

阿達的專欄

目錄視圖 摘要視圖 **RSS** 訂閱

個人資料



s030702614

訪問：10755次
積分：519分
排名：第17582名

原創：40篇 轉載：9篇
譯文：0篇 評論：5條

文章搜索

文章分類

[ubuntu](#) (11)
[網絡爬蟲](#) (8)
[mysql](#) (4)
[編程之美](#) (9)
[美麗心情](#) (2)
[音樂之聲](#) (5)

文章存檔

[2012年08月](#) (1)
[2012年02月](#) (1)
[2011年11月](#) (1)
[2011年08月](#) (1)
[2011年07月](#) (1)

展開

閱讀排行

[larbin編譯、運行與配置](#) (1626)
[larbin的詳細配置](#) (773)
[cURL簡單使用、libcurl編](#) (513)
[tcl語言](#) (436)
[ubuntu下NFS安裝與配置](#) (429)
[larbin主要代碼說明](#) (385)
[C語言連接mysql數據庫](#) (300)
[中文編碼之間的轉換](#) (271)
[ftp報文命令](#) (256)
[linux ubuntu下ftp服務器](#) (236)

評論排行

[larbin中input.cc \(input](#) (2)

新版論壇系列介紹之二——功能介紹篇 公告：CSDN博客頻道博客搬家功能上線！ [JavaEE快速開發平台G4Studio](#)作者熊春專訪
中國最大規模移動開發者高水平盛會 沒有重量只有質量：[iPad版《程序員雜誌》應用上線](#) “第一次親密接觸”——有獎徵文活動

cURL簡單使用、libcurl編程

分類：編程之美 [ubuntu](#)

2011-02-21 21:42 513人閱讀 評論 (1) 收藏 舉報

一、curl簡介

curl是一個利用URL語法在命令行方式下工作的文件傳輸工具。它支持的協議有：FTP, FTPS, HTTP, HTTPS, GOPHER, TELNET, DICT, FILE以及LDAP。curl同樣支持HTTPS認證，HTTP POST方法，HTTP PUT方法，FTP上傳，kerberos認證，HTTP上傳，代理服務器，cookies，用戶名/密碼認證，下載文件斷點續傳，上傳文件斷點續傳，http代理服務器管道（proxy tunneling），甚至它還支持IPv6，socks5代理服務器，通過http代理服務器上傳文件到FTP服務器等等，功能十分強大。

二、curl安裝

下載地址：<http://curl.haxx.se/download.html>

ubuntu安裝：sudo apt-get install curl

libcurl靜態庫：./configure & make & install

windows:下載curl.exe，cmd進入curl.exe所在目錄可以直接使用

三、curl使用

curl的用法為：curl [options] [URL...], 其中options是下載需要的參數，大約有80多個，curl的各個功能完全是依靠這些參數完成的。這裡只介紹幾種簡單的用法，詳細的curl的參數在<http://curl.haxx.se/docs/說明>。

1、讀取網頁

curl <http://www.linuxidc.com>

2、保存網頁、下載文件

以page.html命名下載網頁：curl -o page.html <http://www.linuxidc.com>

以服務器上的文件名 命名網頁：curl -O <http://cgi2.tky.3web.ne.jp/~zzh/screen1.JPG>

批量下載文件：curl -O [http://cgi2.tky.3web.ne.jp/~zzh/screen\[1-10\].JPG](http://cgi2.tky.3web.ne.jp/~zzh/screen[1-10].JPG)

3、使用代理服務器及其端口

curl -x 127.0.0.1:8580 -o page.html <http://www.linuxidc.com>

4、使用cookie來記錄session信息

cookie信息存到cookie1.txt中：curl -o page.html -D cookie1.txt <http://www.linuxidc.com>

使用上次的cookie並生成新的cookie：curl -o page.html -D cookie2.txt -b cookie2.txt <http://www.linuxidc.com>

5、斷點續傳

比如下載screen1.JPG中，突然掉線了，可以這樣開始續傳：curl -c -O <http://cgi2.tky.3web.ne.jp/~zzh/screen1.JPG>

