- **DirectoryWalker (function in fs.c)**

1. Get the inode by function namei() with parameter path. Compute the file name by namiparent() unless the path is just the root.
2. Save required information from inode and name. If the inode is directory, check each file in this dir by dirent struct getting from readi(), and if it has been allocated (inum!=0), call the directoryWalkRecusive again to save information and check this child inode again whether it is also a directory.

- **InodeWalker (function in fs.c)**

1. Read the superblock, and we will know the maximum inode number from it. Then we can check all the inode under max value. Using IBLOCK, we get the block number for specific inode and read the block to buffer.
2. Now we can get dinode form data of bp in position of inum%IPB. Checking whether the dinode number is not 0 (not free), we can get allocated inode and copy the information to our arrays.

- **eraseDirectory (function in fs.c)**

1. Get the inode by function namei() with parameter path.
2. Check this inode whether is a directory.
3. Read each dirent from corresponding block from ip addrs. If the dirent contain same name that we want to erase, set the inum and name as 0 in this dirent and write back to the block.

- **recoveryDirectory (function in fs.c)**

This function is simply recover the link (dirent) from specific inode (inum) to specific path.
1. Get the inode by function namei() with parameter recoverpath.
2. Check this inode whether is a directory.
3. Read each dirent from corresponding block from ip addrs. If the dirent inum is 0, we write the name with inum to this dirent and write dirent back to the block.

The total principle for recovery:

1. Compare the inum information between directoryWalker and inodeWalker, and record the unmatched inums in inodeWalker, which means they are lost in directory.

2. For each unmatched inums, we set a default name "reoveryXX" to pair with it. Then we call recoveryDirectory to link the pair of inum and name to default directory (root). In addition, Since the inode is complete, we do not need repair inode information.

- **Other changed code**

sysfile.c: Add system call for sys_directory_walker, sys_inode_walker, sys_erase_directory, sys_recover_directory. Mainly get parameter from user.
syscall.c: Add declaration for these new system calls.
syscall.h: Add corresponding number (position) to these new system calls.
user.h: Add declaration for user mode functions.
usy.S: Add reflections.
Makefile: Add test code path.