

Projet Philosophers - Sujet traduit en français

Résumé :

Dans ce projet, vous apprendrez les bases de la gestion de threads dans un processus.

Vous verrez comment créer des threads et découvrirez les mutex.

I. Introduction

La philosophie est l'étude des questions fondamentales sur l'existence, la connaissance, la raison, etc.

Ici, c'est une introduction humoristique au problème classique des philosophes.

II. Instructions générales

- Le projet doit être en C et respecter la Norme 42.
- Aucune fuite mémoire ne sera tolérée.
- Un Makefile conforme est requis (règles : all, clean, fclean, re, bonus).
- Aucune variable globale n'est autorisée.
- Votre libft peut être incluse si le sujet le permet.
- Des programmes de test sont recommandés pour votre défense.

III. Aperçu

- Plusieurs philosophes sont assis autour d'une table.
- Ils mangent, dorment ou pensent.
- Pour manger, ils doivent prendre deux fourchettes : une à gauche, une à droite.
- Une fois le repas terminé, ils dorment puis pensent.
- La simulation s'arrête si un philosophe meurt de faim.

- Les philosophes ne communiquent pas entre eux.

IV. Règles globales

- Programme obligatoire et programme bonus séparés.

- Arguments du programme :

nombre_de_philosophes temps_pour_mourir temps_pour_manger temps_pour_dormir

[nombre_de_repas]

- Les philosophes sont numérotés de 1 à n.
- Philosophe 1 est à côté de philosophe n.
- Les messages doivent être clairs et horodatés en millisecondes :
"X a pris une fourchette", "X mange", "X dort", "X pense", "X est mort"
- Les messages ne doivent pas être entremêlés.
- Un philosophe qui meurt doit être affiché dans les 10ms.

V. Partie obligatoire

Nom du programme : philo

Fichiers requis : *.c, *.h, Makefile

Fonctions autorisées :

memset, printf, malloc, free, write, usleep, gettimeofday,

pthread_create, pthread_detach, pthread_join,

pthread_mutex_init/destroy/lock/unlock

Spécificités :

- Chaque philosophe est un thread.
- Une fourchette entre chaque philosophe, protégée par un mutex.

VI. Partie bonus

Nom du programme : philo_bonus

Fonctions supplémentaires autorisées :

fork, kill, exit, waitpid, sem_open/close/post/wait/unlink

Spécificités :

- Chaque philosophe est un processus (sauf le processus principal).
- Les fourchettes sont représentées par des sémaphores.

VII. Remise et évaluation

- Soumettez votre travail sur le dépôt git prévu.
- Le dossier `philo/` contient la partie obligatoire.
- Le dossier `philo_bonus/` contient la partie bonus.

Remarque :

La partie bonus n'est évaluée que si la partie obligatoire est parfaite.