Alexandre Bescond
Master student
alexabes@student.matnat.uio.no

Etienne Bernoux
Master student
etiennb@student.matnat.uio.no

Assigmnent 2

INF5040 - Open distributed processing

Autumn semester 2017

Table of contents

1. EPHEMERAL VS PERSISTENT (REGULAR) ZNODES

1.1 CREATE A ZNODE

```
[zk: 127.0.0.1:2181(CONNECTED) 8] ls /
[zookeeper]
[zk: 127.0.0.1:2181(CONNECTED) 9] create /1/ex1 data
Node does not exist: /1/ex1
[zk: 127.0.0.1:2181(CONNECTED) 10] create /1 data
Created /1
[zk: 127.0.0.1:2181(CONNECTED) 11] ls /
[1, zookeeper]
[zk: 127.0.0.1:2181(CONNECTED) 12] create /1/ex1 data
Created /1/ex1
```

1.2 CREATE ANOTHER PERSISTENT NODE

```
[zk: 127.0.0.1:2181(CONNECTED) 15] create /1/ex1/persistent data
Created /1/ex1/persistent
```

1.3 CREATE NODE EPHEMERAL

```
[zk: 127.0.0.1:2181(CONNECTED) 0] create -e /1/ex1/ephemeral data
Created /1/ex1/ephemeral
```

1.4 LIST THE CHILDREN

```
[zk: 127.0.0.1:2181(CONNECTED) 17] ls /1/ex1
[ephemeral, persistent]
```

1.5 NOW QUIT THE CLIENT BY

```
[zk: 127.0.0.1:2181(CONNECTED) 18] quit
Quitting...
2017-10-19 19:29:21,885 [myid:] - INFO  [main:ZooKeeper@684] - Session: 0x15f359930020000 closed
2017-10-19 19:29:21,886 [myid:] - INFO  [main-EventThread:ClientCnxn$EventThread@519] - EventThread shut down for session: 0x15f359930020000
```

```
WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[zk: 127.0.0.1:2181(CONNECTED) 0] ls /1/ex1
[persistent]
```

The ephemeral is gone :'(, the session who create the node has gone so the node as disappear too. A ephemeral node (create cy -e) is dependent on a session. By quitting the session this node disappear.

1.6 CREATE A NEW EPHEMERAL ZNODE

```
[zk: 127.0.0.1:2181(CONNECTED) 0] create -e /1/ex1/ephemeral1 data
Created /1/ex1/ephemeral1
[zk: 127.0.0.1:2181(CONNECTED) 1] create /1/ex1/ephemeral1/child data
Ephemerals cannot have children: /1/ex1/ephemeral1/child
[zk: 127.0.0.1:2181(CONNECTED) 2] create -e /1/ex1/ephemeral1/child data
Ephemerals cannot have children: /1/ex1/ephemeral1/child
```

Because of if ephemeral behavior, this node cannot have children.

2. SEQUENTIAL SUFFIX

2.1 CREATE THE ZNODE FOR THIS EXERCISE

```
[zk: 127.0.0.1:2181(CONNECTED) 4] create /1/ex2 data
Created /1/ex2
```

2.2 CREATE A FEW ZNODES WITH SEQUENTIAL SUFFIX

```
Created /1/ex2
[zk: 127.0.0.1:2181(CONNECTED) 5] create -s /1/ex2/child someData
Created /1/ex2/child0000000000
[zk: 127.0.0.1:2181(CONNECTED) 6] create -s /1/ex2/child1 someOtherData
Created /1/ex2/child10000000001
[zk: 127.0.0.1:2181(CONNECTED) 7] create -s /1/ex2/child2 someData
Created /1/ex2/child20000000002
```

Q1

```
[zk: 127.0.0.1:2181(CONNECTED) 10] ls /1/ex2
[child10000000001, child20000000002, child0000000000]
```

The sequential mode add an increasing couter at the end of the name. This counter value is unique for this parentZNode. The counter have a 10 digits formats starting at 0. With a 0 padding.

