Lab 14.2: Modifying Filesystem Parameters with tune2fs

We are going to fiddle with some properties of a formatted **ext4** filesystem. This does not require unmounting the filesystem first.

In the below you can work with an image file you create as in:

```
$ dd if=/dev/zero of=imagefile bs=1M count=1024
```

or you can substitute /dev/sdaX (using whatever partition the filesystem you want to modify is mounted on) for `imagefile`.

1. Using **dumpe2fs**, obtain information about the filesystem whose properties you want to adjust.
2. Ascertain the maximum mount count setting (after which a filesystem check will be forced) and modify it to have the value `30`.
3. Set the `Check interval` (the amount of time after which a filesystem check is forced), to three weeks.
4. Calculate the percentage of blocks reserved, and then reset it to 10%.

Solution 14.2

1. `$ dumpe2fs imagefile > dump_results`

2. `$ grep -i "Mount count" dump_results`

   ```
   Mount count:              0
   Maximum mount count:      -1
   ```

   ```
   $ sudo tune2fs  -c  30 imagefile
   $ grep -i "Mount count" dump_results
   ```

   ```
   Mount count:              0
   Maximum mount count:      30
   ```

3. `$ grep -i "Check interval" dump_results`

   ```
   Check interval:           0 (<none>)
   ```

   ```
   $ sudo tune2fs -i 3w imagefile
   $ grep -i "Check interval" dump_results
   ```

   ```
   Check interval:           1814400 (3 weeks)
   ```

4. `$ grep -i "Block Count" dump_results`

   ```
   Block count:              131072
   Reserved block count:     6553
   ```

   ```
   $ echo "scale=4; 6553/131072" | bc
   ```

   ```
   .0499
   ```

   ```
   $ sudo tune2fs -m 10 imagefile
   $ tune2fs 1.42.9 (28-Dec-2013)
   ```

   ```
   Setting reserved blocks percentage to 10% (13107 blocks)
   ```

   ```
   $ grep -i "Block Count" dump_results
   ```

   ```
   Block count:              131072
   Reserved block count:     13107
   ```