1 Introduction

Le but de ce projet-ci tait de dtermin la rapidit d'une machine en faisant des appels aux fonctions systmes en language mais aussi et surtout, comparer le temps d'excution des diffrentes fonctions systmes entre elles.
En effet, chaque fonction systme coute du temps et le but tait entre autre de comparer le temps mis par la combinaison de deux fonctions systmes par rapport l'appel d'une seule fonction systme ralisant la mme chose.

2 Choix d'implementations

2.1 Readdir

L'appel de la fonction readdir permet de parcourir / lister le contenu d'un dossier. Nous avons donc dcid de comparer le temps mis par readdir pour lire un dossier contenant un nombre croissants de fichiers ainsi qu'un nombre croissants de fichiers remplis. Nous voulions essayer de savoir si readdir prenait plus de temps lorsque les fichiers contenaient une chaine de caractre quelconque. C'est pourquoi, sur l'image ci-dessous, il y a 2 graphiques.

2.2 Writev compar 'write' et 'lseek'

Tout d'abord, nous nous sommes renseigns sur les 3 fonctions systmes afin de comprendre comment combiner lseek et write pour savoir comment faire un appel quivalent avec writev. Ceci fait, nous avons d trouver un scnario suffisamment intressant implmenter.

3 Interpretation des rsultats

3.1 Readdir

Voici le graphique obtenu en sortie aprs l'excution du benchmark
+ IMAGE
Nous pouvons clairement apercevoir une suite de segment de longueur horizontale de 1000 fichiers qui s'lvent de plus en plus. De plus, nous pouvons remarquer que les espaces verticaux entre ces segments sont quivalent. En effet, aprs multiples recherches sur internet, cela serait d au fait que readdir est appell rellement 1 fois tous les 1000 (25) fichiers et qu'il ne fait que charger les attributs de chaque fichier ensuite avec getattr. Une fois qu'il atteint 1000 appels getattr, il rappelle readdir. Ceci expliquerait assez bien l'allure du graphe de notre benchmark.

3.2 Writev

4 Conclusion

Au final, ce projet-ci aura t le plus enrichissant car il s'agissait de comprendre rellement ce qu'il y a derrire les fonctions systmes elles-mmes pour comprendre leurs temps d'excution respectifs. Par contre, ce projet n'aura pas t de tout repos, les premiers problmes se posant ds la compilation du fork que l'on a fait du git. Pas mal d'autres problmes de liens sont apparus ar notre projet devait s'intgrer dans un projet dj conu. De plus, pour rajouter nos deux fonctions personnelles au benchmark, il aura fallu ouvrir et comprendre la plupart des fichiers du projet donn afin d'en adapter certains.