Lab2-2 Server-Client

实现思路

两个类 ServerNode、ClientNode

ServerNode

实现服务端，监测客户端即子节点是否活跃

ClientNode

实现客户端，可以有多个实例对应多个客户端

每个客户端都有自己的id，代表是第几个客户端，当客户端运行时，设置节点内容为alive，当一定时间后客户端关闭，设置节点内容为offline，客户端关闭后并不会删除节点，可以等待下次客户端重启。

服务端则监听客户端状态，一旦有客户端启动或关闭，就会打印出当前存活的客户端信息

操作过程

最初添加了三个客户端

```
WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[zk: localhost:2181(CONNECTED) 0] ls /
[election, leader, lxy, registry, servernode, test, zk_test, zookeeper]
[zk: localhost:2181(CONNECTED) 1] ls /servernode
[client-1, client-2, client-3]
[zk: localhost:2181(CONNECTED) 2]
```

打开服务端

最开始没有客户端是活跃的

```
    at org.apache.zookeeper.ClientCnxn$EventThread.proces
    at org.apache.zookeeper.ClientCnxn$EventThread.run(Cl
servernode exists!
[client-1, client-3, client-2]
alive clients:
[]
```

启动客户端1, 2，3

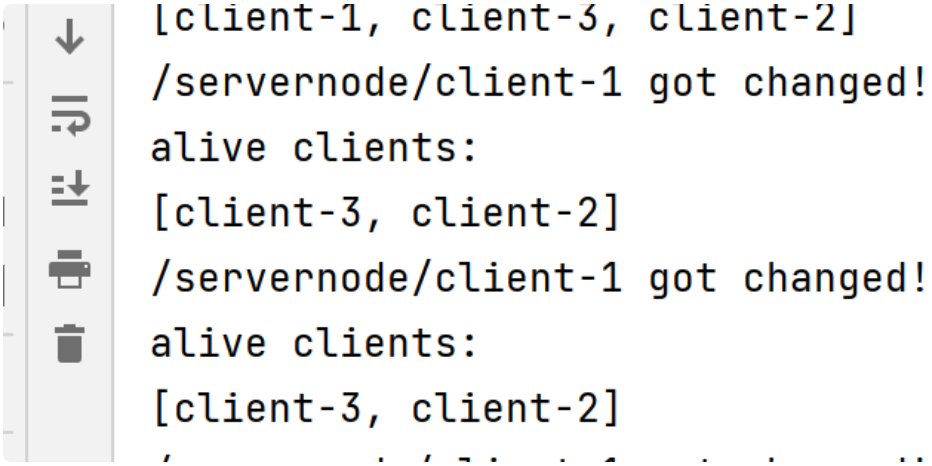
```
/servernode/client-1 got changed!
alive clients:
[client-1, client-3, client-2]
/servernode/client-1 got changed!
alive clients:
[client-1, client-3, client-2]
/servernode/client-1 got changed!
alive clients:
[client-1, client-3, client-2]
```

设置一个定时关闭，和客户端的id有关，若clientId为1，则10s后client–1自动关闭

```
[main-SendThread(localhost:2181)] INFO org.apache.zookeeper.ClientCnx
[main-SendThread(localhost:2181)] INFO org.apache.zookeeper.ClientCnx
SyncConnected
SyncConnected
Closed
client-1 is offline
[main] INFO org.apache.zookeeper.ZooKeeper - Session: 0x10048adf46c0
[main-EventThread] INFO org.apache.zookeeper.ClientCnxn - EventThread

Process finished with exit code 0
```

服务器端变化



最后所有的客户端都关闭

```
alive clients:
[]
/servernode/client-3 got changed!
alive clients:
[]
```

补充说明

这里的实现是无论client是活跃的还是失活的都会保留节点（通过节点内容是alive还是offline区分），当然也可以实现成每有一个客户端是活跃的就新建节点，当客户端失活时，把节点删除。