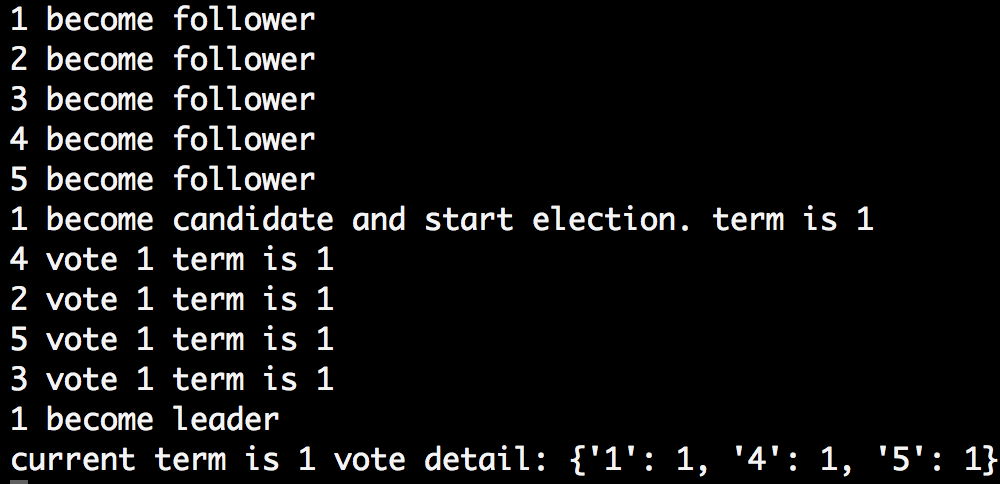
The tests consists of three parts: election, log replication and client-server communication.

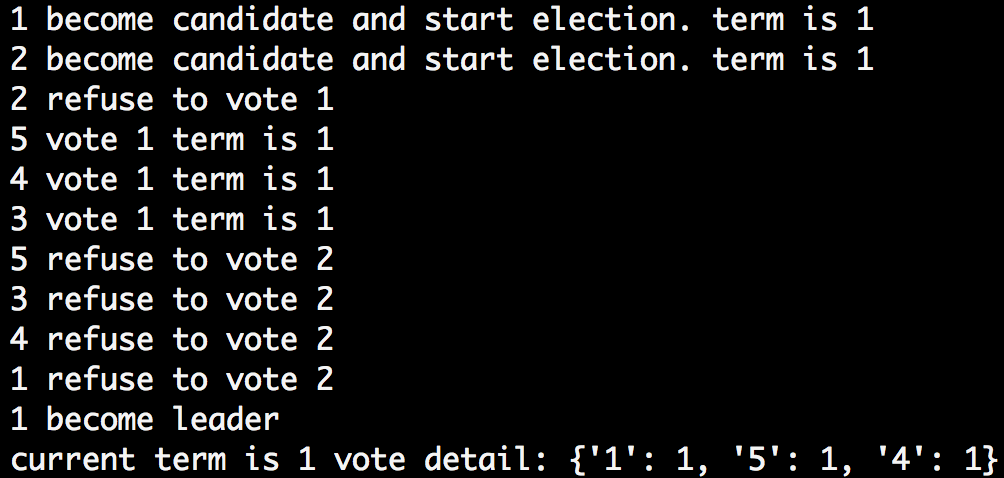
1. Election

The log of test script would indicate the state of node, the vote split and details of transition to leader.

