



Philosophes

Je n'aurais jamais pensé que la philosophie serait si mortelle

Résumé:

Dans ce projet, vous apprendrez les bases du threading d'un processus. Vous verrez comment créer des threads et vous découvrirez les mutex.

Version : 11

Contenu

je	Introduction	2
II	Instructions communes	3
III	Aperçu	5
IV	Règles mondiales	6
V	Partie obligatoire	8
VI	Partie bonus	9
VII	Soumission et évaluation par les pairs	dix

Chapitre I

Introduction

Philosophie (du grec, *philosophie*, 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. Ces questions sont souvent posées comme des problèmes à étudier ou à résoudre. Le terme a probablement été inventé par Pythagore (vers 570 – 495 avant notre ère). 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 ? Qu'est-ce qui 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 l'on peut s'en sortir) ? Les humains ont-ils le libre arbitre ?

Historiquement, la « philosophie » englobait tout ensemble de connaissances. Depuis l'époque du philosophe grec Aristote jusqu'au XIXe siècle, la « philosophie naturelle » englobait l'astronomie, la médecine et la physique. Par exemple, les Principes mathématiques de philosophie naturelle de Newton de 1687 furent plus tard classés comme un livre de physique.

Au XIXe siècle, la croissance des universités de recherche modernes a conduit 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 recherches étroitement liées à l'art, à la science, à la politique ou à d'autres activités sont restées partie intégrante 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 plein d'espoir ou un fantasme désespéré ? Les principaux sous-domaines de la philosophie universitaire comprennent la métaphysique (« concernée par la nature fondamentale de la réalité et de l'être »), l'épistémologie (sur « 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

Instructions communes

- Votre projet doit être écrit en C.
- Votre projet doit être rédigé conformément à la Norme. Si vous disposez de fichiers/fonctions bonus, ils sont inclus dans le contrôle des normes et vous recevrez un 0 s'il y a une erreur de norme à l'intérieur.
- Vos fonctions ne doivent pas s'arrêter de manière inopinée (erreur de segmentation, erreur de bus, double libération, etc.) en dehors de comportements non définis. Si cela se produit, votre projet sera considéré comme non fonctionnel et recevra un 0 lors de l'évaluation.
- Tout l'espace mémoire alloué au tas doit être correctement libéré si nécessaire. Aucune fuite ne sera tolérée.
- Si le sujet l'exige, vous devez soumettre un Makefile qui compilera vos fichiers sources vers la sortie requise avec les indicateurs -Mur, -Wextra et -Ereux, utilisez cc et votre Makefile ne doit pas être lié à nouveau.
- Ton Makefile doit au moins contenir les règles \$(NOM), tout, propre, propre et concernant.
- Pour accorder des bonus à votre projet, vous devez inclure une règle prime à votre Makefile, qui ajoutera tous les différents en-têtes, bibliothèques ou fonctions interdites sur la partie principale du projet. Les bonus doivent être dans un fichier différent _bonus.{c/h} si le sujet ne précise rien d'autre. L'évaluation des parties obligatoires et bonus est effectuée séparément.
- Si votre projet vous permet d'utiliser votre Libft, vous devez copier ses sources et ses associés Makefile dans un libft dossier avec son Makefile associé. Tes projets Makefile doit compiler la bibliothèque en utilisant son Makefile, puis compilez le projet.
- Nous vous encourageons à créer des programmes de test pour votre projet même si cela fonctionne **ne devra pas être soumis et ne sera pas noté**. Cela vous donnera l'occasion de tester facilement votre travail et celui de vos pairs. Vous trouverez ces tests particulièrement utiles lors de votre soutenance. En effet, lors de la soutenance, vous êtes libre d'utiliser vos tests et/ou les tests du pair que vous évaluez.
- Soumettez votre travail à votre référentiel git attribué. Seul le travail dans le dépôt git sera noté. Si Deepthought est chargé de noter votre travail, cela sera fait

après vos évaluations par les pairs. Si une erreur se produit dans une section de votre travail lors de la notation de Deepthought, l'évaluation s'arrêtera.

Chapitre III

Aperçu

Voici les choses que vous devez savoir si vous souhaitez réussir cette mission :

- Un ou plusieurs philosophes sont assis autour d'une table ronde.
Il y a un grand bol de spaghettis au milieu de la table.
- Les philosophes alternativement **manger, pense, ou dormir**.
Pendant qu'ils mangent, ils ne pensent ni ne dorment ; en pensant, ils ne mangent ni ne dorment ;
et, bien sûr, pendant leur sommeil, ils ne mangent ni ne pensent.
- Il y a aussi des fourchettes sur la table. Il y a **autant de fourchettes que de philosophes**.
- Parce que servir et manger des spaghettis avec une seule fourchette est très gênant, un philosophe prend sa fourchette droite et sa fourchette gauche pour manger, une dans chaque main.
- Quand un philosophe a fini de manger, il repose ses fourchettes sur la table et commence à dormir. Une fois réveillés, ils recommencent à réfléchir. La simulation s'arrête lorsqu'un philosophe meurt de faim.
- Tout philosophe a besoin de manger et ne devrait jamais mourir de faim.
- Les philosophes ne se parlent pas.
- Les philosophes ne savent pas si un autre philosophe est sur le point de mourir.
- Inutile de dire que les philosophes doivent éviter de mourir !

