



Lua数据结构 -- TValue (一)

2014/04/11

数据结构的设计，在一定程度上奠定了整个系统的设计，所以决定写一个对Lua主要数据结构的分析文章，本来打算写一篇就好了，但是每个数据类型其实都有点复杂，一篇的话篇幅太长，所以就拆开几篇来写了。

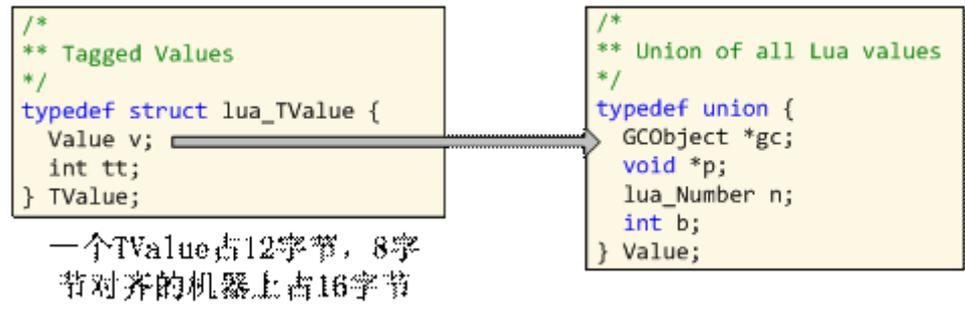
为什么是从TValue说起，**TValue是实现Lua弱数据类型的主要数据结构**，不但在脚本中的值使用了TValue，连Lua的实现中，很多数据结构也依赖于TValue，TValue一定程度上贯穿了整个Lua。先说一下Lua里面的**数据类型**：(lua.h :69)

```

#define LUA_TNONE      (-1)
#define LUA_TNIL       0
#define LUA_TBOOLEAN   1
#define LUA_TLIGHTUSERDATA 2
#define LUA_TNUMBER    3
#define LUA_TSTRING    4
#define LUA_TTABLE     5
#define LUA_TFUNCTION  6
#define LUA_TUSERDATA  7
#define LUA_TTHREAD    8

```

从上面的定义中可以看到，**Lua的值类型有9种**，其中LUA_TNONE是用于判断这个变量是否等于为空使用的，这个是Lua内部使用的，后面再详细说明。现在来看Lua里面的**TValue数据结构**：(lobject.h 71-75)

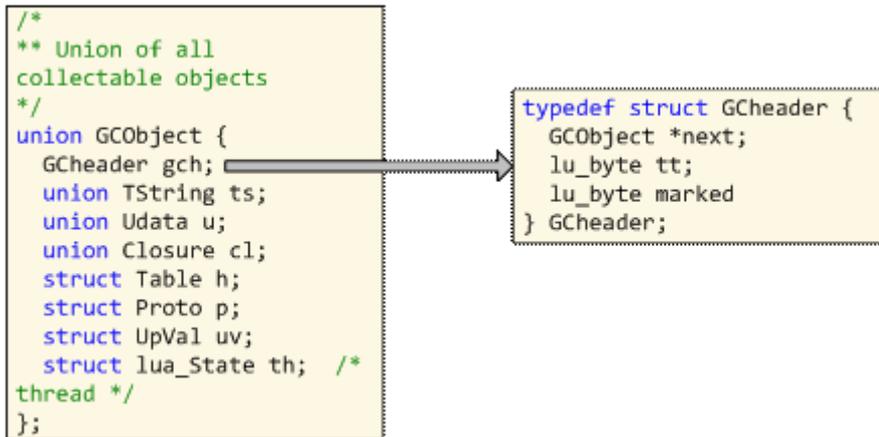


在Lua里面，一个变量使用TValue这个类型来存储的，int tt就是上面宏的类型值（4个字节），而Value则是一个union（8个字节）。在这个union中，其实分工也十分明确：

union value	对使用这个成员的类型值
	LUA_TNIL
GCOBJECT *gc	LUA_TSTRING、LUA_TUSERDATA、LUA_TTABLE、LUA_TTHREAD、LUA_TFUNCTION

union value	对使用这个成员的类型值
Void *p	LUA_TLIGHTUSERDATA
lua_Number n	LUA_TNUMBER
int b	LUA_TBOOLEAN

在Value中, void* p、lua_Number n、int b都是不用回收的值类型, 而GCObject* gc则都是需要回收的对象, 下面是GCObject数据结构: (lstate.h 133-145)



GCObject也是一个union, 存储了一个GCHheader, 这个GCHheader主要用于GC回收机制使用, GC回收机制超出了这次讨论话题, 暂时先忽略。真正存储值的结构是TString、Udata、Closure等等, 每个存储数据的结构都会有GCHheader, 接下来几篇文章将会开始逐个数据类型进行解释。

Name	Value
sizeof(TString)	16
sizeof(Udata)	24
sizeof(Closure)	40
sizeof(Table)	32
sizeof(Proto)	76
sizeof(UpVal)	32
sizeof(lua_State)	120
sizeof(GCObject)	120

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

← Previous (<http://geekluo.com/contents/2014/04/11/1-start-my-blog.html>)

Next → (<http://geekluo.com/contents/2014/04/11/3-lua-table-structure.html>)

评论需要翻墙 for disqus

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