REPAIRING LINUX FILE SYSTEMS
By Mark E. Donaldson

FILESYSTEM AND DATA CORRUPTION

Some disk data is kept in memory temporarily before being written to disk for performance reasons. If the kernel does not have an opportunity to actually write this data, the filesystem can become corrupted. This can happen in several ways, for example:

- The storage device, such as a diskette, can be manually removed before the kernel has finished with it.
- The system might suffer a power loss.
- The Linux kernel locks up or reboots the system.
- The user might mistakenly turn off the power or accidentally press the Reset button.

As part of the boot process, Linux runs the **fsck** program, whose job it is to check and repair filesystems. Most of the time, the boot follows a controlled shutdown wherein the filesystems will have been unmounted before the reboot. In this case, fsck says that they are "clean." It knows this because before unmounting them, the kernel writes a special signature on the filesystem to indicate that the data is good. When the filesystem is mounted again for writing, this signature is removed.

On the other hand, if one of the above disasters strikes, the filesystems will not be marked "clean". When fsck is invoked, it will notice this and begin a full check of the filesystem. This also occurs if you specify the -f flag to fsck. To prevent errors from gradually mounting (no pun intended), fsck also enforces a periodic check. A full check is done at an interval specified on the filesystem even if it were unmounted cleanly.

During the boot process, the root filesystem is mounted read-only by the kernel and fsck asks for a "clean" confirmation. If "clean, it is then mounted read/write and checked. First, the root filesystem is checked with the following command:

fsck -V -a /

The remaining filesystems and then checked by executing this command:

fsck -R -A -V -a

These options specify that all the filesystems should be checked (-A) except the root filesystem, which doesn't need checking a second time (-R), and that operations produce informational messages about what it is doing as it goes (-V), but that the process should not be interactive (-a). The latter is specified because there might not be anyone present to answer questions posed by fsck.

In the case of serious filesystem corruption, the approach breaks down because there are some things that fsck will not do to a filesystem without your permission. In this happens, fsck returns an error value to its caller (the startup script), and the startup script spawns a shell to allow the administrator to run fsck interactively. When this happens, this message appears:

REPAIRING LINUX FILE SYSTEMS
By Mark E. Donaldson

*****An error occurred during the file system check.**
*****Dropping you to a shell; the system will reboot**
*****when you leave the shell.**

Give root password for maintenance
(or type Control-D for normal startup):

This prompt is issued if the root filesystem check failed or the filesystem check failed for any of the other disk filesystems.

When the automatic fsck fails, you need to log in specifying the root password and run fsck manually. When you have typed in the root password, you are presented with the following prompt:

(Repair filesystem) #

You might worry about what command to enter here or indeed what to do at all. At least one of the filesystems needs to be checked, but which one? The preceding messages from fsck should indicate which, but it isn't necessary to go hunting for them. You can give fsck a set of options that tells it to check everything manually, and this is a good fallback:

fsck -A -V ; echo == $? ==

This is the same command as the previous except the -R option is missing in case the root filesystem needs to be checked. The -a option is also missing so fsck enters its interactive mode and a check may succeed because it can now ask questions. The purpose of the echo == $? == command is to unambiguously interpret the outcome of the fsck operation. If the value printed between the equal signs is less than 4, all is well. If this value is 4 or more, more recovery measures are needed. The meanings of the various values follow:

0 No errors
1 Filesystem errors corrected
2 System should be rebooted
4 Filesystem errors left uncorrected
8 Operational error
16 Usage or syntax error
128 Shared library error

If fsck fails, the superblock may be corrupted. By design, the ext2/ext3 filesystem has many backup superblocks scattered throughout the filesystem. Suppose the command announces that it has failed to clean some particular filesystem, such as /dev/sda1. You can start fsck again using a backup superblock with the following command:

fsck -t ext2 -b 8193 /dev/sda1

8193 is the block number for the first backup superblock. This backup superblock is at the start of block group 1. (The first is numbered 0.) There are more backup superblocks at the start of block group 2 (16385) and block group 3 (24577) as they are spaced at intervals of 8,192 blocks. If you

made a filesystem with settings other than the defaults, these might change. **mke2fs** lists the superblocks that it creates as it goes, so that is a good time to pay attention if you're not using the default settings.

After the manual fsck has succeeded, the root shell that the startup scripts provide has done its job. Type **exit** to exit it. At this point, the boot process is started again from the beginning. This second time around, the filesystems should be error-free and the system should boot normally.

USING FSCK LIVE

You can force an immediate filesystem check on reboot with fsck by issuing the command:

shutdown -F -r now

The machine drives will be unmounted cleanly, and the system will reboot. While the machine is rebooting the system will force fsck all drives that are set to mount automatically as specified in /etc/fstab.

SINGLE USER MODE

Ever forget your Linux password? Happens all the time, but you're not out of luck. Single-user mode will let you change the password.

If you use LILO to boot Linux, just type "linux single" at the LILO command prompt to enter single-user mode. If you use Grub as the boot manager, follow these steps:

1. At boot, select the kernel or Linux distribution you want to boot into. When your selection is highlighted, hit the "e" key to edit the boot commands.

2. Scroll down to the kernel command and type "e" to edit it. Go to the end of that line and type the word "single".

3. Continue the boot. Instead of getting the GUI and X Window System windows manager, you'll get a simple little prompt that gives you full access to the system, including changing passwords.

4. Once you're in single-user mode, you can change the root password with the "passwd root" command. Then you can reboot, go into X Window System, and change all the other user passwords as well.

Single-user mode is also handy if you have a corrupt file system. It's basically safe mode for Linux. It doesn't mount as many partitions, it doesn't run as many processes, and it doesn't do as many logins, so you can do many things you couldn't otherwise.

NONINTERACTIVE MODE

Fsck can also be run in noninteractive mode using the -y option:

fsck -y

REPAIRING LINUX FILE SYSTEMS
By Mark E. Donaldson

This will repair all errors encountered without wating for user response. Tthe fsck phase is broken down in a 5 step sequence.

- ** phase 1 - Check Blocks and Sizes
- ** phase 2 - Check Pathnames
- ** phase 3 - Check Connectivity
- ** phase 4 - Check Reference Counts
- ** phase 5 - Check Cylinder Groups