另外可以用-r選項進行分塊下載

6、上傳文件

比如我們向ftp傳一個文件：curl -T localfile -u name:passwd ftp://upload_site:port/path/

PS：對於ftp服務器用-u name:passwd選項

7、http提交一個表單GET與POST模式

GET模式什麼option都不用，只需要把變量寫在url裡面就可以了比如：

\$curl <http://www.linuxidc.com/login.cgi?user=nickwolfe&password=12345>

POST模式的選項是-d

比如：curl -d "user=nickwolfe&password=12345" <http://www.linuxidc.com/login.cgi>

四、curl編程

ubuntu搭建論壇	(1)
cURL簡單使用、libcurl編程	(1)
ruijie	(1)
文本抽取函數比正則表達	(0)
I lay my love on you	(0)
larbin中運用數據庫mysql	(0)
用最簡單的方法解決Too	(0)
小函數集合	(0)
larbin程序重啓方案	(0)

推薦文章

- * [JavaEE 快速开发平台 G4Studio](#)
作者熊春专访
- * [Java 泛型之我见 深入研究](#)
- * [POSIX多线程—异步编程举例](#)
- * [Android应用开发中对Bitmap的内存优化](#)
- * [Android 应用程序签名](#)
- * [Cocos2dX\(2.x\)之Lua与自创类间的访问](#)

最新評論

cURL簡單使用、libcurl編程
Pengluer:

ubuntu搭建論壇
luoweifeng1989: 11年的第一天來頂你

larbin中input.cc (input函數) 源
s030702614: 回復匿名用戶：你是想拿它做什麼啊？

larbin中input.cc (input函數) 源
匿名用戶: 我也在研究larbin的源碼，大家一塊學習啊。願與你交流
neu.loner@gmail.com

ruijie
j307533688: 多謝樓主了。。

1、curl編程流程

LibCurl編程流程在基於LibCurl的程序裡，主要採用callback function（回調函數）的形式完成傳輸任務，用戶在啟動傳輸前設置好各類參數和回調函數，當滿足條件時libcurl將調用用戶的回調函數實現特定功能。下面是利用libcurl完成傳輸任務的流程：

- （1）調用curl_global_init()初始化libcurl
- （2）調用curl_easy_init()函數得到easy interface型指針
- （3）調用curl_easy_setopt設置傳輸選項
- （4）根據curl_easy_setopt設置的傳輸選項，實現回調函數以完成用戶特定任務
- （5）調用curl_easy_perform（）函數完成傳輸任務
- （6）調用curl_easy_cleanup（）釋放內存

在整過過程中設置curl_easy_setopt()參數是最關鍵的，幾乎所有的libcurl程序都要使用它。

2、重要函數

- （1）CURLcode curl_global_init(long flags);

描述：

這個函數只能用一次。（其實在調用curl_global_cleanup函數後仍然可再用）

如果這個函數在curl_easy_init函數調用時還沒調用，它講由libcurl庫自動完成。

參數：flags

CURL_GLOBAL_ALL //初始化所有的可能的調用。

CURL_GLOBAL_SSL //初始化支持 安全套接字層。

CURL_GLOBAL_WIN32 //初始化win32套接字庫。

CURL_GLOBAL_NOHING //沒有額外的初始化。

- （2）void curl_global_cleanup(void);

描述：在結束libcurl使用的時候，用來對curl_global_init做的工作清理。類似於close的函數。

- （3）char *curl_version();

描述：打印當前libcurl庫的版本。

- （4）CURL *curl_easy_init();

描述：curl_easy_init用來初始化一個CURL的指針(有些像返回FILE類型的指針一樣).相應的在調用結束時要用curl_easy_cleanup函數清理.

一般curl_easy_init意味著一個會話的開始.它的返回值一般都在easy系列的函數中.

- （5）void curl_easy_cleanup(CURL *handle);

描述：

這個調用用來結束一個會話.與curl_easy_init配合著用.

參數：

CURL類型的指針.

