



Philosophers

Qui aurait cru que philosopher serait si mortel ?

Résumé: Ce projet est une introduction aux bases du threading de processus, et comment travailler sur le même espace mémoire.

Vous apprendrez comment faire des threads.

Vous découvrirez les mutex, les sémaphores, et la mémoire partagée.

Table des matières

I	Introduction	2
II	Mandatory part	3

Chapitre I

Introduction

La philosophie (du grec, philosophia, littéralement "amour de la sagesse") est l'étude de questions générales et fondamentales sur l'existence, la connaissance, les valeurs, la raison, l'esprit et le langage. De telles questions se posent souvent comme des problèmes à étudier ou à résoudre. Le terme a probablement été inventé par Pythagore (c. 570 - 495 AEC). Les méthodes philosophiques comprennent le questionnement, la discussion critique, l'argumentation rationnelle et la présentation systématique.

Les questions philosophiques classiques incluent : Est-il possible de savoir quelque chose et de le prouver ? Quel est le plus réel ?

Les philosophes posent également des questions plus pratiques et concrètes telles que : Existe-t-il une meilleure façon de vivre ? Vaut-il mieux être juste ou injuste (si on peut s'en tirer comme ça) ? Les humains ont-ils le libre arbitre ?

Historiquement, la "philosophie" englobait tout corpus de connaissances.

De l'époque du philosophe grec Aristote au 19ème siècle, la "philosophie naturelle" englobait l'astronomie, la médecine et la physique. Par exemple, les principes mathématiques de la philosophie naturelle de Newton, établis en 1687, ont par la suite été classés dans un livre de physique.

Au 19ème siècle, la croissance des universités de recherche modernes a amené la philosophie académique et d'autres disciplines à se professionnaliser et à se spécialiser. À l'ère moderne, certaines recherches qui faisaient traditionnellement partie de la philosophie sont devenues des disciplines universitaires distinctes, notamment la psychologie, la sociologie, la linguistique et l'économie.

D'autres enquêtes étroitement liées à l'art, à la science, à la politique ou à d'autres activités font toujours partie de la philosophie.

Par exemple, la beauté est-elle objective ou subjective ? Existe-t-il plusieurs méthodes scientifiques ou une seule ? L'utopie politique est-elle un rêve porteur d'espoir ou une fantaisie sans espoir ? Les principaux sous-domaines de la philosophie académique comprennent la métaphysique ("soucieux de la nature fondamentale de la réalité et de l'être"), l'épistémologie (concernant "la nature et les fondements de la connaissance [et] ... ses limites et sa validité"), l'éthique, l'esthétique, philosophie politique, logique et philosophie des sciences.

Chapitre II

Mandatory part

Vous devez faire 3 programmes respectant le même set de règles basiques :

- Ce projet doit être codé en C. Tout crash, fuite mémoire, comportement indéterminé ou erreur de norme rapporte 0 au projet.
- Un nombre de philosophes sont assis autour d'une table et font trois choses : manger, penser et dormir.
- Lorsqu'ils mangent, ils ne peuvent pas penser ou dormir, lorsqu'ils dorment, ils ne peuvent pas manger ou penser, et enfin, lorsqu'ils pensent, ils ne peuvent pas manger ou dormir.
- La table est circulaire et a un large plat de spaghettis au centre.
- Il y a des fourchettes sur la table.
- Comme il n'est pas aisé de se servir et manger avec une seule fourchette, nous assumerons qu'un philosophe doit manger avec deux fourchettes, une dans chaque main.
- Les philosophes ne doivent jamais être affamés.
- Tous les philosophes ont besoin de manger.
- Les philosophes ne communiquent pas entre eux.
- Les philosophes ne savent pas lorsque l'un d'entre eux est sur le point de mourir.
- Dès qu'un philosophe a fini de manger, il pose ses fourchettes et se met à dormir.
- Quand un philosophe a fini de dormir, il se met à penser.
- La simulation s'arrête dès qu'un philosophe meurt.
- Tous les programmes auront les mêmes options : `number_of_philosopher` `time_to_die` `time_to_eat` `time_to_sleep` [`number_of_time_each_philosophers_must_eat`]
- Each programs should have the same option : `number_of_philosopher` `time_to_die` `time_to_eat` `time_to_sleep` [`number_of_time_each_philosophers_must_eat`]
 - `number_of_philosopher` : correspond au nombre de philosophes et au nombre de fourchettes.

