

[!\[\]\(919a2cb85b99741a73c0c31a427236a8_img.jpg\) \(http://twitter.com/kenlistdev\)](http://twitter.com/kenlistdev)
[!\[\]\(666e09182d4cd268646ea700ea60dcdf_img.jpg\) \(http://github.com/kenlist\)](http://github.com/kenlist)
 (http://geekluo.com/)

 (mailto:rijian.lrj@alibaba-inc.com)

Lua数据结构 -- Udata (五)

2014/04/12

Udata负责存储userdata的数据，这部分其实很简单，但是为了保证系列文章的完整性，还是写一篇出来补全。

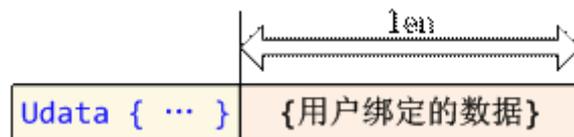
下面是Udata的数据结构：

```
typedef union Udata {
    L_Umaxalign dummy; /* ensures maximum alignment for `local' udata */
    struct {
        CommonHeader;
        struct Table *metatable;
        struct Table *env;
        size_t len;
    } uv;
} Udata; 意义：
```

1. CommonHeader: 和与TValue中的GCHeader能对应起来的部分
2. metatable: userdata的元表，和table的元表一样的
3. env: 创建userdata时，会把当前执行语句的curenv赋给userdata的env，可修改
4. len: 使用userdata的时候绑定对象申请的空间大小

和TString类似，用户绑定的C对象或数据内存紧跟在Udata后面，在luaS_newudata中 (lstring.c 96 - 110) 有如下代码：

```
u = cast(Udata *, luaM_malloc(L, s + sizeof(Udata)));
```



###Udata元表(metatable)的作用：

如果userdata没有元表，那是使用起来将会很麻烦，有元表，可以在脚本这样写：

```
local myuserdata = MyCreateUdata() --MyCreateUdata是一个C函数
myuserdata:MyMethod()           --MyMethod也是一个C函数
```

从C语言层面来看, myuserdata这个变量其实只是个指针, 不像table那样有子元素。但是因为有metatable, 由此可以把成员\函数放到这个metatable中, 在脚本中可以利用它来实现这个类似table的访问方法。

###Udata环境(env)的作用:

env这个成员, 默认是存储创建userdata时的环境table, 而参考Lua官方的文档后, 其实这个env成员在Lua中并没有使用, 它的值时什么并不影响Lua的运行。

这就说明这个成员目前来说是一个用户可以自由操作的table, 在UserDataRefinement (<http://lua-users.org/wiki/UserDataRefinement>)文章中, 告诉了我们一些使用的手段。另外一篇文章<http://lua-users.org/lists/lua-l/2005-08/msg00709.html> (<http://lua-users.org/lists/lua-l/2005-08/msg00709.html>)也对它的使用方法提出了一些建议。

我比较赞成后一篇文章的看法, 对于Udata来说, metatable是一种静态的类型数据(type-common data), 而env则是实例相关的数据(instance-local data)。当然了, 怎么用取决于使用者。

🔖 lua (6) (<http://geekluo.com/tags.html#lua-ref>)

← Previous (<http://geekluo.com/contents/2014/04/12/6-lua-state-structure.html>)

Next → (<http://geekluo.com/contents/2014/04/12/8-auto-jekyll.html>)

评论需要翻墙 for disqus

© 2017 kenlist with Jekyll. Theme: dbyll (<https://github.com/dbtek/dbyll>) by dbtek.