

Example of CENTRALISED MUTEX with a COORDINATOR.

In the next page you will see

TOPLEFT:

The coordinator is keeping track of requests and issuing tokens.

Messages are labelled according to the component that issues them.
For instance

C:mutex ...

says something about the current state of requests - e.g. the buffer. Other messages report when requests have been received and from whom.

These messages are useful to show what the coordinator is doing...
You may have a flag that sets how verbose the logging has to be (e.g. None, Running, Debugging -- optional!).

OTHER THREE WINDOWS

The three windows are reporting the behaviour of the nodes. In particular they show the execution of a critical section (in mutual exclusion). In each of them it is/has to be clear

- the start and
- stop of the mutual exclusion
- where the token is coming from and where it is going to (which nodes)

Please allow enough time in each critical section so as to be able to understand what is going on. Different agent may have different time-to-completion for their critical section.

IMPORTANT: no any two nodes can be inside the critical section at the same time / once one has started its critical section no other node should be inside its own critical section.

>>> END <<<

Token to 127.0.0.1:7001
TOKEN with 127.0.0.1:8003

>>> MUTEX <<<

Returning TOKEN to 127.0.0.1:7001

>>> END <<<

Token to 127.0.0.1:7001

>>> END <<<

Token to 127.0.0.1:7001
TOKEN with 127.0.0.1:8001

>>> MUTEX <<<

Returning TOKEN to 127.0.0.1:7001

>>> END <<<

Token to 127.0.0.1:7001

```
C:mutex Buffer size is 4
C:receiver Coordinator has received a request ...
C:connection IN dealing with request from socket Socket[addr=/127.0.0.1,port=54012,localp
ort=7000]
C:connection OUT received and recorded request from 127.0.0.1:8000 (socket closed)
127.0.0.1 8001 127.0.0.1 8003 127.0.0.1 8000
C:mutex The buffer is 127.0.0.1 8001 127.0.0.1 8003 127.0.0.1 8000
MUTEX: Token to 127.0.0.1:8001
BACK!!
```

```
C:mutex Buffer size is 4
C:receiver Coordinator has received a request ...
C:connection IN dealing with request from socket Socket[addr=/127.0.0.1,port=54021,localp
ort=7000]
C:connection OUT received and recorded request from 127.0.0.1:8001 (socket closed)
127.0.0.1 8003 127.0.0.1 8000 127.0.0.1 8001
C:mutex The buffer is 127.0.0.1 8003 127.0.0.1 8000 127.0.0.1 8001
java.net.ConnectException: Connection refused
CRASH Mutex connecting to the node for granting the TOKENjava.net.ConnectException: Connecti
on refused
```

>>> END <<<

Token to 127.0.0.1:7001
TOKEN with 127.0.0.1:8000

>>> MUTEX <<<

Returning TOKEN to 127.0.0.1:7001

>>> END <<<

Token to 127.0.0.1:7001