Projet sysème
RS
2A TELECOM Nancy - 2022-2023

Rapport de projet

Catch the cat

Membres de l'équipe projet :
BÉNÉ Tom, RICARD Guillaume

Responsable :
RACOUCHOT Maiwenn

Table des matières

1 Choix de conception

1.1 Structures de données

1.1.1 Flags et filtres

Le nombre de flags passés à l'application pouvant varier, nous avons décidé de les stocker dans une liste chaînée. Cela permet de rendre simple l'itération sur l'ensemble des flags et de les convertir simplement en filtres qui permettront de décider quels chemins de fichiers afficher.

1.1.2 Contexte

Afin de faciliter le fonctionnement de `ftc`, nous avons décidé d'introduire un contexte à l'application, qui contient de multiples informations sur l'appel effectué :

— Si on souhaite colorier les résultats

— Si on souhaite suivre les liens symboliques

— Si on souhaite lancer `ftc` en multithread

— Dans le cas où l'on utilise plusieurs flags, s'ils doivent être liés par un ET ou un OU

Le contexte par défaut ne colorie pas les résultats, ne suit pas les liens symboliques, lance `ftc` sur 1 seul thread et considère que les flags doivent être reliés par des ET.

1.2 Librairies

1.2.1 MegaMimes

Afin de réaliser l'option `-mime`, nous avons utilisé la librairie *MegaMimes* de Kobby Owen [1]. Cette dernière nous permet d'obtenir le mime type d'un fichier directement et de mener à bien la comparaison avec le type recherché.

1.2.2 getopt

Afin de rendre les flags passés à l'appication en ligne de commande utilisables, nous avons décidé d'utiliser la librairie `getopt` présente dans la *libc*. Cette librairie permet de lister les flags attendus par l'application et de renvoyer une erreur en cas de flag non présent dans cette liste. Elle permet aussi de préciser quels flags attendent des arguments, et si ceux-ci sont optionnels ou requis.

1.3 Algorithmes

1.3.1 Parcours du système de fichiers

Le parcours du système de fichiers est fait grâce à la librairie `dirent.h`. Nous faisons un parcours en profondeur afin d'avoir le même comportement que la commande `find`.

1.3.2 Application des filtres

Les filtres à appliquer se trouvant dans une liste chaînée, une fonctions va itérer sur cette liste pour chaque fichier parcouru et renvoyer une booléen si le fichier en question correspond à tous les filtres (ou bien au moins un en cas de passage du flag `-ou`).

Afin de gagner en performance, cette fonction renverra `false` dès le premier filtre non correspondant rencontré. En cas de présence du flag `-ou`, elle renverra `true` au premier filtre correspondant. Cela permet d'éviter d'évaluer chaque filtre pour chaque fichier.

1. https ://github.com/kobbyowen/MegaMimes

2 Extensions réalisées

2.1 `-color`

Le flag `-color` permet d'afficher les résultats de la recherche en couleurs. Lorsqu'il est passé à l'application, la valeur `color` du contexte est mise à la valeur `true`.

La fonction d'affichage des chemins de fichier colore les chemins selon les conventions données par la commande `dircolors -p`, basés sur un appel à la fonction `stat` ou l'extension du fichier :

— Bleu pour les dossiers

— Cyan pour les liens symboliques

— Vert pour les fichiers exécutables

— Jaune pour les tubes nommés

— Magenta pour les sockets

— Jaune sur fond noir pour les *block devices* et les *character devices*

— Rouge pour les archives

— Magenta pour les fichiers image et vidéo

— Cyan pour les fichiers audio

Afin d'économiser des performances, les vérification du `stat` et de l'extension du fichier ne sont faites que si le flag `color` a été passé à l'application.

2.2 `-perm`

`ftc` permet également de retrouver les fichiers suivant les permissions. L'appel doit respecter la convention des permissions en octal (644, 700, 754...) sous peine de retourner une erreur.

2.3 `-ou`

Passer le flag `-ou` à `ftc` permet de lister les fichiers correspondant à un des flags passés au lieu de tous. Une option booléenne est mise à `true` dans le contexte lorsque ce flag est passé.

2.4 `-link`

`ftc` est capable de suivre les liens symboliques. Au lieu de renvoyer les informations sur le lien, elle renvoie les informations sur le fichier pointé, exception faite du nom : nous avons considéré qu'il était plus interessant d'avoir le nom du lien que le nom du fichier.

Cette fonctionnalité est implémentée grâce à la fonction `fstatat` est utilisée, avec un flag de `0` ou de `AT_SYMLINK_NOFOLLOW` selon que l'on souhaite suivre les liens symboliques ou non.

2.5 Le chat

Un chat se cache dans le fichier `arbre/economic/pay/trade/fish.css` que l'on trouve via

```
./ftc arbre/ -ctc "chat"
Félicitations, vous avez trouvé le chat. Son nom est AGAMEMNON, original n'est-ce pas (ctc)?
```

3 Répartition des taches

Étapes	Tom	Guillaume
Conception	2h	2h
Implémentation		
Flags	5h	1h
Traversée des fichiers	8h	-
Filtres obligatoires	1h	10h
Extensions	1h	3h
Tests	2h	2h
Rédaction du rapport	3h	3h
TOTAL	22h	21h