Q2

```
WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[zk: 127.0.0.1:2181(CONNECTED) 0] ls /1
[ex2, ex1]
[zk: 127.0.0.1:2181(CONNECTED) 1] ls /1/ex2
[child10000000001, childtoto, childn0000000003, child20000000002, child0000000000]
```

Yes, after restarting the session, we steal have our node

Q3

```
[zk: 127.0.0.1:2181(CONNECTED) 16] create -s -e /1/ex2/child3 data
Created /1/ex2/child30000000019
[zk: 127.0.0.1:2181(CONNECTED) 17] create -s -e /1/ex2/child3 data
Created /1/ex2/child30000000020
```

Yes it's possible, they disappear after restart

Q4

```
[zk: 127.0.0.1:2181(CONNECTED) 19] create -s /1/ex2/seq data
Created /1/ex2/seq0000000027
[zk: 127.0.0.1:2181(CONNECTED) 20] create /1/ex2/nosequ3 data
Created /1/ex2/nosequ3
[zk: 127.0.0.1:2181(CONNECTED) 21] delete /1/ex2/nosequ3
[zk: 127.0.0.1:2181(CONNECTED) 22] create -s /1/ex2/seq data
Created /1/ex2/seq0000000029
```

The counter is incremented at each node creation whatever the properties of node

The counter does not decrease when node delete

Q5

The suffix is 0000000025 (start at 0000000000)

2.3 SCOPE OF SEQUENCE NUMBERS

```
[zk: 127.0.0.1:2181(CONNECTED) 23] create -s /1/ex2/child data
Created /1/ex2/child0000000030
[zk: 127.0.0.1:2181(CONNECTED) 24] create -s /1/ex2/child/toto data
Node does not exist: /1/ex2/child/toto
[zk: 127.0.0.1:2181(CONNECTED) 25] create  /1/ex2/child data
Created /1/ex2/child
[zk: 127.0.0.1:2181(CONNECTED) 26] create -s /1/ex2/child/toto data
Created /1/ex2/child/toto0000000000
[zk: 127.0.0.1:2181(CONNECTED) 27] create -s /1/ex2/child/toto data
Created /1/ex2/child/toto0000000001
[zk: 127.0.0.1:2181(CONNECTED) 28] create -s /1/ex2/child/toto data
Created /1/ex2/child/toto0000000002
[zk: 127.0.0.1:2181(CONNECTED) 29] create -s /1/ex2/child/toto data
Created /1/ex2/child/toto0000000003
```

No, they are not related

2.4 SEQUENCE NUMBERS ACROSS MULTIPLE CLIENTS)

```
Created /1/ex2/child/toto0000000000
[zk: 127.0.0.1:2181(CONNECTED) 27] create -s /1/ex2/child/toto data
Created /1/ex2/child/toto0000000001
[zk: 127.0.0.1:2181(CONNECTED) 28] create -s /1/ex2/child/toto data
Created /1/ex2/child/toto0000000002
[zk: 127.0.0.1:2181(CONNECTED) 29] create -s /1/ex2/child/toto data
Created /1/ex2/child/toto0000000003
[zk: 127.0.0.1:2181(CONNECTED) 30]
```

```
WatchedEvent state:SyncConnected type:None path:null
[zk: 127.0.0.1:2181(CONNECTED) 0] ls /1/ex2/child
[toto0000000002, toto0000000003, toto0000000000, toto0000000001]
[zk: 127.0.0.1:2181(CONNECTED) 1] create -s /1/ex2/child toto
Created /1/ex2/child0000000032
[zk: 127.0.0.1:2181(CONNECTED) 2] create -s /1/ex2/child/toto data
Created /1/ex2/child/toto0000000004
[zk: 127.0.0.1:2181(CONNECTED) 3]
```

Yes are related, the counter remain across the session

3. WATCHES

3.1 CREATE THE ROOT ZNODE FOR THIS EXERCISE WITH PATH

```
[zk: 127.0.0.1:2181(CONNECTED) 33] create /1/ex3 data
Created /1/ex3
```

