

Assignment 6

Gangasagar Patil
Roll No: 201305549

1 Implemented file systems

1. Read-only
2. Read-write

2 Experience

1. Found the documentation at the site.
<http://fuse.sourceforge.net/>
2. Filesystem is maintained to be in the form of Tree hierarchy with its root at mount point. Also I referred fuse.h for function prototype.
3. Maintained a custom structure which contains the fields for path, type of file, few fields of stat structure for storing information about the file and an array which contains all children of that particular file (directory).
4. Maintained an array for all nodes in the file system. The index in this array is used for nodes in subsequent functions. It supports 1000 nodes in the file system. i.e. MAXNODE. Value of MAXNODE can be changed as per need.
5. For read-only system, read only rights are set for files in the file system.
6. Used the system calls like, getuid(), getgid(), time() for loading the data into the structure.
7. Overall it was a nice experience. Learnt a lot of system calls regarding file structure, also file structure implementation. Also came to know about system call dependency in file implementation.

3 Testing

1. Read-only
 - (a) While developing the filesystem, used the sample file structure with 1 file and 1 directory inside root and 2 files inside that child directory.
 - (b) Tested it with long file names. Any length of file can be accessed.
 - (c) Tested with following commands ls, cat, cd.

2. Read-write

- (a) While developing the filesystem, used the sample filestructure with 1 file and 1 directory inside root and 2 files inside that child directory.
- (b) Tested it with long file names. Any length of file can be accessed.
- (c) Functions implemented
 - i. Directory operations: `readdir`, `mkdir`, `rmdir`
 - ii. File operations: `open`, `read`, `write`, `truncate`, `access`, `close`, `create`
 - iii. Meta operations: `getattr`, `statfs`, `symlink`, `link`, `unlink`, `rename` etc.