

计算机二级辅导:正确地killjava进程计算机等级考试 PDF转换  
可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/645/2021\\_2022\\_\\_E8\\_AE\\_A1\\_](https://www.100test.com/kao_ti2020/645/2021_2022__E8_AE_A1_)

[E7\\_AE\\_97\\_E6\\_9C\\_BA\\_E4\\_c97\\_645438.htm](https://www.100test.com/kao_ti2020/645/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_645438.htm) 在linux/unix下，你会怎么中止一个java应用或进程？多数人可能会回答 kill -9 pid，这是一种在多数情况下正确的做法。不过本文打算阐述使用kill -9带来的一些问题，并给出另一种标准的kill方式。标准中断信号在Linux信号机制中，存在多种进程中中断信号

(Linux信号列表)。其中比较典型的有 SIGKILL (9) 和 SIGTERM (15)。SIGKILL (9) 和 SIGTERM (15) 的区别在于：SIGKILL (9) 的效果是立即杀死进程。该信号不能被阻塞, 处理和忽略。SIGTERM (15) 的效果是正常退出进程，退出前可以被阻塞或回调处理。并且它是Linux缺省的程序中中断信号。由此可见，SIGTERM (15) 才是理论上标准的kill进程信号。那使用 SIGKILL (9) 又有什么错呢？

SIGKILL (9) 带来的问题 先看一段程序 Java代码 /\*\* \*

```
Shutdown Hook Presentation */ public class ShutdownHookTest {  
private static final void shutdownCallback() {
```

```
System.out.println("Shutdown callback is invoked."); } public static
```

```
void main(String[] args) throws InterruptedException {
```

```
Runtime.getRuntime().addShutdownHook(new Thread() {
```

```
@Override public void run() { shutdownCallback(); } });
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问

[www.100test.com](http://www.100test.com)