Chapitre IV

Règles mondiales

Vous devez écrire un programme pour la partie obligatoire et un autre pour la partie bonus (si vous décidez de faire la partie bonus). Ils doivent tous deux respecter les règles suivantes :

- Les variables globales sont interdites !
- Votre (vos) programme (s) doit (vos) prendre les arguments suivants :
number_of_philosophers time_to_die time_to_eat time_to_sleep
[number_of_times_each_philosopher_must_eat]
 - number_of_philosophes : Le nombre de philosophes et aussi le nombre de fourchettes.
 - l'heure de mourir (en millisecondes) : Si un philosophe ne commençait pas à manger l'heure de mourir millisecondes depuis le début de leur dernier repas ou le début de la simulation, ils meurent.
 - le temps de manger (en millisecondes) : Le temps qu'il faut à un philosophe pour manger. Pendant ce temps, ils devront tenir deux fourchettes.
 - l'heure de dormir (en millisecondes) : Le temps qu'un philosophe passera à dormir.
 - number_of_times_each_philosopher_must_eat (argument facultatif) : Si tous les philosophes ont mangé au moins number_of_times_each_philosopher_must_eat fois, la simulation s'arrête. Si cela n'est pas précisé, la simulation s'arrête à la mort d'un philosophe.
- Chaque philosophe possède un numéro allant de 1 à nombre_de_philosophes.
- Numéro de philosophe 1 est assis à côté du numéro du philosophe nombre_de_philosophes. Tout autre numéro de philosophe N est assis entre le numéro du philosophe N-1 et numéro philosophe N + 1.

À propos des journaux de votre programme :

- Tout changement d'état d'un philosophe doit être formaté comme suit :
 - timestamp_in_ms X a pris un fork
 - timestamp_in_ms X mange
 - timestamp_in_ms X est en veille
 - timestamp_in_ms X réfléchit
 - timestamp_in_ms X est mort

Remplacer horodatage_in_ms avec l'horodatage actuel en millisecondes et X avec le numéro philosophe.

- Un message d'état affiché ne doit pas être confondu avec un autre message.
- Un message annonçant la mort d'un philosophe ne doit pas être affiché plus de 10 ms après la mort réelle du philosophe.
- Encore une fois, les philosophes devraient éviter de mourir !



Votre programme ne doit avoir aucuncourses aux données.

Chapitre V

Partie obligatoire

Nom du programme	philo
Remettre les fichiers	Makefile, *.h, *.c, dans le répertoire philo/
Makefile	NOM, tout, propre, fclean, re
Arguments	number_of_philosophers time_to_die time_to_eat time_to_sleep [number_of_times_each_philosopher_must_eat]
Fonctions externes.	memset, printf, malloc, gratuit, écrire, usleep, gettimeofday, pthread_create, pthread_detach, pthread_join, pthread_mutex_init, pthread_mutex_destroy, pthread_mutex_lock, pthread_mutex_unlock
Libft autorisé	Non
Description	Philosophes avec des fils et des mutex

Les règles spécifiques à la partie obligatoire sont :

- Chaque philosophe devrait être un fil conducteur.
- Il y a une fourchette entre chaque paire de philosophes. Donc, s'il y a plusieurs philosophes, chaque philosophe a une fourchette à gauche et une fourchette à droite. S'il n'y a qu'un philosophe, il ne devrait y avoir qu'une seule fourchette sur la table.
- Pour empêcher les philosophes de dupliquer les forks, vous devez protéger l'état des forks avec un mutex pour chacun d'eux.

Chapitre VI

Partie bonus

Nom du programme	philo_bonus
Remettre les fichiers	Makefile, *.h, *.c, dans le répertoire philo_bonus/
Makefile	NOM, tout, propre, fclean, re
Arguments	number_of_philosophers time_to_die time_to_eat time_to_sleep [number_of_times_each_philosopher_must_eat]
Fonctions externes.	memset, printf, malloc, gratuit, écrire, fork, kill, exit, pthread_create, pthread_detach, pthread_join, usleep, gettimeofday, waitpid, sem_open, sem_close, sem_post, sem_wait, sem_unlink
Libft autorisé	Non
Description	Philosophes avec processus et sémaphores

Le programme de la partie bonus reprend les mêmes arguments que le programme obligatoire. Il doit répondre aux exigences de la *Règles mondiales* chapitre.

Les règles spécifiques à la partie bonus sont :

- Toutes les fourchettes sont posées au milieu de la table.
- Ils n'ont pas d'états en mémoire mais le nombre de forks disponibles est représenté par un sémaphore.
- Chaque philosophe devrait être un processus. Mais le processus principal ne devrait pas être un philosophe.



La partie bonus ne sera évaluée que si la partie obligatoire est PARFAIT. Parfait signifie que la partie obligatoire a été intégralement réalisée et fonctionne sans dysfonctionnement. Si vous n'avez pas satisfait à TOUTES les exigences obligatoires, votre partie bonus ne sera pas du tout évaluée.

Chapitre VII

Soumission et évaluation par les pairs

Remettez votre devoir dans votreGitréférentiel comme d'habitude. Seuls les travaux réalisés à l'intérieur de votre référentiel seront évalués lors de la soutenance. N'hésitez pas à vérifier les noms de vos fichiers pour vous assurer qu'ils sont corrects.

Répertoire des pièces obligatoires :philo/

Répertoire des pièces bonus :philo_bonus/