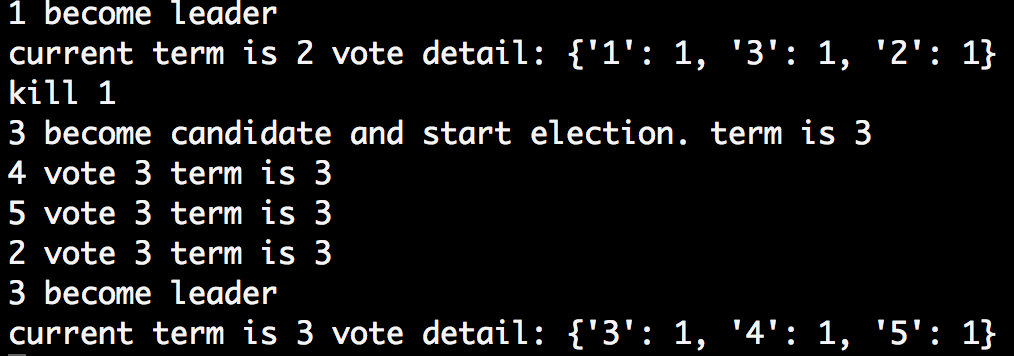
We simulate the scenarios when

i). Only one candidate: test\_election\_one\_candidate.py

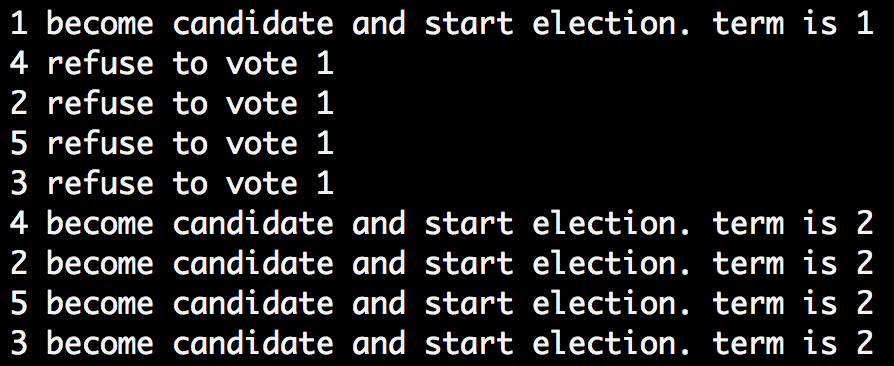
This is the most common case when initializing the system. The follower successfully converts to candidate and send request to each follower. Only if the voters receive vote request, they would log their vote. Then the log indicates the leader successfully receive the response from voter.

ii). Two candidates compete: test\_election\_two\_candidate.py

The log shows the vote split works correctly.

iii). Recover from the death of leader: test\_election\_when\_leader\_die.py

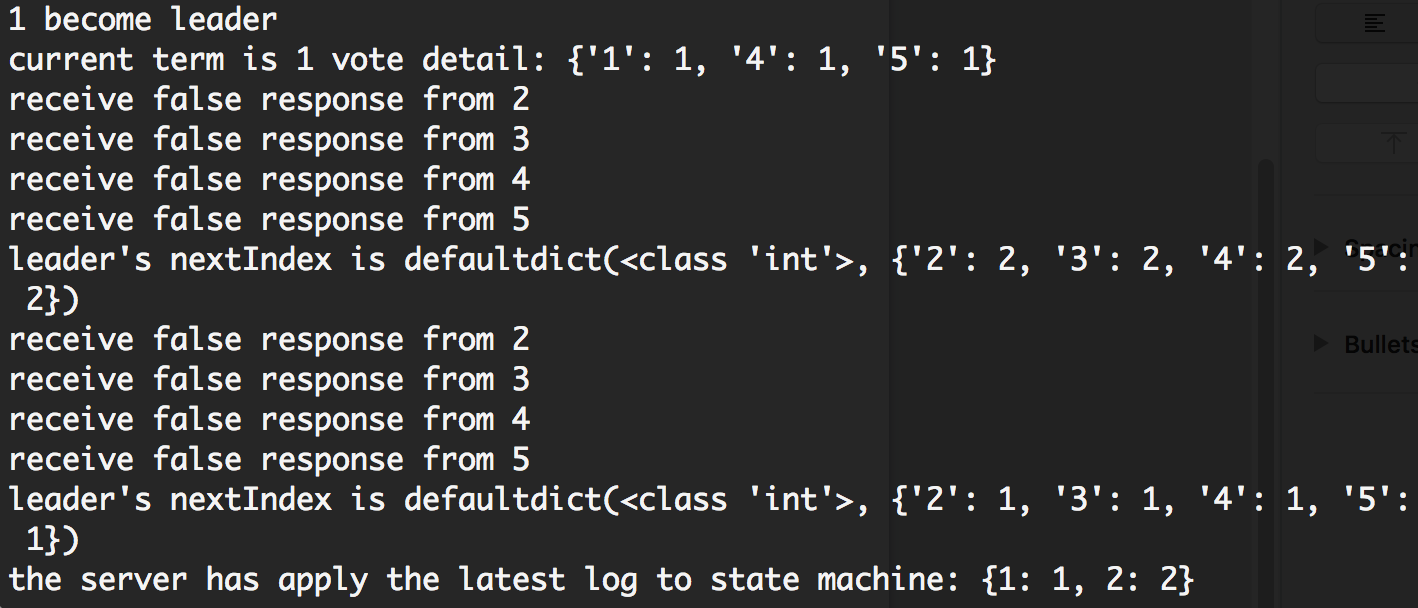
The election timeout is correctly triggered.

