МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное   
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики  
Кафедра технической кибернетики

**Отчет по лабораторной работе №4**

Дисциплина: «Большие Данные»

## Тема: «Введение в ZooKeeper»

Выполнила: Аксенова Д.А.

Группа: 6133-010402D

Самара 2020

**ЗАДАНИЕ**

1. Запустить ZooKeeper;
2. изучить директорию с установкой ZooKeeper;
3. запустить интерактивную сессию ZooKeeper CLI и освоить её команды;
4. научиться проводить мониторинг ZooKeeper;
5. разработать приложение с барьерной синхронизацией, основанной на ZooKeeper;
6. запустить и проверить работу приложения.

**ХОД РАБОТЫ**

Скачаем и распакуем все необходимые для выполнения лабораторной работы файлы. Запустим сервер, а затем клиент.

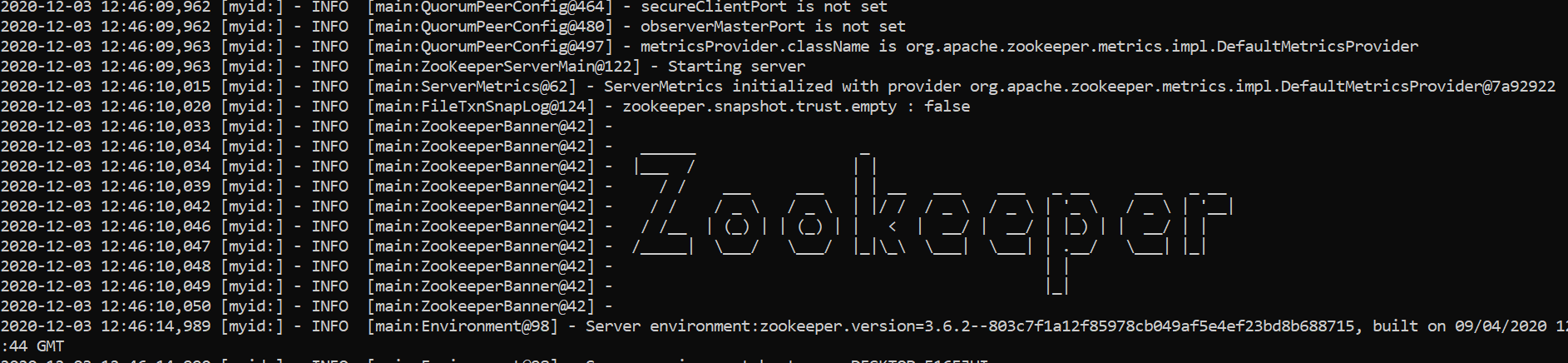


