Rene Sorger - se22m015

Gregor Zeman - se22m048

25.02.2023

Does this strategy resolve the deadlock and why?

No, this strategy, in fact, creates a deadlock by trying to make 2 philosophers pick up the same fork at once.

Measure the total time spent in waiting for forks and compare it to the total runtime:

As expected, the waiting time was far higher than the actual runtime, since multiple threads need to wait for free resources to prevent a deadlock.

Total elapsed time: 18,22 seconds

Total waited time: 26,16 seconds

Interpret the measurement - Was the result expected?

The more shared resources are used, the more likely it is that more time is spent by threads waiting for these shared resources to be freed at any given moment.

Can you think of other techniques for deadlock prevention?

Make resources inaccessible if processes are trying to access them simultaneously through mutual exclusion.

Make a process access one resource while waiting for another resource to get freed in order to access the desired resource. (Hold and wait)

Make sure to always shutdown the program cooperatively and to always cleanup all allocated resources.

C# Garbage Collector, my beloved 💖 💖