OS Project 2 Brainstorm:

Banker’s Algorithm:

Deadlock – 2 or more depend on an event that can’t happen

Starvation – a process never starts

Livelock -2 processes dependant on each other, where we restart it and eventually have to keep restarting it.

Deadlock conditions:

Mutual exclusion – non-sharable resource ( file for writing )

Hold and wait

No preemption – resource cant be freed by os, only by program running

Circular wait

Solutions

- avoidance – bankers algorithm ( Dijkstra )

- prevention

- detection and recovery

Data structures

N – number of processes

M – number of resource types

Safety algorithm

1. Work := available
   1. Finish\_i == false for I = 1,2,…,n
2. Find an I that is both
   1. Finish\_i == false
   2. Need\_i < work
   3. If no i? go to end
3. Work := work + allocationi
   1. Finishi := true
   2. Go to step 2
4. If all I’s are true then end, else lose

Resource allocation algorithm

1. If request I < = need I , otherwise error
2. If request I <= available, otherwise wait
3. Allocate requested resources

Resource algorithm