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

描述：這個函數最重要了.幾乎所有的curl程序都要頻繁的使用它.它告訴curl庫.程序將有如何的行為.比如要查看一個網頁的html代碼等.(這個函數有些像ioctl函數)參數：

1 CURL類型的指針

2 各種CURLOption類型的選項(都在curl.h庫裡有定義,man 也可以查看到)

3 parameter這個參數 既可以是個函數的指針,也可以是某個對象的指針,也可以是個long型的變量.它用什麼這取決於第二個參數.

CURLOption這個參數的取值很多.具體的可以查看man手冊.

- （7）CURLcode curl_easy_perform(CURL *handle);描述：這個函數在初始化CURL類型的指針 以及curl_easy_setopt完成後調用.就像字面的意思所說perform就像是個舞台.讓我們設置的

option運作起來.參數：

CURL類型的指針.

3、curl_easy_setopt函數介紹

本節主要介紹curl_easy_setopt中跟http相關的參數。注意本節的闡述都是以libcurl作為主體，其它為客體來闡述的。

1. CURLOPT_URL

設置訪問URL

2. CURLOPT_WRITEFUNCTION, CURLOPT_WRITEDATA

回調函數原型為：size_t function(void *ptr, size_t size, size_t nmemb, void *stream);函數將在libcurl接收到數據後被調用，因此函數多做數據保存的功能，如處理下載文件。CURLOPT_WRITEDATA用於表明CURLOPT_WRITEFUNCTION函數中的stream指針的來源。

3. CURLOPT_HEADERFUNCTION, CURLOPT_HEADERDATA

回調函數原型為size_t function(void *ptr, size_t size, size_t nmemb, void *stream); libcurl一旦接收到http頭部數據後將調用該函數。CURLOPT_WRITEDATA傳遞指針給libcurl，該指針表明CURLOPT_HEADERFUNCTION函數的stream指針的來源。

4. CURLOPT_READFUNCTION CURLOPT_READDATA

libCurl需要讀取數據傳遞給遠程主機時將調用CURLOPT_READFUNCTION指定的函數，函數原型是：size_t function(void *ptr, size_t size, size_t nmemb, void *stream). CURLOPT_READDATA表明CURLOPT_READFUNCTION函數原型中的stream指針來源。

5. CURLOPT_NOPROGRESS, CURLOPT_PROGRESSFUNCTION, CURLOPT_PROGRESSDATA

跟數據傳輸進度相關的參數。CURLOPT_PROGRESSFUNCTION指定的函數正常情況下每秒被libcurl調用一次，為了使CURLOPT_PROGRESSFUNCTION被調用，CURLOPT_NOPROGRESS必須被設置為false，CURLOPT_PROGRESSDATA指定的參數將作為CURLOPT_PROGRESSFUNCTION指定函數的第一個參數

6. CURLOPT_TIMEOUT, CURLOPT_CONNECTIONTIMEOUT:

CURLOPT_TIMEOUT由於設置傳輸時間，CURLOPT_CONNECTIONTIMEOUT設置連接等待時間

7. CURLOPT_FOLLOWLOCATION

設置重定位URL

CURLOPT_RANGE: CURLOPT_RESUME_FROM:

斷點續傳相關設置。CURLOPT_RANGE指定char *參數傳遞給libcurl，用於指明http域的RANGE頭域，例如：

表示頭500個字節：bytes=0-499

表示第二個500字節：bytes=500-999

表示最後500個字節：bytes=-500

表示500字節以後的範圍：bytes=500-

第一個和最後一個字節：bytes=0-0,-1

同時指定幾個範圍：bytes=500-600,601-999

CURLOPT_RESUME_FROM傳遞一個long參數給libcurl，指定你希望開始傳遞的偏移量。

4、curl_easy_perform函數說明（error狀態碼）

該函數完成curl_easy_setopt指定的所有選項，本節重點介紹curl_easy_perform的返回值。返回0意味一切ok，非0代表錯誤發生。主要錯誤碼說明：

