerWorks, your source for great tutorials

ibm.com/developerWorks

Table of Contents

If you're viewing this document online, you can click any of the topics below to link directly to that section.

1. Tutorial tips 2
2. Introduction 4
3. Downloading and installing JFS 6
4. Using JFS 10
5. Development resources 12
6. Feedback 13

Section 1. Tutorial tips

Should I take this tutorial?

This tutorial will show you how to install and use JFS under Linux. JFS is an enterprise journalling filesystem technology currently used by IBM enterprise servers and now being ported to Linux. This tutorial assumes that you have some experience compiling your own Linux kernel. If not, please take the *Compiling the Linux kernel* tutorial on developerWorks, and then head back here.

While the JFS patches are not yet ready for use on a production server, Linux JFS has basic functionality and is ready to be tested and improved by interested developers. The JFS sources are well organized and freely available, and the JFS development team is actively seeking contributions from the Linux community. If you have the skills and desire to improve upon the kernel, JFS would be an excellent place to start.

Navigation

Navigating through the tutorial is easy:

* Use the Next and Previous buttons to move forward and backward through the tutorial.
* Use the Main menu button to return to the tutorial menu.
* If you'd like to tell us what you think, use the Feedback button.

Getting help

For technical questions about the content of this tutorial, contact the author, Daniel Robbins, at *drobbins@gentoo.org* . For questions directly relating to JFS itself, contact the JFS development team at *linuxjfs@us.ibm.com* .

Residing in Albuquerque, New Mexico, Daniel Robbins is the President/CEO of *Gentoo Technologies, Inc.* , the creator of **Gentoo Linux**, an advanced Linux for the PC, and the **Portage** system, a next-generation ports system for Linux. He has also served as a contributing author for the Macmillan books *Caldera OpenLinux Unleashed*, *SuSE Linux Unleashed* and *Samba Unleashed*. Daniel has been involved with computers in some fashion since the second grade, when he was first exposed to the Logo programming language as well as a potentially dangerous dose of Pac Man. This probably explains why he has since served as a Lead Graphic Artist at **SONY Electronic Publishing/Psygnosis**. Daniel enjoys spending time with his wife, Mary, and his new baby daughter, Hadassah.

Section 2. Introduction

What is JFS?

JFS is an advanced journalling filesystem. It has been designed to provide excellent performance and high availability for server environments. However, JFS is also well suited for Linux workstations and home machines, since everyone can benefit from a reliable filesystem with high performance.

Benefits of JFS

If you have a server that uses an ext2 filesystem, you probably know that an unexpected reboot (due to power loss, for example) can cause your filesystem(s) to be left in an inconsistent state. When the server restarts, it runs the fsck program to restore the filesystems to a usable condition. The problem with the standard ext2 filesystem is that the fsck can be a very time-consuming process, and it is certainly possible that the filesystem will become corrupted beyond repair.

Benefits of JFS, continued

JFS is designed to eliminate these problems. fsck times are dramatically reduced, since JFS keeps a transaction log of the most recent modifications to the filesystem. In the event of a failure, only these most recent transactions need to be checked by fsck. This results in a more reliable filesystem and greatly reduced downtime. JFS also uses other methods to ensure that the filesystem is kept in a consistent state.

What can Linux JFS do right now?

When the JFS developers release a new version of the Linux JFS code, this new version is called a "drop." At the time this tutorial is being written, there have been seventeen drops, and the current version of Linux JFS is 0.0.17. This version of JFS allows you to create, mount, and unmount the filesystem. In addition, you can create, copy, move, link to, and unlink files. Since Linux JFS is still pre-alpha, there are some bugs and quirks with the current implementation. The JFS developers want your feedback on what works and doesn't work so that they can continue to improve the Linux JFS implementation and get it ready for production environments.

What's involved?

Installing JFS involves downloading the latest JFS drop, patching the kernel, reconfiguring and recompiling it, and then installing the new kernel and kernel modules. After this, various JFS userland utilities need to be compiled and copied to /sbin, and the system must be rebooted. Then, you should be able to create and mount JFS filesystems just like any other Linux-supported filesystem.

