云风的 BLOG

思绪来得快去得也快，偶尔会在这里停留

« Lua 5.3 升级注意 | 返回首页 | 在线调试 Lua 代码 »

怎样在运行时插入运行一段 Lua 代码

最近想给 skynet 加一个在线调试器，方便调试 Lua 编写的服务。

Lua 本身没有提供现成的调试器，但有功能完备的 debug api 。通常、我们可以在代码中插入 debug.debug() 就可以进入一个交互环境，输入任何 Lua 指令。当然，你也可以在 debug hook 里调用它。

但这种交互方式有一个缺点：lua 直接用 load 翻译输入的文本，转译为一个 lua 函数并运行。这注定了这个输入的代码中不能直接访问到上下文的局部变量和 upvalue 。

如果想读写上下文中的局部变量或 upvalue ，还得使用 debug.getlocal 等函数。这无疑是相当麻烦的。

有没有办法实现一个增强版的 dostring ，让运行的代码拥有和调用者相同的上下文呢？

可以，但需要一点技巧。

我们不要直接加载要运行的代码，而是给它构造一个类似的环境。比如当前有两个 local 变量 a 和 b 的话，就在代码字符串前加上 local a,b 。然后在运行它之前，把值写进入。

既然我们知道注入了哪些变量，就可以在运行完毕后，读出这些变量再设回当前环境中即可。

对于当前的 upvalue ，要更容易一些。

因为，Lua 5.2 之后提供了 debug.upvaluejoin 可以把当前的 upvalue 关联到你要运行的函数上，这样连事后更新都省了。

处理 ... 这种可变参数要麻烦一些。你需要自己小心的一个个读出来，再传给你要插入的函数。

这套方案说起来简单，实现起来还是比较绕的。ok ，我实现了一份供参考。使用这个版本的 run ，可以运行一个字符串，它拥有和调用者完全相同的环境，就好像代码被嵌在当前位置一样。注意，如果当前环境没有引用某个 upvalue ，即使它可见，你插入的代码也不可能看见。如果想获取它，可以传递恰当的 level ，切到合适的层次上就可以访问了。

有了这个函数，我们可以这样使用：

```
local uv = 2

function f(...)
    local a,b = 1,uv
    print("_ENV ===>", _ENV)
    print("a,b ====>", a,b)
    run[[
        print "=== inject code ==="
        print("\t_ENV ===>", _ENV)
        print("\ta,b,uv ===>", a,b,uv)
        print("\t... ", ...)
        a,b = b,a
        uv = 3
        print "==== level ==="
    ]]
    print("a,b ====>", a,b)
end

f("Hello","world")

print("uv=",uv)
```

运行它可以得到这样的输出：

```
_ENV ===>       table: 0000000000266de0
a,b ====>       1       2
=== inject code ===
_ENV ===>       table: 0000000000266de0
a,b,uv ===>     1       2       2
...     Hello   world
==== level ===
a,b ====>       2       1
uv=     3
```

你可以看到，在 f 函数中插入运行了一段代码，它可以访问到 f 函数可见的 a,b 两个 local 变量， 以及 f 引用的 upvalue uv 。并可以改写它们。可变参数也可以用 ... 正确访问到。

---

run 的实现我放在 gist 上了。

如果在 debug hook 里使用 run ，就需要调用时传入 level （通常是 1 ）。这份实现性能并不高（如果需要高性能，可以用 C 重新实现一遍），推荐用在交互调试器上，而不要用于生产代码中。

云风 提交于 February 10, 2015 03:23 PM | 固定链接

COMMENTS

干得漂亮

Posted by: Zero | (1) February 14, 2015 11:50 AM

POST A COMMENT

非这个主题相关的留言请到：留言本

名字：

Email 地址：

为了验证您是人类，请将六加一的结果（阿拉伯数字七）填写在下面：

URL:

记住我的信息?

留言：
（不欢迎在留言中粘贴程序代码）

提交