Description on how to run/use the code

1. In the project solution inside the debug folder, you will find .exe file run it multiple times equal to the number of processes that you want.
2. In the code “number of processes” is hard coded to handle 5 processes, this limitation can be changed easily by changing the variable that specify this number and the code should be able to handle any number of processes, but then you may need to change the time out value to be increased.
3. You should enter the id for the processes, inside the code the id is also the port that the processes will listen on, so each id should be unique and each id should be in range [1 – 5] as discussed above .
4. The master will always print in console when a child try to connect as part of failure detection, childes take the successes of the connection as indicator that master is alive.
5. In the console you will be notified for “election”, “victory message broad casting” and “assigning new master” inside the processes that initiate these events as
   1. Election process
   2. Victory broadcasting
   3. Assign new master.
6. When processes are receiving message from another processes the simulation will display as required: “message type, origin processes id, time stamp”.
7. You can run the processes, terminate or restart again in any order.
8. **Important note**: console sometimes freezes, not because of the application you just can click with the mouth on it or hit enter
9. id of the processes and id of the master get printed each time the master process complete it’s time out

.

1. \*((int\*)& msg[8]) = master\_id; ->> this expression mean dereferencing integer

pointer starting from the byte at index 8 in the char \*msg array

Documentation about the technology used for the communication between processes.

Socket TCP/IP technology is used for the communication between the processes

Reasons for choosing this TECH:

* Scalable: same code could be used to run on distribute computers with little modification.
* Guarantee that data arrive to its destination.
* another reason is that I was able to put my thoughts into “code logic” to implement the algorithm using the functions I learned on sockets.