

---

# Rapport de projet

## Système

Projet "find the cat"

Edgar LAUGEAIS, Fabien BUCHER, Victor DE  
MOURA NETTO  
18 décembre 2022

# 1 Introduction

Ce mini-rapport est réalisé dans le cadre du module système. Le groupe se compose de trois personnes, Edgar LAUGEAIS, Fabien BUCHER et Victor DE MOURA NETTO.

## 1.1 Objectif

L'objectif de ce projet est d'écrire un programme `ftc` qui permette de réaliser quelques opérations dans un répertoire, comme rechercher un fichier, tous les fichiers avec une taille spécifique, tous les fichiers qui ont été modifiés en une date spécifique, etc. L'application doit fonctionner comme suit :

```
FTC STARTING-POINT [-OPTION [PARAMÈTRE]]
```

## 2 Choix de conception

### 2.1 Structure

Le programme se compose d'un fichier `main.c`. Ce fichier contient la majorité des tests de vérification d'erreur et génère donc les messages d'erreur. Il contient une fonction `main` permettant de vérifier la commande qui a été rentrée, c'est-à-dire quel flag a été utilisé. Le fichier exécute les différentes fonctions pour répondre aux différentes fonctionnalités du projet.

Plusieurs headers `.h` ont été créés pour définir plusieurs fonctions utilisés dans le fichier `main` et répondre aux fonctionnalités demandées. Les fichiers `parameter.c` et `parameter.h` définissent des fonctions qui sont utilisés dans le fichier `main.c`. Ces fonctions permettent de vérifier les paramètres des fichiers. D'autres fichiers ont été créés tels que `file_size.c` permettant d'obtenir la taille d'un fichier ou bien `regex_evaluation`.

Le code a été organisé comme suit :

#### 1. `main.c`

- **VOID** PRINT\_PATH(BOOL COLOR, BOOL ONLY\_PATH, CHAR \*FULL\_PATH, CHAR \*FILE\_NAME)  
Cette fonction fait un print du fichier. Si le flag “-color” a été initialisé, l'impression aura une couleur.
- **INT** SHOW\_FILES\_IN\_SPECIFIC\_PATH(PARAMETER PARAMETER, CHAR \*PATH)  
Va vérifier si dans le chemin, il s'agit d'un fichier ou d'un dossier. Il va aussi vérifier si le flag -dir est initialisé. Dans le cas où le flag -dir est initialisé, seulement les répertoires vont être imprimés, pas les fichiers.
- **VOID** VERIFY\_IF\_CAN\_BE\_PRINTED(PARAMETER PARAMETER, CHAR \*FULL\_PATH, CHAR \*FILE\_NAME)  
Cette fonction essaie tous les paramètres qui sont initialisés dans le struct *Parameter*. Elle permet de vérifier si les conditions des flags ont été respectées. Si les conditions ont été respectées, elle va imprimer le nom du fichier.

#### 2. `file_information.h`

- **TIME\_T** GET\_FILE\_LAST\_ACCESS(CHAR \*FULL\_PATH)  
Cette fonction recherche quelle est la date de la dernière modification du fichier, pour la flag -date.
- **LONG INT** GET\_FILE\_SIZE(CHAR \*FULL\_PATH)  
Cette fonction reçoit le chemin comme paramètre et elle retourne la taille du fichier.
- **INT** GET\_OCTAL\_CHMOD(CHAR \*FULL\_PATH)

#### 3. `regex_evaluation.h`

- **BOOL** EVALUATE\_REGEX\_PERFECT\_MATCH(**CHAR** \*EXPRESSION, **CHAR** \*FILE\_NAME)  
Cette fonction permet de vérifier la correspondance *REGEX* entre une expression et le nom d'un fichier.

