

Rapport TP1 Systèmes d'Exploitation

On cherche à vérifier si il est plus intéressant d'utiliser les primitives système que les fonctions de bibliothèque

Lorsque le fichier est très grand (> 30000 octets), il est plus intéressant d'utiliser des primitives systèmes. En effet, le temps est divisé par deux pour un fichier de 35000 octets (ici, le fichier se nommait test) :

```
$ time ./a2.out 500 < test > poubelle
real    0m0.003s
user    0m0.000s
sys     0m0.003s
```

```
$ time ./a.out < test > poubelle
real    0m0.006s
user    0m0.006s
sys     0m0.000s
```

Après avoir modifié la taille du buffer de la fonction fonction_primitive.c par des puissances de 2, j'ai trouvé qu'à partir de 2^8 la fonction devient vraiment plus rapide et plus efficace.

Pour faire une moyenne du temps d'exécution, j'ai lancé 10 exécutions de chaque programme sur un très grand fichier et j'en ai fait la moyenne :

- fonction_bibliotheque.c :

Exécution numéro 1 : 7 secondes
Exécution numéro 2 : 6 secondes
Exécution numéro 3 : 7 secondes
Exécution numéro 4 : 10 secondes
Exécution numéro 5 : 6 secondes
Exécution numéro 6 : 6 secondes
Exécution numéro 7 : 7 secondes
Exécution numéro 8 : 11 secondes
Exécution numéro 9 : 10 secondes
Exécution numéro 10 : 8 secondes

Donc en moyenne, avec les fonctions de bibliothèques, le temps moyen de l'exécution est de 7,8 secondes.

- fonction_primitives.c (avec un buffer de 2^8)

Exécution numéro 1 : 3 secondes
Exécution numéro 2 : 3 secondes
Exécution numéro 3 : 3 secondes
Exécution numéro 4 : 3 secondes
Exécution numéro 5 : 3 secondes
Exécution numéro 6 : 5 secondes
Exécution numéro 7 : 3 secondes
Exécution numéro 8 : 3 secondes
Exécution numéro 9 : 3 secondes
Exécution numéro 10 : 4 secondes

Donc en moyenne,¹ avec les fonctions primitives, le temps moyen de l'exécution est de 3,3 secondes.

1 DIVRIOTIS Constantin L2 CMI