Zack Chambers

CST 221

Deadlock Avoidance

January 13, 2019

Link to Git repository:

<https://github.com/zchambers3/CST-221/tree/master/DeadAvoidance>

To avoid a deadlock, we must implement certain precautions before a process can simply request a recourse. In this program a timer is implemented so that the process can request the recourse and wait a certain amount of time, in this case a millisecond. The program will attempt to do this up to four times before it is either accepted by the recourse or terminated and forced to begin the entire process again. I wanted to create checkpoints, or recourses that need to be requested before the final recourse becomes available but was unable to accomplish this task. The same timer would be implemented for each checkpoint but when the timeout occurs it would not send it to the back of the line but rather that recourse that caused the timeout.

**Flowchart**

