Ryan Myers

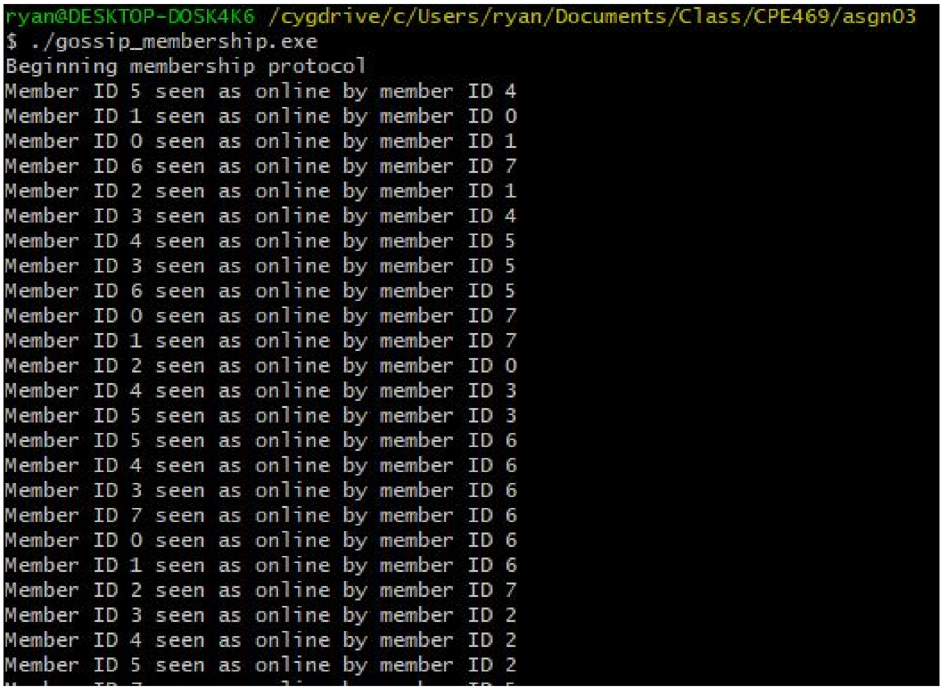
CPE 429

Dr. Maria Pantoja

Assignment 3: Gossip

The code for this assignment is all contained within the "gossip_membership.go" file. It can be built by running "go build gossip_membership.go" and then run by executing the resulting "gossip_membership.exe" file. To implement this algorithm and clearly show it working as expected I designed it so that each process that is spawned will update its own heartbeat every 10ms and share its membership table with one of its neighbors every 20ms. During this time, a random process will be put to sleep every 20 seconds and will stay asleep for 15 seconds before restarting. The program outputs information to the console during various events to show the gossip membership algorithm running as expected. The program will tell the user every time a process is killed and subsequently brought back online, whenever a process sees another come online (a member is added to its table), and whenever a process decides another is offline (a member is deleted from its table). Whenever a member is deleted from a processes membership table, the table is printed after to show its current state.

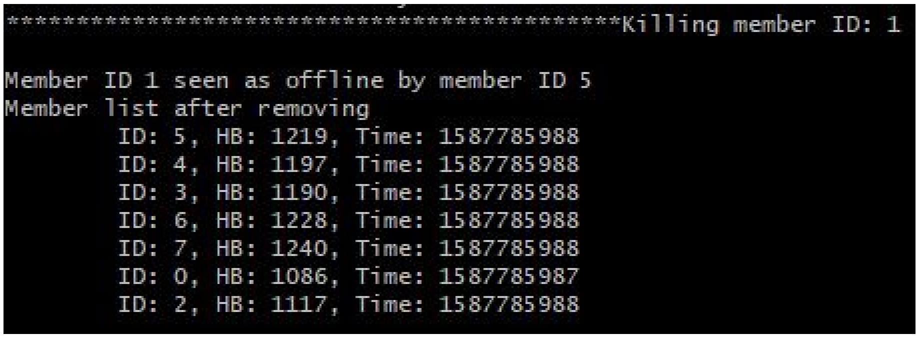
When the program begins, the user will see a large amount of console output as each node begins to receive membership tables from others and sees all the other nodes go online. An example of this can be seen below:



```
ryan@DESKTOP-DOSK4K6 /cygdrive/c/Users/ryan/Documents/Class/CPE469/asgn03
$ ./gossip_membership.exe
Beginning membership protocol
Member ID 5 seen as online by member ID 4
Member ID 1 seen as online by member ID 0
Member ID 0 seen as online by member ID 1
Member ID 6 seen as online by member ID 7
Member ID 2 seen as online by member ID 1
Member ID 3 seen as online by member ID 4
Member ID 4 seen as online by member ID 5
Member ID 3 seen as online by member ID 5
Member ID 6 seen as online by member ID 5
Member ID 0 seen as online by member ID 7
Member ID 1 seen as online by member ID 7
Member ID 2 seen as online by member ID 0
Member ID 4 seen as online by member ID 3
Member ID 5 seen as online by member ID 3
Member ID 5 seen as online by member ID 6
Member ID 4 seen as online by member ID 6
Member ID 3 seen as online by member ID 6
Member ID 7 seen as online by member ID 6
Member ID 0 seen as online by member ID 6
Member ID 1 seen as online by member ID 6
Member ID 2 seen as online by member ID 7
Member ID 3 seen as online by member ID 2
Member ID 4 seen as online by member ID 2
Member ID 5 seen as online by member ID 2
```

Figure 1: Program beginning and all nodes being discovered through the gossip protocol

When a process is put to sleep, the console will print that it is killing a specific member ID, this will be followed by other nodes in the system deciding the node went offline and printing their decision. This is shown in the following image:

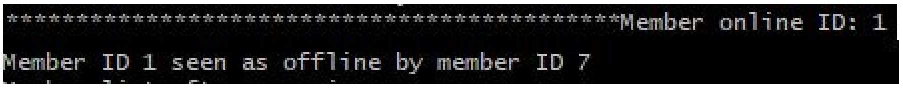


```
********************************************Killing member ID: 1

Member ID 1 seen as offline by member ID 5
Member list after removing
        ID: 5, HB: 1219, Time: 1587785988
        ID: 4, HB: 1197, Time: 1587785988
        ID: 3, HB: 1190, Time: 1587785988
        ID: 6, HB: 1228, Time: 1587785988
        ID: 7, HB: 1240, Time: 1587785988
        ID: 0, HB: 1086, Time: 1587785987
        ID: 2, HB: 1117, Time: 1587785988
```

Figure 2: Example of member 1 going offline and

Lastly, the program will print that a node has come back online and will then announce when the node has been seen as online by other members as shown by:



```
********************************************Member online ID: 1
Member ID 1 seen as offline by member ID 7
```