

华东师范大学数据科学与工程学院实验报告

课程名称： 分布式模型与编程	年级： 2016 级	上机实践成绩：
指导教师： 徐辰	姓名： 张宏伟	
上机实践名称： ZooKeeper 的简单使用	学号： 10165101180	上机实践日期： 20181111
上机实践编号： 08	组号：	上机实践时间：

一、实验目的

熟悉 ZooKeeper 在本地环境下，命令行中的基础操作和在 Java 中的接口；熟悉 ZooKeeper 分布式锁的使用

二、实验任务

安装 ZooKeeper，在命令行下体验基本操作，用 junit

三、使用环境

ZooKeeper3.4.13

四、实验过程

检查 ZooKeeper 是否成功启动

```
hadoop@PC-honwee: /usr/local/zookeeper/conf$ telnet localhost 2181
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
stat
Zookeeper version: 3.4.13-2d71af4dbe22557fda74f9a9b4309b15a7487f03, built on 06/29/2018 04:05 GMT
Clients:
 /127.0.0.1:52682[0](queued=0,recved=1,sent=0)
Latency min/avg/max: 0/13/49
Received: 6
Sent: 5
Connections: 1
Outstanding: 0
Zxid: 0x1
Mode: standalone
Node count: 4
Connection closed by foreign host.
```

- 使用 ls 命令来查看当前 ZooKeeper 中所包含的内容：
 - ls /
- 创建一个新的 znode，使用 create /zk myData。这个点“zk”以及与它关联的字符串：
 - create /zk "myData"
- 我们运行 get 命令来确认 znode 是否包含我们所创建的
 - get /zk

命令行下 get 操作

```
[zk: localhost(CONNECTED) 3] get /zk
myData
cZxid = 0x4
ctime = Sun Nov 11 08:13:46 CST 2018
mZxid = 0x4
mtime = Sun Nov 11 08:13:46 CST 2018
pZxid = 0x4
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 6
numChildren = 0
```

zookeeper命令行操作

- 运行 zkCli.sh -server <ip> 进入

命令行下设置 Watcher 之后使用 set 更改结点存储数据

```
[zk: localhost(CONNECTED) 5] set /zk "zsl"
点"zk"以及与它关联的字符串:
WATCHER::
      ○ create /zk "myData"
WatchedEvent state:SyncConnected type:NodeDataChanged path:/zk
cZxid = 0x4
ctime = Sun Nov 11 08:13:46 CST 2018 认 znode 是否包含我们所创建的字符
mZxid = 0x5
mtime = Sun Nov 11 08:15:52 CST 2018
pZxid = 0x4      ○ get /zk
cversion = 0
dataVersion = 1 • 监听这个节点的变化,当另外一个客户端改变/zk时,它会打出
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 3      ○ get /zk watch
numChildren = 0
```

使用 JUNIT 测试 java 下操作 Zookeepe:

test1()创建/eclipse 结点，绑定字符串"hellozk"

```
@Test
public void test1() throws Exception {
    // 参数1: 要创建的节点的路径 参数2: 节点大数据 参数3: 节点的权限 参数4: 节点的类型
    String nodeCreated = zkClient.create("/eclipse", "hellozk".getBytes(), Ids.OPEN_ACL_UNSAFE, CreateMode.PERSIST
    // 上传的数据可以是任何类型,但都要转成byte[]
}
```

test2()判断结点是否存在

```
@Test
public void test2() throws Exception {
    Stat stat = zkClient.exists("/eclipse", false);
    System.out.println(stat==null?"not exist":"exist");
}
```

test3()打印根目录下的所有子结点

```
4 public void test3() throws Exception {
5     List<String> children = zkClient.getChildren("/", true);
6     for (String child : children) {
7         System.out.println(child);
8     }
9     Thread.sleep(2000);
0 }
```

test4()打印/eclipse 结点所绑定的字符串

```
public void test4() throws Exception {  
    byte[] data = zkClient.getData("/eclipse", false, null);  
    System.out.println(new String(data));  
}
```

