

TP UNIX n°2

ATTENTION : pour compiler les programmes utilisant des threads vous devez ajouter l'option **-lpthread** à la commande gcc.

Fonctions utiles: pthread_create, pthread_attr_init, pthread_attr_setdetachstate, pthread_exit, pthread_join, sleep

1°) Ecrire un programme qui :

- crée deux threads :
- affiche : "les 2 threads sont lancés"
- et se termine normalement

Le premier thread fait une boucle infinie qui affiche un '.' chaque seconde (avec printf avec retour à la ligne après le point).

Remarque : si vous voulez utiliser un printf sans retour à la ligne pour que ce soit plus lisible vous pouvez mettre dans ce thread : **setbuf(stdout, NULL)** qui supprime la bufferisation de l'affichage.

Le second thread affiche "tapez un caractère" puis attend un caractère au clavier par getchar, dès qu'il l'a reçu il se termine.

Exécuter ce programme, Ça ne marche pas ! Pourquoi ?

2°) Modifier le programme du 1°) pour qu'après avoir affiché "les 2 threads sont lancés", il attende la terminaison du second thread (celui qui fait une saisie au clavier) puis affiche : "On va s'arrêter là" et se termine normalement.

Exécuter ce programme, qu'est ce que ça a changé ? Pourquoi ?

3°) Que se passe-t-il si on modifie le programme pour qu'il attende la fin du premier thread au lieu du second ?

4°) Modifier le programme du 2°) pour qu'il crée les deux threads dans l'état 'détaché'. Pour cela il faut utiliser la fonction pthread_attr_setdetachstate pour préparer convenablement les attributs des threads créés.

Exécuter ce programme. Ça ne marche plus ! Pourquoi ?