LINUX FOUNDATION

TRAINING

Troubleshooting Filesystem Issues

A Linux Foundation Training Publication
www.training.linuxfoundation.org

Overview

When an operating system crashes, due for example to a power failure, improper shutdown, or other unexpected event, filesystems mounted on it can become corrupted. Generally filesystem corruption means the superblock (the part of the filesystem that contains information about file system type, size, data blocks, free blocks, and inodes) is not updated and has incorrect information.

Key Ideas

fsck: The fsck utility checks filesystems for inconsistencies and can also repair them. It can be used manually, but may also be run automatically if detection is detected during system boot, or if a value is set in the mount options of an fstab entry for a particular filesystem. Unmount a filesystem before using the fsck utility on it.

dd: The dd command is used to copy and optionally convert data. Used incorrectly, it can cause data loss.

Example Scenario

To simulate an unexpected power outage that corrupts a filesystem, use the dd command to write random data to an empty filesystem. Then use the fsck command to repair the filesystem.

Now Do It

1. Use the dd command to copy random data over top of the the empty filesystem that you are corrupting on purpose.
   # dd if=/dev/zero count=1 bs=4096 seek=0 of=/dev/PARTITION

   Where PARTITION contains the filesystem you are corrupting on purpose. Don't do this on a filesystem that contains important data.
2. Interactively check and repair the corrupted filesystem using the fsck command:
   # fsck /dev/PARTITION
3. Confirm that the filesystem has been repaired using the fsck command.
   # fsck /dev/PARTITION
4. Force a check on the clean partition, just to be sure using the fsck command with the -f argument.
   # fsck -f /dev/PARTITION
5. Use the dd command to corrupt the filesystem again.
   # dd if=/dev/zero count=1 bs=4096 seek=0 of=/dev/PARTITION

6. Check and automatically repair the corrupted filesystem using the fsck command with the -y argument.
   # fsck -y /dev/PARTITION

If you remember nothing else...

Do not use fsck on a mounted filesystem.

Answer Key

1. The output from the dd command:
   1+0 records in
   1+0 records out
   4096 bytes (4.1 kB) copied, 0.00825589 s, 496 kB/s

2. Output from interactive fsck:
   fsck from util-linux 2.23.2
   e2fsck 1.42.9 (28-Dec-2013)
   ext2fs_open2: Bad magic number in super-block
   fsck.ext2: Superblock invalid, trying backup blocks...
   /dev/md0 was not cleanly unmounted, check forced.
   Resize inode not valid. Recreate<y>? yes

   Pass 1: Checking inodes, blocks, and sizes
   Pass 2: Checking directory structure
   Pass 3: Checking directory connectivity
   Pass 4: Checking reference counts
   Pass 5: Checking group summary information

   Free blocks count wrong for group #0 (4883, counted=4884).
   Fix<y>? yes
   Free blocks count wrong (87432, counted=87433).
   Fix<y>? yes
   /dev/md0: ***** FILE SYSTEM WAS MODIFIED *****
   /dev/md0: 11/24096 files (9.1% non-contiguous), 8695/96128 blocks

3. Output from second fsck command:
   fsck from util-linux 2.23.2
   e2fsck 1.42.9 (28-Dec-2013)
   /dev/md0: clean, 11/24096 files, 8695/96128 blocks

4. Output from fsck with force option:
   [root@localhost thildred]# fsck -f /dev/md0
   fsck from util-linux 2.23.2
   e2fsck 1.42.9 (28-Dec-2013)

   Pass 1: Checking inodes, blocks, and sizes
   Pass 2: Checking directory structure
   Pass 3: Checking directory connectivity
   Pass 4: Checking reference counts
   Pass 5: Checking group summary information
   /dev/md0: 11/24096 files (9.1% non-contiguous), 8695/96128 blocks

5. Output from second dd command same as first.

6. Output from fsck with automatic repair:

```
fsck from util-linux 2.23.2
e2fsck 1.42.9 (28-Dec-2013)
ext2fs_open2: Bad magic number in super-block
fsck.ext2: Superblock invalid, trying backup blocks...
/dev/md0 was not cleanly unmounted, check forced.
Resize inode not valid.  Recreate? yes

Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Free blocks count wrong for group #0 (4883, counted=4884).
Fix? yes
Free blocks count wrong (87432, counted=87433).
Fix? yes
/dev/md0: ***** FILE SYSTEM WAS MODIFIED *****
/dev/md0: 11/24096 files (9.1% non-contiguous), 8695/96128 blocks
```

LINUX FOUNDATION

Get Certified!

Get more information at http://training.linuxfoundation.org/certification/lfcs