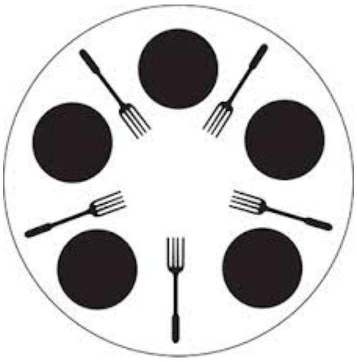
INSTITUT POLYTECHNIQUE DES SCIENCES AVANCEES

Labs: In422 TP N°: 03

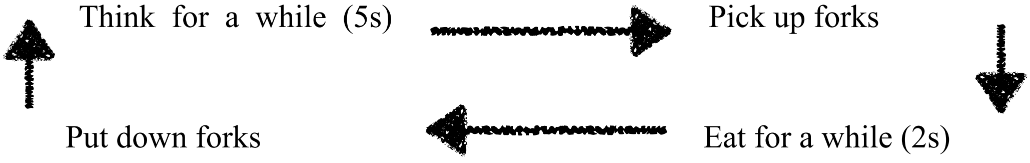
Objectives Programming the dinning Philosophers using C/C++

Exercise :
The dining philosophers problem is invented by E. W. Dijkstra. This problem is related to five philosophers who spend their lives just thinking and eating. In the middle of the dining room, there is a circular table with five plates of spaghetti !! However, there are only five forks available, as shown in the following figure.



Each philosopher thinks. When he gets hungry, he picks up the two forks that are closest to him (left and right). If a philosopher can pick up both forks, he eats for some time (2seconds). After a philosopher finishes eating, he puts down the forks and starts to think.
Each philosopher can eat just when it gets two forks (Left and right).
In our case we will use six philosophers and a table that contains six forks.

1) How do we write a **thread** program to simulate philosophers ?
First, we notice that these philosophers are in a thinking —> picking up forks—>eating —> putting down forks cycle as shown below.
Show the program to your professor.



Think for a while (5s)

Pick up forks

Eat for a while (2s)

Put down forks

2) The most serious problem of this program is that **deadlock** could occur!

Suppose that each philosopher sits down at the same time and picks up his right fork. In this case, all forks are locked and none of the philosophers can successfully lock his left fork. As a result, we have a circular waiting (*i.e.*, every philosopher waits for his left fork that is currently being locked by his left neighbor), and hence a deadlock occurs :(.

- How can we solve this deadlock ?
- Using this solution, implement it in your code.

Remarks:
This TP will be marked:

- 5 points for the autonomy;
- 5 points for the rank (ex. you're the first and everything is correct you will get the total mark);
- 10 points for the contents and the code optimality (in term of code lines).

Good Luck;