

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

Ceci est une introduction

2. Algorithme de déplacement

Explication de notre algo de déplacement.

3. Java vs Posix

Comparaison Java et Posix

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 ?)