
Rapport de projet

Systeme — Find the Cat —

Nicolas Frache, Guillaume Bourgeon

TELECOM Nancy

18 décembre 2022

Table des matières

1	Développement du projet	3
1.1	Extensions développées	3
1.2	Choix de conception	3
1.3	Difficultés rencontrées	4
1.3.1	Difficultés techniques	4
1.3.2	Difficultés externes	5
1.4	Solutions apportées	5
1.5	Déroulement du travail de groupe	5
1.6	Temps de travail	6

1 Développement du projet

1.1 Extentions développées

La première étape de ce projet a été de faire l'option `-test` et le "ls récursif". Cette première étape n'a pas particulièrement posé problème.

```
eleve@TNCY-Linux:~/Documents/rs2022proj245$ ./ftc . -test -name fichier1
La valeur du flag -name est fichier1
```

FIGURE 1.1 – Option test

Nous avons ensuite tenté de développer les options de l'énoncer. L'option `-size` permet de trouver les fichiers en fonctions de leur taille (supérieur, inférieur ou égale). Elle a été difficile à développer mais nous avons fini par y parvenir.

```
eleve@TNCY-Linux:~/Documents/rs2022proj245$ ./ftc tests/_tests_pipeline/arbre -size +9K
tests/_tests_pipeline/arbre/sell/production/meet.css
tests/_tests_pipeline/arbre/sell/production/everybody.bmp
tests/_tests_pipeline/arbre/sell/foreign.mp3
tests/_tests_pipeline/arbre/sell/need.mp4
tests/_tests_pipeline/arbre/sell/light/they.pptx
tests/_tests_pipeline/arbre/sell/light/represent.odp
tests/_tests_pipeline/arbre/sell/light/toward.wav
tests/_tests_pipeline/arbre/put/still/tell/painting.gif
tests/_tests_pipeline/arbre/put/still/tell/eye.webm
tests/_tests_pipeline/arbre/put/still/tell/score.wav
tests/_tests_pipeline/arbre/put/still/tell/performance.webm
```

FIGURE 1.2 – Option size

L'option `ET` permet de réaliser un et logique entre deux options de la fonction `ftc`. De la même manière que `-size`, l'option `-date` permet elle d'afficher les fichiers de l'arborescence en fonction de leur date de dernière modification en affichant les fichiers modifiés avant, après ou au moment de la date. Enfin, les fonctions `-mime` et `-ctc` permettent d'afficher les fichiers selon leur extension et leur contenu avec des expressions régulières.

1.2 Choix de conception

La majorité des choix ont bien sûr été guidés par l'énoncer du sujet qui était très clair sur les fonctionnalités. Le premier choix de développement a été d'écrire des constantes

pour définir les différentes options. Le deuxième choix majeur a été de faire la structure arguments pour pouvoir stocker les options passés en ligne de commande ainsi que les arguments de chacune de ces options. L'utilisation de cette structure a grandement facilité le développement du projet.

Nous nous sommes rapidement rendu compte que devoir commit sur le serveur gitlab pour vérifier que les tests passent bien n'est pas très pratique lors du développement. Donc nous avons téléchargé l'image docker et écrit un script bash pour lancer les tests en local directement.

```
struct arguments_t {
    bool is_def[NB_ARGS];
    char *values[NB_ARGS];
    int nb_args;
} arguments_t;

struct arguments_t arguments;
```

FIGURE 1.3 – Structure pour les options

```
#define W 0
#define TEST 1
#define NAME 2
#define SIZE 3
#define DATE 4
#define MIME 5
#define CTC 6
#define DIR_ARG 7
#define COLOR 8
#define PERM 9
#define LINK 10
#define THREAD 11
#define OU 12
#define TYPE 13
#define NB_ARGS 14
```

FIGURE 1.4 – Constantes pour les options

1.3 Difficultés rencontrées

1.3.1 Difficultés techniques

La principale difficulté de ce projet a été d'appréhender les concepts du langage C utilisés pour implémenter la fonction ftc. Nous avons dû chercher des bibliothèques à utiliser et donc apprendre à les utiliser. La fonction qui convertit l'argument de l'option

-size en entier nous a également posé problème à cause de la conversion de caractère à entier.

1.3.2 Difficultés externes

La difficulté la plus compliquée à gérer a été d'accorder du temps au projet à cause des partiels et des autres projets du semestre qui sont très chronophages. Nous avons également été trois pendant une partie du projet avec Arno Kempf (étudiant en pharmacie). Nous avons essayé de lui expliquer les composants du projet. Ce fut parfois difficile et il a arrêté son année à Telecom. Il n'a pas participé au développement du projet mais a assisté à des réunions. Enfin, Guillaume a eu un problème familial qui a entravé son travail en cette fin de semestre. Nicolas a eu une charge supplémentaire de travail.

1.4 Solutions apportées

Premièrement, nous avons trouvé du temps pour s'organiser au mieux pour le développement du projet. Ensuite, l'entraide entre groupe a permis de palier le problème de la compréhension du C. L'utilisation de bibliothèques C nous a également sauvé de plusieurs heures de développement. Nous avons utilisé la bibliothèque *Dirent* pour avoir accès aux informations sur les fichiers telles que la taille ainsi que pour implémenter le "ls récursif". La deuxième bibliothèque utilisée a été *MegaMime* pour l'implémentation de l'option mime.

```
struct dirent {
    ino_t      d_ino;      /* numéro d'inœud */
    off_t      d_off;      /* décalage jusqu'à la dirent suivante */
    unsigned short d_reclen; /* longueur de cet enregistrement */
    unsigned char d_type;   /* type du fichier */
    char        d_name[256]; /* nom du fichier */
};
```

FIGURE 1.5 – Bibliothèque dirent en C

1.5 Déroulement du travail de groupe

Pour ce projet, nous avons été trois (dont un étudiant en pharmacie qui n'avait jamais fait de C) puis deux. Nous avons développé le projet à deux. Nous avons fait plusieurs réunions pour avancer sur le projet et se répartir les tâches. En plus de cela, nous avons communiqué régulièrement, notamment pour s'aider lors des problèmes de conceptions.

1.6 Temps de travail

Domaines	Nicolas	Guillaume
Conception	4h	3h
Implémentation	32h	28
Réunions	4h	4h
Tests	5h	3h
Rapport final	1h	4h
Total	49h	41h