

Making a C Library for Lua

Androids: Design and Practice

Douglas Blank
Fall 2011

Why a C Library for Lua?

- Speed
- Access some other C or C++ code

```
$ lua  
Lua 5.1.4 Copyright © 1994–2008 Lua.org  
> require('awesome')  
> awesome.func( 23 )  
4376467
```

Writing a C Library for Lua

- Write the C/C++ code
- Map Lua names to the C names
- Register the library

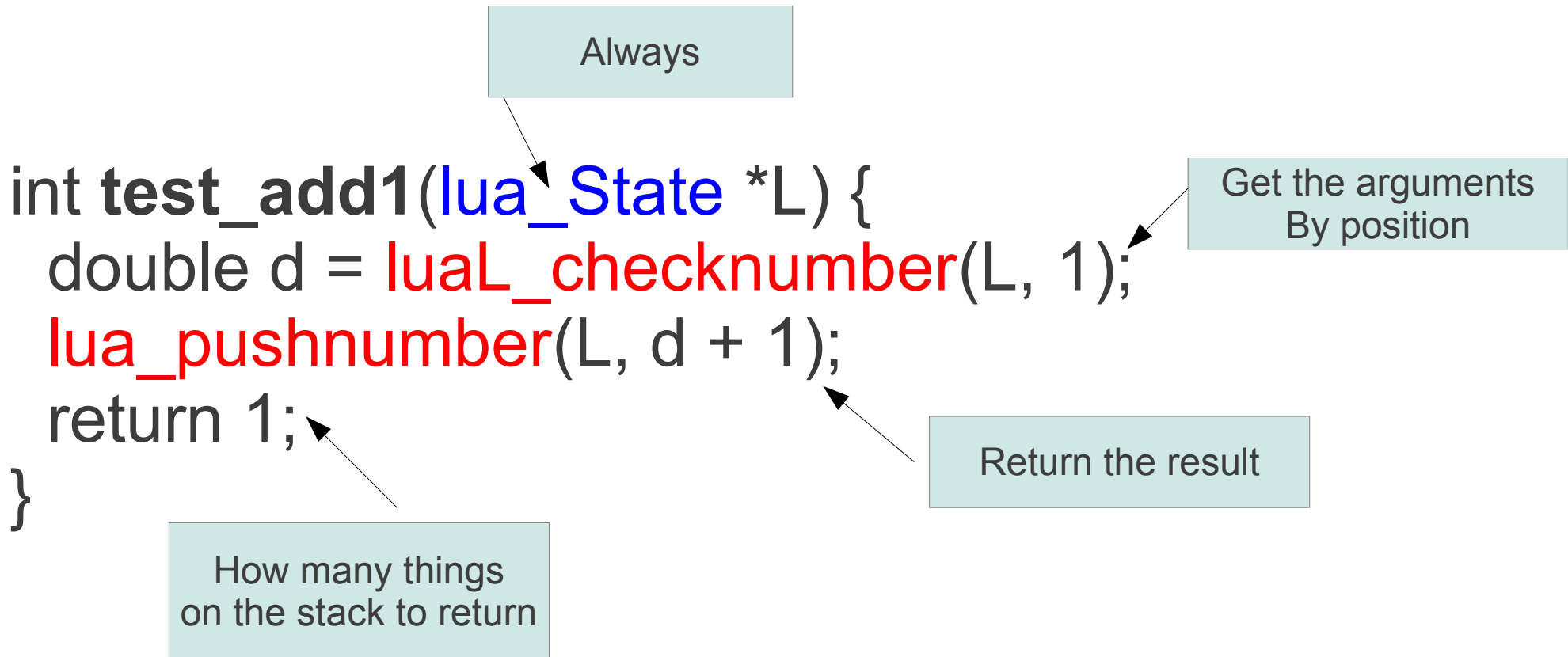
As an example, let's write a `add1` function:

```
> require('test')  
> test.add1(23)  
24
```

Step 1: C Function

```
int test_add1(lua_State *L) {  
    double d = luaL_checknumber(L, 1);  
    lua_pushnumber(L, d + 1);  
    return 1;  
}
```

Step 1: C Function



Step 2: Map Lua names to C names

```
static const struct luaL_reg testlib[] = {  
    {"add1", test_add1},  
    {NULL, NULL}  
};
```

Step 3: Register the library

```
int luaopen_test (lua_State *L) {  
    luaL_register(L, "test", testlib);  
    return 1;  
}
```

Lua name for library

Mapping Struct



If you have C++ code...

```
#ifdef __cplusplus  
extern "C" {  
#endif
```

...

```
#ifdef __cplusplus  
}  
#endif
```


Final C++ Code

```
#ifdef __cplusplus
extern "C" {
#endif

#include "lua.h"
#include "lualib.h"
#include "lauxlib.h"

#ifdef __cplusplus
}
#endif

int test_add1(lua_State *L) {
    double d = luaL_checknumber(L, 1); // get item 1
    lua_pushnumber(L, d + 1);
    return 1; // number of items returned
}

static const struct luaL_reg testlib[] = {
    {"add1", test_add1},
    {NULL, NULL}
};

extern "C"
int luaopen_test (lua_State *L) {
    luaL_register(L, "test", testlib);
    return 1;
}
```

Compile

```
g++ lua_test.cpp -shared -fpic -I/usr/local/include -o test.so
```

Compile with Make

```
INCLUDE = -I/usr/local/include
```

```
LIBOPTS = -shared
```

```
FLAGS = -fpic
```

```
CXX = g++
```

```
test.so: lua_test.cpp
```

```
    $(CXX) $^ $(INCLUDE) $(FLAGS) $(LIBOPTS) -o $@
```

Use your new Library

```
$ lua
Lua 5.1.4 Copyright (C) 1994-2008 Lua.org, PUC-Rio
> require("test")
> print(test);
table: 0xf98cc0
> print(test.add1);
function: 0xf98d40
> test.add1(10);
> print(test.add1(10));
11
```