6: File Sweeper

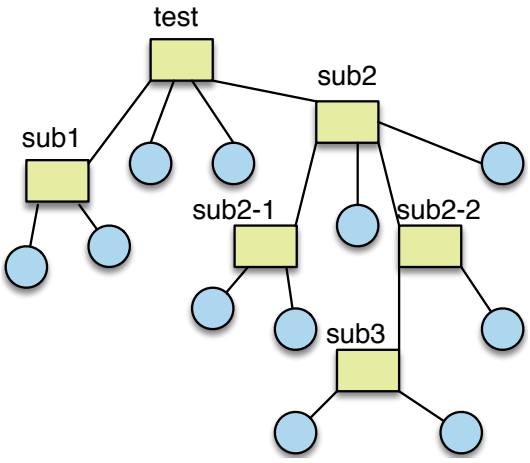
CSCI 4547 / 6647 Systems Programming
Fall 2017

1 Goals

1. To write and use a recursive function.
2. To identify duplicate files in a directory tree.

1.1 Preparation

Create a test directory for this program. Create directories (green) and files (blue) as shown on the right. Then add soft links in at least two directories and add several hard links. Each one should go from one subdirectory to a file in a different directories.



2 Sweeper::run().

Sweeper::run() might be given either an absolute pathname for the starting point, or it might be the simple name of a sub-directory of the current working directory. Whichever it is given, you must calculate the other because you need both.

- Set `current` to the simple name of the starting directory and `path` to the absolute pathname of `current`.
- Call `travel` with `current` as the parameter.
- When all the recursive calls return, you have constructed a table of all the files in your directory tree.
- Sort it by inode# and print it as in P5.

3 The travel() function

Define a `void` recursive function called `travel` in the Sweeper class. The parameters will be the pathname and the simple name of the directory to be processed next. This function will handle directory entries for both files and subdirectories. File entries will be stored in the `vector<FileID>` that is a member of the Sweeper class. Directory names will be processed recursively.

When `travel` is first called, the parameters should be the pathname and simple name of the place you will start the sweep operation. In the body of `travel()`:

- Open the directory `current`.
- Change directory to `current`.
- Read each entry in `current`. If it is not a directory process it as in P5.
- If it is a directory, prepare for and make a recursive call on `travel`. The first parameter should be the concatenation of the path, '/', and the simple name of the new directory.
- By doing the concatenation operation in the call itself, you avoid needing to REMOVE the last link of the path when you return from the recursion.
- When the recursion returns, change directory to `..`