Dining Philosophers

Prof. Darrell D. E. Long
Computer Science Department

Fall 2013

Do what?

Your task is to construct a simulation of the *dining philosophers problem* in C. The following description is due to Hoare¹ and can be found on pg. 75.

In ancient times, a wealthy philanthropist endowed a College to accommodate five eminent philosophers. Each philosopher has a room in which he could engage in his professional activity of thinking; there was also a common dining room, furnished with a circular table, surrounded by five chairs, each labelled by the name of the philosopher who was to sit in it. The names of the philosophers were \( P_0, P_1, P_2, P_3, P_4, \) and they were disposed in this order anticlockwise round the table. To the left of each philosopher there was laid a golden fork, and in the centre stood a large bowl of spaghetti, which was constantly replenished.

A philosopher was expected to spend most of his time thinking; but when he felt hungry, he went to the dining room, sat down in his own chair, picked up his own fork on his left, and plunged it into the spaghetti. But such is the tangled nature of spaghetti that a second fork is required to carry it to the mouth. The philosopher therefore had also to pick up the fork on his right. When he was finished he would put down both his forks, get up from his chair, and continue thinking. Of course, a fork can be used by only one philosopher at a time.

You should solve this problem by using the thread and mutual exclusion facilities of C to simulate each of the philosophers and relevant resources, such as forks.

Keep in mind that there is a serious complication: suppose all five philosophers get hungry at the same time. In that case it is possible that each philosopher will grasp his left fork, only to find his distinguished colleague to the right has also grasped his own fork. Each philosopher, being quite stubborn, is likely to starve rather than relinquish his fork to his eminent colleague.

The solution that Hoare suggests is to add a *footman* to the model. It is the footman’s task to allow only four philosophers into the dining chamber at one time. The result is that the right-most philosopher is able to eat his fill of spaghetti, and then leave the room. A simple inductive argument shows that the other philosophers will then each be able to eat in turn.

There is one other problem: suppose we have a particularly ravenous philosopher (say, St. Thomas Aquinas), who, having finished eating, immediately picks up both forks and begins eating again. Further suppose that his unfortunate colleague to the right (say, Socrates) has a particularly slow left arm. Then this unfortunate scholar could very well starve to death while his colleague to the left grows quite fat. I leave it as an exercise for you to modify the behavior of the footman to prevent this unfortunate circumstance.

The output of your program should be a sequence (of selectable length) of \( P_i \) is seated, \( P_i \) takes left fork, \( P_i \) takes right fork, \( P_i \) eats. Note that several \( P_j \) may very well be doing this concurrently with \( P_i \).

1 Implementation details

- Each philosopher should eat at least 10 times, their eating times should be interleaved.
- The philosopher should print the stage it’s in when it first arrives. The stages are eating, thinking, and waiting.
- Parameterize with the number of philosophers.
  (i.e. n = 10, 10 philosophers or n = 100, 100 philosophers)
- Use sleep in thinking phase. Creativity and other customization in the thinking phase is encouraged.

2 Hints

- You may use any form of deadlock avoidance, choose whichever makes the most sense to you.
- Use the pthread functions from chapter 12 and 13 in the book.
- As always, build your program piece by piece, test each piece separately before you add it in.

When do I have to turn this in?

This assignment will be due at noon on Wednesday, 27 November, 2013.