

## PHILOSOPHER

Ce projet va nous permettre de découvrir les bases de l'enfilage d'un processus et ce qu'est un mutex. /  
*This project will enable us to discover the basics of threading a processus and what is a mutex.*

### I. Partie Obligatoire / Mandatory Part :

- Bien évidemment, ce projet doit être **codé à la Norme**, aucune fuite de mémoire, crash, erreur de norme ou comportement indéterminé sera toléré. / *Obviously, this project will be coded in coherence with the Norme, any leaks, crash, norme error of undefined behaviour will be tolerated.*
- **Une ou plusieurs philosophes sont assis autour d'une table** ne faisant qu'une de ces 3 choses : **manger, penser ou dormir** (lorsqu'ils mangent ils ne peuvent ni dormir ni penser, lorsqu'ils dorment ils ne peuvent ni manger ni penser et quand il pensent ils ne peuvent ni dormir ni manger). / *One or more philosophers are sitting around a table doing one of three things : eating, thinking or sleeping (While eating, they are not thinking or sleeping, while sleeping, they are not eating or thinking and while thinking, they are not eating or sleeping.*
- Les philosophes s'assoient autour d'une table ronde avec un grand bol de spaghetti dans le centre, accompagné de quelques fourchettes. **Pour manger, chaque philosophe doit avoir à disposition 2 fourchettes.** / *The philosophers sit at a circular table with a large bowl of spaghetti in the center, as well as some forks. To eat, a philosopher needs 2 forks.*
- **A chaque fois qu'un philosophe a finis de manger, il lâchera ses fourchettes et dort, une fois réveillé, il commence à penser** / *Each time a philosophes finishes eating, he will drop his forks and start to sleep, once awake he will think.*
- **La simulation s'arrête quand un des philosophes meurent** / *The simulation ends when a philosopher dies.*
- **Chaque philosophe doit être attribué un nombre de 1 au nombre de philosophes** (le philosophe 1 est à côté du philosophe "nombre de philosophes", sinon les autres sont entre le philosophes **N-1** et **N+1**) / *Each philosophers should be given a number from 1 to the numbers of philosophers (philosopher 1 is next to philosopher "number of philosopher", else a philosopher is between the philosopher N-1 and N+1).*

- Tout changement de statut d'un philosophe doit s'écrire de la manière suivante (l'affichage de changement de statut d'un philosophe ne doit pas écraser celui d'un autre et le **temps entre la mort d'un philosophe et l'affichage de sa mort de doit pas dépasser 10 ms**) / *All changement of status of a philosopher must be written as followed (the change of status of a philosopher should not overwrite another's philosophers status changement and the time between a philosopher's death and the print of his death can't exceed 10 ms) :*

Timestamp in ms	Philosopher number X	Action
////	X	has taken a fork
////	X	is eating
////	X	is sleeping
////	X	is thinking
////	X	died

- Autres règles / *Other rules* :
  - **Un philosophe ne doit jamais mourir de faim** / *A philosopher must never be starving ;*
  - **Tous les philosophes doivent manger** / *All philosopher need to eat ;*
  - **Les philosophes ne parlent pas entre eux** / *Philosophers don't speak with each other ;*
  - **Les philosophes ne savent pas quand l'un d'eux va mourir** / *The philosophers don't know when one of them is about to die ;*
  - Bien évidemment, les philosophes doivent éviter de mourir / *Of course philosophers should avoid dying.*
- **Il y a une fourchette entre chaque philosophe, donc une à droite et une à gauche de chacun** (pour éviter la duplication d'une fourchette, il est conseillé de protéger chaque fourchette par un mutex) / *There is one fork between each philosopher, therefore a philosopher has a fork on his right and on his left (to avoid duplicating forks, it is recommended to protect each fork with a mutex.*
- **Chaque philosophe devrait être un "thread"** / *Each philosopher should be a thread.*

## II. Programme / Program :

Nom / <i>Name</i>	philo
Fichier / <i>File</i>	philo/
Arguments	<p>[1] <b>number_of_philosophers</b> : Nombre de philosophes (mais aussi le nombre de fourchette) <i>Numbers of philosophers (but also the number of forks).</i></p> <p>[2] <b>time_to_die</b> : Temps pendant laquelle un philosophe peut ne pas manger avant de mourir en millisecondes à partir du moment où il a commencé son dernier repas / <i>Time a philosopher can not eat before he dies in milliseconds from the moment he started his last meal.</i></p> <p>[3] <b>time_to_eat</b> : Temps qu'un philosophe a pour manger en millisecondes (mais aussi le temps qu'un philosophe aura 2 fourchettes) / <i>Time a philosopher has to eat in milliseconds (but also the time a philosopher has two forks).</i></p> <p>[4] <b>time_to_sleep</b> : Temps qu'un philosophe a pour dormir en millisecondes / <i>Time a philosopher has to sleep in milliseconds.</i></p> <p>[5] <b>number_of_times_each_philosopher_must_eat</b> : Argument optionnel, qui détermine le nombre de fois que chaque philosophe doit manger avant que la simulation s'arrête (si ce n'est pas spécifié le programme s'interrompt quand un philosophe est mort) / <i>Optional argument that determines the amount of times each philosopher should eat before the program end (if not specified the program should end when a philosopher dies).</i></p>
Fonctions Externes / <i>External function</i>	memset, printf, malloc, free, write, usleep, gettimeofday, pthread_create, pthread_detach, pthread_join, pthread_mutex_init, pthread_mutex_destroy, pthread_mutex_lock, pthread_mutex_unlock