Repair a broken Ext4 Partition Superblock

Posted on

In Linux, the entire disk space of a partition is subdivided into multiple file system blocks. The blocks are used for two different purposes. Most blocks stores user data or normal files. Some blocks in every file system store the file-system's metadata. Metadata describes the structure of the file system. The most common metadata structures are superblocks, inodes and directories. Each file-system has a superblock, which contains information about the file-system such as file-system type (ext2, ext4, etc), size of the partition and it's mount status amongst other things. If this information is lost, you are in trouble (data loss!) so Linux maintains multiple redundant copies of the superblock in every file system. This is very important in many emergency situations, for example you can use backup copies to restore damaged primary superblocks.

For this example, let's assume your secondary drive's first partition is corrupt (/dev/sdb1). If your primary root file-system is corrupt, you'll need to boot your system from a live DVD/CD and repair it from the live OS using the root user account or "sudo [command]" on Ubuntu.

So if you see an error like the below when attempting to mount a file-system: –

```
/dev/sdb1: Input/output error
mount: /dev/sdb1: can't read superblock
```

…your superblock is corrupt and the partition file-system is not accessible. You can restore the superblock from a backup but unless you've checked obvious things like SATA cables, your hard disk is probably on the way out and should be replaced as soon as possible, even if you restore the superblock from a backup on the partition.

Anyway, first make sure your partition is UNMOUNTED (umount /mountpoint). I cannot stress this enough. If you attempt to fix the partition whilst it is mounted, you will corrupt the drive even further.

You can try to run an initial file-system check using the "fsck" command.

```
fsck.ext4 -v /dev/sdb1
```

This will probably return something like: –

```
fsck /dev/sdb1
fsck 1.41.4 (27-Jan-2009)
e2fsck 1.41.4 (27-Jan-2009)
fsck.ext4: Group descriptors look bad... trying backup
blocks...
fsck.ext4: Bad magic number in super-block while trying to
open /dev/sdb1

The superblock could not be read or does not describe a
correct ext4
filesystem.  If the device is valid and it really contains
an ext4
filesystem (and not swap or ufs or something else), then the
superblock
is corrupt, and you might try running e2fsck with an
alternate superblock:
e2fsck -b 8193 <device>
```

Next, recover the list of backup superblocks from the partition like so: –

```
dumpe2fs /dev/sdb1 | grep superblock
```

This will produce a list of alternate superblocks you can use.

```
Primary superblock at 0, Group descriptors at 1-6
Backup superblock at 32768, Group descriptors at 32769-32774
Backup superblock at 98304, Group descriptors at 98305-98310
Backup superblock at 163840, Group descriptors at
163841-163846
Backup superblock at 229376, Group descriptors at
229377-229382
Backup superblock at 294912, Group descriptors at
294913-294918
Backup superblock at 819200, Group descriptors at
819201-819206
```

```
Backup superblock at 884736, Group descriptors at
884737-884742
Backup superblock at 1605632, Group descriptors at
1605633-1605638
Backup superblock at 2654208, Group descriptors at
2654209-2654214
Backup superblock at 4096000, Group descriptors at
4096001-4096006
Backup superblock at 7962624, Group descriptors at
7962625-7962630
Backup superblock at 11239424, Group descriptors at
11239425-11239430
Backup superblock at 20480000, Group descriptors at
20480001-20480006
Backup superblock at 23887872, Group descriptors at
23887873-23887878
```

Now you can use a alternate superblock and attempt to repair the file-system.

```
fsck -y -b 32768 /dev/sdb1
```

This will produce output similar to the below: –

```
fsck 1.40.2 (12-Jul-2007)
e2fsck 1.40.2 (12-Jul-2007)
/dev/sdb1 was not cleanly unmounted, check forced.
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 #241 (32254,
counted=32253).
Fix? yes
Free blocks count wrong for group #362 (32254,
counted=32248).
Fix? yes
Free blocks count wrong for group #368 (32254,
counted=27774).
Fix? yes
```

```
..........
/dev/sdb1: ***** FILE SYSTEM WAS MODIFIED *****
/dev/sdb1: 59586/30539776 files (0.6% non-contiguous),
3604682/61059048 blocks
```

You should now be able to mount the file-system as normal (or reboot if it’s the primary root file-system): –

```
mount /dev/sdb1 $HOME/mount
```

Here, I’m mounting the file-system on the mount subdirectory of my user’s (/root in this case) home directory. If this doesn’t work, run through the fsck command above trying each backup superblock number in turn until you find one that works. Once you can successfully mount the file-system at a directory mount point, you can access your files.

Now would be the time to backup those files before the disk fails completely. Sometimes superblocks get corrupted and the disk will be fine for a while longer, but I take no chances :-)