前提:

1.需要一个有lua环境的c++项目(lua-5.2.XX)

2.下载luasocket 3.0-rc1

下载luasocket:https://github.com/diegonehab/luasocket/tags

    最新版本v3.0-rc1

Lua集成LuaSocket:

|  |  |  |
| --- | --- | --- |
| 文件名 | 替换前 | 替换后 |
| lualib.h | /\*  \*\* $Id: lualib.h,v 1.43.1.1 2013/04/12 18:48:47 roberto Exp $  \*\* Lua standard libraries  \*\* See Copyright Notice in lua.h  \*/  #ifndef lualib\_h  #define lualib\_h  #include "lua.h"    LUAMOD\_API int (luaopen\_base) (lua\_State \*L);  #define LUA\_COLIBNAME   "coroutine"  LUAMOD\_API int (luaopen\_coroutine) (lua\_State \*L);  #define LUA\_TABLIBNAME  "table"  LUAMOD\_API int (luaopen\_table) (lua\_State \*L);  #define LUA\_IOLIBNAME   "io"  LUAMOD\_API int (luaopen\_io) (lua\_State \*L);  #define LUA\_OSLIBNAME   "os"  LUAMOD\_API int (luaopen\_os) (lua\_State \*L);  #define LUA\_STRLIBNAME  "string"  LUAMOD\_API int (luaopen\_string) (lua\_State \*L);  #define LUA\_BITLIBNAME  "bit32"  LUAMOD\_API int (luaopen\_bit32) (lua\_State \*L);  #define LUA\_MATHLIBNAME "math"  LUAMOD\_API int (luaopen\_math) (lua\_State \*L);  #define LUA\_DBLIBNAME   "debug"  LUAMOD\_API int (luaopen\_debug) (lua\_State \*L);  #define LUA\_LOADLIBNAME "package"  LUAMOD\_API int (luaopen\_package) (lua\_State \*L);  /\* open all previous libraries \*/  LUALIB\_API void (luaL\_openlibs) (lua\_State \*L);    #if !defined(lua\_assert)  #define lua\_assert(x)   ((void)0)  #endif  #endif | /\*  \*\* $Id: lualib.h,v 1.43.1.1 2013/04/12 18:48:47 roberto Exp $  \*\* Lua standard libraries  \*\* See Copyright Notice in lua.h  \*/  #ifndef lualib\_h  #define lualib\_h  #include "lua.h"    LUAMOD\_API int (luaopen\_base) (lua\_State \*L);  #define LUA\_COLIBNAME   "coroutine"  LUAMOD\_API int (luaopen\_coroutine) (lua\_State \*L);  #define LUA\_TABLIBNAME  "table"  LUAMOD\_API int (luaopen\_table) (lua\_State \*L);  #define LUA\_IOLIBNAME   "io"  LUAMOD\_API int (luaopen\_io) (lua\_State \*L);  #define LUA\_OSLIBNAME   "os"  LUAMOD\_API int (luaopen\_os) (lua\_State \*L);  #define LUA\_STRLIBNAME  "string"  LUAMOD\_API int (luaopen\_string) (lua\_State \*L);  #define LUA\_BITLIBNAME  "bit32"  LUAMOD\_API int (luaopen\_bit32) (lua\_State \*L);  #define LUA\_MATHLIBNAME "math"  LUAMOD\_API int (luaopen\_math) (lua\_State \*L);  #define LUA\_DBLIBNAME   "debug"  LUAMOD\_API int (luaopen\_debug) (lua\_State \*L);  #define LUA\_LOADLIBNAME "package"  LUAMOD\_API int (luaopen\_package) (lua\_State \*L);  /\* open all previous libraries \*/  LUALIB\_API void (luaL\_openlibs) (lua\_State \*L);  #define LUA\_SOCKETLIBNAME "socket"  LUAMOD\_API int (luaopen\_socket\_core)(lua\_State \*L);  #define LUA\_MIMELIBNAME "mime"  LUAMOD\_API int (luaopen\_mime\_core)(lua\_State \*L);    #if !defined(lua\_assert)  #define lua\_assert(x)   ((void)0)  #endif  #endif |
| Linit.c | /\*  \*\* $Id: linit.c,v 1.32.1.1 2013/04/12 18:48:47 roberto Exp $  \*\* Initialization of libraries for lua.c and other clients  \*\* See Copyright Notice in lua.h  \*/  /\*  \*\* If you embed Lua in your program and need to open the standard  \*\* libraries, call luaL\_openlibs in your program. If you need a  \*\* different set of libraries, copy this file to your project and edit  \*\* it to suit your needs.  \*/  #define linit\_c  #define LUA\_LIB  #include "lua.h"  #include "lualib.h"  #include "lauxlib.h"  /\*  \*\* these libs are loaded by lua.c and are readily available to any Lua  \*\* program  \*/  static const luaL\_Reg loadedlibs[] = {    {"\_G", luaopen\_base},    {LUA\_LOADLIBNAME, luaopen\_package},    {LUA\_COLIBNAME, luaopen\_coroutine},    {LUA\_TABLIBNAME, luaopen\_table},    {LUA\_IOLIBNAME, luaopen\_io},    {LUA\_OSLIBNAME, luaopen\_os},    {LUA\_STRLIBNAME, luaopen\_string},    {LUA\_BITLIBNAME, luaopen\_bit32},    {LUA\_MATHLIBNAME, luaopen\_math},    {LUA\_DBLIBNAME, luaopen\_debug},    {NULL, NULL}  };  /\*  \*\* these libs are preloaded and must be required before used  \*/  static const luaL\_Reg preloadedlibs[] = {    {NULL, NULL}  };  LUALIB\_API void luaL\_openlibs (lua\_State \*L) {    const luaL\_Reg \*lib;    /\* call open functions from 'loadedlibs' and set results to global table \*/    for (lib = loadedlibs; lib->func; lib++) {      luaL\_requiref(L, lib->name, lib->func, 1);      lua\_pop(L, 1);  /\* remove lib \*/    }    /\* add open functions from 'preloadedlibs' into 'package.preload' table \*/    luaL\_getsubtable(L, LUA\_REGISTRYINDEX, "\_PRELOAD");    for (lib = preloadedlibs; lib->func; lib++) {      lua\_pushcfunction(L, lib->func);      lua\_setfield(L, -2, lib->name);    }    lua\_pop(L, 1);  /\* remove \_PRELOAD table \*/  } | /\*  \*\* $Id: linit.c,v 1.32.1.1 2013/04/12 18:48:47 roberto Exp $  \*\* Initialization of libraries for lua.c and other clients  \*\* See Copyright Notice in lua.h  \*/  /\*  \*\* If you embed Lua in your program and need to open the standard  \*\* libraries, call luaL\_openlibs in your program. If you need a  \*\* different set of libraries, copy this file to your project and edit  \*\* it to suit your needs.  \*/  #define linit\_c  #define LUA\_LIB  #include "lua.h"  #include "lualib.h"  #include "lauxlib.h"  /\*  \*\* these libs are loaded by lua.c and are readily available to any Lua  \*\* program  \*/  static const luaL\_Reg loadedlibs[] = {    {"\_G", luaopen\_base},    {LUA\_LOADLIBNAME, luaopen\_package},    {LUA\_COLIBNAME, luaopen\_coroutine},    {LUA\_TABLIBNAME, luaopen\_table},    {LUA\_IOLIBNAME, luaopen\_io},    {LUA\_OSLIBNAME, luaopen\_os},    {LUA\_STRLIBNAME, luaopen\_string},    //{LUA\_BITLIBNAME, luaopen\_bit32},    {LUA\_MATHLIBNAME, luaopen\_math},    {LUA\_DBLIBNAME, luaopen\_debug},  #if defined(LUA\_COMPAT\_BITLIB)    { LUA\_BITLIBNAME, luaopen\_bit32 },  #endif    { LUA\_SOCKETLIBNAME, luaopen\_socket\_core },    { LUA\_MIMELIBNAME, luaopen\_mime\_core },    {NULL, NULL}  };  /\*  \*\* these libs are preloaded and must be required before used  \*/  static const luaL\_Reg preloadedlibs[] = {    {NULL, NULL}  };  LUALIB\_API void luaL\_openlibs (lua\_State \*L) {    const luaL\_Reg \*lib;    /\* call open functions from 'loadedlibs' and set results to global table \*/    for (lib = loadedlibs; lib->func; lib++) {      luaL\_requiref(L, lib->name, lib->func, 1);      lua\_pop(L, 1);  /\* remove lib \*/    }    /\* add open functions from 'preloadedlibs' into 'package.preload' table \*/    luaL\_getsubtable(L, LUA\_REGISTRYINDEX, "\_PRELOAD");    for (lib = preloadedlibs; lib->func; lib++) {      lua\_pushcfunction(L, lib->func);      lua\_setfield(L, -2, lib->name);    }    lua\_pop(L, 1);  /\* remove \_PRELOAD table \*/  } |

3.集成好后重新编译出lua.lib/luasocket.lib即可正常使用