链滴

在 Lua 中实现 StringBuffer

作者：zaccn
原文链接：https://ld246.com/article/1531101232639
来源网站：链滴

在 Lua 中，字符串是一个常量，如果用字符串连接符“..”把 2 个字符串连接起来，例如 first_str = first_str .. second_str，那么原来的 first_str 和 second_str 就会作为垃圾等待回收，first_str 引用是一个新的字符串，如果在程序里面有大量的字符串连接操作的话，性能会十分低下。Lua 是一个很洁的语言，他没有 StringBuffer 的实现，但是其实我们可以动手写一个简单的 StringBuffer 实现，避免性能的问题。
首先定义一个叫 StringBuffer 的 table，使得这个 StringBuffer 被调用的时候看起来像是面向象的样子：）
然后分别定义两个方法 append 和 tostr，实现的原理就是：append 用 table 来保存所有字符串，to tr 把保存了字符串的 table 用 concat 转成真正的字符串。
StringBuffer = {}
StringBuffer.append = function(t, str)
if t and str then
table.insert(t, str)
end
end
StringBuffer.tostr = function(t)
if t then
return table.concat(t)
end
end
StringBuffer.new = function() return {} end
调用的时候大概如下，摘录了一段代码。。。
all_assets = StringBuffer.new()
for asset in ctx:allassets() do
StringBuffer.append(all_assets, asset:id())
StringBuffer.append(all_assets, ', ')
end
result = StringBuffer.tostr(all_assets)
print (result)
在 Lua 中实现这样的一个 StringBuffer，既可以避免潜在的性能问题，又可以使得代码看起来加易懂~好了，重构以前的代码去了。。。