Matthew "u4125668" Joiner

COMP3300 Assignment 2 Summary

The file system was given the ability to unlink files. This required removing an entry from the directory inode, and marking the files inode as empty on disk. The required function pointer to vvsfs_unlink() was added to vvsfs_dir_inode_operations.

File truncation required that permissions be granted for the MAY_WRITE mask, passed from vfs_permission(). A function pointer to vvsfs_permission() was added to vvsfs_file_inode_operations to allow this.

The file can now record additional information in the file inodes such as the user and group ids, permissions (in the form of a mode parameter), and timestamps (to one second precision). Additional fields were added to the vvsfs_inode structure, and 2 existing flags, is_directory and is_empty were merged into a new flags variable. These variables are read and written from and to disk, in the vvsfs_iget() (called from vvsfs_dir_inode_operations: lookup function pointer), and vvsfs_write_inode called from the super operations vvsfs_ops: write_inode pointer.

Due to several errors in the initial implementation, checks are made on a folder before dentries added, to ensure enough room exists in the directory inode on disk. Furthermore, no extra filesystem information is recorded for the directory inode, the VFS fills out a default inode which has generic timestamps and permissions or seems to sometimes reuse the ones on the mount dentry.

Directories are limited to holding 25 files, and as there is no way to add additional directories, this is the file limit for the super block. If an attempt is made to add more than 25 files, the ENOSPC error is return, which does not appear to the correct error value, but the closest available. (ENOSPC is intended for failure to retrieve a new inode, which would occur at the use of all 100 inode sectors in the case of vvsfs.)

Most development on vvsfs, was done on a 2.6.26.5 vanilla kernel. Of noteworthy difference, several locking mechanisms have changed during the interval from 2.6.20, and the read_inode, and put_inode function points in the super_operations struct have been removed, apparently as they were no longer wanted, and only several filesystems were still depending on them. By 2.6.26, these functions were removed.

Added functions:

- `static inline int has_perm(int, short unsigned);`
- `static int vvsfs_permission(struct inode *, int, struct nameidata *);`
- `static struct inode *vvsfs_iget(struct super_block *, long unsigned);`
- `static int vvsfs_write_inode(struct inode *, int);`
- `static int vvsfs_unlink(struct inode *, struct dentry *);`

Removed/unused functions:

- `static void vvsfs_read_inode(struct inode *);`

The central change was in vvsfs_inode, where types are matched in precision to those in the

kernel, but using standard library available types, so as to allow mkfs.vvsfs and friends to compile without special kernel header file requirements.

```
struct vvsfs_inode {
      int32_t atime, mtime, ctime;
      uint32_t flags, uid, gid;
      uint16_t mode, size;
      char data[MAXFILESIZE];
};
```