# Programmation Concurrente

César COLLÉ

Rémy KALOUSTIAN

POLYTECH NICE-SOPHIA

SI4

10/10/2016

1. Introduction
2. Algorithme de déplacement
3. Java vs Posix
4. Algorithme de création des processus fils
5. Algorithme de terminaison de l'application
6. Conclusion

# 1. Introduction

L'objectif de ce projet est de simuler le déplacement de personnes sur une grille et ce, à l'aide de threads qui géreront le déplacement des personnes. Outre l'objectif fonctionnel, nous avons un objectif pédagogique qui est de bien nous rendre compte de l'utilité et du fonctionnement des threads.

Nous avons choisi d'implémenter ce projet en C, car personne ne peut rivaliser avec le charisme de César.

# 2. Algorithme de déplacement

Explication de notre algo de déplacement.

# 3. Java vs Posix

Création : En C, on appelle thread_create() en lui passant en paramètre le thread, la fonction à exécuter, et des informations complémentaires sur le comportement du thread. Une majeure partie de la création se fait via des paramètres et on a juste à déclarer une variable de type pthread_t.

En Java, on doit tout d'abord créer une classe héritant de Thread, qui représentera notre thread. La fonction à exécuter est déjà dans la classe(run()). Contrairement au C, il n'y a pas de paramètres à passer lors de la création, qui se fait comme une instanciation d'objet classique.

Démarrage :

Arrêt :

Destruction :

## 4. Algorithme de création des processus fils

Comment on crée nos fils.

## 5. Algorithme de terminaison de l'application

Comment on termine notre application

## 6. Conclusion

Le rapport doit être rédigé comme un rapport et donc comporter outre les éléments attendus une introduction et une conclusion. Dans cette première étape, il doit insister et décrire : - l'algorithme utilisé pour déplacer une personne (rappel : un algorithme n'est pas le code C). Pour ceux qui ne savent pas ce qu'est un algorithme vous pouvez lire l'article : https://interstices.info/jcms/c_5776/qu-est-ce-qu-un-algorithme.

- comparer la manipulation des threads en Java (que vous avez vu en cours en SI3) et la manipulation des threads Posix (création, démarrage, arrêt, destruction, passages de paramètres, terminaison) ;

- pour la thread principale (i.e. celle associée au main de l'application), vous devez donner l'algorithme de création des threads filles (option –t1 et –t2) et celui lié à la terminaison de l'application (i.e. attente de création des threads filles précédemment créées) ;

- analyser la correction de chacun des scénarios proposés.

- analyser de manière comparative les divers scénarios corrects proposés, cette analyse doit nécessairement utiliser les mesures effectuées.

+ Dis en cours ( titre, date, auteurs, introduction, ce qu'on a fait(gné) Bilan (bien, moins bien, lent, rapide, on a détecté…))

+En java, pour chaque objet est déclaré un verrou (peut être utile ?)