int \_tmain(int argc, \_TCHAR\* argv[])

{

lua\_State \*L = lua\_open();

luaL\_openlibs(L); **// 初始化lua库，添加了io、math等等**

lua\_pushstring(L, "print0");

lua\_pushcfunction(L, print0);

lua\_settable(L, LUA\_GLOBALSINDEX);

luaL\_dofile(L, "Test.lua");

system("pause");

return 0;

}

注册全局函数**lua\_settable**

LUA\_API void lua\_settable (lua\_State \*L, int idx) {

StkId t;

lua\_lock(L);

**api\_checknelems**(L, 2); **// 检测当前元素是否超过2个，top指向栈中第一个空元素**

t = **index2adr**(L, idx); **// 获取当前位置对应的TValue指针**

**api\_checkvalidindex**(L, t); **// 检测当前TValue指针是否指向nil元素**

**luaV\_settable**(L, t, L->top - 2, L->top - 1);

**L->top -= 2;** /\* pop index and value \*/ **// 将当前的两个值弹出，实际上元素还在栈中**

lua\_unlock(L);

}

#define api\_checknelems(L, n) api\_check(L, (n) <= (L->top - L->base))

#define api\_checkvalidindex(L, i) api\_check(L, (i) != luaO\_nilobject)

#define LUA\_REGISTRYINDEX (-10000)

#define LUA\_ENVIRONINDEX (-10001)

#define LUA\_GLOBALSINDEX (-10002)

#define lua\_upvalueindex(i) (LUA\_GLOBALSINDEX-(i))

static TValue \*index2adr (lua\_State \*L, int idx) {

if (idx > 0) { **// 大于0表示是正常表，然后通过栈的位置取的相应的TValue**

TValue \*o = L->base + (idx - 1);

api\_check(L, idx <= L->ci->top - L->base); **// 判断索引是否在当前的合理区间**

if (o >= L->top) return cast(TValue \*, luaO\_nilobject); **// 如果超出了，则返回nil对象**

else return o;

}

else if (idx > LUA\_REGISTRYINDEX) { **// 支持负数索引**

api\_check(L, idx != 0 && -idx <= L->top - L->base);

return L->top + idx;

}

else switch (idx) { /\* pseudo-indices \*/

case LUA\_REGISTRYINDEX: return registry(L);

case LUA\_ENVIRONINDEX: {

Closure \*func = curr\_func(L);

sethvalue(L, &L->env, func->c.env);

return &L->env;

}

case LUA\_GLOBALSINDEX: return gt(L);

default: {

Closure \*func = curr\_func(L);

idx = LUA\_GLOBALSINDEX - idx;

return (idx <= func->c.nupvalues)

? &func->c.upvalue[idx-1]

: cast(TValue \*, luaO\_nilobject);

}

}

}

**下面这个函数看看Table的实现**

void luaV\_settable (lua\_State \*L, const TValue \*t, TValue \*key, StkId val) {

int loop;

for (loop = 0; loop < MAXTAGLOOP; loop++) {

const TValue \*tm;

if (ttistable(t)) { /\* `t' is a table? \*/

Table \*h = hvalue(t);

TValue \*oldval = luaH\_set(L, h, key); /\* do a primitive set \*/

if (!ttisnil(oldval) || /\* result is no nil? \*/

(tm = fasttm(L, h->metatable, TM\_NEWINDEX)) == NULL) { /\* or no TM? \*/

setobj2t(L, oldval, val);

luaC\_barriert(L, h, val);

return;

}

/\* else will try the tag method \*/

}

else if (ttisnil(tm = luaT\_gettmbyobj(L, t, TM\_NEWINDEX)))

luaG\_typeerror(L, t, "index");

if (ttisfunction(tm)) {

callTM(L, tm, t, key, val);

return;

}

t = tm; /\* else repeat with `tm' \*/

}

luaG\_runerror(L, "loop in settable");

}