翻译人员: 铁锚

翻译日期: 2013年11月18日

原文链接: Why string is immutable in Java ?

这是一个老生常谈的话题(This is an old yet still popular question). 在Java中将String设计成不可变的是综合考虑到各种因素的结果,想要理解这个问题,需要综合内存,同步,数据结构以及安全等方面的考虑. 在下文中,我将为各种原因做一个小结。

**1. 字符串常量池的需要**

字符串常量池(String pool, String intern pool, String保留池) 是Java堆内存中一个特殊的存储区域, 当创建一个String对象时,假如此字符串值已经存在于常量池中,则不会创建一个新的对象,而是引用已经存在的对象。

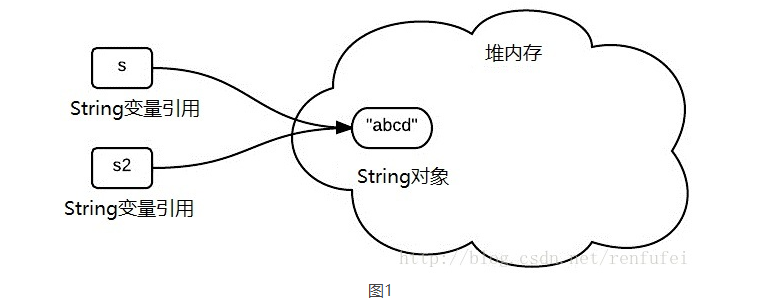
如下面的代码所示,将会在堆内存中只创建一个实际String对象.

[java] view plain copy

String s1 = "abcd";

String s2 = "abcd";

示意图如下所示:



假若字符串对象允许改变,那么将会导致各种逻辑错误,比如改变一个对象会影响到另一个独立对象. 严格来说，这种常量池的思想,是一种优化手段.

请思考: 假若代码如下所示，s1和s2还会指向同一个实际的String对象吗?

[javascript] view plain copy

String s1= "ab" + "cd";

String s2= "abc" + "d";

也许这个问题违反新手的直觉, 但是考虑到现代编译器会进行常规的优化, 所以他们都会指向常量池中的同一个对象. 或者,你可以用 jd-gui 之类的工具查看一下编译后的class文件.

**2. 允许String对象缓存HashCode**

Java中String对象的哈希码被频繁地使用, 比如在hashMap 等容器中。

字符串不变性保证了hash码的唯一性,因此可以放心地进行缓存.这也是一种性能优化手段,意味着不必每次都去计算新的哈希码. 在String类的定义中有如下代码:

[javascript] view plain copy

private int hash;//用来缓存HashCode

**3. 安全性**

String被许多的Java类(库)用来当做参数,例如 网络连接地址URL,文件路径path,还有反射机制所需要的String参数等, 假若String不是固定不变的,将会引起各种安全隐患。

假如有如下的代码:

[javascript] view plain copy

boolean connect(string s){

if (!isSecure(s)) {

throw new SecurityException();

}

// 如果在其他地方可以修改String,那么此处就会引起各种预料不到的问题/错误

causeProblem(s);

}

总体来说, String不可变的原因包括 设计考虑,效率优化问题,以及安全性这三大方面. 事实上,这也是Java面试中的许多 "为什么" 的答案。

相关文章 :

String对象不可改变的特性

String is passed by “reference” in Java

十大常见Java String问题

Java中Set的contains()方法