[lua-5.1 和 lua-5.2 之间的一些差异。](http://blog.csdn.net/hacking_in_my_mind/article/details/7670533)

标签： [table](http://www.csdn.net/tag/table)[lua](http://www.csdn.net/tag/lua)[null](http://www.csdn.net/tag/null)

2012-06-17 11:39 17384人阅读 [评论](http://blog.csdn.net/hacking_in_my_mind/article/details/7670533#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/hacking_in_my_mind/article/details/7670533#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

Lua（3） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录[(?)[+]](http://blog.csdn.net/hacking_in_my_mind/article/details/7670533)

**1. luaL\_register 2 luaL\_setfuncs**

lua5.1 之中有一个luaL\_register,用于把luaL\_Reg 数组中的所有函数注册到lua中。但在lua5.2中已经不支持这一函数了，

lua5.2的手册中建议使用luaL\_setfuncs来替代luaL\_register.

使用如下。

**[cpp]** [view plain](http://blog.csdn.net/hacking_in_my_mind/article/details/7670533) [copy](http://blog.csdn.net/hacking_in_my_mind/article/details/7670533)

[print?](http://blog.csdn.net/hacking_in_my_mind/article/details/7670533)

1. **int** luaopen\_libname(lua\_State \*L)
2. {
3. **static** **const** luaL\_Reg funcs[] = {
4. {"func1", func1},
5. {"func2", func2},
6. {NULL, NULL}
7. };
8. #ifdef lua5.1
9. //5.1 下直接使用luaL\_register 就好
10. luaL\_register(L, "libname", funcs);
11. #else //lua5.2
12. lua\_newtable(L);
13. //先把一个table压入VS，然后在调用luaL\_setfuncs就会把所以的func存到table中
14. //注意不像luaL\_register这个table是个无名table，可以在的使用只用一个变量来存入这个table。
15. //e.g local clib = require "libname". 这样就不会污染全局环境。比luaL\_register更好。
16. luaL\_setfuncs(L, funcs, 0);
17. #endif
18. **return** 1;  //返回table
20. }

luaL\_register, 和 luaL\_setfunc 实现很想，这不过luaL\_register会先创建一个全局table在把函数注册到这个table中。

具体实现请参考lua源马。

**2. LUA\_GLOBALINDEX 2 registry LUA\_RIDX\_GLOBAS**

在5.1中经常会把一些C函数注册在LUA\_GLOBALINDEX 全局table中。

e.g.

lua\_pushstring(L, name)

lua\_pushcfunction(L, func)

lua\_settable(L, LUA\_GLOBALSINDEX);

在5.2中已经移除了LUA\_GLOBALSINDEX,去而带之的是注册表。

在5.1中lua\_setglobal() 和 lua\_getgloba() 都用是LUA\_GLOBALINDEX 伪索引。

在5.2中上面两个函数都是使用的注册标中的LUA\_RIDX\_GLOBAS伪索引（索引注册表的全局环境）

LUA\_RIDX\_GLOBAS是LUA state 注册表中与定义的两个值之一。另一个是

LUA\_RIDX\_MAINTHREAD 索引的是lua state的主线程状态。

**3. luaL\_getn 2 lua\_objlen 2 lua\_rawlen**

5.0->5.1->5.2

都是得到一个table的大小和#table的值相等，或一个userdata （lua 为指定的userdata分配的原始内存大小，或一个string的长度。

都与其他类型，返回0