**CS 6378.002**

**Project -2: Roucairol and Carvalho’s Distributed Mutual Exclusion Algorithm**

**Team Members:**

* MANEESH ABRAHAM (mea130130)
* RUCHIR KEMNAIK (rrk130330)
* KARAN LUNIYA (kal130230)

**Key Distribution:**

* Developed a program to generate arbitrary distribution of keys based on the number of processes provided as input.

**Algorithm Testing Mechanism:**

* **Detecting Critical Section violations:**
  + Using a lock file (as it is a shared file system) (During execution of the algorithm)
    - When a node enters the Critical Section, a lock file is created.
    - When a node leaves the Critical Section, the lock file is deleted.
    - Each process that needs access to the Critical Section, first checks if a lock already exists. Incase multiple nodes try to enter Critical Section at the same time, it will be detected as a lock file would exist and hence Critical Section violations will be detected.
  + Using Vector Timestamp (if there is no shared file system) (After execution of the algorithm)
    - The Vector Timestamp is logged when a node enters the Critical Section as well as when the node leaves the Critical Section.
    - Testing script parses the log file and compares the Vector Timestamps to check for any Critical section overlaps.
* **Verifying all Critical Section requests were satisfied:**
  + A line is written to the resource file each time a node enters Critical Section and similarly a line is written to the resource file each time a node leaves Critical Section.
  + Testing script parses the config file to determine how many number of lines is expected if the algorithm executed correctly.
  + Testing script then checks the number of lines in the resource file. If the Expected No. of lines matches the Actual No. of lines in the file, the algorithm executed correctly.