

Operating Systems

CSCI 5806

Term Project — Filesystem Integrity Checker

Due date: Friday, April 28, 2017

Goal

Develop a program that checks the integrity of an ext2 filesystem, reporting any errors found and possibly correcting the errors.

Details

Your program should examine a VDI file containing a virtual disk with a single partition formatted as an ext2 filesystem. The filename must be given on the command line; do not prompt the user for it and do not hardcode it.

The program must display the following general statistics about the filesystem:

- Total filesystem size in bytes
- Size available for files (used and unused)
- Amount of space currently used
- Number of possible files and directories (number of inodes)
- Number of existing files
- Number of existing directories
- Number of block groups, with the information for each block group
- Block size in bytes
- State of the filesystem

Your program must also check the following items:

- The superblock must have the appropriate magic number
- All copies of the superblock must be consistent (*spoiler alert* — they won't be, but that's okay. Report what's different.)
- All copies of the block group table must be consistent (see above)
- Every inode marked as used must be reachable from the root directory
- No unused inode may be reachable from the root directory
- Every directory entry must reference an in-use inode

- Every used data block must be referenced by exactly one inode, either as data or as an indirect block
- No unused data block may be referenced by any inode
- The number of existing files reported by the superblock must be correct
- The number of existing directories reported by the superblock must be correct

Your program must work properly for 1KB, 2KB and 4KB block sizes.

Example VDI files that are correct (except for inconsistent superblock copies) and ones with various errors will be made available. These will be a mix of 1KB, 2KB and 4KB block sizes.

Extra credit options

For extra credit, you may do almost any combination of the following:

- For 3%, correct inconsistent copies of the superblock and block group descriptor tables. That is, make them match the master copy in block group 0.
- For 3%, mark all unreachable inodes and data blocks as unused. Adjust free and used block and inode counts to match. *You cannot do this along with either of the next two options.*
- *Graduate students must implement this.* For 5%, for each unreachable inode marked in-use, see if its data blocks are orphans — unreachable but marked in-use.
 - If they are orphans, create an entry in the **lost+found** directory using this inode
 - Otherwise, mark the inode as unused
 - Recheck the filesystem after doing this
- For 10%, for any other orphaned data blocks — unreachable but marked in-use — create an entry in **lost+found** using a new inode and place the orphaned data blocks under that inode. Allocate indirect blocks if necessary. Make sure that counts are correct and recheck afterward.
- *Graduate students must implement this.* For 5%, adapt your program to work with variable-sized VDI files.

What to turn in

Turn in your source code and instructions on how to build and run your program.