Dining Philosophers Encyclopedia Article

Dining Philosophers

The following sections of this BookRags Literature Study Guide is offprint from Gale's For Students Series: Presenting Analysis, Context, and Criticism on Commonly Studied Works: Introduction, Author Biography, Plot Summary, Characters, Themes, Style, Historical Context, Critical Overview, Criticism and Critical Essays, Media Adaptations, Topics for Further Study, Compare & Contrast, What Do I Read Next?, For Further Study, and Sources.

(c)1998-2002; (c)2002 by Gale. Gale is an imprint of The Gale Group, Inc., a division of Thomson Learning, Inc. Gale and Design and Thomson Learning are trademarks used herein under license.

The following sections, if they exist, are offprint from Beacham's Encyclopedia of Popular Fiction: "Social Concerns", "Thematic Overview", "Techniques", "Literary Precedents", "Key Questions", "Related Titles", "Adaptations", "Related Web Sites". (c)1994-2005, by Walton Beacham.

The following sections, if they exist, are offprint from Beacham's Guide to Literature for Young Adults: "About the Author", "Overview", "Setting", "Literary Qualities", "Social Sensitivity", "Topics for Discussion", "Ideas for Reports and Papers". (c)1994-2005, by Walton Beacham.

All other sections in this Literature Study Guide are owned and copyrighted by BookRags, Inc.

Contents

Dining Philosophers Encyclopedia Article.....1

Contents.....2

Dining Philosophers.....3

Dining Philosophers

In a famous 1971 paper titled "Hierarchical Ordering of Sequential Processes," **Edsger Dijkstra** proposed a problem that has become a canonical example of difficulties in **concurrency**, multithreading, multi-process synchronization, and **mutual exclusion**. Dijkstra's formulation is as follows. Five philosophers sit at a circular dining table. Each has before him a plate with spaghetti on it. The table is set with one fork beside each plate, five forks in all. The philosophers spend their time alternating between thinking and eating, and two forks are needed for any philosopher to eat.

When a philosopher wishes to eat, it is necessary for him to obtain both forks, one on each side, which can only happen if both are available. If a philosopher is unable to obtain both forks, he will be unable to eat, and will eventually die of starvation. The resulting competition for forks is a representation of the competition for resources that occurs in operating systems. Specifically, if there are multiple threads of execution and threads need to **access** their critical sections without getting deadlocked, the problem is equivalent to that of the dining philosophers.

The trivial single-threaded solution is to traverse the table allowing one philosopher to eat at a time. This solution does not maximize concurrency. A common multi-threaded solution is to have a separate **thread** per philosopher executing code:

- while (true)
- {
- Think();
- Lock_Left_Fork();
- Lock_Right_Fork();
- Eat();
- Release_Right_Fork();
- Release_Left_Fork();
- }

However, the solution given above has a potential **deadlock** situation if each philosopher executes the locking of the left fork, prior to the locking of the right fork. (One correct solution is to have odd numbered philosophers lock the left fork before the right fork, while even numbered philosophers lock the right fork before the left.)