Section 3. Downloading and installing JFS

Downloading

To download the JFS patches, visit *http://oss.software.ibm.com/developerworks/opensource/jfs/* . On the main page, you'll see a listing of the most recent drop, currently 0.0.17. There will be two files that you can download; one called "jfs-0.0.x.tar.gz", and the other called "jfs-0.0.x-patch.tar.gz". Download the most recent version of the "patch" tarball and save it to disk.

Extracting the tarball

To patch the kernel, I recommend making a /usr/src/linux/extras directory, and then making a jfs directory inside the extras directory. After entering the jfs directory, extract the jfs-0.0.x-patch.tar.gz file as follows:

```
$ tar xzvf jfs-0.0.x-patch.tar.gz
```

Extracting the tarball, part 2

When I extracted version 0.0.17 of the JFS patch, the following files were created in my current directory:

```
 README
 jfs-2.2.14-v0.0.17-patch
 jfs-2.2.16-v0.0.17-patch
 jfs-2.4.0-test10-v0.0.17-patch
 jfs-common-v0.0.17-patch
```

The README file contains installation instructions and a Changelog, while the rest of the files are patches. The jfs-common patch is applied first, and then one of the kernel version-specific patches is applied.

Patching the kernel

Patching the kernel is easy. First, make sure that there is a kernel version-specific patch available for your current kernel source version (with drop 17, kernel versions 2.2.14, 2.2.16, and 2.4.0-test10 are supported.)

If there isn't a version-specific patch available for your kernel, now would be a good time to upgrade your kernel. The JFS development team provides patches for the most popular "stable" kernels, as well as the most recent 2.4 series kernels.

Patching the kernel, part 2

To patch the kernel, do the following:

```
 # cd /usr/src/linux
 # patch -p1 < extras/jfs/jfs-common-v0.0.x-patch
```

This will apply the "common" patch that works across all kernel versions. Next, apply the kernel version-specific JFS for your particular kernel as follows:

```
 # patch -p1 < /extras/jfs/jfs-2.y.y-v0.0.x-patch
```

Patching the kernel, part 3

The kernel should now be successfully patched. To ensure that all the patching completed successfully, type the following while in /usr/src/linux:

```
 # find -iname *.rej
```

If the find command finds any files ending in .rej, then you know that part of the kernel patch failed. You should try to track down the problem, or report it to the bugtracker or JFS mailing list (see the Remember, JFS is pre-alpha on page 12section later in this tutorial.) However, if all went well (as it should have), then it's time to configure your kernel.

Configuring the kernel

Now you must configure the kernel, making sure that you enable support for the JFS filesystem. JFS can be compiled directly into the kernel or as a module. I recommend that you use either the "make menuconfig" or "make xconfig" commands to configure your kernel -- use "make config" and you risk losing your sanity.

To enable JFS, first enter the "Code maturity level options" category and make sure that the "Prompt for development and/or incomplete code/drivers" item is selected. Then, exit that section and head over to the "File systems" section. You should see an item that reads "JFS filesystem support (EXPERIMENTAL)". Make sure that this item is enabled (either as a module or added to the main kernel image.) After configuring the rest of the kernel to your liking, exit and save your changes.

Compiling and installing kernel/modules

If you're using a 2.4 series kernel and you want to compile JFS as a module, you can just recompile and reinstall your kernel modules by typing "make modules; make modules_install". Otherwise, you'll need to recompile your kernel and reboot to enable JFS functionality.

If you're using a 2.2 series kernel or you want to compile JFS support directly into your 2.4 kernel image, a kernel recompile will be necessary. Go ahead and compile a new kernel and modules with a "make dep; make clean; make bzImage; make modules". Then, install the new kernel modules with a "make modules_install", and configure your boot loader to boot the new kernel, /usr/src/linux/arch/i386/boot/bzImage. Now it's time to install the userspace tools.

Installing userspace JFS tools

The userspace tools include the mkfs.jfs and fsck.jfs commands, which allow you to create and fix JFS filesystems; there are other JFS-related utilities as well, along with man pages. To compile these tools, enter the /usr/src/linux/fs/jfs/utils directory:

```
# cd /usr/src/linux/jfs/fs/utils
```

Then, compile the tools:

```
# make
```

Installing userspace JFS tools, part 2

After compilation is finished, you'll find the compiled tools in the /usr/src/linux/jfs/fs/utils/output directory. You can install these tools by copying them to /sbin:

```
 # cp output/* /sbin
```

Each tool has a corresponding man page. To install the man pages, make sure you're in the /usr/src/linux/jfs/fs/utils directory, and then enter the following commands. Note that we are using backquotes for these commands (the key above Tab on US keyboards), not single quotes:

```
 # gzip -9 `find -iname *.1`
 # cp `find -iname *.1.gz` /usr/man/man1
```

If you had to compile a new kernel, now's the time to reboot. If you installed JFS as a module, you can just type "modprobe jfs" to immediately add JFS functionality to your system.

