NEU Lab 3

Implement a solution to the Dining Philosophers problem (as explained in Chapter 6.6) using message passing through MPI. Your solution must utilize a “Fork Master” who monitors and authorizes the use of the shared resources (forks). The Fork Master must ensure that hungry philosophers eventually get to eat and that two philosophers cannot use the same fork at the same time.

This can be accomplished with only needing the standard MPI Send and Recv commands. These are the “blocking” point-to-point message passing commands outlined in the online MPI tutorials, but note that “blocking send” in this case only requires that the message successfully be processed into the send queue (so it will seem to immediately proceed to execute the next command – but the message will not be “lost” in that when an appropriate Recv is called, that message will be available).

You are responsible for determining implementation details (such as if messages indicate single forks are being “passed” or if a message indicates two forks are “assigned”, how to handle a case when a philosopher is hungry but the forks aren’t immediately available, avoiding deadlock, etc.)

An example of the bare bones structure along with a sending and receiving messages is provided as example code.