|  |
| --- |
| 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

[Introduction 2](#_Toc495571948)

[1. Spreading the word 2](#_Toc495571949)

[2. Tracing non-projectivity 2](#_Toc495571950)

[3. Using parsers 3](#_Toc495571951)

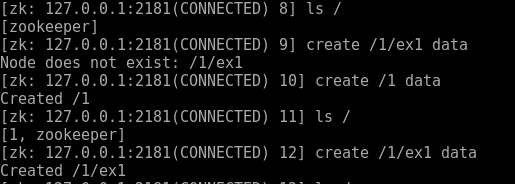
[3.1 Training 3](#_Toc495571952)

[3.2 Evaluation 3](#_Toc495571953)

[4. Parsing one language with another 6](#_Toc495571954)

# Ephemeral vs Persistent (Regular) znodes

## Create a znode



## Create another persistent node



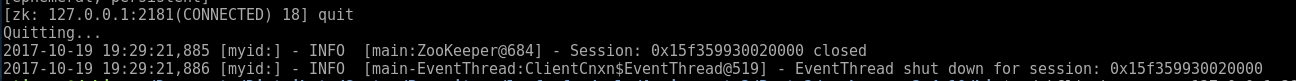
## Create node ephemeral

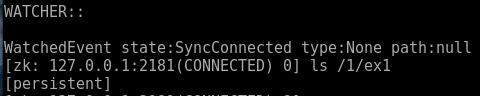


## List the children



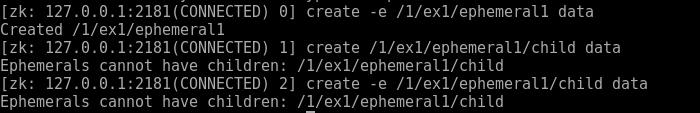
## Now quit the client by





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.

## Create a new ephemeral znode



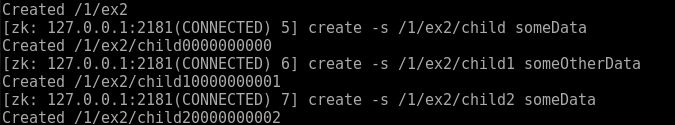
Because of if ephemeral behavior, this node cannot have children.

# Sequential suffix

## Create the znode for this exercise



## Create a few znodes with SEQUENTIAL suffix

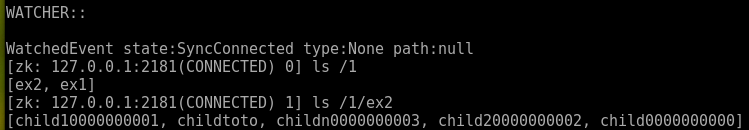


#### Q1



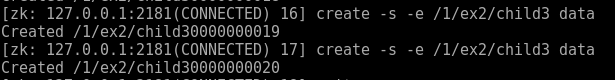
The sequential mode add an increasing couter at the end of the name. This counter value is unique for this parentZNode. The counter have a 10 digits formats starting at 0. With a 0 padding.

#### Q2



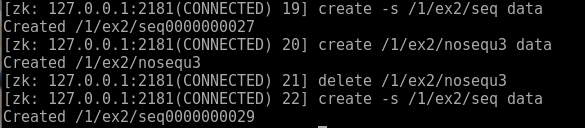
Yes, after restarting the session, we steal have our node

#### Q3



Yes it’s possible, they disappear after restart

#### Q4



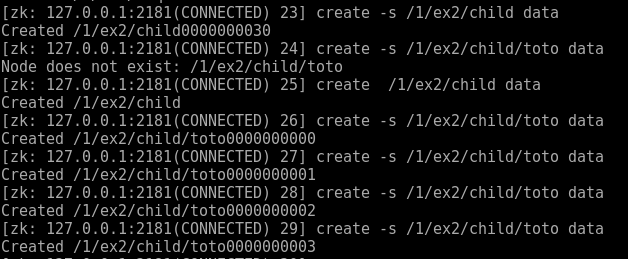
The counter is incremented at each node creation whatever the properties of node

