

Compte-rendu TP4

Temps estimé : 24 heures

Temps réel : 24heures

Pour l'exécution des fichiers, il faut entrer en argument 1 ou 2, pour effectuer les tests avec (2) ou sans ordonnancement (1)

- **Etapes pour la création/destruction**

- **Quel outil pour quelle mesure ?**

Si l'on a un mètre avec seulement des graduations en centimètres à notre disposition, et que l'on veuille mesurer l'épaisseur d'une feuille de papier issue de ramette(s) également à notre disposition, comment procédera-t-on?
Cet exemple a-t-il un rapport avec la question posée dans ce TP? Pourquoi?

Cet Exemple a un rapport avec le TP, car il faut choisir la bonne unité de mesure. Si On décide de mesurer le temps d'exécution en heures, on n'aura pas de mesure possible ; il faudrait plutôt choisir une mesure en millisecondes par exemple.

- **Question optionnelle**

Identifiez les fonctions de mesure de temps disponibles dans un environnement Linux, s'il s'agit d'un appel système (section 2 du manuel) ou d'une bibliothèque (section 3 du manuel), l'unité de temps utilisée par cette fonction, la précision de la mesure permise par cette fonction sur le système où l'on réalisera les mesures de performance. Il vous est donc demandé d'établir quel est le plus petit intervalle de temps observable au moyen des fonctions que vous aurez identifiées. Cet intervalle n'est pas forcément l'unité de la valeur renvoyée par la fonction. Une mesure effectuée par un décimètre pourrait être exprimée en mètres, il n'en reste pas moins qu'un décimètre ne peut pas mesurer une longueur avec une précision inférieure à 10 mètres. Pour les fonctions renvoyant une durée écoulée depuis un moment dans le passé, vous tenterez de définir/trouver ce point dans le passé. Vous tenterez de vérifier si la valeur renvoyée peut « déborder ».

- La commande `time` est une commande du shell. Elle fournit les temps réels (temps total), utilisateurs (durée nécessaire au processeur pour exécuter les ordres du programme) et systèmes (durée nécessaire au processeur pour traiter les ordres du système d'exploitation) en minutes et secondes, avec une précision allant jusqu'à la milliseconde.

Il existe également une fonction `time` en C qui renvoie le nombre de secondes écoulées depuis le 01/01/1970 à 00:00. L'unité minimale est la seconde.

- La fonction C clock renvoie une approximation du temps processeur (ticks) que le programme a mis.
- La fonction C gettimeofday remplit une structure qui nous permet d'avoir le temps en nanosecondes.
- **Écrivez un programme (en C) qui mesure le temps d'exécution des opérations**
Pour les 2 opérations mesurées (création de processus et création de thread), décrivez précisément ce que vous mesurez:
 - Quel est le point initial de mesure du temps ?
Pour la création de processus, on commence à mesurer juste avant le fork, et pour la thread juste avant sa création.
 - Quel est le point final de mesure du temps ?
Pour le processus/thread on fait un wait/join pour attendre que le processus/la thread se termine, et on arrête le « chronomètre ».
 - Quelle séquence d'opérations est ainsi mesurée ? Est-ce bien ce que vous vouliez mesurer pour satisfaire à la demande faite dans ce sujet ?
Pour le processus, il y a donc fork et wait, et pour la thread create et join.
Si on n'utilise pas wait et join, le « chronomètre peut s'arrêter avant la terminaison.
 - Commentez les résultats obtenus.
La création d'une thread semble un peu plus rapide que la création d'un processus.
- **Répétez votre mesure**

Le résultat obtenu est-il constant? Quelles méthodes statistiques devraient être utilisées pour « publier » des résultats ?

```
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 470 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 1210 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 1408 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 798 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 1035 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 555 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 1182 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 529 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 1309 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 1507 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 1217 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 1104 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 1480 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 404 µs
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_thread
Temps en µs pour 1 thread : 569 µs
Temps en µs pour 1 fork : 1436 us
mhk@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork
Temps en µs pour 1 fork : 1073 us
```

- Énumérez les phénomènes, facteurs qui peuvent influencer la mesure effectuée.

