



Rapport Projet "find the cat"
RS - Réseaux et Systèmes
Deuxième année

Caroline Wang, Maxime Wirth

Decembre 2022



Introduction

Le projet avait pour but de créer une commande *ftc*, devant reproduire le comportement de la commande bash *find*, pour permettre de retrouver un fichier dans une arborescence. Un certain panel de fonctionnalités était obligatoire, et des extensions étaient possibles.

1 Fonctionnement de notre commande *ftc*

La commande *ftc* devait pouvoir être appelée avec une commande de la forme : **ftc starting-point [-option [paramètre]]**.

Nous avons implémenté toutes les fonctions obligatoires, et les flags *-perm*, *-ou* et *-color*.

1.1 Fonctionnement général de la commande

Quand la commande est appelée, on teste déjà si le premier flag est *-test*. Si c'est le cas, on ne regarde que les 2 flags suivants. Il se passe déjà ici un premier test sur les flags, où l'on regarde si le flag après *-test* existe. Si le premier flag n'est pas *test*, alors nous faisons le parsing de la ligne de commande. Nous utilisons pour cela 2 structures de données : *flag* et *nb.Flags*.

Une structure *flag* contient l'information sur un flag : quel flag il est, dans un champ *flag*, la valeur de son paramètre (s'il n'en a pas, on met ici le nom du flag) dans un champ *valeur* et son type (nous avons défini -2 si une erreur se produit lors du parsing, -1 pour le flag de fin, 0 s'il est inactif, 1 si le flag a un paramètre et 2 sinon) dans un champ *flag.type*. Une erreur se produit pendant le parsing si le paramètre sur lequel on est n'existe pas, ou si la valeur d'un flag est fausse dans la ligne de commande.

Une structure *nb.Flags* contient le nombre de flags *-color*, *-ou*, *-name*, et *-dir* avec et sans paramètres. Elle nous sert à savoir si le flag *-ou* est présent, sans avoir à rechercher dans la ligne de commande, ou à pouvoir rendre les autres flags inopérants au cas où on a au moins un flag *-dir* sauf en présence d'un flag *-ou*.

Après avoir obtenu ces 2 structures, on fait un parcours récursif sur **starting-point** et on fait les tests sur tous les flags de la structure *flag*. Au début des tests sur les flags, on instancie une variable *tests_a_valider* (qui va compter le nombre de tests subis) et *tests_valides* (qui va compter le nombre de tests réussis). À chaque flag sur lequel on doit tester un chemin (donc *-size*, *-date*, *-name*, *-mime*, *-ctc* et *-dir*), on incrémente *tests_a_valider* et on incrémente *tests_valides* que si le test est réussi.

Enfin, on imprime le chemin en ligne de commande si *tests_a_valider == tests_valides*. Dans le cas de la présence d'un flag *-ou*, il faut que *tests_a_valider == 0 || tests_valides >= 1*.

Toutes les extensions rajoutées ont une fonction principale qui les traite, appelée *check_"nom du paramètre"*, qui incrémente *test_valides* de 1 si le test

est positif, 0 sinon.

1.2 Option -size

Lorsque nous testons pour l'option -size, nous regardons d'abord si le paramètre d'entrée correspond à la forme souhaitée : possiblement + ou - au début, avec un nombre (correct, donc qui ne commence pas par 0), et une lettre, c, k, M ou G, grâce à la fonction *is_size_param*. Si l'argument est correct, l'argument est converti en bits, puis on le compare avec la vraie taille du fichier que l'on regarde (grâce à la bibliothèque sys/stats).

1.3 Option -date

Pour l'option -date, nous testons aussi de la validité de l'argument : potentiellement un +, puis un nombre valide et enfin une lettre, m, h ou j avec la fonction *is_date_param*. Si l'argument est correct, on utilise la fonction *check_date* : on convertit l'argument en secondes et on effectue le test voulu par le paramètre sur le chemin, grâce à la bibliothèque time.h.

1.4 Option -name

Ici, on compare directement l'argument de -name avec le nom du fichier que nous sommes en train de traiter. Nous avons étendu le fonctionnement de -name aux regex, donc nous testons la présence de certains caractères (comme) dans le paramètre pour savoir comment le traiter.

1.5 Option -mime

Ici, on utilise la bibliothèque MegaMimes.h pour reconnaître les types mime associés à une extension de fichier. Donc, pour un fichier, on regarde d'abord si l'option en entrée est un type mime correct (grâce à la table dans MegaMimes.c, dans la fonction *is_mime_param*), on récupère l'extension du fichier (s'il n'en a pas, le test est négatif), et on compare les types mime associés à cette extension avec le paramètre de -mime.

1.6 Option -ctc

Pour cette option, on regarde les lignes du fichier texte une à une en regardant si le mot recherché se trouve dedans. Nous avons aussi implémenté les regex pour -ctc, donc si le paramètre est une regex, on essaie de matcher chaque ligne du fichier.

1.7 Option -dir

Cette option a 2 formes : avec et sans paramètres. Si -dir est appelé sans paramètres, alors nous faisons le parcours récursif en imprimant juste les dossiers.

Sinon, on applique aussi *check_name*, mais sur les dossiers cette fois. Aussi, si *dir* est appelé avec paramètres, tous les autres flags sont rendus inopérants et on ne les teste pas, sauf en présence d'un flag *-ou*. Par exemple, la commande *./ftc . -dir src -name ftc.c* affiche tous les dossiers nommés "src", trouvés en partant du répertoire courant, et annule le flag *-name*, tandis que la commande *./ftc . -dir src -ou -name ftc.c* affiche tous les dossiers nommés "src" et tous les fichiers nommés "ftc.c", trouvés en partant du répertoire courant.

2 Extensions

2.1 Option *-perm*

L'option *-perm* est appliquée pour les arguments d'un format particulier : une suite de trois chiffres entre 0 et 7. On vérifie tout d'abord que le paramètre donné est bien de ce format. Ensuite, on utilise la librairie *sys/stat* pour récupérer les permissions du fichier en cours de traitement, et on teste la compabilité de ce résultat avec le paramètre en entrée.

2.2 Option *-ou*

Son fonctionnement a été décrit au-dessus : au lieu d'imprimer un chemin uniquement si *tests_a_valider == tests_valides*, on imprime un chemin si *tests_a_valider == 0 || tests_valides >= 1*.

2.3 Option *-color*

Si l'option *-color* est présente, alors au lieu d'imprimer tout de suite les chemins, on vérifie si le chemin renvoie vers un dossier ou un fichier : si c'est un dossier, on l'imprime en vert, sinon on l'imprime normalement.

3 Difficultés rencontrées

La conception de quelques options laissait place à des incertitudes au niveau de leur fonctionnement ou bien de leur application. Elles ont été confirmées ou infirmées grâce aux tests effectués. Par exemple, l'option *-date* prenait en compte la dernière date de modification, et non d'accès. Pour le savoir, il a d'abord fallu implémenter la fonction correspondante, puis la tester.

4 Bilan du projet

	Caroline	Maxime
Conception	2h	3h
Implémentation	20h	35h
Tests	2h	4h
Rédaction du rapport	0h	1h
Total	24h	43h