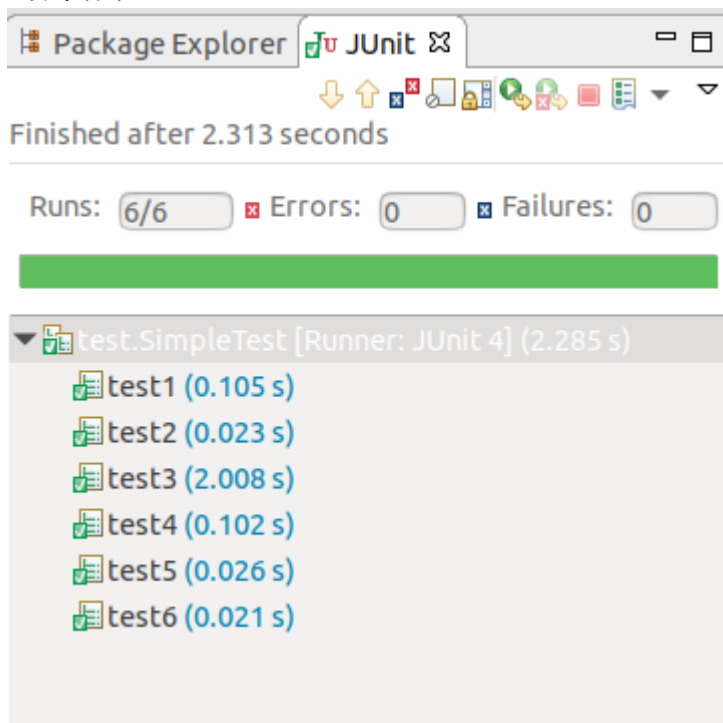
test5()改变/eclipse 结点绑定的字符串并打印

```
public void test5() throws Exception {  
    zkClient.setData("/eclipse", "imissyou angelababy".getBytes(), -1);  
    byte[] data = zkClient.getData("/eclipse", false, null);  
    System.out.println(new String(data));  
}
```

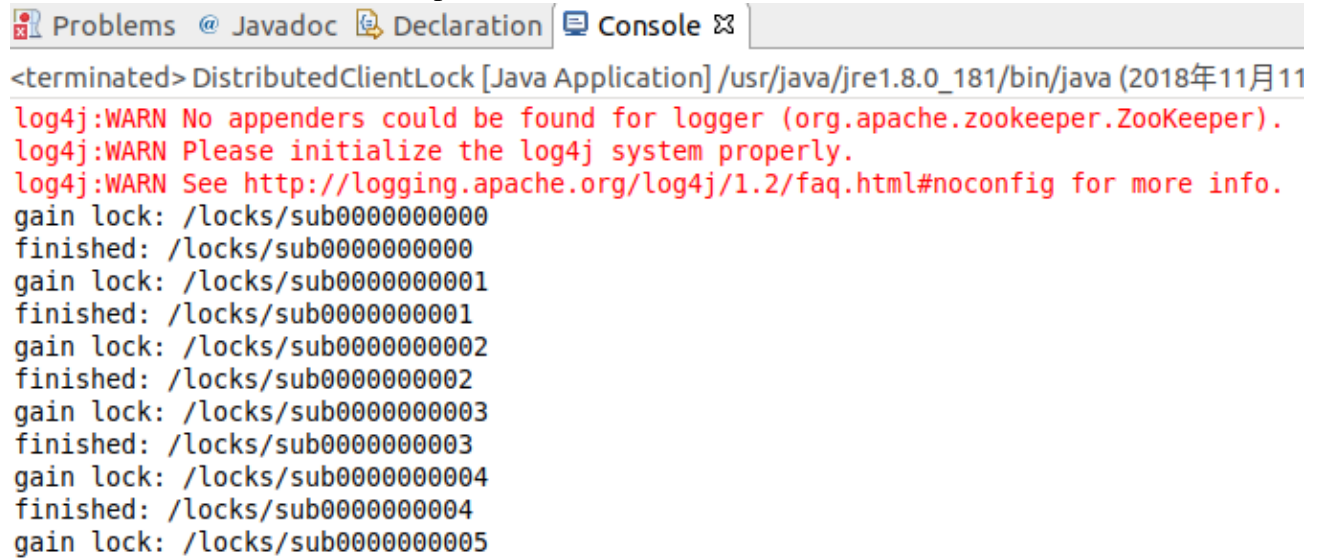
test6()删除/eclipse 结点

```
public void test6() throws Exception {  
    zkClient.delete("/eclipse", -1);  
}
```

测试结果



运行附件的代码，通过 Zookeeper 实现分布式共享锁



```
<terminated> DistributedClientLock [Java Application] /usr/java/jre1.8.0_181/bin/java (2018年11月11
log4j:WARN No appenders could be found for logger (org.apache.zookeeper.ZooKeeper).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
gain lock: /locks/sub0000000000
finished: /locks/sub0000000000
gain lock: /locks/sub0000000001
finished: /locks/sub0000000001
gain lock: /locks/sub0000000002
finished: /locks/sub0000000002
gain lock: /locks/sub0000000003
finished: /locks/sub0000000003
gain lock: /locks/sub0000000004
finished: /locks/sub0000000004
gain lock: /locks/sub0000000005
```

五、总结

用类似文件系统的简单结点树模型就可以解决几个分布式中的问题，略神奇。