# ######### 编译

|  |  |
| --- | --- |
| 依赖库 |  |
|  | libcurld.lib ws2\_32.lib winmm.lib wldap32.lib |
| 需要Preprocessor Definitions定义宏BUILDING\_LIBCURL | |
|  |  |
| 或code里指定： | |
| pragma comment( lib, "D:\\\_WorkSrc\\\_git\\MultiCurl\\multiCurl\\libcurld.lib" ) | |
| #pragma comment( lib, "ws2\_32.lib" ) | |
| #pragma comment( lib, "winmm.lib" ) | |
| #pragma comment( lib, "wldap32.lib" ) | |
| stdafx.h里#define BUILDING\_LIBCURL | |
|  |  |
| ### 编译curl | |
| Open Visual Studio 2008 Command Prompt | |
| E:\\_WorkSrc\curl-7.50.0\winbuild>nmake /f Makefile.vc mode=dll DEBUG=yes ENABLE\_WINSSL=no ENABLE\_IPV6=no VC=9 ENABLE\_IDN=no ENABLE\_SSPI=no GEN\_PDB=yes | |
|  |  |
| ### 编译curl\_7.50 (curl\_multi\_wait在7.26里尚未存在) | |
| E:\\_WorkSrc\curl-7.50.0\winbuild>nmake /f Makefile.vc mode=static VC=9 DEBUG=yes USE\_IPV6=no USE\_IDN=no GEN\_PDB=yes RTLIBCFG=static | |
| 最后RTLIBCFG=static表示静态链接运行时库，即/MT静态链接相应的libcmt.lib | |
|  |  |
| 编译dll |  |
| E:\\_WorkSrc\curl-7.50.0\winbuild>nmake /f Makefile.vc mode=dll VC=9 DEBUG=no USE\_IPV6=yes USE\_IDN=no | |
| E:\\_WorkSrc\curl-7.50.0\winbuild>nmake /f Makefile.vc mode=dll VC=9 DEBUG=yes USE\_IPV6=yes USE\_IDN=no | |
|  |  |
|  |  |
|  |  |
|  |  |
| ######################### ubuntu编译curl | |
| git clone https://github.com/curl/curl.git | |
| vi GIT-INFO | |
| ./buildconf | |
| ./configure --help | |
| ./configure | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| ######################### c-ares; cares | |
| git clone https://github.com/c-ares/c-ares.git | |
| vi INSTALL.md | |
| ### Windows | |
| ./buildconf.bat | |
| vc/vc6aws.dsw => vc6aws.sln (2008) | |
| # curl使用cares (libcares.lib为静态库, cares.lib为动态库lib) | |
| 将cares所有的头文件（测试其实只需要ares.h, ares\_build.h, ares\_rules.h, ares\_version.h）拷贝到curl同级目录deps/include下，生成的libcares.lib, cares.dll, cares.lib拷贝到curl同级目录deps/lib下 | |
| libcurl使用cares要定义宏: USE\_ARES | |
| libcurl编译dll静态编译链接cares库: 定义宏CARES\_STATICLIB, 链接libcares.lib | |
| libcurl编译dll动态编译链接cares库: 链接cares.lib | |

# ######### tanjl测试结果

|  |  |  |
| --- | --- | --- |
|  | **Thread-Based** | **c-ares** |
| **线程数** | 100并发HTTP时，每个DNS解析请求一个线程, 观察到会有几十个线程同时存在 | 100个并发请求时，最多观察到4个线程; 1~2个请求时，观察到有1~2个线程不等 |
| **退出时间** | CURL multi方式，退出时间为几百毫秒，但在大量并发请求过程中拔掉网线时会有崩溃 | 退出时不会崩溃，但退出时间不确定, 也不等于连接超时时间，大量并发测试发现几百毫秒到几秒不等，少量请求时为几百毫秒 |
| **官方说明** | 所有平台都支持 | 有些平台可能不支持 (windows我们测试可用) |
| **其他说明** | 采用系统实现的DNS解析方式，更可靠 | DNS超时时间不准确. 解析结果没有thread-based方式可靠  (<best practices: c-ares vs threaded resolver> https://curl.haxx.se/mail/lib-2013-09/0098.html) |
| **参考例子** | None | <c-ares 一个C语言的异步DNS解析库> [www.ithao123.cn/content-8677653.html](http://www.ithao123.cn/content-8677653.html)  <[gevent源码分析] c-ares异步DNS请求 > blog.csdn.net/yueguanghaidao/article/details/39301269 |

### 其它相关参考

基于curl的测试客户端:

<http://curl-loader.sourceforge.net/>