Simple File System
Operating Systems, 2021

Dario Petrillo, 1839609

1 Introduction

This project implements a simple userspace filesystem. Allocated blocks are kept in linked lists for each file/directory, and a bitmap stored at the start of the disk image keeps information on whether each block is allocated or not. The bitmap is then used to find a new unallocated block (or report that none exists) when necessary

2 Files

The project is subdivided in various files:

- `include/bitmap.h, src/bitmap.c` - Bitmap implementation, used in the disk driver to keep track of allocated blocks and to search for free ones
- `include/disk_driver.h, src/disk_driver.c` - Low level disk driver, handles block allocation and reading/writing to individual blocks
- `include/simplefs.h, src/simplefs.c` - Main filesystem implementation. Handles opening and formatting filesystem images, file/folder creation/deletion, reading/writing from files and moving in the folder structure
- `include/util.h` - Error handling macros
- `shell/shell.c` - Interactive shell
- `tests/*` - Unit tests

3 How to run

Compile: `make`

Run tests: `./run_tests.sh`

Run interactive shell: `./run_shell.sh`

3.1 Interactive shell

mkdir <dir>	create directory <dir> in the current directory
touch <file>	create empty file <file> in the current directory
cd <dir>	move in directory <dir> from the current directory
ls	print the contents of the current directory
tree	recursively print the contents of the current directory
cat <file>	print the contents of file <file>
write <file> <data>	append <data> at the end of <file>, creating it if necessary
rm <file|dir>	remove the specified file or directory
format	format the filesystem
help	print this message
exit	exit the shell

Table 1: Interactive shell commands

4 Source code

The source code for the project is hosted at https://github.com/dp1/simple-file-system