1. Introduction
   1. Project overview

In this project two programs were implemented: my-cat and my-grep. They are simple versions of commonly used commands of cat and grep.

* + 1. my-cat

The program reads a file as specified by the user and prints its contents.

* + 1. my-grep

The program looks through a file, line by line, trying to find a user-specified search term in the line. If a line has the word within it, the line is printed out, otherwise it is not.

* 1. Features
     1. my-cat

The program can be invoked as follows:

prompt> ./my-cat file.txt

The program has following details:

* The program can be invoked with one or multiple files on the command line; it just prints out each file in turn.
* In all non-error cases the return code is 0.
* If no files are specified on the command line, the program just exits and returns 0.

The program handless the following errors:

* If opening a file with *fopen()* fails, the program prints out the message “my-cat: cannot open file” and exits with return code 1. If multiple files are specified on the command line, the files should be printed out in order until the end of the file list is reached or an error opening a file is reached.
  + 1. my-grep

The program can be invoked as follows (searcing term foo from the file bar.txt):

prompt> ./my-grep foo bar.txt

The program has following details:

* The program is always passed a search term and zero or more files.
* The matching is case sensitive.
* Lines can be arbitrarily long.
* In all non-error cases the return code is 0.
* If a search term, but no file, is specified the program reads from the standard input.
* If the program is passed an empty string as a search term it matches ALL lines.

The program handless the following errors:

* If no command-line arguments are passed, the program prints an error message “my-grep: searchterm [file ...]” and exits with error code 1.
* If a file cannot be opened the program prints an error message “my-grep: cannot open file” and exits with error code 1.

1. Implementation
   1. my-cat

The program my-cat checks in the main function if more than one argument is given and sends each file one by one to be printed out to the *readFile()* function. If no files are specified (argc = 1) the program exits with return code 0.

A computer screen with text

Description automatically generated

The program then in the *readFile()* function opens a file with *fopen()*, reads lines one by one with *getline()*, and prints each line to *stdout* with *fprintf()*. If an error occurs when opening a file, the program exits with return code 1.

A computer code on a black background

Description automatically generated

An example of the output of the program with on file:

A screen shot of a computer

Description automatically generated

* 1. my-grep

The program works by reading all the lines that have the search term in them to a linked list and then prints them out.

A screen shot of a computer code

Description automatically generated

Example of the program when passed a file with the following text in it:

Hello

this

is

a file

