

B_H_L的专栏

个人资料



B_H_L

访问：726073次

积分：7752

等级：BLOG*>*6

排名：第1679名

原创：53篇

转载：331篇

译文：0篇

评论：61条

文章搜索

文章分类

java (60)

MFC (22)

JSP (7)

DB (2)

data structure (3)

AJAX (1)

oracle (6)

C (67)

ubuntu (2)

linux (66)

算法 (9)

C++ (6)

杂谈 (12)

杀毒引擎 (7)

软件工程 (4)

百度地图 (2)

Mysql (1)

data mining (17)

python (38)

shell (7)

git (8)

KVM (2)

libcurl源码编译及使用

2014-02-26 16:2512491人阅读评论(5)收藏举报

分类：C (66)

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

1:linux下源码编译

./configure --prefix=/usr/local/curl --disable-shared --enable-static --without-libidn --without-ssl --without-librtmp --without-gnutls --without-nss --without-libssh2 --without-zlib --without-winidn --disable-rtsp --disable-ldap --disable-ldaps --disable-ipv6

注意：不需要其它功能就disable，不然会在连接时报错，如果你没有disable掉，但你的机器上又没有安装相应的库，link时会报错。

2: windows下源码编译

libcurl 7.21以后的版本在Windows下的编译比较简单，自帶了MinGW和VC环境的Makefile文件，首先去Curl官网下载源代码：<http://curl.haxx.se/download.html>，任选一个下载即可，推荐这个：[curl-7.21.2.tar.gz](#)，下载完成后解压开，打开命令行进入curl源码目录，(在此之前请先设置好MinGW的环境变量)：
cd curl-7.21.2
编译libcurl库文件：
cd lib
make -f Makefile.m32
等待编译完成即可
(若需编译生成curl可执行文件，则执行：
cd ../src
make -f Makefile.m32)
编译完成后，我们需要复制include头文件和库文件到一个目录供程序开发用
1. 新建curllib目录
2. 新建curllib/include目录，将源代码include目录里的curl文件夹复制到curllib/include目录，这些是使用libcurl需要的头文件
3. 新建curllib/lib目录，将源代码lib目录里编译好的库文件libcurl.a，libcurl.dll，libcurl.dll复制到curllib目录
4. 将MinGW安装目录下lib文件夹里的libwldap32.a和libws2_32.a复制到curllib/lib目录
至此，curllib就是我们开发中要使用到的libcurl的全部文件，下面新建一个测试程序main.c:

[cpp]

01. #include <stdio.h>

02. #include <stdlib.h>

03. #include <string.h>

04. #include "../curllib/include/curl.h"

05.

06. size_t callback_write_file(void *ptr, size_t size, size_t nmemb, void *userp)

07. {

08. //fprintf(stdout,"size=%d\n",size);

09. fwrite(ptr,size,nmemb,(FILE *)userp);

10. return size * nmemb; //必须返回这个大小，否则只回调一次，不清楚为何。

11. }

12.

http://blog.csdn.net/b_h_l/article/details/19985061

1/6

nginx (3)

postgresql (2)

逆向 (21)

os (1)

安全研究相关 (5)

驱动开发 (2)

文章存档

2016年04月 (2)

2015年10月 (1)

2015年09月 (3)

2015年08月 (2)

2015年07月 (5)

展开

阅读排行

根据两点经纬度计算距离 (39490)

nginx做静态文件下载服务 (31504)

java 从jar包中读取资源 (23734)

git 如何让单个文件回退到指定版本 (21602)

char与TCHAR相互转化 (20607)

java socket通信 (17954)

如何在网页中调用百度地图 (15453)

shell中uniq与sort -u 两种方法 (15337)

libcurl源码编译及使用 (12490)

java jTable排序和过滤 (11543)

评论排行

java jTable排序和过滤 (10)

java socket通信 (7)

libcurl源码编译及使用 (5)

根据两点经纬度计算距离 (4)

JDIC 中利用WebBrowser (4)

java 从jar包中读取资源 (4)

ubuntu12.04 修改mysql (3)

C结构体边界对齐2 (3)

Linux画图工具gnuplot (2)

决策树CART (2)

