A + B C – D E + F

(A+B)\*(C-D)

(A + B)\*(C – D) + (E + F)

Dans le premier cas, avec les process, on utilise la fonction fork() afin de créer trois process différents : le premier et le deuxième process enfant, ainsi que le process père. Dans chaque process on va pouvoir calculer les différents membres de l’équation, sachant qu’ils s’exécutent simultanément, on va donc pouvoir réaliser parallèlement ce calcul.

Dans le second cas, en utilisant la méthode des threads, on va d’abord calculer E+F et on va créer un premier thread enfant à partir du thread principal. Dans ce premier thread enfant on va calculer C-D puis on va créer un deuxième thread enfant dans lequel on va calculer A+B. Ainsi au sein du même processus on aura donc trois threads différents qui vont s’exécuter simultanément, on va donc pouvoir réaliser le calcul en parallèle. En effet, les threads permettent d'exécuter plusieurs instructions en même temps. On parle de « programmation parallèle », car au lieu de développer selon un seul flux d'instruction, on développe plusieurs flux en parallèle.

Lorsqu’on crée un nouveau processus le code, les données, les fichiers, les registre et la pile de ce nouveau processus (fils) sont alors dissocies de son créateur (père). Le nouveau processus commence « sa vie » à l’endroit où il a été créé.

Un thread lui partage le code, les données et les fichiers avec son créateur mais « possède » ses propres registres et sa propre pile d’exécution.

Les différences :

* les processus sont plus lourd que les threads à créer
* les processus sont réellement indépendants
* les threads peuvent se partager des informations plus rapidement
* les threads doivent eux-mêmes faire attention à ne pas se marcher sur les pieds

Donc dans notre cas, s’agissant d’opérations légères, les threads sont plus adaptés

3.

Afin de mesurer le temps d’exécution des deux programmes nous avons utilisé les fonctions fournies dans l’exemple. Nous avons pu obtenir des mesures différentes en ajoutant dans les deux cas une boucle de 1000 répétitions, on peut ainsi conclure que l’utilisation des threads est plus rapide que celle des process.

4.