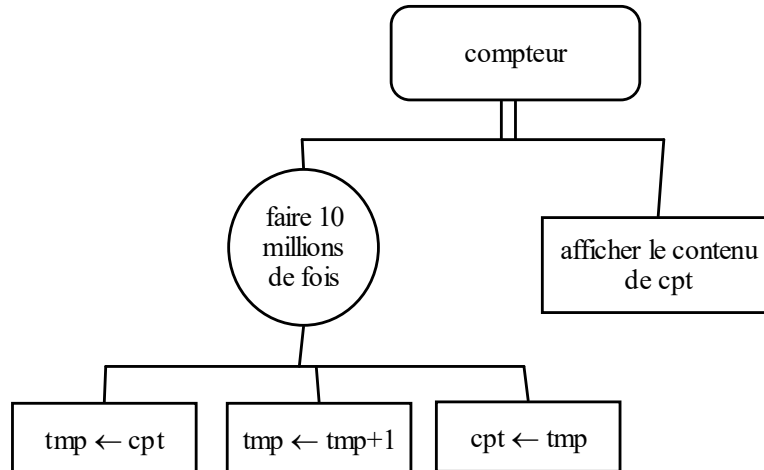


## TP UNIX n°3

Fonctions utiles : *pthread\_create*, *pthread\_attr\_init*, *pthread\_mutex\_lock*, *pthread\_mutex\_unlock*, *pthread\_mutex\_destroy*

1°) Ecrire un programme qui déclare une variable `cpt` de type unsigned long et l'initialise à 0 puis crée deux threads qui effectuent tous les deux la fonction 'compteur' dont l'algorithme est le suivant :



Dans cet algorithme la variable 'tmp' est une variable locale à la fonction 'compteur'

Le programme attendra la fin des 2 threads puis affichera : "valeur finale du compteur = x" où x est le contenu de la variable `cpt` lorsque les deux threads sont terminés.

Remarque : pour afficher un unsigned long avec printf il faut utiliser le format `%lu` et non `%d`.

Exécuter plusieurs fois le programme et relever les valeurs affichées obtient-on toujours la même valeur finale ? Pourquoi ?

Pourquoi la valeur finale du compteur n'est-elle pas de 20 millions alors que chaque thread a ajouté 10 millions à ce compteur ?

2°) On va maintenant améliorer le programme du 1°) de façon à contrôler les accès concurrents à la variable partagée 'cpt'. Pour ce faire le programme va déclarer en variable globale un sémaphore de type `MUTEX` initialisé par la constante `PTHREAD_MUTEX_INITIALIZER`. Les deux threads utiliseront ce sémaphore pour protéger les 3 instructions de modification de 'cpt' placées dans la boucle.

Exécuter plusieurs fois ce programme et relever les valeurs affichées. Vérifier que la valeur finale est bien maintenant de 20 millions à tous les coups.

3°) On peut mesurer le temps d'exécution d'un programme (prog) grâce à la commande UNIX 'time' pour cela il suffit de taper : **time prog**

Exécuter à nouveau le programme du 1°) avec mesure du temps puis faites la même chose avec le programme du 2°) et comparer les temps. Refaites cette comparaison plusieurs fois pour avoir des valeurs moyennes. Pouvez-vous expliquer cette différence ?