

Programming Assignment 4

CS450 Fall, 2021

1. This assignment is a pair programming effort. It is due on 12/02/2021
2. **Purpose:** Develop a set of tools that can help to recover a damaged file system. For many reasons, the information in a directory file, in an inode etc. may get damaged and cannot be read. As a result, a user program will not be able to access some parts of the file system. In this exercise, we ask you to develop a set of tools and show that you can use them to recover a file system if one or more of its directory inode is damaged.
3. **Requirements: (60%)**
 - 1) Your tools and recovery program will work on the xv6 file system.
 - 2) (10%) Write a program, called the `directoryWalker`, that prints out the names of each file and directory in a file system tree, starting at a given point in the tree. The program will also write the inodes associated with each file and directory.
 - 3) (10%) Write a program, called the `inodeTBWalker`, that prints out all the allocated inodes. You obtain this set of inodes from the inode table.
 - 4) (10%) Write a program that will compare the inode related output from the two Walkers.
 - 5) (10%) Write a program that can erase the information (including the block pointers) in a directory inode so that you can test and demonstrate your results.
 - 6) (20%) If one or more of the directory inodes of the file system is damaged and therefore cannot be read, the `directoryWalker` can traverse only part of the file system tree. Write a program that will fix the damaged file system. Can you do this just by having the Walkers as specified above? What is the best you can do in terms of the number of damaged directory inodes?
 - 7) **Bonus:** (Additional 5% or more) Can you recover the file system if other parts of the file system data structure is damaged? Implementation not necessary but will count.
4. **Deliverables:**
 - 1) (15%) A document that describes your design and why it works, the various existing xv6 functions and header files that you changed, and the manual pages for the system calls that you added to the system. The document should not be more than 6 pages with at least a 11 point font size.
 - 2) (25%) The test data (i.e. the test file system) that you use and explanation on why the test data is of good quality. If you use the equivalence partitioning method to generate your test data, describe your equivalence partitions.
 - 3) Source(with complete xv6 source code) and executable objects with a README on how to build and execute them. If you use an version of xv6 other than the

one in <http://github.com/mit-pdos/xv6-public> you need to provide the details of how to make it work and how to compile it.

- 4) Upload all files as a **zip** archive with GroupID_PA4.zip. Documents and readme only supports: txt, doc, docx and pdf format.
- 5) (Mandatory, i.e. no point if not submitted) Self-evaluation is due 24 hours after submission.