Рисунок 1 – Баннер запущенного сервера Zookeeper

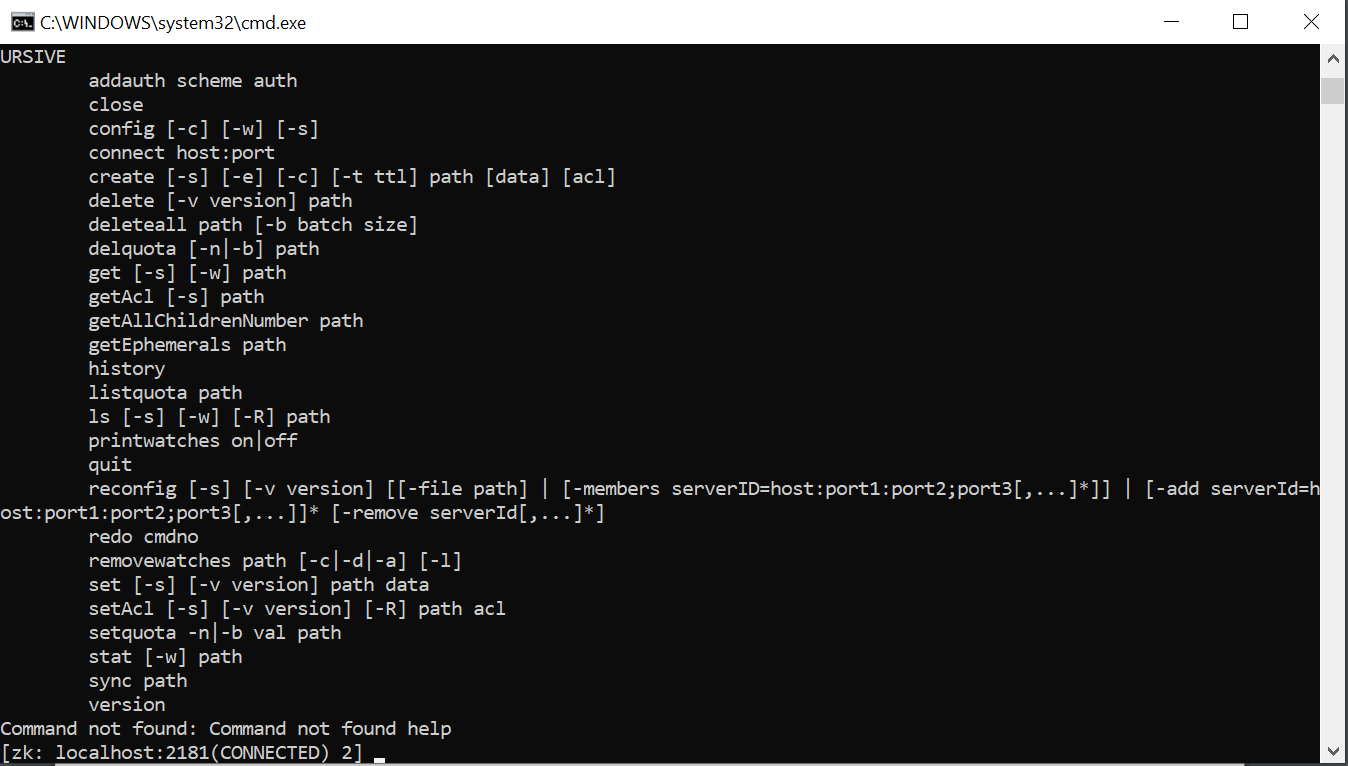


Рисунок 2 – Запущенный клиент Zookeeper

Изучим директорию установки.

Далее изучим возможности CLI интерфейса. При помощи команды ls просмотрим узлы и выведем список дочерних узлов /zookeeper.

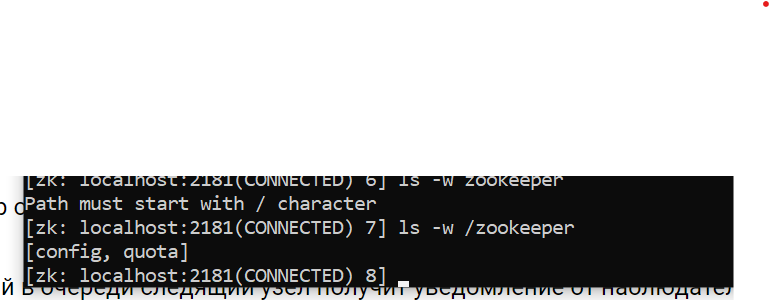


Рисунок 3 – Список дочерних узлов zookeeper

Затем Теперь в корне создайте свой узел /mynode с данными "first\_version" следующей командой: змените данные узла на "second\_version": Теперь создайте два нумерованных (sequential) узла в качестве дочерних mynode

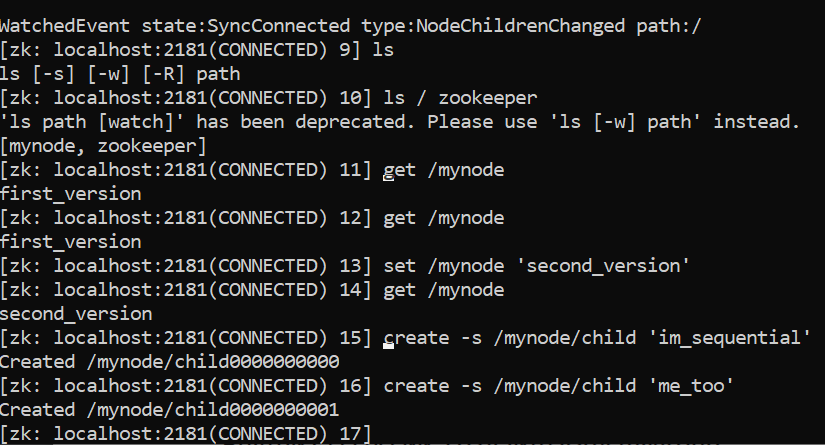


Рисунок 4 – Создание дочерних узлов для mynode

Внутри CLI сессии создадим узел mygroup. Откроем две новых CLI консоли и в каждой создадим по дочернему узлу в mygroup.

