Assignment 0: Getting Started with `file(1)`

Software Development 2017
Department of Computer Science
University of Copenhagen

Peter, Tobias & Xueying

10. september 2017

1 Compiling and running the code

1. Unzip `src.zip`.
2. Navigate to the `src` directory.
3. Type the command: `make` in a UNIX-like command line to compile the `file`-program.
4. Type the command: `bash test.sh` in a UNIX-like command line to run the tests.

2 Our implementation

The purpose of `file` is to determine the type of a given file.

The program should take one argument only, the path to a file, and open the file in read-only mode, if the file exists. For some cases that the file exists, yet not accessible, an error message will be delivered. When the program takes no argument or more than one argument, it returns a usage message.

2.1 Handling empty files

If `file` is able to open a stream of the given file, it will first determine if the file is greater than a certain size, thus resolving whether the file is empty. If the file isn't empty, it will reset the pointer and examine the contents of the file as described below.

2.2 Discerning between ASCII and data via a while loop

The while loop iterates through the file and with the help of two if statements checks the following:

- 1. That the character being checked, is not at the end of the file, so the EOF character is not included in the next if statement.

- 2. That the character is a non-ASCII character. If it is the while loop terminates, and the file will be considered a data file.

When the while loop is over and no non-ASCII characters have been found, the file will be considered an ASCII file.

3 Testing

We have been supplied `test.sh` to handle our tests and written additional tests to supplement those provided. Specifically, tests have been written to test the behaviour of `file` with files containing various combinations of hexadecimal, octal and decimal.

Worth mentioning is that most files are written to using `printf`, but in one case `echo` had to be used in order to produce an `ASCII`-file (and not a "very short file"). In another case we create an empty file using `touch`.