Simple Filesystem Implementation Report

Marco La Gamba

July 2018

FileSystem Implementation

This project consists in a simple implementation of a filesystem. First i implemented a bitmap, to store the used blocks on disk and tested it. Then i proceeded with the disk driver modules implementation. With a simple API they let me write/read a block on disk, which is represented by a simple file, initialized with the disk size on the the disk driver init function, which also mmaps the metadata containing the bitmap and other info about the filesystem to store them in the first blocks on disk. I also provided some easy functions, which calls the bitmap methods to get a free block or set a block in the bitmap in disk. After that i implemented the filesystem functions: create, open, remove, read, write, seek in a file and mkdir, changedir, remove with some useful helper functions, such as findfileindir, that must be called before creating a file, to check it doesn't already exist. I also added a funcion printTree to show the content of the filesystem. I marked with "AGG:" all the methods added to the initial specifics.

I provided two structures to make operations on file and directories, FileHandle and DirectoryHandle, which stores an in memory copy of the FirstFileBlock and CurrentBlock (set to null, if it is the first file block) , in addition to other useful fields, such as pos in file that store the current position in a file, or block num which store the block number on disk of the current block.

Test

To run tests:

-bitmap tests: in directory "bitmap" run

```
./bitmap-main
```

-diskdriver tests: in "/" run

```
./disk_driver_test
```

-simplefs tests: in "/" run

```
./simplefs_test -h
```

-simplefs weird cases: in "/" run

```
./simplefs_test_weird_cases
```