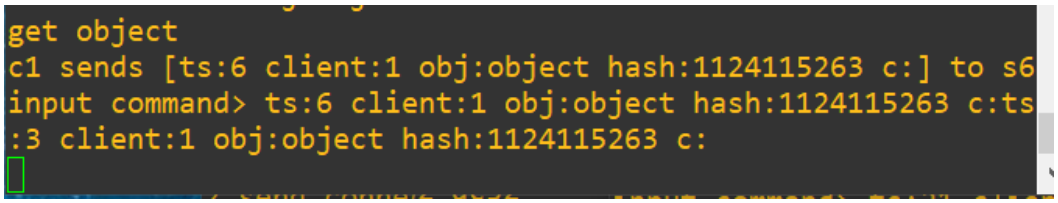
Start with:

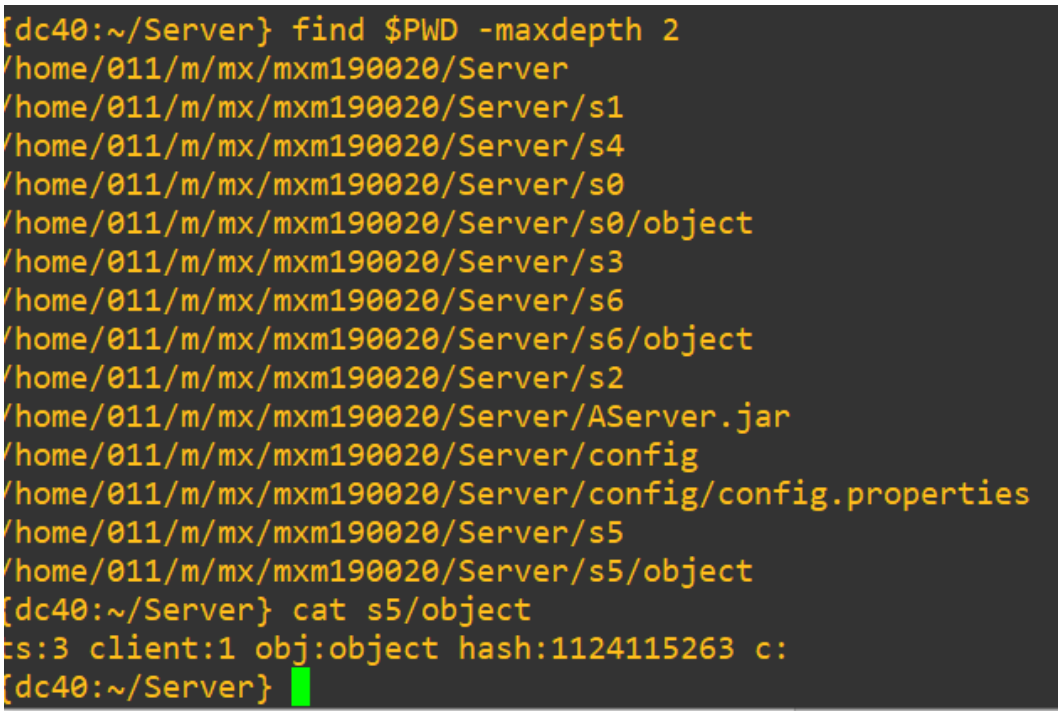
```
{dc40:~/Server} find $PWD -maxdepth 2
/home/011/m/mx/mxm190020/Server
/home/011/m/mx/mxm190020/Server/s1
/home/011/m/mx/mxm190020/Server/s4
/home/011/m/mx/mxm190020/Server/s0
/home/011/m/mx/mxm190020/Server/s3
/home/011/m/mx/mxm190020/Server/s6
/home/011/m/mx/mxm190020/Server/s2
/home/011/m/mx/mxm190020/Server/AServer.jar
/home/011/m/mx/mxm190020/Server/config
/home/011/m/mx/mxm190020/Server/config/config.properties
/home/011/m/mx/mxm190020/Server/s5
```

