



链滴

# Java 线程的优先级

作者: [guihuo](#)

原文链接: <https://ld246.com/article/1543397741650>

来源网站: [链滴](#)

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



文章摘抄自《图解多线程设计模型》

## Java的优先级只能在特定的JAVA平台运行环境中起作用

Java规范中几乎没有规定任何通用于所有Java平台运行环境的优先级。因此,优先级只在特定的Java平台运行环境(Java VM的实现和版本,以及操作系统)中起作用。

Java规范中并没有写明优先级的值是否会对线程的运行产生影响,以及如果会产生影响,具体是什么样影响。虽然其中写着“优先级高的线程会优先执行”、但是因为并没有写明这里的“优先”到底是什么意思,所以没有太大意义。

在某个Java平台运行环境中,高优先级的线程可能会比低优先级的线程得到更长的CPU时间但是在其他Java平台运行环境中,最高优先级的线程则可能会使用全部CPU时间。这时,只要高

优先级的线程没有被阻塞,低优先级的线程就无法执行。此外,还有可能在另外一个Java平台运行环境中,不管线程是什么优先级,都不会对线程的执行产生任何影响。

可能会有读者认为“不管怎样,将想要优先执行的线程的优先级设置得高一点总没有坏处吧”。这其实错误的。因为在有些Java平台运行环境中,只要线程的优先级稍微高一点,就有可能一直不停地执行下去。这种情况下,如果提高某个线程的优先级,就可能会导致只有该线程被执行,其他线程则全部被停止。也就是说,程序会变为类似单线程执行的状态。

如果无论如何都必须使用优先级,需要先确定Java平台运行环境,然后调查该运行环境是如何处理优先级的。但是,这种做法无法确保这段程序在其他运行环境中执行时与在当前运行环境中执行时程序行为是相同的。

## 表示优先级的静态字段

虽然Java定义了这些值,但这些值并不一定直接表示操作系统的优先级。

- Thread.MAX\_PRIORITY 表示最高的优先级

- Thread.NORM\_PRIORITY 表示默认的优先级
- Thread.MIN\_PRIORITY 表示最低的优先级

## API

### **setPriority(Int)**

用于设置优先级

### **getPriority()**

用于获取优先级