CS6378 Advanced Operating System Project 3 Report

Hanlin He (hxh160630)
Tao Wang (txw162630)

Sunday April 30, 2017

Contents

1 Compilation

The directories structure is as follow

```
/
├── META-INF
│   └── MANIFEST.MF
├── Main.java
├── core
│   ├── CLIEngine.java
│   ├── Message.java
│   ├── MessageProcessorThread.java
│   ├── MessageType.java
│   ├── Node.java
│   └── VoteData.java
└── log
```

```
    ├── EventType.java
    └── log.java
├── makefile
└── net
    ├── CommunicationThread.java
    ├── ConnectionInitiator.java
    └── ListenerThread.java
```

To compile, simply execute the `make` command in *src* directory would generate the executable jar file named `DynamicVoting.jar`. A example is shown as follow:

```
--- project3/src > make
javac -g Main.java
jar cvfm DynamicVoting.jar META-INF/MANIFEST.MF Main.class core/*.class net
    /*.class log/*.class
added manifest
adding: Main.class(in = 647) (out= 432)(deflated 33%)
adding: core/CLIEngine.class(in = 3802) (out= 1995)(deflated 47%)
adding: core/Message.class(in = 1430) (out= 677)(deflated 52%)
adding: core/MessageProcessorThread$1.class(in = 924) (out= 564)(deflated
    38%)
adding: core/MessageProcessorThread.class(in = 2475) (out= 1349)(deflated
    45%)
adding: core/MessageType.class(in = 1211) (out= 679)(deflated 43%)
adding: core/Node$1.class(in = 813) (out= 523)(deflated 35%)
adding: core/Node.class(in = 15385) (out= 7483)(deflated 51%)
adding: core/VoteData.class(in = 1609) (out= 763)(deflated 52%)
adding: net/CommunicationThread.class(in = 4271) (out= 2141)(deflated 49%)
adding: net/ConnectionInitiator.class(in = 2614) (out= 1370)(deflated 47%)
adding: net/ListenerThread.class(in = 2397) (out= 1279)(deflated 46%)
adding: log/EventType.class(in = 1116) (out= 637)(deflated 42%)
adding: log/log$1.class(in = 791) (out= 522)(deflated 34%)
adding: log/log.class(in = 2905) (out= 1458)(deflated 49%)
--- project3/src > make install
mv DynamicVoting.jar ..
```

After the execution, the executable jar file `DynamicVoting.jar` would be in the root directory of the project.

2 Execution

The command to execute the jar file is as follow:

`java -jar DynamicVoting.jar` *<label of the node>*

in which *<label of the node>* should avoid using the same label for different node.

There are seven commands that can be executing:

- help: Display the helping information.
- init: Initialize the voting data, must execute this command when all nodes are online for the first time.
- display: There are three kinds of information can be displayed, status, vote and connection.
- connect: Connect current to the nodes with specific node.
- disconnect: Disconnect current to the nodes with specific node.
- write: Write command for the object.
- quit/exit/q: Exit the program.

You can use `help` command to check the function of each command as follow

```
> help
Usage:
  <command> [options]

Commands:
  help                    Show help for commands.
  init                    Initialize vote data for all connected nodes.
  display [options]       Display current status.
                          Available options:
                              status,
                              vote,
                              connection/network.
  connect [options]       Connect to some nodes.
  connect [options]       Need to specify the label of the nodes.
                          Example: connect B C D
  disconnect [options]    Disconnect with some nodes.
  disconnect [options]    Need to specify the label of the nodes.
                          Example: disconnect B C D
  write                   Write to the object.
  quit/exit/q             Safe exit.
```

3 Result

The log file of each node are as follow.

```
2017-04-30 18:22:22: Write request success.
        Current Vote Data:VN(2) USC(8) DS([A])
2017-04-30 18:22:25: Write request success.
        Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:13: Write operation success.
        Current Vote Data:VN(4) USC(4) DS([A])
2017-04-30 18:23:16: Write request success.
        Current Vote Data:VN(5) USC(4) DS([A])
2017-04-30 18:23:37: Write operation failed, since node not in distinguished
    partition.
        Current Vote Data:VN(5) USC(4) DS([A])
```

Listing 1: Node A log

```
2017-04-30 18:22:22: Write request success.
        Current Vote Data:VN(2) USC(8) DS([A])
2017-04-30 18:22:25: Write request success.
        Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:13: Write request success.
        Current Vote Data:VN(4) USC(4) DS([A])
2017-04-30 18:23:16: Write request success.
        Current Vote Data:VN(5) USC(4) DS([A])
2017-04-30 18:23:44: Write request success.
        Current Vote Data:VN(6) USC(3) DS([D, C, B])
2017-04-30 18:23:47: Write operation success.
        Current Vote Data:VN(7) USC(3) DS([B, C, D])
2017-04-30 18:24:16: Write request success.
        Current Vote Data:VN(8) USC(6) DS([B])
2017-04-30 18:24:20: Write operation success.
        Current Vote Data:VN(9) USC(6) DS([B])
```

