Distributed Systems Assignment 5

James Reinke and Will Tachau

* The Algorithm in Summary
  + The algorithm we have implemented for saving this Peer to Peer network takes as simplistic of a model as possible. Each node in our environment is represented by one non-storage process. Each of these non-storage processes is responsible for storing the dictionaries of the node above them in sequential order. This ensures that when a node goes down, those nodes are migrated to the new node that already holds their data. This allows us to efficiently transfer the data back to the migrated storage processes. When a node goes down our monitor warns us and we send a backup to be stored on the preceding node. Messages sent from this node can reach the preceding node in O(log(2^m)) time using our chord algorithm. We implemented in the chord algorithm in our function recipient. It finds the biggest jump a process can make without passing the target process while still obeying the rule of message passing neighbors. Storage processes are responsible for their own dictionary and no one else’s. They are also responsible for sending a store message to non-storage processes on the correct node. For snapshots, we send a message around in a ring starting with the process that received the message that requires a snapshot (first key, last key). The message runs around the circle, moving from each process node one number at a time. They pass along the message and send their dictionary information item by item to the process that started the snapshot algorithm.
* States
  + Non-Storage Processes
    - Remain in a single listening state. Unless terminated, they resume listening after performing a finite number of actions
  + Storage Processes
    - Has two listening states. One for listening to the outside controller and other processes. Another is a listener for the snapshot algorithm, where the process determines if it has collected the full snapshot or not.
* Information Stored by Each Process
  + Non-Storage processes store a backup of every single processes’ information located on the next node (ascending order by ID). They can backup process data when deleted processes are migrated to the hosting node. This allows for good locality of information.
* Message Types