链滴

程序 kill -9 与 kill -15 的区别，以及回调函数的作用

作者：mingan

原文链接：https://ld246.com/article/1520244253951

来源网站：链滴

在Linux/unix下，中止一个Java

进程有两种方式，一种是kill -9 pid，一种是kill -15 pill（默认）。

两种方式的区别是：

SIGNKILL（9） 的效果是立即杀死进程. 该信号不能被阻塞, 处理和忽略。

SIGNTERM（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缺省程序中断信号。

标准中断信号

在Linux信号机制中，存在多种进程中断信号（Linux信号列表）。其中比较典型的有 SIGNKILL（9）和 SIGNTERM（15）.

由此可见，SIGNTERM（15） 才是理论上标准的kill进程信号。

那使用 SIGNKILL（9） 又有什么错呢?

SIGNKILL（9） 带来的问题

看下面一段程序：

```
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();

}

});

Thread.sleep(10000);

}

}
```

在上面这段程序中，我使用Runtime为当前java进程添加了一个ShutdownHook，它的作用是在java常退出时，执行shutdownCallback()这个回调方法。

此时，如果你试验过在java进程未自动退出前，执行 kill -9 pid，即发送 SIGNKILL 信号，会发现这回调接口是不会被执行的。这是SIGNKILL信号起的作用。**

对于我这个简单的测试用例来说，不被执行也无大碍。但是，如果你的真实系统中有需要在java进程出后，释放某些资源。

而这个释放动作，因为SIGNKILL被忽略了，那就可能造成一些问题。

所以，推荐大家使用标准的kill进程方式，即 kill -15 pid。

---》》此外，如果想在程序被kill之前执行某些操作，就可以用到这个例子中的Runtime.getRuntime(
.addShutdownHook函数了，它是一个回调函数，在被kill之前会执行这个函数里面的内容。

本文转载自http://blog.csdn.net/ungoneless/article/details/53191719