BITS STUDENT-ID - 2020mt12098

Student /Developer NAME- Somnath Khamaru

**Chandy-Mishra-Haas-Algorithm (OR Model) for Diffusion Based Deadlock Detection in Distributed Systems**

* The Source code is written in Java 8 and tested in Windows10 CMD. Source files InitiateProcess.java , Process.java ,Initiator.java and dependencies like Dependencies.txt should be kept in same folder.
* The input WFG is present in the Dependencies.txt file, currently the Wait-For Graph (WFG) is as per the diagram present in WFG.jpg. It is designed for 7 processes. User can increase the no of processes as per wish and can automatically scale the system to use as many processes required.
* The code consist of a process class named Process.java and the Process is initiated by class InitiateProcess.java . The class InitiateProcess.java takes 3 command line input for process name, process ID and total No of process. The use should use the same Process name as defined in Dependencies.txt i.e Process 1 is named P1 and ID is 1.
* The user needs to compile the source code by command **javac \*.java** using jdk 8 compiler.
* The user need to start each process separately in separately commad prompt windows .

Eg Say for current WFG the for starting Process P1 , use the following command

***Java InitiateProcess P1 1 7***

Similarly the use as to initiate rest if the 6 process also.

* The User needs to choose the Initiator Process i.e.. The process which initiates the enquiry for dead lock . For choosing say process P2 use command

***Java Initiator 2*** .

* Once deadlock detection is initiated, if dead lock is found then in the corresponding Initiator Process’s logs will print Deadlock Detection message.
* A sample video of the deadlock detection process and a screen shot have been attached to show the demo testing done by the developer.