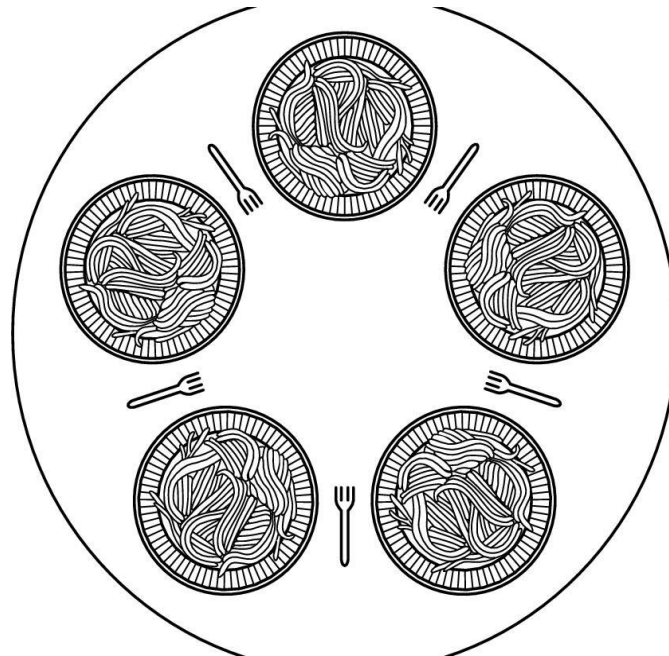


## 1) Aufgabe (Dining Philosophers):

Gegeben ist die folgende Aufgabenstellung, die auf Dijkstra zurückgeht. Der Tagesablauf eines Philosophen besteht aus einer alternierenden Folge der Tätigkeiten: Essen und Denken. Es befinden sich 5 Philosophen an einem Tisch. An diesem Tisch befinden sich zudem 5 Teller mit Spaghetti und 5 Gabeln:



Zum Essen benötigt jeder Philosoph 2 Gabeln, die sich links und rechts von seinem Teller befinden. (Es existiert auch eine chinesische Variante mit 2 Essstäbchen.) Beachten Sie auch, dass die Philosophen an festen Plätzen am Tisch sitzen.

- a) Skizzieren Sie eine Lösung der Synchronisation der Philosophen mit Semaphoren mit einer von Ihnen gewählten Beschreibungsform (Pseudocode, Diagramm, oder ähnliches). Diskutieren Sie die Lösungsskizze.  
Hinweis: Berücksichtigen Sie hierbei, dass bei der folgenden Programmierung bei den IPC-Semaphoren auch Manipulation von Semaphore-Gruppen als eine atomare Aktion möglich sind. Somit eine atomare Operation  $P(S1, S2)$  möglich ist. Dies wird dann mit der Operation `semop` mit 2 Elementen implementiert.
- b) Setzen Sie die Lösungsskizze in ein C-Programm um. Gehen Sie hierfür wie folgt schrittweise vor:
  - i) Erstellen Sie den Rahmen des Programms indem Sie die Kernelemente des Handelns der Philosophen beschreiben. Achten Sie auch auf unterschiedliche Essens und Denkzeiten. Verwenden Sie hierzu die Zufallszahlenfunktion (`srand()` bzw `rand()`); Ignorieren Sie hierzu die Gabeln. Sorgen Sie auch für geeignete Ausgaben. Testen Sie den Ablauf.
  - ii) Erweitern Sie nun den Algorithmus um Synchronisationselemente. Verwenden Sie hierzu geeignete Hilfsfunktionen, die sie auch aus Blatt1 übernehmen können.
- c) Testen Sie den Algorithmus. Entwickeln Sie hierfür eine geeignete Teststrategie.

Diskutieren Sie Teststrategie.

## **2) Aufgabe (Dining Philosophers mit einzelnen Semaphoreoperationen):**

Verzichten Sie nun bei der Lösung aus Aufgabe 1 auf die Spezialität des IPC-Paktes der Verwendung von Semaphoreoperationen in einem Array. Sie dürfen nun bei jeder Semaphoreoperation nur genau ein Semaphore manipulieren.

- a) Skizzieren Sie eine geeignete Lösung in Pseudocode
- b) Untersuchen Sie die Lösung auf mögliche Deadlocks, Livelocks. Beachten Sie hierbei, dass die Ressourcen (Gabeln) möglichst gut genutzt werden.
- c) Diskutieren Sie die Lösung mit Ihrem Nachbarn
- d) Fleißaufgabe, gerne auch nach dem Praktikum: Setzen Sie die Lösung in ein Programm um, indem Sie das Programm aus Aufgabe 1 modifizieren