A0 - Getting started with file(1)

Bjørn Christian Vedel Bennetsen &
Kalle Kromann

10 September 2017

1 Compilation and tests

To compile the code,

```
./src/$ make
```

To run the tests,

```
./src/$ bash test.sh
```

Including some other tests, we test all the bytes from 1 to 127 each in separate files. These files are named accordingly to the expected result.
Some of our tests fail on purpose to showcase some of the places where file(1) prints a string we were not asked to handle. Some of the failed results are,

- ASCII text, with no line terminators
- very short file (no magic) - We get this result with single byte files
- ASCII text, with very long lines
- ASCII text, with very long lines, with no line terminators
- UTF-8 Unicode text
- Sendmail frozen configuration - We get this result with byte 38 = &
- And a long message when we test on the binary ./file describing which system it was compiled on.

We know that a empty file contains 0 bytes. We have verified that `printf` can create such a file, and thus it has already been tested. We also created a couple alternatives to create an empty file and verified that we get the same result.
At the end of the test, the total number of tested files are printed to the console.

2 Design and implementation

We use `fread()` to read from a **FILE**-stream that we get from opening the file with `fopen()`. If the file is not empty, we read one byte at a time through to the end of the file and simply check if the byte is in the ASCII set. If it isn't, the file is of the type **data**. If an error occurs upon using `fread()`, the error is handled using the set `errno` variable in `<errno.h>`.