1. Test functions: create, update, get
   Client1: create object, update object

```
input command> create object
c1 sends [ts:0 client:1 obj:object hash:1124115263 c:] to s
5
c1 sends [ts:0 client:1 obj:object hash:1124115263 c:] to s
6
c1 sends [ts:0 client:1 obj:object hash:1124115263 c:] to s
0
input command> ts:0 client:1 obj:object hash:1124115263 c:
ts:0 client:1 obj:object hash:1124115263 c:
ts:0 client:1 obj:object hash:1124115263 c:
update object
c1 sends [ts:3 client:1 obj:object hash:1124115263 c:] to s
5
c1 sends [ts:3 client:1 obj:object hash:1124115263 c:] to s
6
c1 sends [ts:3 client:1 obj:object hash:1124115263 c:] to s
0
input command> ts:3 client:1 obj:object hash:1124115263 c:
ts:3 client:1 obj:object hash:1124115263 c:
ts:3 client:1 obj:object hash:1124115263 c:
```

Client1: get object



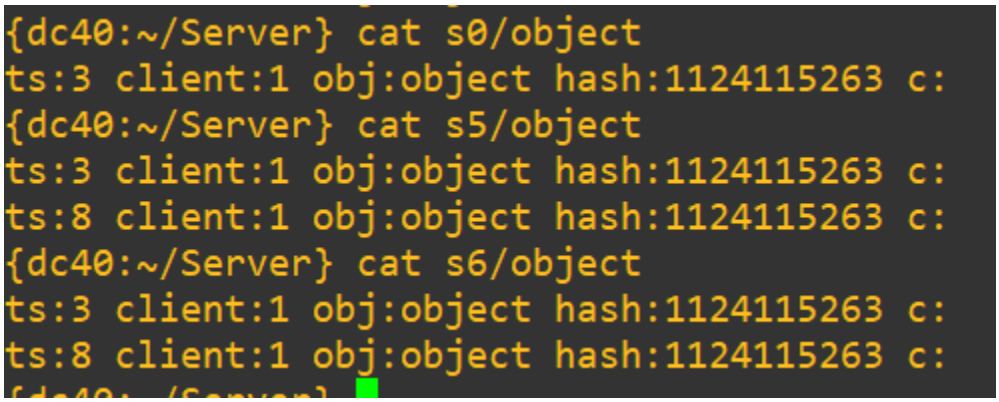


After object2 updated by client0, client5 and client6:

```
{dc40:~/Server} cat s0/object2
ts:3 client:3 obj:object2 hash:487834835 c:
ts:6 client:3 obj:object2 hash:487834835 c:
ts:0 client:2 obj:object2 hash:487834835 c:
{dc40:~/Server} cat s6/object2
ts:3 client:3 obj:object2 hash:487834835 c:
ts:6 client:3 obj:object2 hash:487834835 c:
ts:0 client:2 obj:object2 hash:487834835 c:
{dc40:~/Server} cat s5/object2
ts:3 client:3 obj:object2 hash:487834835 c:
ts:6 client:3 obj:object2 hash:487834835 c:
ts:0 client:2 obj:object2 hash:487834835 c:
{dc40:~/Server}
```

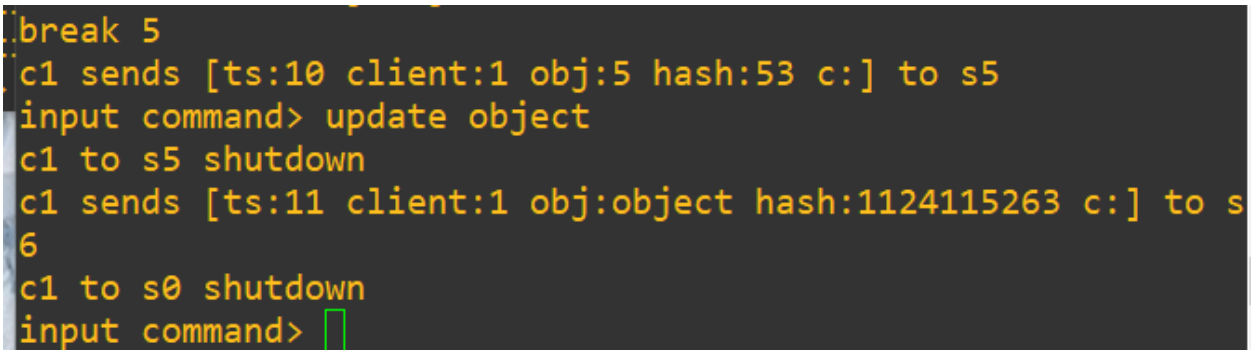
2. Test break a channel between a client and a server

   Client1 breaks its channel with server0, then sends an update command. Below is what object's content would look like on server 0, 5 and 6.

