SOFE 3950: Lab 5 Banker's algorithm

Group Member:
Umar Ehasn 100634240
Diba Shojaeigoradel 100621768
Yin Zhou 100314426
Thomas Jansz 100652111

Introduction:

The banker's algorithm is used to allocate resources between different processes. It works on the principle that at any given time, the system should have enough resources available for at-least one future resource request from one process. The core of Banker's algorithm is the safety algorithm.

The Safety Algorithm:

The overall algorithm steps are:

1. Initialize a "work" structure equal to the available resources..
2. Assign a "finish" attribute to each process with a value of false
3. Find a process in the process list such as:
    a. Finish attribute of the process is false and
    b. Need attribute of the process <= work.
4. If such a process in step 3 is found, then update its finish attribute to true and add its allocated attribute to the work structure.
5. If all processes have finish attributes = true, then system is safe, else check the processes n x n times where n is the number of processes. If finish attribute of one or more processes is false, then system is unsafe.
6. Resource request is only granted if resulting system state is safe.

Resource Allocation Algorithm:

The resource allocation algorithm has the following steps:

1. If resources requested are greater than needed or available, then stop, else goto step 2.
2. Allocate the resources.
3. Call the safety algorithm to check resulting system state.
4. If the system is safe then return success, else do not allocate resources and return failure.

Overall Structure:

After the program is initialized, we create the threads, one each for each process/customer and the threads then request the resources. The mutex locks prevent the threads for modifying the shared resources (in this case the allocation, need, available, and process/customer global variable). Each thread follows the above Banker's algorithm to safely request resources.

Conclusion:

This lab allowed us to learn the following:

- Multi-processing along with POSIX threads, mutex locks.
- Banker's algorithm