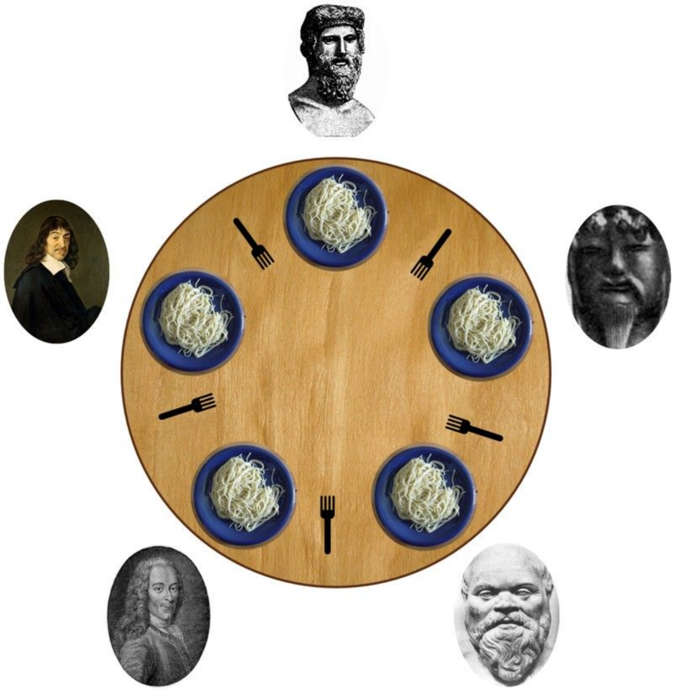
Exercise 09

1 Dining Philosophers



Implement a program that is running an instance of the dining philosophers problem. N Philosophers sit around a table with plates of noodles in front of them. There is exactly one fork placed between two plates, respectively.
Philosophers only know of two activities: discussing and eating. Within a fixed interval, each Philosopher talk for a random amount of time, then switches to eating for an again random amount of time. However: he can only begin eating when both forks beside his plate are available.
It is your task to make sure that no fork is accessed by more than one Philosopher at a time. Be aware of deadlocks and thread-starvation(pun intended).
Use OpenMP to introduce parallelization to this problem. Feel free to use either standard C++11 synchronization constructs or OpenMP synchronization primitives to coordinate ressource access among the threads.
Count the total number of discussion periods made by all the philosophers. Btw: it is completely fine for a philosopher to talk with himself, there is no need to assign pairs of philosophers to coordinate a meaningful conversation.
Compare your performance within a fixed time frame with that of your fellow students.