**Team:** Livar Cunha, Luciano Mandryk, Sergio Clemente

# Compiling & Executing

Just run ./compile.sh or ./execute.sh

If you run into any issues make sure you set your java home to something like the following:

|  |
| --- |
| [~] JAVA\_HOME=/Library/Java/JavaVirtualMachines/1.7.0.jdk/Contents/Home/bin  [~] export JAVA\_HOME |

Learner

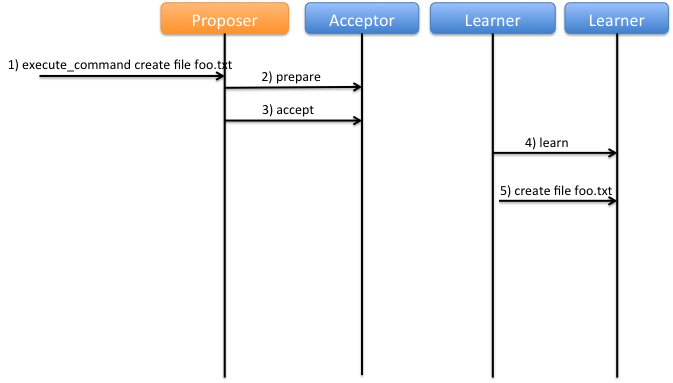
Also, make sure the *classes* folder exist.

# Storage System & Paxos

The proposer, accepter, learner and storage system are being implemented in the same node. The class name is PaxosNode. The proposer has a command called “execute\_command” that tries to add the given command as argument in the next available slot of the state machine. It basically runs the SYNOD (prepare/accept/learn).

If multiple proposers try to propose at the same time, only one will succeed, but the ones that fails will try into the next available slot.

Once the command is learned, it sends the command to the storage systems to be executed. You can imagine you have many storage systems that execute all commands once they are learned.



# Example of running

|  |
| --- |
| start 0  start 1  start 2  0 execute\_command create foo.txt  0 execute\_command put foo.txt hey\_foo!  1 execute\_command create bar.txt  1 execute\_command put bar.txt hey\_bar!  2 execute\_command create hello.txt  2 execute\_command put hello.txt hello world! |

# Assumptions

* We made all roles (proposer, learner, acceptor, storage system) in the same address. This was done just for simplification. The only hard requirement of the code is that the learner and storage system are located in the same address. This is to avoid implementing any extra protocol between the learner and storage system and also avoiding rpc costs.
* We used reliably delivery node. We think we could have optimized some parts of it but it was not required for the assignment.

# Failure Handling and Recovery

Serialized information:

* The proposer do not have any persistent state. If it crashes it will simply try to propose to a small number and discover the highest proposal number.
* The acceptor stores the following information:
  + Promised numbers
  + Accepted numbers
  + Accepted values
* The learner stores the following information:
  + The learned values