Exercise 14.2 fs_mark

The **fs_mark** benchmark gives a low level bashing to file systems, using heavily asynchronous I/O across multiple directories and drives. It's a rather old program written by Rick Wheeler that has stood the test of time.

It can be downloaded from http://sourceforge.net/projects/fsmark/ Once you have obtained the tarball, you can unpack it and compile it with:

```
$ tar zxvf fs_mark-3.3.tgz
$ cd fs_mark
$ make
```

Read the README file as we are only going to touch the surface.

If the compile fails with an error like:

```
$ make
....
/usr/bin/ld: cannot find -lc
```

it is because you haven't installed the **static** version of **glibc**. You can do this on **Red Hat**-based systems by doing:

```
$ sudo yum install glibc-static
```

and on **SUSE**-related systems with:

```
$ sudo zypper install glibc-devel-static
```

On **Debian**-based systems the relevant static library is installed along with the shared one so no additional package needs to be sought.

For a test we are going to create 1000 files, each 10 KB in size, and after each write we'll perform an **fsync** to flush out to disk. This can be done in the /tmp directory with the command:

```
$ fs_mark -d /tmp -n 1000 -s 10240
```

While this is running, gather extended **iostat** statistics with:

```
$ iostat -x -d /dev/sda 2 20
```

in another terminal window.

The numbers you should surely note are the number of files per second reported by **fs_mark** and the percentage of CPU time utilized reported by **iostat**. If this is approaching 100 percent, you are I/O-bound.

Depending on what kind of filesystem you are using you may be able to get improved results by changing the **mount** options. For example, for **ext3** or **ext4** you can try:

```
$ mount -o remount,barrier=1 /tmp
```

or for **ext4** you can try:

```
$ mount -o remount,journal_async_commit /tmp
```

See how your results change.

Note that these options may cause problems if you have a power failure, or other ungraceful system shutdown; i.e., there is likely to be a trade-off between stability and speed.

Documentation about some of the **mount** options can be found with the kernel source under Documentation/filesystems and the **man** page for **mount**.