1. CURLE_OK

任務完成一切都好

2 CURLE_UNSUPPORTED_PROTOCOL

不支持的協議，由URL的頭部指定

3 CURLE_COULDNT_CONNECT

不能連接到remote主機或者代理

4 CURLE_REMOTE_ACCESS_DENIED

訪問被拒絕

5 CURLE_HTTP_RETURNED_ERROR

Http返回錯誤

6 CURLE_READ_ERROR

讀本地文件錯誤

五、curl編程例子

1、獲取html網頁

```
[cpp]
01. #include <stdio.h>
02.
03. #include <curl/curl.h>
04.
05. #include <stdlib.h>
06.
07. int main( int argc, char *argv[])
08. {
09.
10.
11.     CURL *curl;           //定義CURL類型的指針
12.
13.     CURLcode res;         //定義CURLcode類型的變量，保存返回狀態碼
```

```

14.
15.     if (argc!=2)
16.     {
17.
18.         printf( "Usage : file <url>;/n" );
19.
20.         exit(1);
21.     }
22.
23.     curl = curl_easy_init();           //初始化一個CURL類型的指針
24.
25.     if (curl!=NULL)
26.     {
27.
28.         //設置curl選項. 其中CURLOPT_URL是讓用戶指定url. argv[1]中存放的命令行傳進來的網址
29.
30.         curl_easy_setopt(curl, CURLOPT_URL, argv[1]);
31.
32.         //調用curl_easy_perform 執行我們的設置.並進行相關的操作. 在這裡只在屏幕上顯示出來.
33.
34.         res = curl_easy_perform(curl);
35.
36.         //清除curl操作.
37.
38.         curl_easy_cleanup(curl);
39.
40.     }
41.
42.     return 0;
43. }
44.
45.
46.
47.

```

編譯gcc get_http.c -o get_http -lcurl

./get_http www.baidu.com

2、網頁下載保存



[cpp]

```

01. // 採用CURLOPT_WRITEFUNCTION 實現網頁下載保存功能
02.
03. #include <stdio.h>;
04.
05. #include <stdlib.h>;
06.
07. #include <unistd.h>;
08.
09. #include <curl/curl.h>;
10.
11. #include <curl/types.h>;
12.
13. #include <curl/easy.h>;
14.
15. FILE *fp; //定義FILE類型指針
16.
17. //這個函數是爲了符合CURLOPT_WRITEFUNCTION而構造的
18.
19. //完成數據保存功能
20.
21. size_t write_data( void *ptr, size_t size, size_t nmemb, void *stream)
22. {
23.
24.     int written = fwrite(ptr, size, nmemb, ( FILE *)fp);
25.
26.     return written;
27.
28. }
29.
30. int main( int argc, char *argv[])
31. {
32.
33.     CURL *curl;
34.
35.

```

```

36.
37.     curl_global_init(CURL_GLOBAL_ALL);
38.
39.     curl=curl_easy_init();
40.
41.     curl_easy_setopt(curl, CURLOPT_URL, argv[1]);
42.
43.     if ((fp=fopen(argv[2], "w" ))==NULL)
44.     {
45.
46.         curl_easy_cleanup(curl);
47.
48.         exit(1);
49.
50.     }
51.
52.
53.     ///CURLOPT_WRITEFUNCTION 將後繼的動作交給write_data函數處理
54.
55.     curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_data);
56.
57.     curl_easy_perform(curl);
58.
59.     curl_easy_cleanup(curl);
60.
61.     exit(0);
62.
63. }

