

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：吴双	
上机实践名称： Zookeeper 的简单使用	学号： 10164102141	上机实践日期：
上机实践编号： #8	组号： 23	上机实践时间：

一、实验目的

使用 Zookeeper 在本地环境下的基础操作，熟悉分享锁的使用。

二、实验任务

安装 Zookeeper，使用 Zookeeper，使用 Junit。

三、使用环境

Ubuntu LTS 18.04
Hadoop 2.7.3
Zookeeper-3.4.13
Eclipse IDE for Java Developers 2018-09(4.9.0)

四、实验过程

1. 本地部署 Zookeeper，并测试：

a) 启动 Zookeeper

```
$ zkServer.sh start
```

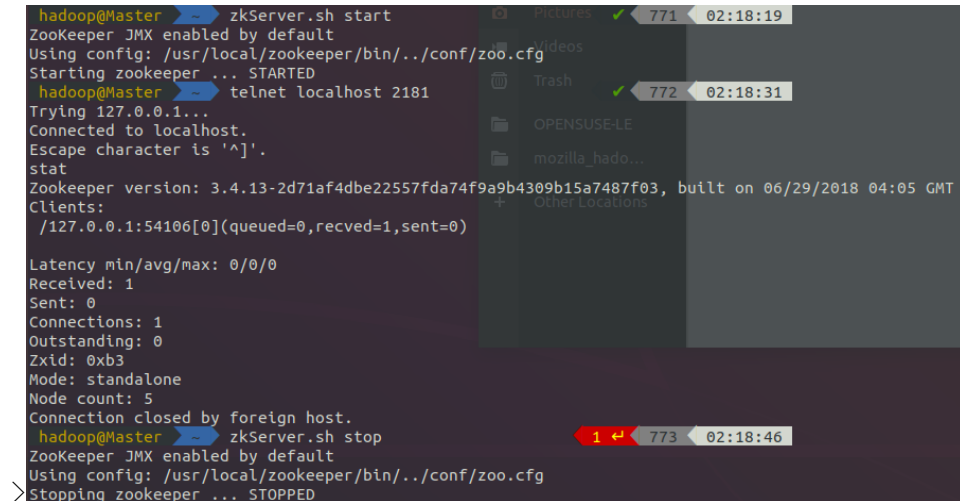
b) 验证 Zookeeper 是否启动.

```
$ telnet localhost 2181
```

```
$ stat
```

2. 关闭 Zookeeper

```
$ zkServer.sh stop
```



```
hadoop@Master ~$ zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
hadoop@Master ~$ 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:
 /127.0.0.1:54106[0](queued=0,recved=1,sent=0)

Latency min/avg/max: 0/0/0
Received: 1
Sent: 0
Connections: 1
Outstanding: 0
Zxid: 0xb3
Mode: standalone
Node count: 5
Connection closed by foreign host.
hadoop@Master ~$ zkServer.sh stop
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg
Stopping zookeeper ... STOPPED
```

3. 使用 Junit 测试 Java 下操作 Zookeeper:

a) 相关代码补充

i. 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[]
}
}
```

ii. 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");
}
>
```

iii. 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 }
```

iv. 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));
> }
```

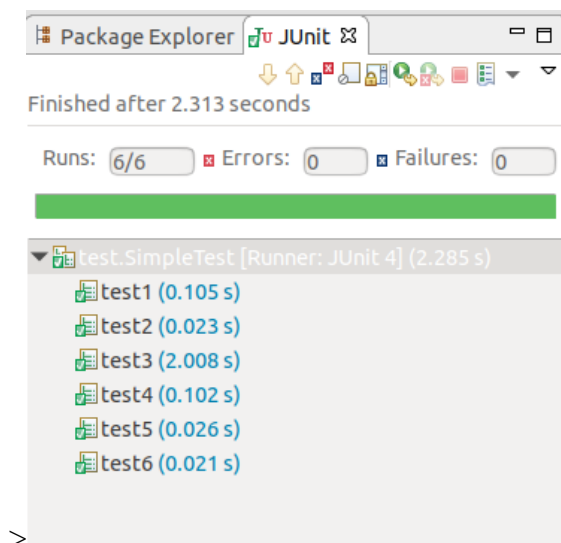
v. test5()改变/eclipse 结点绑定的字符串并打印

```
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));
> }
```

vi. test6()删除/eclipse 结点

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

b) 测试结果



4. 运行附件的代码，通过 Zookeeper 实现分布式共享锁：

```
Problems @ Javadoc Declaration Console
<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

Zookeeper 在托管方面还是很有效的，比较好用。