The counter does not decrease when node delete

#### Q5

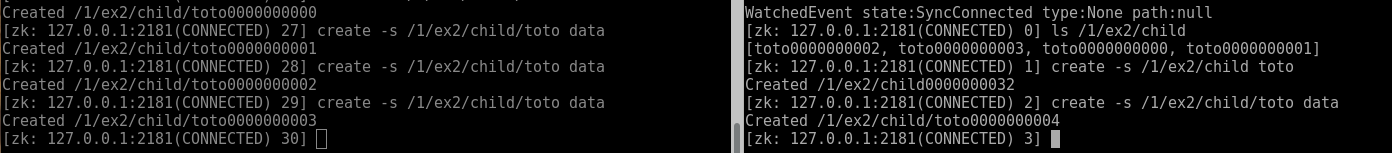
The suffix is 0000000025 (start at 0000000000)

## Scope of sequence numbers



No, they are not related

## Sequence numbers across multiple clients)



Yes are related, the counter remain across the session

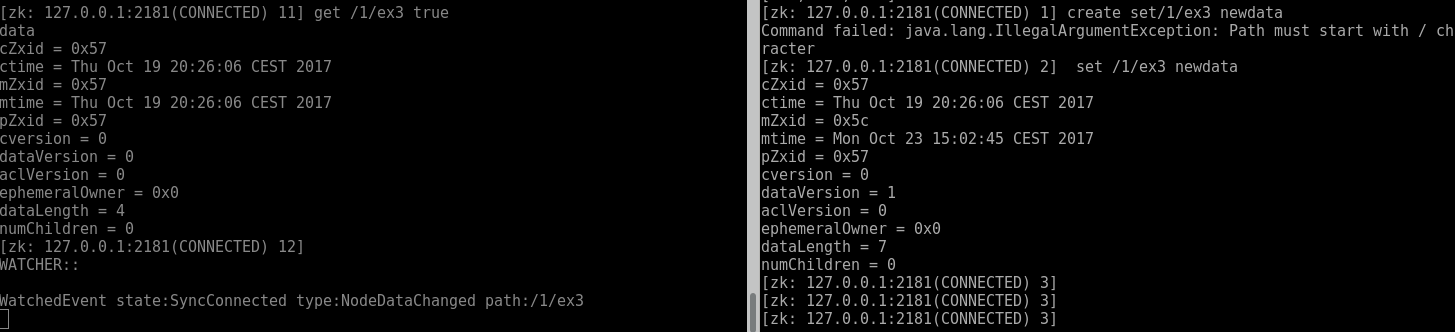
# Watches

## Create the root znode for this exercise with path

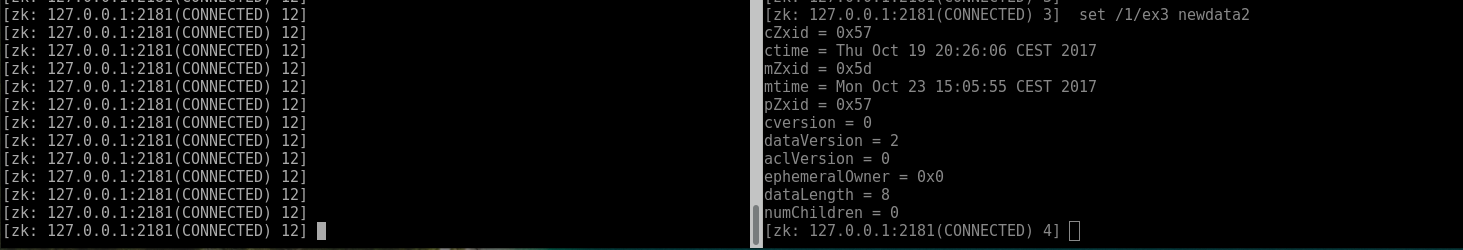


## First watch

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

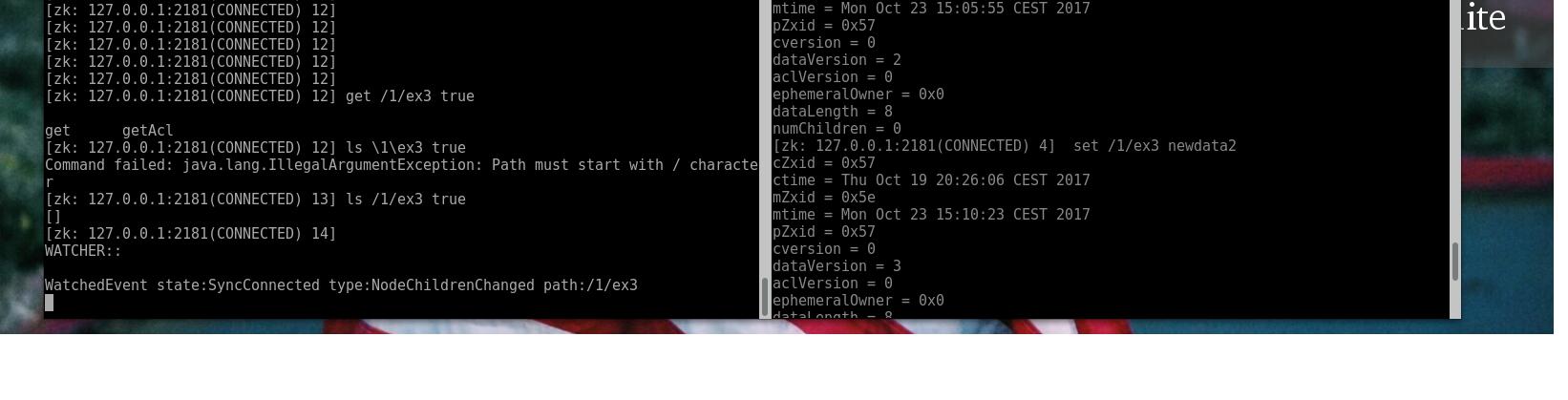