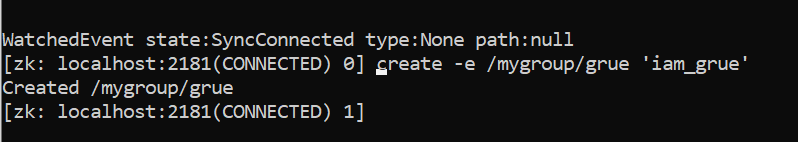


Рисунок 5 – Дочерний узел grue в узле mygroup

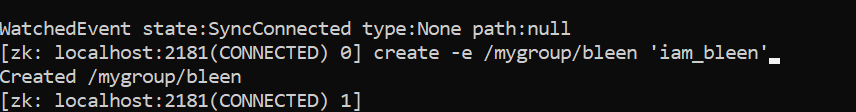


Рисунок 6 – Дочерний узел bleen в узле mygroup

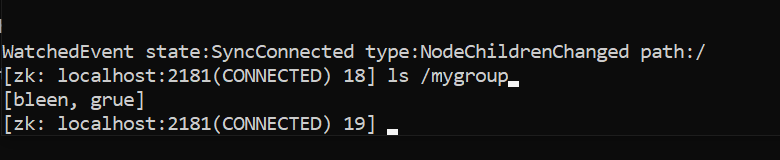


Рисунок 7 – Проверка из исходной консоли

Проверим в исходной консоли, что grue и bleen являются членами группы mygroup. Выберем консоль grue и обратимся к информации узла bleen. Эмулируем аварийное отключение клиента grue. Проверим, что он пропал из mygroup.

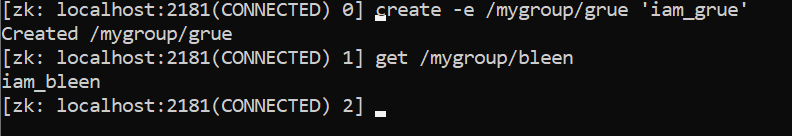


Рисунок 8 – Обращение к узлу bleen из узла grue

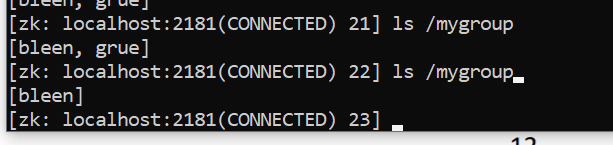


Рисунок 9 – Удаление узла grue с точки зрения основного узла

Таким образом клиенты могут получать информацию о появлении и отключении других клиентов.

Создадим в корне узел "myconfig" в задачу которого будет входить хранение конфигурации. Откроем новую консоль и подключимся к ZooKeeper. Установим watch-триггер.

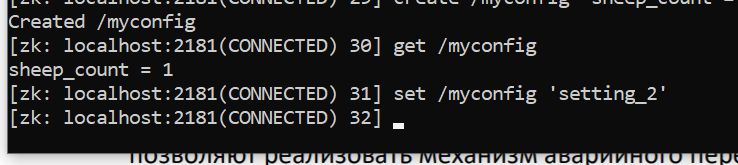


Рисунок 10 – Создание и изменение узла myconfig

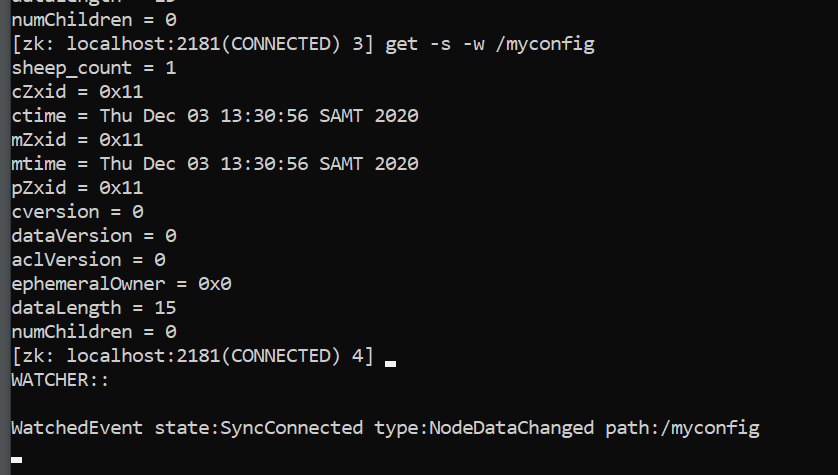


Рисунок 11 – watcher для узла myconfig и реакция на его изменение

Изменим значение myconfig в первом терминале. Во втором терминале появилось оповещение.

Создадим приложение зоопарк, который будет представлен корневым узлом /zoo/ (создан заранее). Каждое животное (приложение) входит в зоопарк, создает дочерний эфемерный узел в зоопарке со своим именем. После того, как все животные будут в сборе, каждое начинает бежать и останавливается через определённый период времени. В конце приложения эфемерные узлы удаляются.