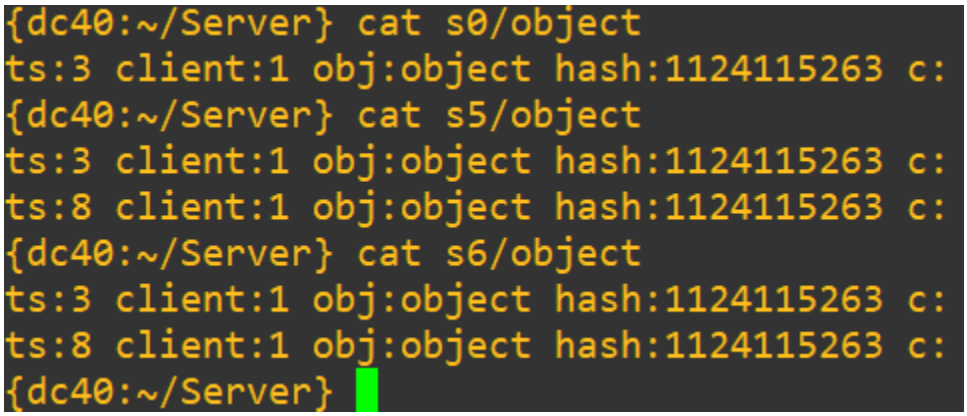


3. Test break two channels between a client and a server

   Client1 then breaks its channel with server5, then sends an update command.

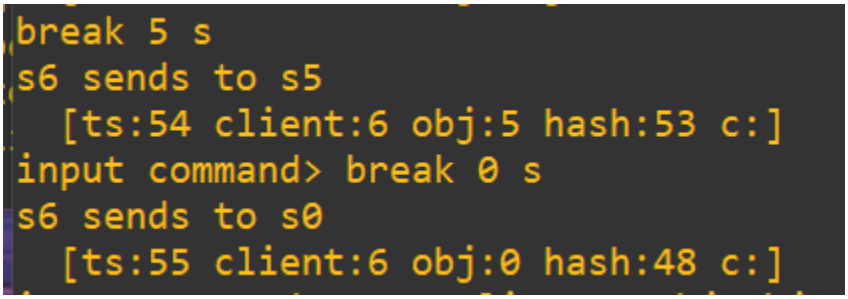


   Below is what object's content would look like on server 0, 5 and 6. Noting is changed.