- `time_to_die` : en millisecondes. Si un philosophe ne commence pas à manger dans '`time_to_die`' millisecondes après avoir commencé son dernier repas ou le début de la simulation, il meurt.
- `time_to_eat` : en millisecondes. C'est le temps que prend un philosophe à manger. Il garde les deux fourchette pendant ce temps.
- `time_to_sleep` : en millisecondes. C'est le temps que prend un philosophe à dormir.
- `number_of_time_each_philosophers_must_eat` : Optionnel, si tous les philosophes mangent au moins '`number_of_time_each_philosophers_must_eat`' la simulation s'arrête. Si ce temps n'est pas spécifié, la simulation s'arrête dès qu'un philosophe meurt.
- Chaque philosophe se verra assigné un nombre de 1 à '`number_of_philosopher`'.
- Philosophe 1 est à côté de philosophe '`number_of_philosopher`'. Tous les autres philosophes sont N sont à côté de leur N + 1 et N - 1.
- Tout changement de statut d'un philosophe doit être écrit comme suit (X remplaçant le numéro de philosophe et `timestamp_in_ms` le timestamp, en millisecondes)
 - `timestamp_in_ms X has taken a fork`
 - `timestamp_in_ms X is eating`
 - `timestamp_in_ms X is sleeping`
 - `timestamp_in_ms X is thinking`
 - `timestamp_in_ms X died`
- Le statut affiché ne doit pas être mélangé avec le statut d'un autre philosophe.
- Vous ne pouvez pas avoir plus de 10ms de retard entre la mort d'un philosophe et l'affichage de sa mort.
- Les philosophes doivent éviter de mourir !

Nom du programme	<code>philo_one</code>
Fichiers de rendu	<code>philo_one/</code>
Makefile	oui
Arguments	<code>number_of_philosopher time_to_die time_to_eat time_to_sleep [number_of_time_each_philosophers_must_eat]</code>
Fonctions externes autorisées	<code>memset, malloc, free, write, usleep, gettimeofday, pthread_create, pthread_detach, pthread_join, pthread_mutex_init, pthread_mutex_destroy, pthread_mutex_lock, pthread_mutex_unlock</code>
Libft autorisée	non
Description	philosopher avec des threads et des mutex

Dans cette version, les règles supplémentaires seront :

- Il y a une fourchette entre chaque philosophe, donc il y aura une fourchette à gauche et à droite de chaque philosophe.
- Pour éviter aux philosophes d'accidentellement dupliquer des fourchettes, vous devez protéger l'état des fourchettes à l'aide de mutex.
- Vous devez attacher chaque philosophe à un thread différent.

Nom du programme	philo_two
Fichiers de rendu	philo_two/
Makefile	Oui
Arguments	number_of_philosopher time_to_die time_to_eat time_to_sleep [number_of_time_each_philosophers_must_eat]
Fonctions externes autorisées	memset, malloc, free, write, usleep, gettimeofday, pthread_create, pthread_detach, pthread_join, sem_open, sem_close, sem_post, sem_wait, sem_unlink
Libft autorisée	Non
Description	philosopher avec des threads et des semaphore

Dans cette version, les règles supplémentaires seront :

- Toutes les fourchettes sont au milieu de la table.
- Elles n'ont pas d'état spécifique en mémoire, mais le nombre de fourchettes disponibles est représenté par un semaphore
- Vous devez attacher chaque philosophe à un thread différent.

Nom du programme	philo_three
Fichiers de rendu	philo_three/
Makefile	Yes
Arguments	number_of_philosopher time_to_die time_to_eat time_to_sleep [number_of_time_each_philosophers_must_eat]
Fonctions externes autorisées	memset, malloc, free, write, fork, kill, exit, pthread_create, pthread_detach, pthread_join, usleep, gettimeofday, waitpid, sem_open, sem_close, sem_post, sem_wait, sem_unlink
Libft autorisée	No
Description	philosopher avec des processus et des semaphores

Dans cette version, les règles supplémentaires seront :

- Toutes les fourchettes sont au milieu de la table.
- Elles n'ont pas d'état spécifique en mémoire, mais le nombre de fourchettes disponibles est représenté par un semaphore
- Chaque philosophe est représenté par un processus différent. Le processus principal ne devrait pas être un philosophe.