

Programmation Système et Multi-Threading

Le TP est long, il se peut que vous n'ayez pas le temps de tout faire. Le sujet étant nouveau chaque année, c'est difficile de la dimensionner correctement. Pensez à commenter et à tester ce qui vous implémentez au fur et à mesure. La présentation de vos listing sera prise en compte (indentation, commentaire, lisibilité, etc). Il est important que vous fournissiez un listing qui compile. Si vous n'avez pas fini une fonctionnalité dans le temps imparti mettez la en commentaire, elle sera lue et prise en compte dans la notation mais vous faciliterez le travail de votre correcteur qui va commencer par compiler votre programme.

Mettez tous dans un seul fichier que vous nommerez ainsi : nom_prenom_groupeTP_promo.c et qui pourra donc être compilé en tapant :
gcc -pthread -Wall nom_prenom_groupeTP_promo.c.

La remise des fichiers se fait via moodle (moodle.insa-toulouse.fr). Afin de s'assurer qu'il n'y aura pas de problème lors de la remise, connectez vous de suite a moodle, allez dans le cours programmation système (clé sys2008) et dans examen TP et laissez la connexion pendante. Vous avez jusqu'à 18h30 pour déposer un fichier après le dépôt est bloqué. Vous pouvez faire plusieurs dépôts c'est le dernier dépôt qui est conservé (attention donc à ne pas vous tromper de fichier car c'est celui stocké sous moodle qui sera évalué).

le sujet :

On désire mettre en place un ensemble de processus coopérant pour créer le jeu : « Gary Pouter et ses amis les magiciens ». Dans ce jeu, **chaque joueur est représenté par un processus UNIX**. Il y a des magiciens bons et des magiciens mauvais, ces derniers définissent le camp du bien et du mal. Chaque camp s'affronte pour détruire l'autre camp. L'affrontement se fait en jetant des sorts. Il existe 2 types de sortilège :

- des sorts négatifs que l'on envoie à un magicien du camp adverse et qui font perdre des forces à ce magicien
- des sorts positifs que l'on envoie aux magiciens de son propre camp pour leur redonner des forces

Un magicien meurt quand il est à bout de force (niveau de force à 0). Chaque magicien **entre dans la bataille avec un volume de force égale à 10**. La réception d'un **sort positif augmente de 1** ses forces. La réception d'un **sort négatif fait perdre 2 unités** de force.

Le choix du camp des processus se fait après la création d'un processus. Si la **somme des chiffres constituant le PID du processus est pair c'est un bon magicien** sinon il appartient au camp du mal.

Exemple : si le processus a le PID 1353 alors il appartient au camp du bien (1+3+5+3=12)

Les magiciens sont créés initialement tous ensemble (à dt prêt). Ils commencent ensuite le combat.

L'activité des magiciens est la suivante :

- **attente x secondes que tous les magiciens soient là**
- **initialisation(....)**
- **boucle jusqu'à ce que mort s'en suive**
 - o **attendrealeatoirement(....)**
 - o **jetersort(....)**
- **enterrement(...)**

Parallèlement à son activité volontaire, un magicien voit **ses forces augmenter ou diminuer à travers les sorts qu'il reçoit. Il est le seul à connaître et à pourvoir incrémenter ou décrémenter ses forces**. C'est à dire la variable représentant ses forces est une variable locale au processus. Elle n'est pas accessible aux autres magiciens.

Remarque : il est conseillé de lire toutes les questions de cet exercice avant de commencer à répondre.

- afin d'augmenter la lisibilité de votre programme chaque fonctionnalité sera codé en créant une fonction.
- afin de simplifier :
 - on ne remettra pas à jour en cours d'exécution la liste des magiciens encore vivant. Il se peut donc qu'un sort soit lancé sur un magicien déjà mort.
 - Si plusieurs sorts sont lancés simultanément sur le même magicien, on peut perdre des sorts ou pas

1) création des magiciens

Fonction : creerXmagiciens(....) :

Afin de simplifier la création du système, on suppose que le premier magicien créé (appelé Gary Pouter) se charge de créer tous les processus représentant les autres magiciens en appelant cette fonction creerXmagiciens. Dès que Gary Pouter a fini de créer tous les autres processus des magiciens il redevient un magicien normal et il se lance dans la guerre.

2) les magiciens dans la guerre

a) Fonction : ...initialisation(....)

Proposer et implémenter un système pour permettre aux différents magiciens de se connaître (phase d'initialisation). **Vous devez justifier votre choix dans le listing** (pas de justification = moins de points).

b) Fonction : ...attendrealeatoirement(....)

Donnez la fonction utilisée pour mettre en attente le magicien pendant un temps aléatoire. Cette attente sera considéré comme maximum à dt prêt. Le magicien peut éventuellement attendre un peu moins.

c) Fonction :jetersort(.....)

Proposer et implémenter un mécanisme permettant de choisir aléatoirement un magicien et de lui envoyer un sort positif ou négatif en fonction de son camps. Justifier votre choix (pas de justification = peu de points).

d) Fonction : réceptionsort(.....)

Proposer et implémenter un mécanisme permettant de traiter la réception d'un sort et son implication sur son compteur de force.

e) Fonction :enterrement(.....)

Proposer et implémenter la phase de destruction d'un magicien quand son total de force est inférieur ou égal à 0.

3) Question joker :

En fait les magiciens ont 2 mains et sont donc à même s'ils possèdent 2 baguettes magiques de lancer indépendamment 2 sorts. On se propose dans cette question de permettre aux magiciens d'utiliser leur 2 mains simultanément en parallélisant. Chaque main suit donc indépendamment un cycle : attendre aléatoirement puis jeter sort.

Barème indicatif qui sera éventuellement réévalué:

item	points
Question 1	2
Question 2.a	2
Question 2.b	0,5
Question 2.c	2
Question 2.d	2
Question 2.e	1,5
Question 3	Bonus +3
Propreté du code : - indentation - commentaires - noms des variables	4 points
Simplicité et pertinence des mécanismes mis en oeuvre	4 points
Execution : compile et fait bien la partie qui a été codé	2 point