

Check_when_a_fsck_is_due

Note: this is for Debian/Ubuntu Systems.

When applying kernel upgrades, typically a reboot is required. Typically a FSCK (disk check) is scheuled every 6 months (or number of mount times). A fsck can take quite a long time to scan the disks to see if there are any errors, especially with large disks. Wouldn't it be nice to know if a fsck will happen at the next reboot? And then you can plan for more downtime.

```
tune2fs -l /dev/sda1
```

It tells you when the next fsck will be run (Next check after:). See below for an example.

```
twister:~# tune2fs -l /dev/sda1
tune2fs 1.41.3 (12-Oct-2008)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         451286b3-e9b0-43fe-8684-7fbc1dc611ed
Filesystem magic number: 0xEF53
Filesystem revision #:   1 (dynamic)
Filesystem features:     has_journal filetype needs_recovery sparse_super
Default mount options:   (none)
Filesystem state:        clean
Errors behavior:         Continue
Filesystem OS type:      Linux
Inode count:             393216
Block count:             786432
Reserved block count:   31457
Free blocks:             370519
Free inodes:             326179
First block:             0
Block size:              4096
Fragment size:          4096
Blocks per group:       32768
Fragments per group:   32768
Inodes per group:       16384
Inode blocks per group: 512
Filesystem created:      Tue Sep 12 17:23:59 2006
Last mount time:         Wed Apr 29 23:02:37 2009
Last write time:         Wed Apr 29 23:02:37 2009
Mount count:             5
Maximum mount count:    32
Last checked:            Fri Jan 30 08:28:42 2009
Check interval:         15552000 (6 months)
Next check after:        Wed Jul 29 09:28:42 2009
Reserved blocks uid:    0 (user root)
Reserved blocks gid:    0 (group root)
First inode:            11
Inode size:             128
Journal inode:          8
First orphan inode:     182499
Default directory hash: tea
Directory Hash Seed:    1602904f-b5d3-4e6b-aac2-2e9b3a9964da
Journal backup:         inode blocks
```