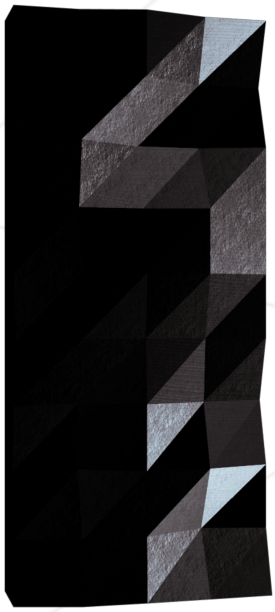

Philosophers

I've never thought philosophy would be so deadly.

Summary: In this project, you will learn the basics of threading a process and how to work on the same memory space. You will learn how to make threads. You will discover the mutex, semaphore and shared memory.

Contents

Chapter I

Introduction

Philosophy (from Greek, philosophia, literally "love of wisdom") is the study of general and fundamental questions about existence, knowledge, values, reason, mind, and language. Such questions are often posed as problems to be studied or resolved. The term was probably coined by Pythagoras (c. 570 – 495 BCE). Philosophical methods include questioning, critical discussion, rational argument, and systematic presentation. Classic philosophical questions include: Is it possible to know anything and to prove it? What is most real? Philosophers also pose more practical and concrete questions such as: Is there a best way to live? Is it better to be just or unjust (if one can get away with it)? Do humans have free will?

Historically, "philosophy" encompassed any body of knowledge. From the time of Ancient Greek philosopher Aristotle to the 19th century, "natural philosophy" encompassed astronomy, medicine, and physics. For example, Newton's 1687 Mathematical Principles of Natural Philosophy later became classified as a book of physics. In the 19th century, the growth of modern research universities led academic philosophy and other disciplines to professionalize and specialize. In the modern era, some investigations that were traditionally part of philosophy became separate academic disciplines, including psychology, sociology, linguistics, and economics.

Other investigations closely related to art, science, politics, or other pursuits remained part of philosophy. For example, is beauty objective or subjective? Are there many scientific methods or just one? Is political utopia a hopeful dream or hopeless fantasy? Major sub-fields of academic philosophy include metaphysics ("concerned with the fundamental nature of reality and being"), epistemology (about the "nature and grounds of knowledge [and]...its limits and validity"), ethics, aesthetics, political philosophy, logic and philosophy of science.

Chapter II

Mandatory part

You will have to write 3 different programs but they will have the same basic rules:

- This project is to be coded in C, following the Norm. Any leak, crash, undefined behavior or norm error means 0 to the project.
- A number of philosophers are sitting at a round 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 of course, while thinking, they are not eating or sleeping.
- The philosophers sit at a circular table with a large bowl of spaghetti in the center.
- There are some forks on the table.
- As spaghetti is difficult to serve and eat with a single fork, it is assumed that a philosopher must eat with two forks, one for each hand.
- The philosophers must never be starving.
- Every philosopher needs to eat.
- Philosophers don't speak with each other.
- Philosophers don't know when another philosopher is about to die.
- Each time a philosopher has finished eating, he will drop his forks and start sleeping.
- When a philosopher is done sleeping, he will start thinking.
- The simulation stops when a philosopher dies.
- Each program should have the same options: number_of_philosophers time_to_die time_to_eat time_to_sleep [number_of_times_each_philosopher_must_eat]
    - number_of_philosophers: is the number of philosophers and also the number of forks
    - time_to_die: is in milliseconds, if a philosopher doesn't start eating 'time_to_die' milliseconds after starting his last meal or the beginning of the simulation, it dies

  - time_to_eat: is in milliseconds and is the time it takes for a philosopher to eat. During that time he will need to keep the two forks.
  - time_to_sleep: is in milliseconds and is the time the philosopher will spend sleeping.
  - number_of_times_each_philosopher_must_eat: argument is optional, if all philosophers eat at least 'number_of_times_each_philosopher_must_eat' the simulation will stop. If not specified, the simulation will stop only at the death of a philosopher.

- Each philosopher should be given a number from 1 to 'number_of_philosophers'.
- Philosopher number 1 is next to philosopher number 'number_of_philosophers'. Any other philosopher with number N is seated between philosopher N - 1 and philosopher N + 1
- Any change of status of a philosopher must be written as follows (with X replaced with the philosopher number and timestamp_in_ms the current timestamp in milliseconds)
  - 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
- The status printed should not be scrambled or intertwined with another philosopher's status.
- You can't have more than 10 ms between the death of a philosopher and when it will print its death.
- Again, philosophers should avoid to die!

Program name	`philo_one`
Turn in files	`philo_one/`
Makefile	`Yes`
Arguments	`number_of_philosophers time_to_die time_to_eat time_to_sleep [number_of_times_each_philosopher_must_eat]`
External functs.	`memset, malloc, free, write, usleep, gettimeofday, pthread_create, pthread_detach, pthread_join, pthread_mutex_init, pthread_mutex_destroy, pthread_mutex_lock, pthread_mutex_unlock`
Libft authorized	`No`
Description	`philosopher with threads and mutex`

In this version the non common rules will be:

- One fork between each philosopher, therefore there will be a fork at the right and at the left of each philosopher.
- To avoid philosophers duplicating forks, you should protect the forks state with a mutex for each of them.
- Each philosopher should be a thread.

Program name	`philo_two`
Turn in files	`philo_two/`
Makefile	`Yes`
Arguments	`number_of_philosophers time_to_die time_to_eat time_to_sleep [number_of_times_each_philosopher_must_eat]`
External functs.	`memset, malloc, free, write, usleep, gettimeofday, pthread_create, pthread_detach, pthread_join, sem_open, sem_close, sem_post, sem_wait, sem_unlink`
Libft authorized	`No`
Description	`philosopher with threads and semaphore`

In this version the non common rules will be:

- All the forks are in the middle of the table.
- They have no states in memory but the number of available forks is represented by a semaphore.
- Each philosopher should be a thread.

Program name	`philo_three`
Turn in files	`philo_three/`
Makefile	`Yes`
Arguments	`number_of_philosophers time_to_die time_to_eat time_to_sleep [number_of_times_each_philosopher_must_eat]`
External functs.	`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 authorized	`No`
Description	`philosopher with processes and semaphore`

In this version the non common rules will be:

- All the forks are in the middle of the table.
- They have no states in memory but the number of available forks is represented by a semaphore.
- Each philosopher should be a process and the main process should not be a philosopher.