Listing 2: Node B log

```
2017-04-30 18:22:22: Write request success.
        Current Vote Data:VN(2) USC(8) DS([A])
2017-04-30 18:22:25: Write operation success.
        Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:13: Write request success.
        Current Vote Data:VN(4) USC(4) DS([A])
2017-04-30 18:23:16: Write operation success.
        Current Vote Data:VN(5) USC(4) DS([A])
2017-04-30 18:23:44: Write request success.
        Current Vote Data:VN(6) USC(3) DS([D, C, B])
2017-04-30 18:23:47: Write request success.
        Current Vote Data:VN(7) USC(3) DS([B, C, D])
2017-04-30 18:24:16: Write request success.
        Current Vote Data:VN(8) USC(6) DS([B])
2017-04-30 18:24:20: Write request success.
        Current Vote Data:VN(9) USC(6) DS([B])
```

Listing 3: Node C log

```
2017-04-30 18:22:22: Write request success.
         Current Vote Data:VN(2) USC(8) DS([A])
2017-04-30 18:22:25: Write request success.
         Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:13: Write request success.
         Current Vote Data:VN(4) USC(4) DS([A])
2017-04-30 18:23:16: Write request success.
         Current Vote Data:VN(5) USC(4) DS([A])
2017-04-30 18:23:44: Write operation success.
         Current Vote Data:VN(6) USC(3) DS([D, C, B])
2017-04-30 18:23:47: Write request success.
         Current Vote Data:VN(7) USC(3) DS([B, C, D])
2017-04-30 18:24:16: Write request success.
         Current Vote Data:VN(8) USC(6) DS([B])
2017-04-30 18:24:20: Write request success.
         Current Vote Data:VN(9) USC(6) DS([B])
```

Listing 4: Node D log

```
2017-04-30 18:22:22: Write request success.
         Current Vote Data:VN(2) USC(8) DS([A])
2017-04-30 18:22:25: Write request success.
         Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:05: Write operation aborted by coordinate node.
         Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:07: Write operation aborted by coordinate node.
         Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:39: Write operation failed, since node not in distinguished
     partition.
         Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:42: Write operation aborted by coordinate node.
         Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:24:16: Write request success.
         Current Vote Data:VN(8) USC(6) DS([B])
2017-04-30 18:24:20: Write request success.
         Current Vote Data:VN(9) USC(6) DS([B])
```

Listing 5: Node E log

```
2017-04-30 18:22:22: Write operation success.
         Current Vote Data:VN(2) USC(8) DS([A])
2017-04-30 18:22:25: Write request success.
         Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:05: Write operation failed, since node not in distinguished
     partition.
         Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:07: Write operation aborted by coordinate node.
         Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:39: Write operation aborted by coordinate node.
         Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:23:42: Write operation aborted by coordinate node.
         Current Vote Data:VN(3) USC(8) DS([A])
2017-04-30 18:24:16: Write request success.
         Current Vote Data:VN(8) USC(6) DS([B])
2017-04-30 18:24:20: Write request success.
```

```
16          Current Vote Data:VN(9) USC(6) DS([B])
```

Listing 6: Node F log

```
 1 2017-04-30 18:22:22: Write request success.
 2          Current Vote Data:VN(2) USC(8) DS([A])
 3 2017-04-30 18:22:25: Write request success.
 4          Current Vote Data:VN(3) USC(8) DS([A])
 5 2017-04-30 18:23:05: Write operation aborted by coordinate node.
 6          Current Vote Data:VN(3) USC(8) DS([A])
 7 2017-04-30 18:23:07: Write operation aborted by coordinate node.
 8          Current Vote Data:VN(3) USC(8) DS([A])
 9 2017-04-30 18:23:39: Write operation aborted by coordinate node.
10          Current Vote Data:VN(3) USC(8) DS([A])
11 2017-04-30 18:23:42: Write operation failed, since node not in distinguished
       partition.
12          Current Vote Data:VN(3) USC(8) DS([A])
13 2017-04-30 18:24:16: Write operation success.
14          Current Vote Data:VN(8) USC(6) DS([B])
15 2017-04-30 18:24:20: Write request success.
16          Current Vote Data:VN(9) USC(6) DS([B])
```

Listing 7: Node G log

```
 1 2017-04-30 18:22:22: Write request success.
 2          Current Vote Data:VN(2) USC(8) DS([A])
 3 2017-04-30 18:22:25: Write request success.
 4          Current Vote Data:VN(3) USC(8) DS([A])
 5 2017-04-30 18:23:05: Write operation aborted by coordinate node.
 6          Current Vote Data:VN(3) USC(8) DS([A])
 7 2017-04-30 18:23:07: Write operation failed, since node not in distinguished
       partition.
 8          Current Vote Data:VN(3) USC(8) DS([A])
 9 2017-04-30 18:23:34: Write operation failed, since node not in distinguished
       partition.
10          Current Vote Data:VN(3) USC(8) DS([A])
```

Listing 8: Node H log

Note that:

- For each `write` operation on one node, *all the connected nodes* would record an event whether that write operation is success or not.
- `Write operation success.` event on a node's log indicates that the `write` operation was on that node, i,e current node is the coordinator of the `write` operation.
- `Write request success.` event on a node's log indicates that the `write` operation was on another node, i,e current node is the cohort of the `write` operation.