HOMEWORK DESIGN

HierarchicalFS.py Things tested on Command line and GUI:

1) Done extensive testing of the code for directory creation by making upto 7 to 10 levels of directories in fusemount hierarchical as well as on a given level through command line as well as GUI.
2) Tested ls and la commands over the levels of directories as well as files.
3) Created files on every level, modified contents (write) and displayed it on terminal.
4) Renamed files and folders using move command. The contents remain the same for a rename on same level in case of files and the contents don't change for any level in case of directories.
5) Remove command used both on files and directory and its contents too get deleted.
6) Chmod command used to change the permissions of files as well as directories successfully on any level.
7) Content can be written to a file as well as appended using cat > or cat >> command.

Functions modified:

1) MKDIR: modified the self. Files dictionary that stores the metadata of the directories and files to create a kind of nested directory structure which adds the new directory name as a key with its value as a dictionary of all its attributes. Further if any directory is to made inside this directory, the name of the new directory is set as key along with the attributes of the parent directory and its value is a dictionary with attributes of new directory. It also increments the link count in st_nlink of the parent directory.
2) Getattr: the getattr function traverses the path name and reaches the final directory whose attribute is requested. It then filters out the attributes from other directory keys that may be present and outputs the attributes.
3) Rmdir: It follows the path till the parent directory in self.Files and pops out the key value pair of the mentioned directory in path or file from self.Files.
4) Readdir: it follows the path provided as an argument to the parent directory an filters out and lists all the keys that are related to a directory on that particular level as an output.
5) Create: the function is modified to create files kindof dictionaries in self.Files which keeps the metadata of the files and directories. It has been modified to work on multiple level. Also it increments the file descriptor variable fd.

6) Write: modified to store the input data into self.Data dictionary and to traverse the path and make changes to the metadata of the file in self.Files.
7) Utimens: changed the code to follow the path in the argument in self.Files and make changes to the time attributes of the metadata.
8) Truncate: changed to traverse the path in self.Files and modify the size data in the attributes.
9) Symlink: changed to traverse the path in self.Files to go till target and change the mode and add the soruce to self.Data
10) Setxattr: follow the path till the required directory and give back the values of the attributes.
11) Rename: changed to include file as well as directories. The function first checks the if the path is for a file or a directory and accordingly modifies the data. If it is for path it changes both the self.Data and self.Files to rename the file as well as contain the data whereas in case of a directory it just changes the self.Files metadata.
12) Removeattr: modified to follow the path and remove the attributes that are provided in argument on multi-level.
13) Listxattr: modified for multilevel path following and listing all the attributes in metadata.
14) Getxattr: modified to work for multilevel path
15) Chown: modified to change uid and gid on a multilevel scale by traversing self.Files.
16) Chmod: changed to work on multilevel and assigning different modes of permissions by changing the st_mode attribute in self.Files.
17) Helper function: add_path: this is a helper function used in every modified program to split the path provided by system into list whose elements are dictionary or file names.

The file is handling these cases perfectly:

1) A directory with the same name as its parent directory.
2) Number of hard links of parent directory properly set when the list of sub directories changes
3) Rename of a file retains its contents on a same level.
4) Rename of a folder retains its contents on all levels.
5) Remove directory removes all its contents.