

Révision du C :

Exercice1 :

Soit les déclarations de variables suivantes :

long n = 15 ;

long *p ;

long **pp ;

Imaginez que p pointe sur n et que pp pointe sur p.

En partant du principe que vous connaissez les adresses de n, p et pp, quelles seront :

- les valeurs de p et pp ?
- la valeur de *p ?
- la valeur de *pp ?
- la valeur de **pp ?

Ecrivez un programme permettant de vérifier vos hypothèses.

Note : pour afficher un long ou une adresse de variable, utilisez le format %ld.

Exercice2 : Simulation du remplissage d'une baignoire avec fuite – Approche sans threads

Cet exercice vise à simuler le remplissage d'une baignoire par un robinet tout en prenant en compte une fuite. Le programme doit gérer l'évolution du niveau d'eau en fonction des débits respectifs du robinet et de la fuite. Dès que la baignoire est entièrement vide, la fuite est colmatée. Le programme s'arrête lorsque la baignoire est pleine.

L'objectif est d'explorer une première approche de **programmation concurrente sans outils spécifiques** et d'identifier les limitations inhérentes à une exécution séquentielle.

1. **Gestion du temps** : Introduction d'une boucle de simulation qui ajuste le niveau d'eau à chaque itération pour représenter le passage du temps.
2. **Événements aléatoires** : Possibilité que la fuite s'aggrave ou que le débit du robinet diminue de manière aléatoire.
3. **Affichage en temps réel** : Mise à jour dynamique de l'état de la baignoire pour une meilleure visualisation.

Vous pourriez vous inspirer d'une version écrite en Java à l'adresse suivante :

<http://rmdiscala.developpez.com/cours/LesChapitres.html/Java/Cours3/Chap3.3.htm#2.0>

Essayer de relever les problèmes potentiels, quel modèle nous pourrions utiliser dans cet exercice comme modèle parmi ceux cités en cours :

- boss/worker
- Le modèle pair
- Le modèle pipeline