iv). Vote request from outdated log: test\_election\_old\_log.py

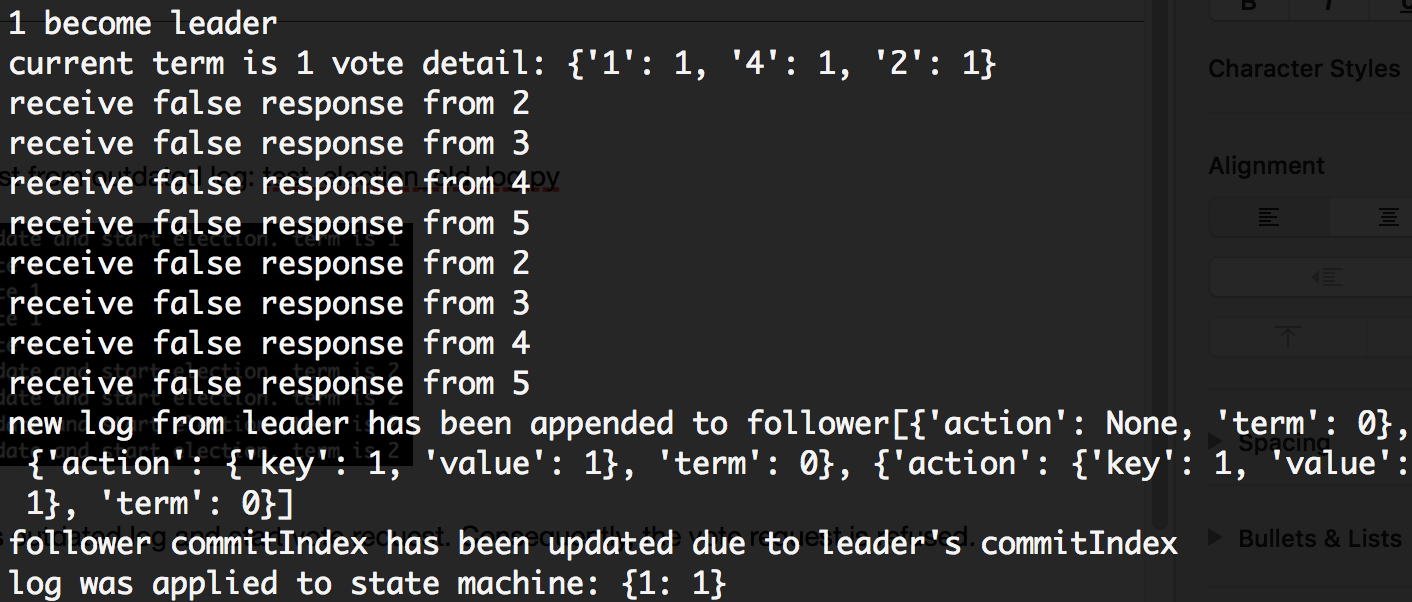
Node 1 carries outdated log and start vote request. Consequently, the vote request is refused.

2. Log replication

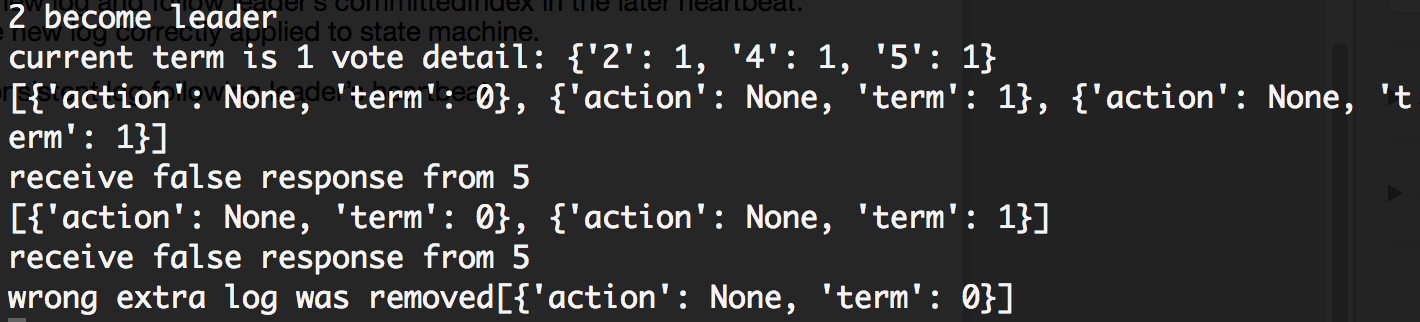
We test the workflow:

i). Leader update track of follower: test\_log\_replication\_leader\_update\_commitedIndex.py

After 1 becomes leader, it would correct the nextIndex for each follower according to the false response from follower. Then it would update the commit index and apply to state machine.

ii). Follower would update itself after receiving heartbeat: test\_log\_replication\_append\_request\_from\_leader.py

Then follower would insert the new log and follow leader’s committedIndex in the later heartbeat. With updated commitIndex, the new log correctly applied to state machine.

iii). Follower would remove inconsistent log following leader’s heartbeat :test\_log\_replication\_remove\_conflict\_log.py

Node 5 carries two more log of incorrect term (for purpose of test):

The log shows that inconsistence logs are removed by each heartbeat from leader.

3. Client-Server

We generate random put-get request to distributed system to test: test\_client.py