推荐文章

* 浅谈android中异步加载之"取消异步加载"二

*Delta - 轻量级JavaWeb框架使用文档

*Nginx正反向代理、负载均衡等功能实现配置

* 浅析ZeroMQ工作原理及其特点

*android源码解析（十九）-->Dialog加载绘制流程

*Spring Boot 实践折腾记（三）：三板斧，Spring Boot下使用Mybatis

最新评论

java jTable排序和过滤(JDK 6.0 Michael5937: 请问，return Value = getValueAt(0, column).getClass();...

libcurl源码编译及使用 : 可行的，开始找不到mingw32-

```
13. void *func_download_file(FILE *fp)
14. {
15.     CURL *curl = NULL;
16.     CURLcode res;
17.
18.     curl_global_init(CURL_GLOBAL_DEFAULT);
19.     curl = curl_easy_init();
20.     curl_easy_setopt(curl, CURLOPT_URL, "http://10.255.64.30:8082/20140225/vt/01B5CE3D7366577E5A1");
    设置下载的URI
21.     curl_easy_setopt(curl, CURLOPT_TIMEOUT, 20); //设置超时
22.     curl_easy_setopt(curl, CURLOPT_NOSIGNAL, 1); //屏蔽其它信号
23.     //curl_easy_setopt(curl, CURLOPT_HEADERDATA, 1); //下载数据包
24.     //curl_easy_setopt(curl, CURLOPT_RANGE, "0-500"); //用于断点续传，设置下载的分片
25.
26.     //char buffer[MAXHEADLEN] = {0x0};
27.     curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback_write_file); //设置下载数据的回调函数
    数
28.     curl_easy_setopt(curl, CURLOPT_WRITEDATA, fp);
29.
30.     res = curl_easy_perform(curl);
31.     if(res != CURLE_OK)
32.     {
33.         fprintf(stdout, "curl execute fail\n");
34.     }
35.     curl_easy_cleanup(curl);
36.     curl_global_cleanup();
37.     return 0;
38. }
39.
40. int main()
41. {
42.     FILE *fp = fopen("./kaka", "wb");
43.     if(!fp)
44.     {
45.         fprintf(stdout, "fopen fail \n");
46.     }
47.     func_download_file(fp);
48.     fclose(fp);
49.     return 0;
50. }
```

方法一、命令行编译使用libcurl的程序
假设测试代码curltest.c位于e:/project
假设curlib文件夹的位置为c:/curlib
命令行运行下列命令编译这个测试程序：
cd e:/project
gcc -I. -Ic:/curlib/include -g -O2 -DCURL_STATICLIB -c main.c
gcc -o main.exemain.o -Lc:/curlib/lib -lcurl -lws2_32
这时可以看到main.c目录下生成了一个main.exe文件
接着在命令行输入：
main.exe
方法二、Code::Blocks中使用libcurl静态库
1. 新建工程，在工程里添加代码同上的main.c文件
2. 将上面curlib/include目录下的curl文件夹复制到MinGW安装目录的include目录，或在工程目录下建立以目录存放include，然后打开Build Options选项，在Search directories选项卡下添加include目录
3. 工程名上右键打开Build Options选项，在Compiler Settings选项卡下的#defines里面输入CURL_STATICLIB，（这表示使用静态库）
4. 在Linker Settings选项卡下面的link libraries里添加上面curlib/lib目录里的四个文件：
C:\curlib\lib\libcurl.a
C:\curlib\lib\libcurl.dll.a
C:\curlib\lib\libws2_32.a
C:\curlib\lib\libws2_32.a
然后回到工程页面，点击Build即可
需要注意的是，本文中编译的是不带ssl和zlib支持的libcurl，如果需要编译支持ssl和zlib的curl，还需要先编译openssl，zlib和libssh，编译zlib比较简单，直接使用源码自带的makefile文件即可，编译openssl需要安装MSYS和Perl，还需要修改一些代码，libssh的编译依赖openssl，网上都可以找到方法，也可以看源码的README文件。
附一篇在C语言中使用libcurl库的文章供参考：
使用 cURL 和 libcurl 通过 Internet 进行对话
下面几篇是Curl的文档和教程：
Scripting HTTP Requests Using Curl
Curl Man Page
Curl Mannul

make.exe文件，无法重命名。后来解决了。

java socket通信
吴士龙: 很具备参考性，感谢楼主分享

java 从jar包中读取资源文件
lmb55: 讲的好详细啊，学习了~

C结构体边界对齐2
SherlockReal: 学习了

决策树CART
矮王子: 你这表不完整吧。怎么算恒温的也只有五个而已。没有七个。

JFinal是一个基于Java的极速Web
小木学堂_Michael: CSDN学院有JFinal视频教程
<http://bbs.csdn.net/topics/391898...>

