



# Chapter 21: Maintaining Integrity



# Maintaining Filesystem Integrity

- Most common causes of filesystem damage is improper shutdown

Operation	Traditional SysV	Systemd
Shutdown	<code># init 0</code> <code># halt</code>	<code># systemctl poweroff</code>
Restart	<code># init 6</code> <code># reboot</code>	<code># systemctl reboot</code>



# Displaying filesystem data

- There are filesystem commands that can:
  - Display filesystem usage on a filesystem-by-filesystem basic
  - Display filesystem usage on a directory basic
  - View filesystem metadata
  - Modify filesystem metadata



# The df command

- The `df` command displays filesystem usage:
  - Displays free and used data block space
  - Displays free and used inodes
- By default, `df` shows data block usage in 1-K data block size
- Use `-h` to show in "human readable" size
- Use `-i` to show free and used inodes



# The df command

- Example:

```
# df -hT
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/sda2	ext4	6.3G	3.2G	2.9G	53%	/
tmpfs	tmpfs	351M	84K	351M	1%	/dev/shm
/dev/sda1	ext4	485M	52M	408M	12%	/boot
/dev/sda5	ext4	2.2G	69M	2.0G	4%	/home



# The du command

The `du` command displays disk space usage for a directory structure

Use `sort` to have largest files displayed last:

```
$du | sort -n
```

- Use `-h` option to display sizes as "human readable"
- Use `-s` option to see a summary (not the size of every item in the directory)



# The du command

- Example:

```
$ du | sort -n | tail -10
324  ./kde/share/config
336  ./mozilla/firefox/yj3gb7ak.default/thumbnails
452  ./kde/share
460  ./kde
712  ./mozilla/firefox/yj3gb7ak.default/Cache
924  ./mozilla/firefox/yj3gb7ak.default/startupCache
14428 ./mozilla/firefox/yj3gb7ak.default
14436 ./mozilla/firefox/
14452 ./mozilla/
16020 .
```



# The `tune2fs` command

- The `tune2fs` command can view and change to some of the settings for `ext2`, `ext3` and `ext4` filesystems
- Good idea to back up filesystem before modifying



# The tune2fs options

Option	Effect
-l	Used to list the superblock information for a filesystem.
-c	Used to change the maximum number of times that a filesystem may be mounted before it is required to have a full filesystem check.
-i	Used to change the maximum time interval between when a filesystem is forced to have a full filesystem check.
-j	Using this option will create a journal file for an ext2 filesystem, allowing it to be mounted as an ext3 or ext2 filesystem.
-m	Specify the percentage of space to be reserved for the root user or privileged processes.
-o	Used to specify default mount options.



# The dumpe2fs command

- The `dumpe2fs` command displays superblock and group block information for `ext2`, `ext3`, or `ext4` filesystems:

```
# dumpe2fs /dev/sda1 | head
dumpe2fs 1.41.12 (17-May-2010)
Filesystem volume name: <none>
Last mounted on:      /boot
Filesystem UUID:      974e2406-eeec-4a49-9df7-c86d046e97f9
Filesystem magic number: 0xEF53
Filesystem revision #: 1 (dynamic)
Filesystem features:  has_journal ext_attr resize_inode dir_index filetype
                    needs_recovery extent flex_bg sparse_super huge_file uniting_bg dir_nlink extra_isize
Filesystem flags:      signed_directory_hash
Default mount options: user_xattr acl
Filesystem state:      clean
Errors behavior:       Continue
```



# The dumpe2fs command

- Output is similar to the `tune2fs` command  
Use `-h` to see only superblock information
- Use `-b` to see bad block information



# Introduction to filesystem integrity

- Filesystem can become corrupt when not unmounted properly:
  - Power loss
  - Removing a USB drive without running the `umount` command
  - Pulling the plug on a system
- It is important to gracefully shut down the system as it correctly unmounts partitions



# The fsck command

- The `fsck` command is a front end command that launches commands that repairs filesystems
  - Use `-y` option to answer yes to all "do you want to fix this" questions
  - Use `-n` option to answer no to all "do you want to fix this" questions
- Do not run `fsck` on mounted filesystems



# The fsck command

- Runs non-interactively during boot on select filesystems
- Which filesystems depend on filesystem options, the `/etc/fstab` file and how long since last check
- To force `fsck` checks on next reboot, create a `/forcefsck` file:  

```
#touch /forcefsck
```



# The force option

- To force a check, use the `-f` option:  
# `fsck -f /dev/sdb1`  
fsck from util-linux-ng 2.17.2  
e2fsck 1.41.12 (17-May-2010)  
Pass 1: Checking indoors, 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/sdb1: 11/12824 files (9.1% non-contiguous),`  
`6532/51200 blocks`



# Using an alternate superblock

- When a superblock is damaged, the `mount` command is unable to read the filesystem type causing the `mount` command to fail
- If the superblock is damaged, `fsck` will not be able to check or fix the filesystem
- `fsck` will attempt to use backup superblock



# Using an alternate superblock

- `fsck` expects backup superblocks to be in specific places
- Based on how the filesystem was created, the backup superblocks could be in a different location



# Using an alternate superblock

- List the location of backup superblocks with the `dumpe2fs` command:

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

```
dumpe2fs 1.41.12 (17-May-2010)
```

```
Primary superblock at 1, Group descriptors at 2-2
```

```
Backup superblock at 8193, Group descriptors at 8194-8194
```

```
Backup superblock at 24577, Group descriptors at 24578-24578
```

```
Backup superblock at 40961, Group descriptors at 40962-40962
```



# Using an alternate superblock

- Use the `e2fsck` command to specify the location for the backup superblock:

```
#e2fsck -b 8193 /dev/sdb1
```

- `fsck` could also be used:

```
#fsck -b 8193 /dev/sdb1
```



# The lost+found directory

- An "unreferenced file" error means a file was "lost"
- This means the file data exists, but the location of where the file is suppose to go has been lost
- By looking in the `lost+found` directory in the mount point of the filesystem, you can see these files



# The lost+found directory

- Files in lost+found will not have a name, just an inode number:

```
# ls -l /data/lost+found
```

```
total 5
```

```
-rw-r--r--. 1 root root 83 Mar 4 06:37 #34234
```

```
-rw-r--r--. 1 root root 1911 May 20 2013 #4596
```



# The lost+found directory

- Look at the attributes of the file (owner, size, timestamps) to determine where it goes or who it belongs to
- Use the `file` command to determine if it is a text file - if so, look at its contents to determine where it goes or who it belongs to