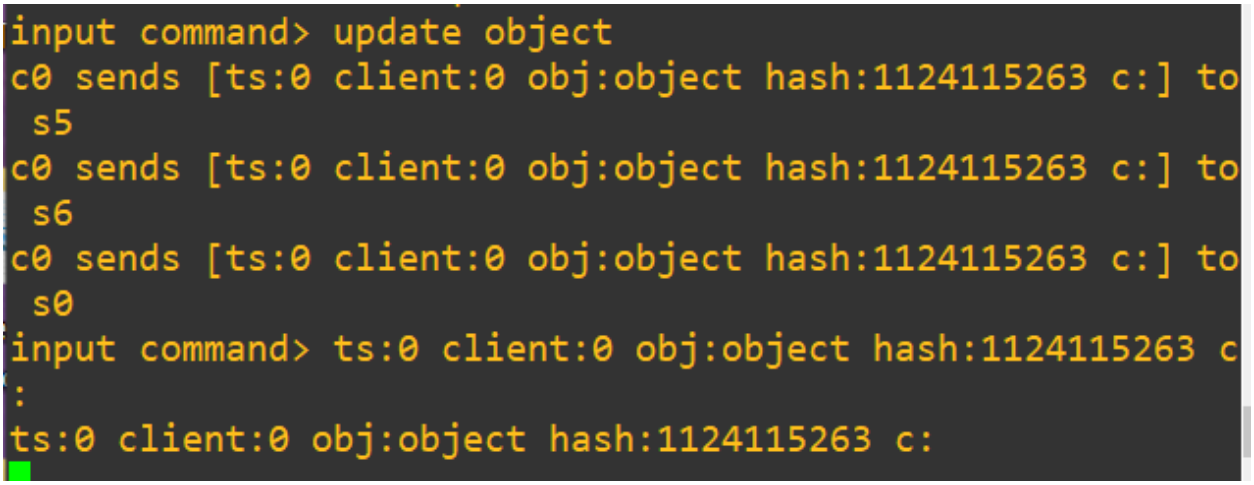


4. Test a server break with both of its neighbors

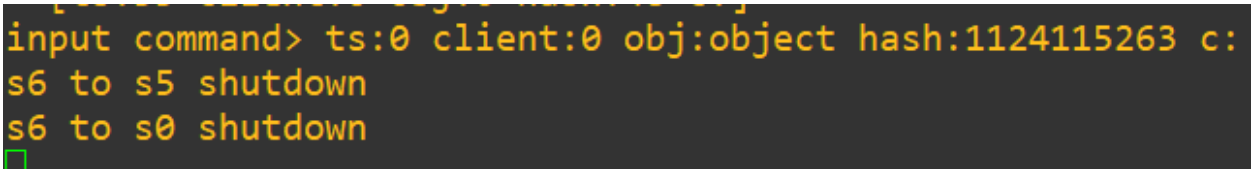
Server6 breaks its channel with s0 and s5.



Then Client0 sends an update command of object.



Server6 receives this command and tries to send message to s0 and s5.



Server5 receives command and tries to forward it to s0 and s6. S5 receives the message forwarded by s0, so object is successfully appended. Server0 is the same.

```
ts:0 client:0 obj:object hash:1124115263 c:
s5 to s6 shutdown
s5 sends to s0
  [ts:0 client:0 obj:object hash:1124115263 c:]
ts:0 client:0 obj:object hash:1124115263 c:
object appended!
s5 sends to c0
  [ts:0 client:0 obj:object hash:1124115263 c:]
```

S0 and s5 successfully append the message to object, s6 does not.

```
{dc40:~/Server} cat s0/object
ts:3 client:1 obj:object hash:1124115263 c:
ts:0 client:0 obj:object hash:1124115263 c:
{dc40:~/Server} cat s5/object
ts:3 client:1 obj:object hash:1124115263 c:
ts:8 client:1 obj:object hash:1124115263 c:
ts:0 client:0 obj:object hash:1124115263 c:
{dc40:~/Server} cat s6/object
ts:3 client:1 obj:object hash:1124115263 c:
ts:8 client:1 obj:object hash:1124115263 c:
```

5. Test breaking servers and clients into two components

   Separation:
   {c0, c1, c2, s0, s5, s6}.
   {c3, c4, s1, s2, s3, s4}

   An object named “star” was stored in {s4, s5,s6}. It was created and updated once by client4 before the separation.

   After separation, c3 tried to update star, which failed.

```
c3 sends [ts:3 client:3 obj:star hash:3540562 c:] to s4
c3 to s5 shutdown
c3 to s6 shutdown
```

   Then client2 sent an update command, which was completed by s5 and s6.

```
input command> update star
c2 to s4 shutdown
c2 sends [ts:4 client:2 obj:star hash:3540562 c:] to s5
c2 sends [ts:4 client:2 obj:star hash:3540562 c:] to s6
input command> ts:4 client:2 obj:star hash:3540562 c:
ts:4 client:2 obj:star hash:3540562 c:
```

```
{dc40:~/Server} cat s4/star
ts:3 client:4 obj:star hash:3540562 c:
{dc40:~/Server} cat s5/star
ts:3 client:4 obj:star hash:3540562 c:
ts:4 client:2 obj:star hash:3540562 c:
{dc40:~/Server} cat s6/star
ts:3 client:4 obj:star hash:3540562 c:
ts:4 client:2 obj:star hash:3540562 c:
```