Parallel Programming

Exercise 1, Concurrency - Dining Philosophers

Johann Wimmer
se18m010@technikum-wien.at

FHTW — March 13, 2019

1 A naive implementation

2 Deadlock prevention

2.1 What are the necessary conditions for deadlocks?

2.2 Why does the initial solution lead to a deadlock?

2.3 Switch the order in which the philosophers take the forks

2.4 Does this strategy resolve the deadlock and why?

2.5 Measure the time spent in waiting for fork and compare it to total runtime

2.6 Can you think of other techniques for deadlock prevention?

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