INFO0940-1: Operating Systems

Assignment 4: faster cmp command

Academic year 2018-2019

Abstract

In this assignment, students will improve the performance of the `cmp` command by extending the ext4 file system.

Students will work in teams of 2 students.

1 Motivations

The `cmp` command compares 2 different files and shows the differences between them. This sort of workloads need to `read( )` the entire those 2 files in the worst case. When the file sizes are sufficiently large, it takes several seconds to minutes for just `read( )`ing the file data from the disk. Furthermore, the `cmp` must compare whole contents of those 2 files. This workload consumes a lot of CPU cycles. Because of these reasons, it may take long time for the `cmp` command to finish the comparison of 2 different files.

In this project, we solve this problem by extending the ext4 file system. The core idea is to maintain a hash tree (Merkle tree) for every file. When we compare 2 files, we start from checking the hash trees of them first. If the root nodes' hashes are the same, it means that those 2 files' contents are same. If there is any difference between the 2 files, corresponding nodes of the hash trees' values will be different. After the comparison of the hash trees, we can know *where* are the differences. The last thing we need to is to `read( )` the different blocks and confirm which byte is how different. In this procedure, we `read( )` only the file blocks which contain differences. As a result, we can reduce the amount of read from the disk and amount of data to be compared with. By decreasing these disk I/O and CPU overheads, we try to improve the performance the `cmp` command.

2 Requirements

The following requirements must be satisfied:

- Please implement a user-space application named `cmp` whose output is similar to the default `cmp` command.

- We only ask you to implement the `-l` option of the default `cmp` command. (Please refer the man page.)
- Your `cmp` command start from comparing hash trees of 2 files in order to find out differences.
- Please maintain a hash tree for every file by your ext4 extension.
- Please save the hash trees on the disk.

3 Specification

- The file name to be uploaded to the submission platform : src.tar.gz
- The directory name to be compressed by tar : src
- The binary file name of your ext4 kernel module : ext42.ko
- The location of your `cmp` application : src/apps/cmp

Please provide a Makefile in the `src` directory. The Makefile should produce the `ext42.ko` binary at `src/ext42.ko`. Please prepare another Makefile at `src/apps/Makefile` for compiling your `cmp` binary. Our test script will perform `make -C src` and `make -C src/apps`.

4 Evaluation and tests

Your program can be tested on the submission platform. A set of automatic tests will allow you to check if your program satisfies the requirements. Depending on the tests, a **temporary** mark will be attributed to your work. Note that this mark does not represent the final mark. Indeed, another criteria such as the structure of your code, the memory management, the correctness and your report will also be considered.