决策树CART
张睿晖2: 写的不错奥、

java JTable排序和过滤(JDK 6.0
lywsir: 正是吾所用到的

Docker使用心得
qq_30439625: 博主 这里有win7 64位的虚拟机吗，防封的那种

Using The libcurl C Interface
libcurl – small example snippets
programming with libcurl

2: 生成DLL

```
[cpp] C {  
01. #include <stdio.h>  
02. #include <stdlib.h>  
03. #include "../curl/lib/include/curl.h"  
04. size_t callback_write_file(void *ptr, size_t size, size_t nmemb, void *userp)  
05. {  
06.     //fprintf(stdout,"size=%d\n",size);  
07.     fwrite(ptr,size,nmemb,(FILE *)userp);  
08.     return size * nmemb;    //必须返回这个大小，否则只回调一次，不清楚为何。  
09. }  
10.  
11. <span style="color:#ff0000;">__declspec(dllexport) </span>void *  
    <span style="color:#ff0000;">__cdecl</span> func_download_file(FILE *fp)  
12. {  
13.     CURL *curl = NULL;  
14.     CURLcode res;  
15.  
16.     curl_global_init(CURL_GLOBAL_DEFAULT);  
17.     curl = curl_easy_init();  
18.     curl_easy_setopt(curl, CURLOPT_URL, "http://10.255.64.30:8082/20140225/vt/01B5CE3D7366577E5A");  
    设置下载的URI  
19.     curl_easy_setopt(curl, CURLOPT_TIMEOUT, 20);    //设置超时  
20.     curl_easy_setopt(curl, CURLOPT_NOSIGNAL, 1);    //屏蔽其它信号  
21.     //curl_easy_setopt(curl, CURLOPT_HEADERDATA, 1);    //下载数据包  
22.     //curl_easy_setopt(curl, CURLOPT_RANGE, "0-500");    //用于断点续传，设置下载的分片  
23.  
24.     //char buffer[MAXHEADLEN] = {0x0};  
25.     curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback_write_file); //设置下载数据的回调函数  
26.     curl_easy_setopt(curl, CURLOPT_WRITEDATA, fp);  
27.  
28.     res = curl_easy_perform(curl);  
29.     if(res != CURLE_OK)  
30.     {  
31.         fprintf(stdout,"curl execute fail\n");  
32.     }  
33.     curl_easy_cleanup(curl);  
34.     curl_global_cleanup();  
35.     return 0;  
36. }
```

在codeblocks中加入libcurl需要的.a文件，然后在Compiler Settings选项卡下的#defines里面输入
CURL_STATICLIB，（这表示使用静态库）
执行后可生成.dll

3: 调用DLL

```
[cpp] C {  
01. #include <stdio.h>  
02. #include <stdlib.h>  
03. #include <windows.h>  
04.  
05. typedef int (*func_download_file_p)(FILE *fp);  
06.  
07.  
08.  
09. int main()  
10. {  
11.     HINSTANCE hinstance=NULL;  
12.     FILE *fp = NULL;  
13.  
14.     fp = fopen("../download","wb");  
15.     if(!fp)  
16.     {  
17.         fprintf(stdout,"fopen fail \n");  
18.     }  
19.  
20.     func_download_file_p func_download_file;  
21.     //加载动态链接库
```

```
22.         hinstance = LoadLibrary("F:\\Program Files\\CodeBlocks\\workspace\\TestCurlDll2\\libTestCurlI
23.         func_download_file = (func_download_file_p)GetProcAddress(hinstance, "func_download_file");
24.         func_download_file(fp);
25.         return 0;
26.     }
```

4: libcurl的使用

一、常用函数

1) libcurl的全局初始化及释放

```
CURLcode curl_global_init(long flags)
    flags: CURL_GLOBAL_ALL      //初始化所有的可能的调用。
           CURL_GLOBAL_SSL      //初始化支持 安全套接字层。
           CURL_GLOBAL_WIN32    //初始化win32套接字库。
           CURL_GLOBAL_NOHING  //没有额外的初始化。
void      curl_global_cleanup(void)
```

应该在程序开始时调用初始化函数。虽然不调用这个初始化函数，libcurl会在curl_easy_init()函数中自动调用。但在多线程处理时，可能会出现多次自动调用的情况。

2) 初始化下载handle及释放

```
CURL *curl = curl_easy_init();
curl_easy_cleanup(curl);
```

3) 设置下载属性. 及常用参数.

