Relatório de projeto checkFile de Programação Avançada

Micael Ferreira Marques nº2201743

Sara Filipa dos Santos Martins, nº2201757

Micael Ferreira Marques (2201743) e por Sara Filipa dos Santos Martins (2201757) declaram sob compromisso de honra que o presente trabalho (código, relatórios e afins) foi integralmente realizado por nós, sendo que as contribuições externas se encontram claramente e inequivocamente identificadas no próprio código. Mais se declara que os estudantes acima identificados não disponibilizaram o código ou partes dele a terceiros.

O checkFile é um programa criado em C, que permite analisar a extensões e detetar se são de facto o que a extensão diz ser. É compatível com as extensões PDF, GIF, JPG, MP4, ZIP e HTML. Se não for nenhum destes, o mesmo vai indicar que não é suportado pelo programa.

Modo -f, --file	Totalmente Operacional
Modo -b, --batch	Totalmente Operacional
Modo -d, --dir	Totalmente Operacional
Modo -h, --help	Totalmente Operacional

Modo -f:

Neste modo, é possível analisar mais do que um ficheiro de cada vez, utilizando novamente o -f <ficheiro> seguido de outro -f <ficheiro>.

- e.g: ./checkfile -f <ficheiro1> -f <ficheiro2>

No entanto, não é possível analisar ficheiros de modos diferentes como -f junto com -b, ou, -f junto com -d.

A principal função utilizada neste modo, é a "optionFile" que após analisar se o ficheiro realmente existe (com a função "checkIfFileExists") vai fazer a função "extensionfile" e utilizar o fork() para criar um novo processo e conseguirmos usar o execlp(), sem alterar o processo pai, para assim conseguirmos continuar o programa. No processo pai, é analisada a extensão e o tipo do ficheiro que depois vai dar o resultado [OK], [MISMATCH], [INFO]. Neste modo, é possível utilizar o sinal SIGQUIT (Ctrl + \), para obter o PID do processo.

Modo -b:

Neste modo, apenas é possível analisar um ficheiro de cada vez. Neste modo também é possível utilizar o sinal SIGUSR1, com o **comando** 'kill -s SIGUSR1 <pid>' numa outra cmd. Este <pid> é obtido quando se manda qualquer tipo de comando para o checkFile, que vai informar o utilizador de imediato qual é o PID do processo. O PID também pode ser obtido ao fazer uso do sinal SIGQUIT (Ctrl + \).

Ao fazer -b <ficheiro>, o programa vai abrir e ler o ficheiro linha a linha, como anteriormente referenciado, a cada linha vai analisar se o ficheiro

existe, e se existir, que tipo e extensão o mesmo tem, utilizando as funções “optionBatch”, “checkIfFileExists”, “extensionfile”, por fim, apresenta um sumário com o número de ficheiros lidos, os ficheiros [OK], ficheiros [MISMATCH] e os erros ocorridos.

Quando se faz o SIGUSR1, o processo vai ser parado e vai indicar o número e o nome do ficheiro que foi lido.

Modo -d:

Neste modo, apenas é possível analisar uma pasta de cada vez. Ao abrir o diretório, é analisado cada ficheiro, usando a base dos outros modos (“checkIfFileExists”, “extensionfile”), e ignora outros subdiretórios. No final também apresenta um sumário com o número de ficheiros lidos, os ficheiros [OK], ficheiros [MISMATCH] e os erros ocorridos.

A função principal neste modo é a “optionDirectory” que “abre” a pasta do diretório e analisa um a um. Caso este não consiga abrir apresenta sempre uma mensagem de erro para o utilizador saber.

Modo -h:

O modo -h apresenta um breve resumo do que cada um dos modos faz, como também apresenta o nosso nome e número de estudante.

Observações:

Foi acrescentado uma linha que ajuda o utilizador a saber qual é o PID do processo, caso necessite para utilização dos sinais.

Ficheiros vazios não contam para o total de contagem de ficheiros, pois estes não chegam a ser analisados.

Para a contagem dos erros e dos ficheiros para fazer o sumário, não são utilizamos dois casos (ambos os casos não são suportados pelo checkFile e o último nem permite analisar o ficheiro):

a) quando a extensão do ficheiro não é suportada, isto é, quando aparece a mensagem [INFO];

b) quando o ficheiro não tem extensão no nome como, por exemplo: “makefile”.

Webgrafia:

https://stackoverflow.com/questions/7292642/grabbing-output-from-exec

https://stackoverflow.com/questions/26747590/sigint-and-sigquit

https://www.delftstack.com/pt/howto/c/c-check-if-file-exists/

https://www.gnu.org/software/gengetopt/gengetopt.html#Parser-function-additional-parameters

https://stackoverflow.com/questions/7788934/how-to-convert-char-pointer-into-char-in-c-open-vms

https://www.decodeschool.com/C-Programming/File-Operations/C-Program-to-read-all-the-files-located-in-the-specific-directory

https://www.tutorialspoint.com/c_standard_library/c_function_strftime.htm

https://stackoverflow.com/questions/423626/get-mime-type-from-filename-in-c

https://www.lifewire.com/file-linux-command-unix-command-4097142

https://stackoverflow.com/questions/13566082/how-to-check-if-a-file-has-content-or-not-using-c

Material teórico e prático de Sistemas Operativos e Programação Avançada.