# 监听服务器节点动态上下线案例

## 需求

某分布式系统中，主节点可以有多台，可以动态上下线，任意一台客户端都能实时感知到主节点服务器的上下线。

## 需求分析

2．需求分析，如图5-12所示



图5-12 服务器动态上下线

## 具体实现

### 创建server节点

（0）先在集群上创建/servers节点

[zk: localhost:2181(CONNECTED) 10] create /servers "servers"

Created /servers

### 服务器端向Zookeeper注册代码

|  |
| --- |
| package com.atguigu.zkcase;  import java.io.IOException;  import org.apache.zookeeper.CreateMode;  import org.apache.zookeeper.WatchedEvent;  import org.apache.zookeeper.Watcher;  import org.apache.zookeeper.ZooKeeper;  import org.apache.zookeeper.ZooDefs.Ids;  public class DistributeServer {  private static String connectString = "hadoop137:2181,hadoop138:2181,hadoop139:2181";  private static int sessionTimeout = 2000;  private ZooKeeper zk = null;  private String parentNode = "/servers";    // 创建到zk的客户端连接  public void getConnect() throws IOException{    zk = new ZooKeeper(connectString, sessionTimeout, new Watcher() {  @Override  public void process(WatchedEvent event) {  }  });  }    // 注册服务器  public void registServer(String hostname) throws Exception{  String create = zk.create(parentNode + "/server", hostname.getBytes(), Ids.OPEN\_ACL\_UNSAFE, CreateMode.EPHEMERAL\_SEQUENTIAL);    System.out.println(hostname +" is online "+ create);  }    // 业务功能  public void business(String hostname) throws Exception{  System.out.println(hostname+" is working ...");    Thread.sleep(Long.MAX\_VALUE);  }    public static void main(String[] args) throws Exception {    // 1获取zk连接  DistributeServer server = new DistributeServer();  server.getConnect();    // 2 利用zk连接注册服务器信息  server.registServer(args[0]);    // 3 启动业务功能  server.business(args[0]);  }  } |

### 客户端代码

|  |
| --- |
| package com.atguigu.zkcase;  import java.io.IOException;  import java.util.ArrayList;  import java.util.List;  import org.apache.zookeeper.WatchedEvent;  import org.apache.zookeeper.Watcher;  import org.apache.zookeeper.ZooKeeper;  public class DistributeClient {  private static String connectString = "hadoop137:2181,hadoop138:2181,hadoop139:2181";  private static int sessionTimeout = 2000;  private ZooKeeper zk = null;  private String parentNode = "/servers";  // 创建到zk的客户端连接  public void getConnect() throws IOException {  zk = new ZooKeeper(connectString, sessionTimeout, new Watcher() {  @Override  public void process(WatchedEvent event) {  // 再次启动监听  try {  getServerList();  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  // 获取服务器列表信息  public void getServerList() throws Exception {    // 1获取服务器子节点信息，并且对父节点进行监听  List<String> children = zk.getChildren(parentNode, true);  // 2存储服务器信息列表  ArrayList<String> servers = new ArrayList<>();    // 3遍历所有节点，获取节点中的主机名称信息  for (String child : children) {  byte[] data = zk.getData(parentNode + "/" + child, false, null);  servers.add(new String(data));  }  // 4打印服务器列表信息  System.out.println(servers);  }  // 业务功能  public void business() throws Exception{  System.out.println("client is working ...");  Thread.sleep(Long.MAX\_VALUE);  }  public static void main(String[] args) throws Exception {  // 1获取zk连接  DistributeClient client = new DistributeClient();  client.getConnect();  // 2获取servers的子节点信息，从中获取服务器信息列表  client.getServerList();  // 3业务进程启动  client.business();  }  } |