Caleb Miller

CST-221

20171008

Deadlock Avoidance

To Avoid 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 wait a certain amount of time, in this case a millisecond. It will try up do four times before it is either accepted by the recourse or terminated and forced to begin the entire process again. Given this was along the lines of our Topic 3 DQ 2, something else would need to be implemented. What I had in mind (but couldn’t produce) was checkpoints, or recourses that need to be requested before the final recourse becomes available to request. 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.

Flow-Chart

START

Timer

Recourse

CP3

CP2

CP1