

GSoC Final Document

Adam Burkepile

August 18, 2008

1 Progress made

1. Rethought some design decisions
2. Created some needed flags in the superblock and inode
3. Modified newfs and dumpfs to modify/read the flag in the superblock to recognize whether a file system has subfile support
4. Created a new system call (`subfile_open`) stub and I am using that stub to build into the system call
5. Open the basefile and get needed attr
6. Recognize if the basefile is a VREG file and needs to have a subfile dir created
7. Made changes to `VOP_MKDIR` to recognize creating a subfile dir, which doesn't install it into a parent directory and make `.` and `..` point to the subfile dir
8. Set the flag that I created in the inode of the basefile and the subfile dir that tells the inode number of the subfile dir
9. `VGET` to retrieve the subfile directory
10. Do a `VOP_LOOKUP` inside of the retrieve subfile dir for the requested subfile

2 Code Changes / Walk Through

Does not include little changes that were needed, i.e. files that needed something added for the new system call.

2.1 src/sys/ufs/ffs/fs.h

1. Added a macro to use one of the spare variables in the ffs2 superblock to show whether the partition supports subfiles

2.2 src/sys/ufs/ufs/dinode.h

1. Added a flag to store the inode number of the subfile directory of the basefile in the basefile's inode

2.3 newfs() - src/sbin/newfs/mkfs.c

1. I changed this to set the subfile support flag if ffs2 was selected as the newfs (with the -O 2 argument)

2.4 dumpfs() - src/usr.sbin/dumpfs/dumpfs.c

1. Print out the subfile fs support flag for the "dumped" fs

2.5 src/sys/ufs/ufs/ufs_vnops.c

1. Changed ufs_getattr to get the subfile directory field from the inode attributes
2. Changed ufs_setattr to set the subfile directory field for the requested inode if it is indeed a ffs2 fs
3. Made changes to ufs_mkdir so if the subfile directory flag is set special, it knows it should make a subfile directory. When this happens it does three things different from the regular call. It does not increase the link count of the parent directory (since the parent directly is the base regular file and the directory is not attached to it), makes .. point to itself (the same as .), and it doesn't install the directory in the parent.

2.6 subfile_open() - src/sys/kern/vfs_subfile.c

1. A slightly modified read-only open on the basefile.
2. Once I have the basefile, use VOP_GETATTR to get the inode number and the subfile directory number of the basefile.

3. Close the basefile.
4. Examine the vnode type of the basefile and make sure its a regular file, return if its isn't.
5. Check the subfile directory number, if it is 0, we know that not subfile directory exists yet for that basefile so we need to create it.
 - (a) Lock the basefile.
 - (b) Create a blank attribute and set up the default modes and type.
 - (c) Call VOP_MKDIR.
 - (d) Get the inode number of the newly created subfile_directory.
 - (e) Set the subfile directory number of the subfile directory and the basefile to the inode number of the subfile directory.
 - (f) Unlock the basefile and the subfile directory.
6. Lock the basefile.
7. Get the subfile directory number from the basefile.
8. Unlock the basefile.
9. Preform VFS_VGET with the subfile directory number to get the vnode of the subfile directory.
10. Preform a VOP_LOOKUP with the subfile directory for the requested subfile to determine if the subfile exists or not.

3 What I'd like to do next

1. Create the file in the subfile dir if it isn't there
I'm a bit hung up on this part at the moment
2. Open the file and return the fd

4 Something I would of done different

I planned on making two system calls, `subfile_open` and `subfile_fopen`. `subfile_open` opened the basefile based on the path and `subfile_fopen` opened a previously opened basefile based on the file descriptor. I began with the path-based `subfile_open` first because I figured that made more sense and the two would be similar and need few changes, but as I worked on it I think it might of been smarter to create the `subfile_fopen` and then have `subfile_open` do a regular open then call `subfile_fopen`. This would kept repeating code from being in the two syscalls and simplify things. I'll have to discuss this but I think it would be worth pursuing changing my code to this.

5 After GSoC

We have some time before our school begins again so I was planning to continue working with this since the overall project will still be far from done. When school starts back up I imagine that I will still continue to work on it, less though due to time constraints.