

## DAY 4——线程池

问题 4：如何在客户线程过多时实现排队？

### 4.1 学习内容

掌握线程池的使用，请学习教程的第 3 章 线程池。

### 4.2 任务清单

- ① 运行并阅读程序，描绘出主要函数的步骤和程序结构。
- ② 学习和运用线程池技术，改写一客户一线程的服务器程序。
- ③ 理解程序内部的通信协议。
- ④ 在小组内，抽签阐述服务器或客户端的软件结构，并录制视频。

### 4.3 任务分解

- ① 线程池功能测试。将线程最大值设为 2，启动 3 台客户程序，观察现象。
- ② 特别注意并记录程序的善后工作步骤。

### 4.4 关键技术

#### 4.4.1 获取当前线程号

```
long currentId=Thread.currentThread().getId();
```

## 4.4.2 线程池

### 1、声明线程池

```
private ExecutorService fixedPool; //线程池
```

### 2、创建线程池

```
int processors=Runtime.getRuntime().availableProcessors();//获取 CPU 数量  
fixedPool=Executors.newFixedThreadPool(processors*2); //创建固定大小线程池
```

### 3、将新建的线程加入线程池

```
Thread clientThread=new ClientThread(toClientSocket,clientCounts); //创建客  
户线程
```

```
fixedPool.execute(clientThread); //用线程池调度客户线程运行
```

## 4.5 问题讨论

- ① 如果是更加复杂的内部协议,且动作顺序不确定,如何解决?
- ② 关于线程池的问题讨论:

<http://www.ty2y.com/study/wsmjlbysyexecutorscjxcc.html>