Запустим несколько клиентов распределённого приложения и проверим его работу. Так же проверим случай для одной обезьянки и вечеринки на одного. Работает, даже в случае вечеринки на одного

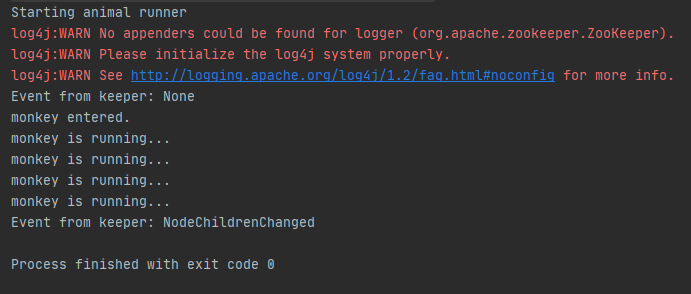


Рисунок 12 – Вечеринка

Решим задачу об обедающих философах. Пять философов располагаются за круглым столом. Философ может размышлять или кушать. Для приема пищи в центре стола большое блюдо с неограниченным количеством спагетти и тарелки, по одной перед каждым философом. Поесть спагетти можно только с использованием двух вилок. Для этого на столе располагается ровно пять вилок – по одной между тарелками философов.

В качестве решения создадим массив вилок-семафоров. Каждый философ сначала берет левую вилку, затем – правую, принимает пище, кладет правую вилку, затем – левую и прекращает прием пищи, начинает размышления. Для того, чтобы избежать ситуации, в которой все философы взяли по левой вилке и ждут (блокировка), для 0 философа сменим порядок, в котором он берет вилки. Теперь нулевой философ начинает прием пищи с поднятия правой вилки.

Результат работы программы:

Philosopher 0 is going to eat

Philosopher 4 is going to eat

Philosopher 1 is going to eat

Philosopher 2 is going to eat

Philosopher 3 is going to eat

Philosopher 2 picked up the left fork

Philosopher 4 picked up the left fork

Philosopher 3 picked up the left fork

Philosopher 1 picked up the left fork

Philosopher 4 picked up the right fork

Philosopher 4 put the right fork

Philosopher 4 put the loft fork and finished eating

Philosopher 3 picked up the right fork

Philosopher 4 is thinking

Philosopher 4 is going to eat

Philosopher 3 put the right fork

Philosopher 4 picked up the left fork

Philosopher 3 put the loft fork and finished eating

Philosopher 2 picked up the right fork

Philosopher 4 picked up the right fork

Philosopher 3 is thinking

Philosopher 2 put the right fork

Philosopher 2 put the loft fork and finished eating

Philosopher 1 picked up the right fork

Philosopher 2 is thinking

Philosopher 3 is going to eat

Philosopher 3 picked up the left fork

Philosopher 2 is going to eat

Philosopher 4 put the right fork

Philosopher 4 put the loft fork and finished eating

Philosopher 3 picked up the right fork

Philosopher 4 is thinking

Philosopher 1 put the right fork

Philosopher 2 picked up the left fork

Philosopher 1 put the loft fork and finished eating

Philosopher 0 picked up the right fork

Philosopher 1 is thinking

Philosopher 0 picked up the left fork

Philosopher 1 is going to eat

Philosopher 3 put the right fork

Philosopher 3 put the loft fork and finished eating

Philosopher 2 picked up the right fork

Philosopher 3 is thinking

Philosopher 0 put the right fork

Philosopher 1 picked up the left fork

Philosopher 0 put the loft fork and finished eating

Philosopher 0 is thinking

Philosopher 2 put the right fork

Philosopher 2 put the loft fork and finished eating

Philosopher 1 picked up the right fork

Philosopher 2 is thinking

Philosopher 0 is going to eat

Philosopher 1 put the right fork

Philosopher 1 put the loft fork and finished eating

Philosopher 0 picked up the right fork

Philosopher 1 is thinking

Philosopher 0 picked up the left fork

Philosopher 0 put the right fork

Philosopher 0 put the loft fork and finished eating

Philosopher 0 is thinking

Process finished with exit code 0

Код решения: <https://github.com/darinaksena/2020.12.03_BD_L4_2>

**ЗАКЛЮЧЕНИЕ**

В результате лабораторной работы был запущен ZooKeeper; была изучена директория с установкой ZooKeeper; была запущена интерактивная сессия ZooKeeper CLI и были освоены её команды; было разработано приложение с барьерной синхронизацией, основанной на ZooKeeper; оно было запущено, его работоспособность была проверена.