3.2 FIRST WATCH

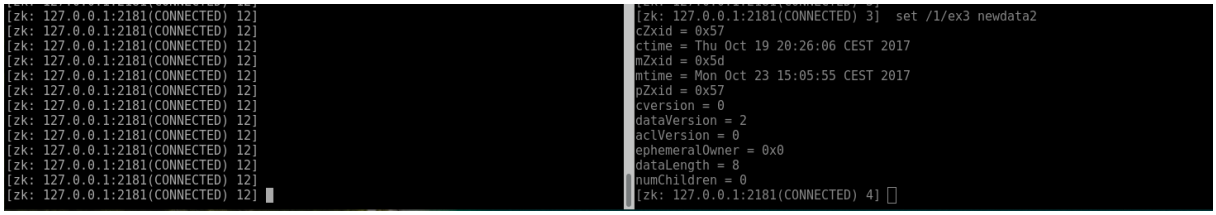
After setting the watch, client 1 receive an event telling that the data has been modified.

```
[zk: 127.0.0.1:2181(CONNECTED) 11] get /1/ex3 true
data
cZxid = 0x57
ctime = Thu Oct 19 20:26:06 CEST 2017
mZxid = 0x57
mtime = Thu Oct 19 20:26:06 CEST 2017
pZxid = 0x57
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 4
numChildren = 0
[zk: 127.0.0.1:2181(CONNECTED) 12]
WATCHER::

WatchedEvent state:SyncConnected type:NodeDataChanged path:/1/ex3
```

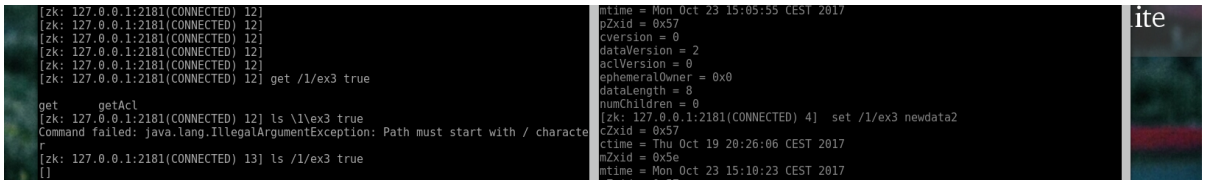
```
[zk: 127.0.0.1:2181(CONNECTED) 1] create set/1/ex3 newdata
Command failed: java.lang.IllegalArgumentException: Path must start with / ch
racter
[zk: 127.0.0.1:2181(CONNECTED) 2]  set /1/ex3 newdata
cZxid = 0x57
ctime = Thu Oct 19 20:26:06 CEST 2017
mZxid = 0x5c
mtime = Mon Oct 23 15:02:45 CEST 2017
pZxid = 0x57
cversion = 0
dataVersion = 1
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 7
numChildren = 0
[zk: 127.0.0.1:2181(CONNECTED) 3]
[zk: 127.0.0.1:2181(CONNECTED) 3]
[zk: 127.0.0.1:2181(CONNECTED) 3]
```

3.3 DURABILITY OF WATCHES



Nothing happen, it's a "one-time" trigger by default. Therefore, each trigger is trigged only once.

3.4 WATCH ON CHILDREN



Nothing happen because we are watching for the child and not the nodes itself

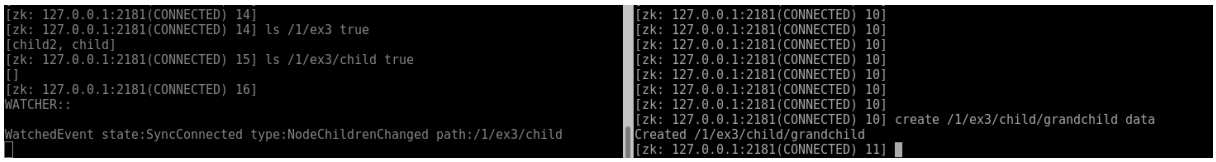
3.5 CHILD WATCHES

The child creation event is trigger

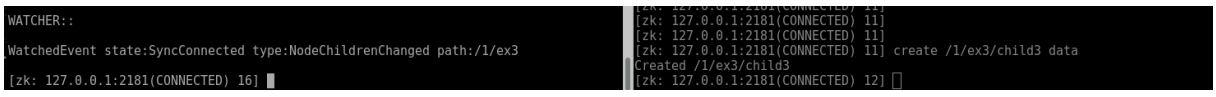
3.6 SUMMARIZE WATCH TYPE

command	*node*	*child*
get	Get trigger node change	Nothing happen
ls	Nothing happen	Ls trigger child node change

3.7 CHILD WATCHING AND NESTING



It trigger only the immediate parent above the new child. So only one trigger raise an event.



Yes, it trigger the watcher on root child, but no event is raise on child one.

3.8 WATCH ZNODE DELETION

They are all triggered

3.9 MORE WATCHES

Good review 😊