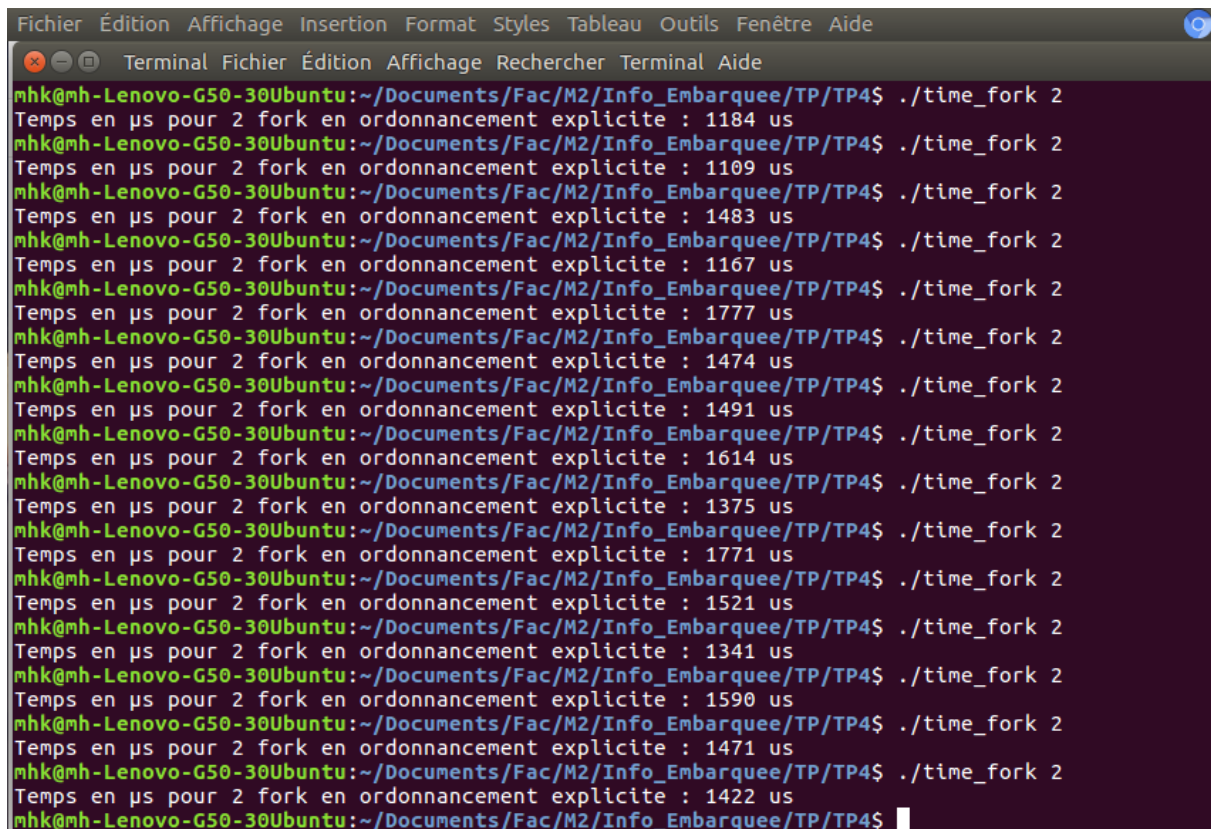
La composition de la machine, l'état de la mémoire...

Les résultats que vous avez obtenus, vous permettent-ils de fournir un temps minimum, et un temps maximum? Seriez-vous prêts à garantir ces indications de temps pour une utilisation dans un système critique?

On ne peut qu'approximer les bornes de temps, car on ne connaît pas réellement l'état de la mémoire entre deux calculs.

- **Changement de contexte**

- Donnez les résultats de vos mesures,



```
Fichier Édition Affichage Insertion Format Styles Tableau Outils Fenêtre Aide
Terminal Fichier Édition Affichage Rechercher Terminal Aide
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1184 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1109 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1483 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1167 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1777 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1474 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1491 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1614 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1375 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1771 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1521 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1341 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1590 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1471 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$ ./time_fork 2
Temps en µs pour 2 fork en ordonnancement explicite : 1422 us
mhh@mh-Lenovo-G50-30Ubuntu:~/Documents/Fac/M2/Info_Embarquee/TP/TP4$
```

- Comparerez ces résultats et expliquerez les origines possibles des différences observées,
- Indiquerez brièvement comment vous avez procédé pour obtenir l'ordonnancement désiré, et en quoi votre choix peut avoir influé sur les résultats...
J'ai utilisé un tube pour les processus et un sémaphore pour les threads.
- Décrirez ce qui peut perturber les mesures que vous avez effectuées