Section 4. Using JFS

Creating a new partition

To use JFS, you'll need a spare partition. If there's unpartitioned space on your disk, you'll want to fire up fdisk or cfdisk and create a partition that will be used to hold a JFS filesystem. After creating a new partition, you may be told that you need to reboot. If you receive this message, it's important that you do so -- otherwise, the filesystem creation step will flake out.

Creating the filesystem

Once you have an empty partition available, you need to put a JFS filesystem on it. To do this, type the following:

```
# mkfs.jfs /dev/hdax
```

Replace hdax with the name of the blank partition. For my particular system, I typed "mkfs.jfs /dev/hda6".

Creating the filesystem, continued

Before mkfs.jfs creates the new JFS filesystem, it'll ask for confirmation. Make sure you typed in the correct partition name (!) and then hit Y.

```
mkfs.jfs development version: $Name: v0_0_17 $

Warning!  All data on device /dev/hda6 will be lost!

Continue? (Y/N) Y
   |

Format completed successfully.

1951866 kilobytes total disk space.
```

Mounting JFS

After the filesystem has been created, it's time to mount it. After creating a new empty directory to serve as the mount point (such as "/mnt/jfs"), mount as follows:

```
 # mount /dev/hdax /mnt/jfs -t jfs
```

Now you should be able to create, modify, and delete files on your new JFS filesystem mounted at /mnt/jfs. Enjoy!

Section 5. Development resources

Remember, JFS is pre-alpha

It's really important to remember that JFS is currently in a pre-alpha stage. This means that you shouldn't put any important data on a JFS filesystem -- JFS should be used *only* for testing. Make sure your system isn't doing anything important while your JFS filesystem is mounted. After you're done testing out JFS, make sure that you unmount your JFS filesystem as a safety precaution.

Reporting bugs

So, while the bad news is that JFS isn't yet ready for production use, the good news is that you can help the JFS development process by finding new bugs and reporting them. Whenever I install a new JFS drop, I always put my new JFS filesystem through a series of "torture tests" to see if I can trigger any bugs in the code. I post any bugs I find to the IBM JFS jitterbug bug tracker, found at *http://oss.software.ibm.com/developer/opensource/jfs/bugs* .

The JFS developers pay attention to these bug reports and try to fix them by the next drop if possible. From my experience, the JFS developers have made themselves very available to the community, and that's a wonderful thing.

Resources

To keep up to date on the latest JFS developments, it's a good idea to join the JFS discussion mailing list. To do so, send an e-mail to *majordomo@oss.software.ibm.com* with a subject line of "subscribe" and "subscribe jfs-discussion" in the body. You can view the mailing list archives at *http://www.mail-archive.com/jfs-discussion%40oss.software.ibm.com/* .

At the *JFS project Web site* , you'll find information on how to check out the latest JFS code from CVS, in-depth JFS technology papers, a FAQ, and more. Remember to visit the IBM JFS project Web site often, keep patching your kernel, and keep testing!

Section 6. Feedback

Your feedback

Please let us know whether this tutorial was helpful to you and how we could make it better. We'd also like to hear about other tutorial topics you'd like to see covered. Thanks!

For technical questions about the content of this tutorial, contact the author, Daniel Robbins, at *drobbins@gentoo.org* . For questions directly relating to JFS itself, contact the JFS development team at *linuxjfs@us.ibm.com* .

Colophon

This tutorial was written entirely in XML, using the developerWorks Toot-O-Matic tutorial generator. The Toot-O-Matic tool is a short Java program that uses XSLT stylesheets to convert the XML source into a number of HTML pages, a zip file, JPEG heading graphics, and two PDF files. Our ability to generate multiple text and binary formats from a single source file illustrates the power and flexibility of XML.