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPToption option, parameter);

1. 设置下载数据的回调函数.
option:
CURLOPT_WRITEFUNCTION //设置回调函数
    回调函数原型为: size_t function( void *ptr, size_t size, size_t nmemb, void *userp);
在libcurl接收到数据后被调用。
void *ptr是下载回来的数据。
void *userp是用户指针，用户通过这个指针传输自己的数据。
CURLOPT_WRITEDATA
    设置回调函数中的void *userp指针的来源。

2. 下载进度控制.
option:
CURLOPT_NOPROGRESS
    为了使CURLOPT_PROGRESSFUNCTION被调用。CURLOPT_NOPROGRESS必须被设置为false。
CURLOPT_PROGRESSFUNCTION
    CURLOPT_PROGRESSFUNCTION 指定的函数正常情况下每秒被libcurl调用一次。
CURLOPT_PROGRESSDATA
    CURLOPT_PROGRESSDATA指定的参数将作为CURLOPT_PROGRESSFUNCTION指定函数的参数。
整个处理与下载数据回调的处理相同。

3. 其它常用属性.
option:
CURLOPT_URL
    设置访问的URI。
CURLOPT_NOSIGNAL
    屏蔽其它信号。
CURLOPT_HEADER
    取数据时连同HTTP头部一起取回。
CURLOPT_HEADERFUNCTION
CURLOPT_HEADERDATA
    只取HTTP头部数据，处理与下载数据回调的处理相同。
CURLOPT_TIMEOUT
    超时时间。
CURLOPT_CONNECTIO NTIMEOUT
    连接等待时间。
CURLOPT_FOLLOWLOCATION
    设置支持302重定向
CURLOPT_RANGE
    断点续传，指定传输分片，格式:"0-200"
```

4) 开始下载

```
CURLcode curl_easy_perform(CURL *handle);
```

5) URL encodes the given string(http://curl.haxx.se/libcurl/c/curl_escape.html)

```
char *curl_escape( char * url , int length );
```

This function will convert the given input string to an URL encoded string and return that as a new allocated string. All input characters that are not a-z, A-Z or 0-9 will be converted to their "URL escaped" version (%NN where NN is a two-digit hexadecimal number).

If the 'length' argument is set to 0, curl_escape() will use strlen() on the input 'url' string to find out the size.

You must curl_free() the returned string when you're done with it.

顶

1

踩

0

上一篇

python模块subprocess学习

下一篇

gcc编译动态和静态链接库

我的同类文章

C（66）

• 哈夫曼树与哈夫曼编码

2016-04-19

阅读 28

• 轻量级C语言实现的xml解析...

2014-12-01

阅读 3887

• select函数详解及实例分析

2014-10-22

阅读 277

• win32下使用pthread

2014-06-20

阅读 1031

• 深入分析Windows和Linux...

2014-03-25

阅读 449

• 红黑树

2015-08-11

阅读 201

• socket编程模型

2014-10-22

阅读 373

• 纯C语言INI文件解析

2014-10-17

阅读 457

• C循环队列

2014-05-12

阅读 365

• JSON的C语言编解码器—...

2014-03-21

阅读 4634

更多文章

猜你在找

《C语言/C++学习指南》Linux开发篇	libcurl的使用
iOS8-Swift开发教程	ubuntu 安装libcurl库
Windows CE车载应用的实现与相关技术点	curl-7390 for android 编译以及jni中的使用
Linux环境C语言编程基础	mesos下载准备安装环境编译以及测试
从此不求人:自主研发一套PHP前端开发框架	mesos下载准备安装环境编译以及测试

查看评论

5楼 [游客] 2016-05-10 14:53发表

C


可行的，开始找不到mingw32-make.exe文件，无法重命名。后来解决了。

4楼 Miseli 2016-04-06 01:37发表

C

博主分享精神伟大？
但是window下的编译过程写错了？还是根本就说的不对？

3楼 dqf88 2015-11-17 08:47发表



谢谢博主，很详细

2楼 暗里着迷GIS 2015-01-12 19:25发表