## Durability of watches



Nothing happen, it’s a “one-time” trigger by default. Therefore, each trigger is trigged only once.

## Watch on children



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

## Child watches

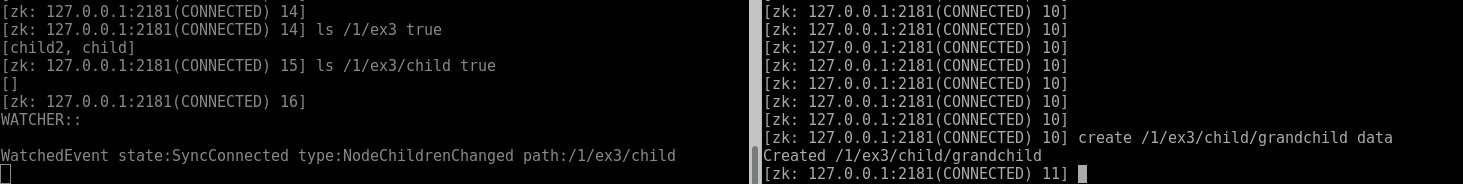


The child creation event is trigger

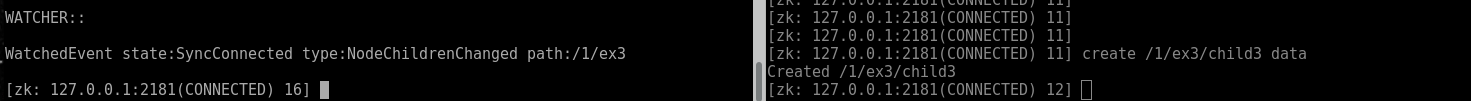
## Summarize watch type

|  |  |  |
| --- | --- | --- |
| command | node | child |
| get | Get trigger node change | Nothing happen |
| ls | Nothing happen | Ls trigger child node change |

## 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.

## Watch Znode deletion



They are all triggered

## More watches

|  |
| --- |
| Good review 😊 |