Monster Mash

A UNIX-like shell and filesystem

Alok Hota & Mohammad Raji

March 5, 2015

1 About

Monster Mash is a UNIX-like shell and virtual filesystem. Its name was originally MASH, which stood for Mohammad and Alok's SHell. This name was taken however, so we added the word monster, referencing the hit 1962 single by Bobby Pickett. It supports various simple commands similar to basic UNIX commands. The underlying filesystem is based on UNIX's inodes and data block structure.

2 Installation

The repository is available on Github (`https://github.com/ahota/monstermash`). To build Monster Mash, simply run `make`. This will create two executables, `mmash` and `client`.

3 Usage

The `mmash` executable will start the shell and filesystem. The following options are supported:

- -s [port]: Runs `mmash` as server on [port]. If port is not specified, `mmash` will default to port 1962 (the year *Monster Mash* was released as a single.)
- -d: Runs `mmash` in debug mode

Running `mmash` with no options starts it locally.

The following commands are supported by `mmash`:

- mkfs: Formats and (re)generates the filesystem. This is run automatically when starting `mmash` locally or when a client connects. This creates a root directory (/).
- open <filename> <r|w|rw>: Opens a file for reading (r), writing (w) or both (rw) depending on the flag given. If a file is opened with the w or rw flag and does not exist, it will be created. After a file is opened, a file descriptor is printed for later accesses.
- read <fd> <size>: Reads size bytes from an open file with file descriptor fd.
- write <fd> <string>: Writes string to an open file with file descriptor fd.
- seek <fd> <offset>: Seeks to offset (bytes) in file with file descriptor fd.
- close <name>: Closes file in the current directory called name.
- mkdir <name>: Makes a directory called name in the current directory. The name can have spaces if it's surrounded by quotes.
- rmdir <name>: Removes the directory called name from the current directory. The directory must be empty.
- cd <path>: Changes the current directory to path. Path must be relative and can have spaces if surrounded with quotes.
- link <target> <link name>: Creates a link called link name to target. Paths to link and target can be relative.
- unlink <name>: Unlinks a link called name or deletes a file called name. If the target data has zero links to it, it is deleted.
- stat <name>: Provides name on disk, inode ID, type, number of links, blocks allocated and the total data size of element called name.

- ls: Lists all elements in the current directory.
- cat <name>: Prints the contents of the file called name.
- cp <src> <dest>: Copies the file or link called src to a new file or link called dest. src and dest can be relative paths with spaces if surrounded by quotes
- tree: Lists the contents of all directories hierarchically from the current directory along with their total file size.
- import <src> <dest>: Imports a file called src from the host to a file called dest on the `mmash` filesystem.
- export <src> <dest>: Exports a file called src from `mmash` to the host filesystem as dest.

4 Architecture

5 Limitations

Currently `mmash` supports 1024 inodes as a maximum. This means there cannot be more than 1024 elements on the filesystem, including the root directory. The total inode table size is 8KB. The disk itself is 100MB total, less the inode table and block meta-data size, leaves 99.06% of the disk for data.

In the current version, the filesystem is not persistent across runs and is erased every time `mmash` is started.