

The Dining Philosopher Problem

The dining philosopher problem is considered a classic synchronization problem as it is an example of large class of concurrency-control problem. It is a simple representation of the need to allocate several resources among several processes in a deadlock-free and starvation-free manner. In 1965, Edsger Dijkstra set an examination question on a synchronization problem where five computers competed for access to five shared tape drive peripherals. Soon afterwards the problem was retold by Tony Hoare as the dining philosophers problem.

The program illustrates five philosophers who share a circular table and in the center of the table is a bowl of rice, and the table is laid with five chopsticks. When a philosopher thinks, he does not interact with his colleagues. So, if a philosopher gets hungry, he tries to pick up the two chopsticks that are closest to him. When a hungry philosopher has both his chopsticks at the same time, he eats without releasing his chopsticks. However, a situation of deadlock may occurs if all five philosophers become hungry simultaneously and grabs his left chopstick. When each philosopher tries to grab his right chopstick, he will be delayed forever.

Several possible remedies to the problem have been demonstrated that ensures freedom from deadlock.

Programmed in JAVA

NetBeans IDE 6.9.1

Java SE Development Kit 6

Gagandeep Singh

Guru Tegh Bahadur Institute of Technology

00413202709

Semester 5

© 2011 Gagandeep Singh | gagandeep.taurus91@gmail.com | 9717568636

**PC****CD****GAGANDEEP SINGH****PRESENTS**

The Dining Philosopher Problem

(A JAVA TECHNOLOGY PRODUCT)

Project By :**Gagandeep Singh**

PC
CD

GAGANDEEP SINGH
PRESENTS

The Dining Philosopher Problem
(A JAVA TECHNOLOGY PRODUCT)

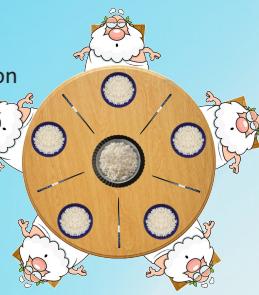
Project By :
Gagandeep Singh

© 2011 Gagandeep Singh | gagandeep.taurus91@gmail.com | 9717568636

The Dining Philosopher Problem

© 2011 Gagandeep Singh

The Dining Philosopher Problem



The Dining Philosopher Problem

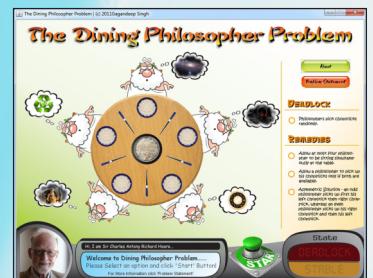
The dining philosopher problem is considered a classic synchronization problem as it is an example of large class of concurrency-control problem. It is a simple representation of the need to allocate several resources among several processes in a deadlock-free and starvation-free manner. In 1965, Edsger Dijkstra set an examination question on a synchronization problem where five computers competed for access to five shared tape drive peripherals. Soon afterwards the problem was retold by Tony Hoare as the dining philosophers problem.

The program illustrates five philosophers who share a circular table and in the center of the table is a bowl of rice, and the table is laid with five chopsticks. When a philosopher thinks, he does not interact with his colleagues. So, if a philosopher gets hungry, he tries to pick up the two chopsticks that are closest to him. When a hungry philosopher has both his chopsticks at the same time, he eats without releasing his chopsticks. However, a situation of deadlock may occurs if all five philosophers become hungry simultaneously and grabs his left chopstick. When each philosopher tries to grab his right chopstick, he will be delayed forever.

Several possible remedies to the problem have been demonstrated that ensures freedom from deadlock.

Programmed in JAVA
NetBeans IDE 6.9.1
Java SE Development Kit 6

This Project is part of summer training in Java Programming Language
20 July - 8 August 2011



Gagandeep Singh
Guru Tegh Bahadur Institute of Technology
00413202709
Semester 5