Q: Why does `fcntl(fd, F_SETLK, &lockstruct)` give `EINVAL`?

Q: What is the preferred method of file locking?

A: The current release of Sun's NFS file system has a number of problems regarding file locking. Sun has implemented a "first crack" at solving the problems associated with extending Unix's file locking to NFS. NeXT decided, upon looking at this, that there were too many hidden deadlock conditions inherent in the implementation to release. Instead, the current version of `fcntl(2)` will fail with `EINVAL` if the directives `F_GETLK` or `F_SETLK` are used. These directives will work when a new version of NFS file locking is available, in our 4.0 release.

In the mean time, if a file locking system is necessary for your application, the best system to

use is to write out a file whenever your application accesses a file for writing. The easiest thing to do is to create a file of the same name with an added extension, like "LCK" for instance. Your application should check to see if this file exists before opening a file, and create the lock file, if one doesn't exist.

Here's one idea: if you open(2) a file with both CREATE and EXCLUSIVE turned on, then open(2) will return an error if the file exists, and will create the file if it doesn't. Something like

```
#include <sys/file.h>
int    fd;

fd = open(lockfile, O_CREAT | O_EXCL, 0600);
```

```
if (fd < 0) {                       /* Something didn't work */
       if (errno != EEXIST) {       /* Real error occurred */
              perror("Open failed");
              exit(1);
       }
       else {        /* Lock file is there      */
                     /* await lock release */
       }
}
else /* Things are OK */
```

This still has some problems when systems crash with files open, or the if the Workspace exits with your application still running. To deal with this, your application should erase these lock files (with the unlink(2) system call, for example) in an appDidTerminate: method.

Also, when one of these lock files exist, you should display an alert panel asking if the user would like to override the lock.

For applications that run on a single machine (without any network file access) you can use `flock(2)` if you are concerned that your application may be run multiple times on the same machine. This scheme works for network files too, when that support is added.

Note: the `fcntl`(2) directives not associated with file locking:

```
F_{DUP,SET,GET}FD
F_{GET,SET}FL
F_{GET,SET}OWN
```

work fine.

QA590

Valid for 1.0, 2.0, 3.0, 3.1