



链滴

多线程内存泄漏

作者: [huihui](#)

原文链接: <https://ld246.com/article/1488529823811>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

事实证明只要在静态函数中没有处理多线程共享数据，就不存在着多线程访问同一个静态方法会出现源冲突的问题，如果在静态方法中有访问多线程共享数据，就要小心多线程同步问题。

同时，在使用线程池进行开发时，要特别注意线程实例的资源释放问题，不要想当然的认为没有创建量名来引用所创建的线程实例，线程在运行完就会自动释放资源，事实证明并没有释放。如果没有显创建变量名引用线程实例，系统会自动创建一个隐藏变量来应用这个实例，会造成线程池运行完线程依然没有释放资源的问题。

下面是我进行内存泄漏分析后的结果。

项目中使用了EDU.oswego.cs.dl.util.concurrent的线程池进行开发，使用线程池运行线程 ProcessThread 。

```
public class ProcessThread implements Runnable{ 略}
```

如果就这样使用线程池运行线程会造成严重的内存泄漏问题：

```
ThreadPool.getInstance().execute( new ProcessThread(messages[i]) );
```

代码更改后（如下所示），暂时就没有发现内存泄漏了

```
ProcessThread thread = new ProcessThread(messages[i]);
```

```
ThreadPool.getInstance().execute( thread );
```

```
thread = null;
```