



UNIVERSITÉ
DE LORRAINE



Lorraine
INP

MODULES RS

RAPPORT : CONCEPTION DE TESH

Matta Karczewski, Guillaume Rozier

Date de rendu : 28 Mai 2017

Sommaire

I. Introduction	2
1. Rappel du sujet	2
2. Contenu	2
II. Déroulement et problèmes rencontrés	3
1. Makefile	3
2. Parsing de la ligne de commande	3
3. Pipe et redirection	3
4. Mise en background/foreground	4
4. Chargement dynamique de dlopen	4
III. Conclusion	4
Répartition du temps de travail	5
Références	5

I. INTRODUCTION

Ce rapport est la synthèse du projet de RS, dont la date de rendu est fixée au 22 décembre 2017.

1. Rappel du sujet

Comme mentionné dans l'énoncé : "L'objectif général du projet est de réaliser un interpréteur de commandes, appelé tesh, très fortement inspiré des shells Unix classiques comme sh, dash, bash, tcsh, zsh."

L'objectif est donc, globalement, d'analyser des commandes entrées par l'utilisateur, de les traiter afin d'obtenir le résultat voulu. Il faut implémenter un certain nombre de commandes (comme la gestion des pipes, de la mise en background de commande, des redirections, de l'historique de commandes...).

2. Contenu

Ce document contient une brève explication de comment le programme fonctionne et de la manière dont nous avons programmé les différentes parties. Ce document n'est donné qu'à titre indicatif car le code est bien commenté et la compréhension n'en est pas difficile.

II. DÉROULEMENT ET PROBLÈMES RENCONTRÉS

Le projet s'est déroulé de manière assez succincte, nous utilisions chacun une branche et nous mettions la branche Master à jour de manière régulière dès que les fonctionnalités que l'on essayait d'implémenter nous semblaient fonctionnelles.

Nous prenions en compte les tests blancs qui nous aidaient à identifier les possibles bugs et nous corrigions/adaptions notre code pour que notre interpréteur de commandes effectue le travail demandé. De manière générale nous avons effectué énormément de tests qui ne sont pas répertoriés mais qui correspondaient aux usages de base des fonctions de récupération et exécution de la commande.

1. Makefile

Afin de pouvoir profiter des premiers tests blancs, nous avons en tout premier naturellement codé le makefile. Pour ne pas avoir à trop nous soucier des dépendances pour ce projet, nous avons codé tout le code dans le fichier tesh.c.

Nous avons inclus l'ouverture dynamique de la bibliothèque Dlopen.

2. Parsing de la ligne de commande

L'étape du parsing se découpe en deux sous étapes, une première étape qui consiste à récupérer la commande tapée et de séparer les mots de la commande dans un tableau, une seconde étape qui permet de couper ce tableau en deux sous tableaux en fonction de la possible rencontre d'un token comme le || , ; , && ...

La commande entrée par l'utilisateur est donc analysée en fonction des tokens.

Cette partie correspond aux premières difficultés auxquelles nous avons été confrontés : lire la commande entrée, séparer en différents mots, détecter les tokens, lancer l'exécution de chaque sous commande...

3. Pipe et redirection

Les redirections et la gestion des pipes ont probablement été la partie la plus difficile et à laquelle nous avons passé le plus de temps.

Nous avons commencé par la gestion d'un seul pipe dans une commande. Ensuite, on a traité le cas où plusieurs pipes étaient présents dans une commande. Finalement nous avons

implémenté les deux redirections.

Concernant les pipes, trois méthodes différentes ont été utilisées. La première, inspirée par une solution proposée par un site internet, consistait à sauvegarder en permanence deux pipes (le pipe courant et le précédent) et à traiter ces pipes suivant leur position, leur index (pair ou impair) dans la commande. Ensuite on a décidé, pour simplifier, d'enregistrer l'ensemble des pipes créés dans un tableau d'éléments (dont la taille est le nombre de pipes dans la commande multiplié par deux). Cette solution était un peu plus propre, mais aussi plus lourde. On l'a finalement simplifiée pour obtenir la version finale : les deux données enregistrées sont le pipe courant, et le file descriptor correspondant à la sortie du pipe précédent (auquel il faut raccorder l'entrée du pipe courant). C'est un peu plus simple et plus léger.

4. Mise en background/foreground

La mise en background et foreground ainsi que la gestion du "&" nous a aussi pris un certain temps. On a dû être assez soigneux afin de ne pas nous emmêler les pinceaux dans la gestion des processus, notamment pour gérer la commande "fg".

5. Chargement dynamique de dlopen

Nous avons passé quelques heures à bien comprendre le fonctionnement du chargement dynamique de bibliothèque. Nous avons rencontré quelques problèmes, notamment pour bien comprendre le fonctionnement de "dlopen" ainsi que pour trouver le nom de la bibliothèque sur les différents OS.

III. CONCLUSION

Tout d'abord ce projet nous a permis de nous rendre compte des difficultés liées au développement en C avec des notions liées au système. Les test blancs réguliers et limités étaient un moyen efficace de nous confronter aux nombreuses deadlines et à la régularité du travail qui pourra nous être demandé lors de notre stage ou dans nos travaux futurs.

Personnellement, nous avons découvert le fonctionnement de base d'un tesh. Cela nous a aussi permis de mieux maîtriser les cours de Système (gestion des tubes, des PID, des dup2 etc.) et de Shell.

Au final nous sommes globalement satisfait de notre travail : nous estimons être arrivé à un résultat convenable, une majorité de fonctions étant implémentées.

Répartition du temps de travail

	Guillaume	Matta
Makefile	< 1	< 1
Prompt	1	0
Parsing	8	4
Commandes internes	1	3
Enchaînement de commandes	3	9
Pipe	9	6
Redirection	4	2
Mode interactif/non-interactif	2	1
Background	4	2
Dlopen et chargement dynamique	2	0
Total	34	27

Références

Lors de la programmation, les sources suivantes ont pu nous être utiles pour comprendre les différents objets en jeu.

- Le site internet de Lucas Nussbaum qui regroupait un grand nombre d'aides sur différents points ainsi que les TD et TP réalisés à Télécom : <https://members.loria.fr/lnussbaum/rs.html>
- le Site Open Classroom, un cours sur les pipes : <https://openclassrooms.com/courses/la-programmation-systeme-en-c-sous-unix/les-tubes>
- Des explications sur les flux et redirection en bash : https://fr.wikibooks.org/wiki/Programmation_Bash/Flux_et_redirections
- Source dlopen : <http://manpagesfr.free.fr/man/man3/dlopen.3.html>
- fonctionnement de chdir : <http://manpagesfr.free.fr/man/man2/chdir.2.html>
- Fonctionnement de wait, WEXITSTATUS : <http://manpagesfr.free.fr/man/man2/wait.2.html>
- Comprendre comment bash réalise le parsing de la ligne de commande : <http://mywiki.woledge.org/BashParser>

- Une explication du principe de foreground/background : <https://linuxconfig.org/understanding-foreground-and-background-linux-processes>