1. **Aufgabenstellung (Englisch)**

The **Dining Philosophers** problem was proposed by Dijkstra in 1965, when dinosaurs ruled the earth. It appears in several variations, but the standard features are a table with five plates, five forks (or chopsticks) and a big bowl of spaghetti. Five philosophers, who represent interacting threads, come to the table and execute the following loop:

While True:

*Think()*

*Get\_forks()*

*Eat()*

*Put\_forks()*

The forks represent resources that the threads must hold exclusively in order to make progress. The thing that makes the problem interesting, unrealistic, and unsanitary, is that the philosophers need two forks to eat, so a hungry philosopher might have to wait for a neighbour to put down a fork.

Assume that the philosophers have a local variable **i** that identifies each philosopher with a value (0…4). Similarly, the forks are numbered from 0 to 4, so that philosopher **i** has fork **i** on the right and fork **i+1** on the left. Here is a diagram of the situation:



If the philosophers know how to think and eat, our job is to write a version of get forks and put forks that satisfies the following constraints:

* 1. Only one philosopher can hold a fork at a time.
  2. It must be impossible for a deadlock to occur.
  3. It must be impossible for a philosopher to starve waiting for a fork.
  4. It must be possible for more than one philosopher to eat at the same time.

The last requirement is one way of saying that the solution should be efficient – that is, it should allow the maximum amount of concurrency.

We make no assumption about how long eat and think take, expect that eat must terminate eventually. Otherwise, the third constraint is impossible – if a philosopher keeps one of the forks forever, nothing can prevent the neighbours from starving.

Weiters soll das Projekt mit **GIT** verwaltet werden.

1. **GIT-Verwaltung**

Zunächst musste ein GitHub-Account angelegt werden und die dazugehörige Arbeitsumgebung installiert werden.

1. **Kommentierter Programmcode**

Der Programmcode befindet sich im angehängten File **Diner2.c**.

1. **Programmausführung**