

Exercice à rendre 2

On souhaite réaliser un programme utilisant des signaux avec $n + 1$ threads (le thread principal et n threads fils numérotés de 0 à $n - 1$). L'objectif est que chaque nouvelle pression sur Ctrl-C (signal `SIGINT`) provoque la terminaison du thread fils suivant.

Pour ce faire, on demande que le signal `SIGINT` soit reçu exclusivement par le thread principal. À la k -ème réception de `SIGINT` ($1 \leq k \leq n$), le thread principal doit envoyer le signal `SIGUSR1` au thread numéro $k - 1$. Celui-ci doit afficher son numéro, puis se terminer.

On demande donc de réaliser le programme de la manière suivante :

1. le programme doit admettre un seul argument, le nombre n de threads fils à générer ;
2. au démarrage, après l'initialisation et la génération des n threads fils, les $n + 1$ threads doivent se synchroniser avec une barrière de façon qu'aucun thread ne fasse une action avant que tous ne soient prêts ;
3. une fois la barrière ouverte, chaque thread attend l'arrivée d'un signal ;
4. le thread principal, quant à lui, prend en compte l'arrivée de `SIGINT` : à chaque réception de `SIGINT`, il envoie `SIGUSR1` au suivant des threads fils, puis il en attend la terminaison ;
5. à la réception de `SIGUSR1`, le thread fils affiche son numéro (vous n'oublierez pas d'utiliser `fflush` pour forcer `printf` à écrire les données sans bufferiser) et se termine ;
6. le thread principal se termine lorsque les n threads fils sont terminés.

Dans cet exercice, vous n'utiliserez aucune variable globale.