

SYSTÈME D'EXPLOITATION

MINIX-FS

June 12, 2015

Axel Fahy - Benjamin Ganty
hepia - ITI

Contents

1	Introduction	3
1.1	Répartition des tâches	3
2	État du projet	3
3	Difficultés rencontrées	3
4	Conclusion	4

1 INTRODUCTION

L'objectif de ce projet est d'implémenter un système de fichier MINIX accessible en lecture et en écriture depuis un fichier image formaté au format MINIX version 1. MINIX est basé sur le système de fichiers UNIX, sans les aspects complexe de ce dernier.

Le projet a été séparé en deux partie. La première consistait à manipuler les structures de données de l'image, récupérée au préalable par des lectures de blocs, a été effectuée en Python 2 sur la base du canevas des classes fournis avec le projet.

La deuxième partie consistait à implémenter un accès distant sur ce système de fichier. Pour ceci, le client (Python) se connecte à un serveur de bloc écrit en langage C.

1.1 Répartition des tâches

Pour le projet, Axel Fahy c'est occupé de la partie Python (sans les sockets).

Quant à lui, Benjamin Ganty a implémenté le serveur en C ainsi que la partie client en Python. Pour synchroniser les différentes modifications, nous avons utilisé un dépôt git.

2 ÉTAT DU PROJET

Le projet est fonctionnel et réussi à exécuter le *tester* pour le Python ainsi que pour la partie avec le serveur C. Les tests ont été effectués sur une machine, en utilisant l'adresse ip *127.0.0.1* pour la connexion.

3 DIFFICULTÉS RENCONTRÉES

Concernant la première partie, la méthode *ialloc_bloc()* a été difficile à réaliser, car cette fonction n'est pas entièrement expliquée dans les documents du cours. Par ailleurs, les tests unitaires ne vérifiait pas le fonctionnement de la méthode dans le cas d'un bloc indirect ou d'un bloc doublement indirect. C'est pourquoi, je ne suis pas sûr de son comportement, plus particulièrement dans le cas de blocs doublement indirects.

Les fichiers de tests mis à disposition ont créé beaucoup de problèmes pour vérifier le fonctionnement de la manipulation des structures de données via le serveur de blocs. Il

n'était précisé nulle part que les fichiers de tests rechargeais l'image de base pour les tests d'après. Les fichiers de tests ne pouvait donc pas fonctionner avec un serveur distant pour ces raisons. C'est pour cela que nous avons du modifier ces fichiers de tests de manière à effectuer une copie de sauvegarde de l'image courante pour pouvoir la réinitialiser ensuite. Il aurait du y avoir un test unitaire à disposition spécialement prévu pour tester la communication avec les sockets.

4 CONCLUSION

En conclusion, ce projet nous a permis de découvrir le fonctionnement d'un système de fichier. Cela a permis de donnée un aspect pratique à la théorie étudiée durant ce semestre. Il nous a également donnée l'opportunité de découvrir un nouveau langage, le Python.