

## CIS 452 Project 2: Resource Manager

Aron Sunuwar

For the extra credit, I decided to implement the detect and recover policy in order to handle the deadlocks. The algorithm is really simple and utilizes methods that I had already in place.

The steps are:

1. Detect deadlock
2. Find the process that holds the resource, which caused the recent deadlock
3. Force the process to release deadlock causing resource
4. Resource is held by the next process that requested that resource
5. Recheck for deadlock
6. Repeat

This algorithm is the Resource Preemption method of recovery from deadlock. Typically, Resource Preemption can cause starvation for the process that causes deadlock, however from my thought process with a single instance resources using resource preemption on the latest resource won't lead to starvation because the next process requesting that resource will be in line and eventually will get it. Although the algorithm could definitely be improved if a priority based preemption was used, however I did not implement that method.