



链滴

# 游戏架构思考：如何优雅的关闭游戏服务器

作者：simon

原文链接：<https://ld246.com/article/1541409375188>

来源网站：[链滴](#)

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



## 1、优雅的姿态是什么样

当我们触发关闭的操作时，程序接收到关闭进程的指令，程序在关闭之前需要完成以下步骤：

- 停止接收客户端的连接
- 执行线程池中未执行完毕的任务
- 持久化缓存中的数据
- 清理临时的文件等

这就是我们对于优雅安全的关闭游戏服的目标，按照以上顺序关闭服务器不会造成数据的丢失。

## 2、如何处理

JVM提供了关闭钩子用来满足以上的需求。关闭钩子实际上是一种hook线程，用来监听JVM的关闭。当然，该hook线程只有在JVM正常关闭的情况下才会触发相关操作，强制关闭下无法执行hook操作。主要的触发情形有以下几种：

- 程序正常退出
- 调用system.exit()方法
- 使用ctrl+c命令中断
- 系统关闭
- OutOfMemory引起的程序退出
- 使用kill pid 命令中断

注意：kill -9 pid命令无法触发shutdown hook执行，该命令属于强制关闭

### 3、如何使用钩子

简单使用实例：

```
public class Test {  
  
    public static void main(String[] args) {  
  
        System.out.println("main exec...");  
  
        Runtime.getRuntime().addShutdownHook(new Thread(new Runnable() {  
  
            @Override  
            public void run() {  
                System.out.println("监听到JVM关闭操作，执行hook的相关操作...");  
            }  
        }));  
  
    }  
}
```

对于一个完整的项目来说，关闭时往往需要注册多个关闭钩子进行处理，这些钩子会并发执行操作，JVM并不能控制其执行的顺序，当钩子执行时JVM延迟关闭，所有的钩子操作执行完毕后JVM才会退出。所以我们应当尽量减少hook操作的时间。另外钩子并发执行的特点，很容易出现死锁和竞态条件等问题，原则上我们应当在一个钩子中执行一系列的相关操作。

以下还有一些需要注意的地方：

- hook线程中抛出的异常需要进行处理，否则会中断执行
- hook执行逻辑中不能调用system.exit()方法，否则会造成死循环
- system.exit()方法后注册的钩子无法执行
- 不能再hook中进行hook的注册和移除，这会引发非法操作异常
- 当JVM收到SIGTERM指令时，hook操作在一定的时间内没有处理完成，则会被终止