Matthew "u4125668" Joiner.

COMP3300 Assignment 2 Progress Report

Thus far I've modified the vvsfs code to work on linux-2.6.25.17, the kernel I operate at home. This required changing:

- calls to iget() -> iget_locked()
- struct super_operations {.read_inode -> .put_inode}
- Several function prototypes, which were conveniently receiving incorrect and unused parameters or returning the wrong types.

I've compiled and run the tests successfully, and created an SVN repository to track my changes.

I plan to enable recording of information such as permissions. This will require storing said information in the vvsfs inode structure (struct vvsfs_inode), and add several functions to struct file_operations. Additional extensions will be added, time permitting.

Testing and evaluation of this extension will involve rigorously attempting combinations of scenarios triggering predictable access and permissions errors, and setting and querying of file status through use of stat(), access() and other system calls, from userspace. Additionally changes may be required to the userspace filesystem tools, to support new features.

Bugs I've found so far, mostly consist of incorrect or out of date variable names, and return of values in the incorrect types, or passing of values as the wrong type, where they're not used (and thus don't cause any problems... yet).

I believe testing of permissions against those found in the inode for a file may be difficult, further research, and development will reveal how extensive the generic kernel support for this is (and how much I will have to reinvent). Default function callbacks, and the very layered abstractions used in the kernel should enable much of this to be done easily.