#### 4. parameter.h

- **VOID** VERIFY\_NAME(**INT** \*SUCCESSFUL\_PARAMETERS, **CHAR** \*FILE\_NAME, **CHAR** \*VALUE)  
Cette fonction permet de vérifier si le file\_name (nom du fichier) est égal au value. Le value peut être le nom exact du fichier ou si une expression régulière.
- **VOID** VERIFY\_MIME(**INT** \*SUCCESSFUL\_PARAMETERS, **CHAR** \*FULL\_PATH, **INT** SIZE\_TYPE, **CHAR** \*VALUE)  
Le mime est un type de médias, à l'origine appelé type MIME, est un identifiant de format de données sur internet en deux parties. Dans cette fonction, on va vérifier si le value (qui peut être le type ou le type/sous-type) est égal au type MIME. Pour faire cela, on utilise la bibliothèque MegaMimes, qui va vérifier quel est le type et comparer avec le value.
- **VOID** VERIFY\_CTC(**INT** \*SUCCESSFUL\_PARAMETERS, **CHAR** \*FULL\_PATH, **CHAR** \*VALUE)  
Cette fonction va ouvrir tous les fichiers et elle va chercher la valeur *value*, qui est une expression régulière.
- **VOID** VERIFY\_SIZE(**INT** \*SUCCESSFUL\_PARAMETERS, **CHAR** \*FULL\_PATH, **INT** FILE\_SIZE, **CHAR** \*VALUE)  
Cette fonction permet de vérifier le paramètre rentré en flag correspond à la taille d'un fichier.
- **VOID** VERIFY\_DATE(**INT** \*SUCCESSFUL\_PARAMETERS, **CHAR** \*FULL\_PATH, **CHAR** \*VALUE)  
Cette fonction vérifie si la valeur du drapeau est valide. Pour ce faire, il vérifie quelle est la date de modification du fichier pour la comparer à la date actuelle. Si le drapeau est "now", "today", "yesterday" et "this month", la valeur sera traitée d'une manière différente, en comparant des chaînes de caractères avec les dates.
- **VOID** VERIFY\_PERMISSION(**INT** \*SUCCESSFUL\_PARAMETERS, **CHAR** \*FULL\_PATH, **INT** INT\_SIZE, **CHAR** \*VALUE)  
Cette fonction compare la valeur du flag -perm avec la permission du fichier, en format d'octet.

## 2.2 Bibliothèque MegaMimes

La bibliothèque choisie pour trouver le type MIME d'un fichier était MegaMimes. L'idée d'avoir une bibliothèque pour réaliser cette fonction est de gagner du temps afin d'optimiser les autres parties du code. Pour l'implémenter, le code .h et .c a été téléchargé sur [git](#), et mis dans le projet.

### 3 Extensions développées

- **-color** : ce flag permet l'écriture en couleur du nom du fichier par rapport à son path et colore différemment les fichiers des dossiers  
Cette extension permet de très facilement faire la différence entre les dossiers, les fichiers et leurs chemins. Elle facilite la visualisation et l'utilisation du programme.
- Extension du comportement du flag **-date** par les mots clé now, today, yesterday, this month....  
Cette extension, plutôt simple à mettre en place est très pratique notamment pour retrouver un fichier. Permet de gagner du temps et
- **-perm** : ce flag permet de rechercher selon le type de permission du fichier  
La permission est un autre moyen intéressant pour retrouver un fichier à intégrer au programme.
- **-ou** : ce flag indique au programme de faire un "OU" logique sur l'ensemble des conditions au lieu d'un "ET"  
Cette extension facile à implémenter peut-être très utiles pour certaines recherches.

## 4 Difficultés rencontrées

- Le groupe se compose de 3 personnes à la demande de madame Collin parce qu'Edgar LAUGEAIS qui vient de la faculté de pharmacie n'avait jamais codé en C avant cette année. Cependant nous avons eu des difficultés pendant le projet parce que Fabien a eu des problèmes de santé durant l'année. Il a malgré tout fait son maximum pour aider le groupe. Edgar LAUGEAIS avec ses lacunes en C a dû se contenter des fonctionnalités les plus simples du programme et s'est concentré sur la rédaction de ce rapport. Victor a donc dû gérer beaucoup de choses pendant le projet au niveau de l'implémentation.
- Nous avons parfois eu des difficultés pour comprendre ce que devait faire certaines des fonctionnalités de l'outil.
- Nous avons demandé l'aide de certains autres groupes afin de nous assurer qu'ils avaient compris les consignes comme nous.
- L'une de nos difficultés fut de passer dans les pipelines du gitlab. En effet, on ne savait pas exactement quel était le format correct pour faire les print.
- Une autre difficulté fut de trouver une bibliothèque pour trouver le type MIME d'un fichier. Nous avons finalement opté pour MegaMimes.

## 5 Temps passé

### 5.1 Conception

- Victor : 4h
- Edgar : 4h
- Fabien : 4h

### 5.2 Implémentation

- Victor : 30h
- Edgar : 20h
- Fabien : 10h

### 5.3 Tests

- Victor : 3h
- Edgar : 2h
- Fabien : 2h

### 5.4 Rédaction du rapport

- Victor : 2h
- Edgar : 6h
- Fabien : 0h