Project 01

Started: Oct 30 at 3:17pm

Quiz Instructions

Introduction

In the next several class sittings you will be working on implementing an in-memory file system simulator. You will then integrate it with **FUSE** **(http://fuse.sourceforge.net/)** to allow users to interact with your simulated file system using the Linux file system interface. There will be no distinction between interacting with the simulated file system and any other file system available to the users in the operating system. You will be able to explore the simulated file system, create and delete files and folders, inspect the folders and files, open and close the files and folders, and to write and read the files.

The whole project is worth 100 points distributed evenly among four steps; i.e., you will get 25 points for completing each of of the steps.

Step 1

25 points

The file **simfs.zip** contains declarations of the data structures and functions that you must use in this assignment.

The file system volume consists of three parts:

- the superblock,
- a bit vector for maintaining the free storage on the volume, and
- the storage comprised of blocks.

The seed code defines sizes of all the components of the volume.

The superblock holds information about the file system such as the number of blocks, the size of a block, and the reference to the block holding the root folder of the file system.

The bitvector that spans a number of blocks holds bits that indicate whether the corresponding blocks in the volume are free or not; hence, there are as many bits in the bitvector as there are blocks in the storage. You will need to use fast bit-wise operations to search and modify bits in the bitvector.

The storage consists of blocks that are used by folders and files. Each block can hold either a folder or a file descriptor, an index to other blocks, or raw data (file content in bytes).

A file-descriptor node holds meta-data information about a file or a directory such as its size, access rights, creation time, access time, and modification time, along with a pointer to the actual content of the file or the directory. The superblock contains a reference to the block holding a file descriptor for the root folder of the volume.

In a block with the type of a folder, the size field indicates the number of files in the folder, and the block reference field points to the index block that holds references to all files or folders in this folder. Each reference is an index to the block holding the file descriptor of the corresponding file or a folder. Of course, each of the subfolders may in turn hold other files or folders, and so on. There is an upper limit on the number of files or folders in a single folder.

In case of a block with a type of a file, the size field in the file descriptor indicates the actual size of the file. The block reference either points to a single data block if the size of the file is less than the size of a block (less the space needed for the type), or to an index block holding references to data blocks for larger files.

In this step, implement functions to:

- create the file system,
- create a file or a folder,
- delete a file or a folder, and
- obtain file information.

The files are predefined for you in the code with extensive descriptions.

Next ▸

Quiz saved at 2:44pm Submit Quiz