```

編譯gcc save_http.c -o save_http -lcurl

./save_http www.baidu.com /tmp/baidu

3、進度條實例（顯示文件下載進度）

```

[cpp]
01. // 採用CURLOPT_NOPROGRESS， CURLOPT_PROGRESSFUNCTION CURLOPT_PROGRESSDATA 實現文件傳輸進度提
    示功能
02.
03. //函數採用了gtk庫，故編譯時需指定gtk庫
04.
05. //函數啓動專門的線程用於顯示gtk 進度條bar
06.
07. #include <stdio.h>
08.
09. #include <gtk/gtk.h>
10.
11. #include <curl/curl.h>
12.
13. #include <curl/types.h> /* new for v7 */
14.
15. #include <curl/easy.h> /* new for v7 */
16.
17. GtkWidget *Bar;
18.
19. ///這個函數是爲了符合CURLOPT_WRITEFUNCTION而構造的
20.
21. //完成數據保存功能
22.
23. size_t my_write_func( void *ptr, size_t size, size_t nmemb, FILE *stream)
24.
25. {
26.
27.     return fwrite(ptr, size, nmemb, stream);
28.
29. }
30.
31. //這個函數是爲了符合CURLOPT_READFUNCTION而構造的
32.
33. //數據上傳時使用
34.
35. size_t my_read_func( void *ptr, size_t size, size_t nmemb, FILE *stream)
36.
37. {
38.
39.     return fread(ptr, size, nmemb, stream);
40.

```

```

41.     }
42.
43.     //這個函數是爲了符合CURLOPT_PROGRESSFUNCTION而構造的
44.
45.     //顯示文件傳輸進度，t代表文件大小，d代表傳輸已經完成部分
46.
47.     int my_progress_func(GtkWidget *bar,
48.
49.         double t, /* dltotal */
50.
51.         double d, /* dlnow */
52.
53.         double ultotal,
54.
55.         double ulnow)
56.
57.     {
58.
59.     /* printf("%d / %d (%g %%)\n", d, t, d*100.0/t);*/
60.
61.         gdk_threads_enter();
62.
63.         gtk_progress_set_value(GTK_PROGRESS(bar), d*100.0/t);
64.
65.         gdk_threads_leave();
66.
67.         return 0;
68.     }
69.
70.
71.
72.
73. void *my_thread( void *ptr)
74.
75. {
76.
77.     CURL *curl;
78.
79.     CURLcode res;
80.
81.     FILE *outfile;
82.
83.     gchar *url = ptr;
84.
85.     curl = curl_easy_init();
86.
87.     if (curl)
88.     {
89.
90.
91.         outfile = fopen( "test.curl" , "w" );
92.
93.         curl_easy_setopt(curl, CURLOPT_URL, url);
94.
95.         curl_easy_setopt(curl, CURLOPT_WRITEDATA, outfile);
96.
97.         curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, my_write_func);
98.
99.         curl_easy_setopt(curl, CURLOPT_READFUNCTION, my_read_func);
100.
101.         curl_easy_setopt(curl, CURLOPT_NOPROGRESS, 0L);
102.
103.         curl_easy_setopt(curl, CURLOPT_PROGRESSFUNCTION, my_progress_func);
104.
105.         curl_easy_setopt(curl, CURLOPT_PROGRESSDATA, Bar);
106.
107.         res = curl_easy_perform(curl);
108.
109.         fclose(outfile);
110.
111.         /* always cleanup */
112.
113.         curl_easy_cleanup(curl);
114.
115.     }
116.
117.     return NULL;
118.
119. }

```

```

120.
121. int main( int argc, char **argv)
122.
123. {
124.
125.     GtkWidget *Window, *Frame, *Frame2;
126.
127.     GtkAdjustment *adj;
128.
129.     /* Must initialize libcurl before any threads are started */
130.
131.     curl_global_init(CURL_GLOBAL_ALL);
132.
133.     /* Init thread */
134.
135.     g_thread_init(NULL);
136.
137.     gtk_init(&argc, &argv);
138.
139.     Window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
140.
141.     Frame = gtk_frame_new(NULL);
142.
143.     gtk_frame_set_shadow_type(GTK_FRAME(Frame), GTK_SHADOW_OUT);
144.
145.     gtk_container_add(GTK_CONTAINER(Window), Frame);
146.
147.     Frame2 = gtk_frame_new(NULL);
148.
149.     gtk_frame_set_shadow_type(GTK_FRAME(Frame2), GTK_SHADOW_IN);
150.
151.     gtk_container_add(GTK_CONTAINER(Frame), Frame2);
152.
153.     gtk_container_set_border_width(GTK_CONTAINER(Frame2), 5);
154.
155.     adj = (GtkAdjustment*)gtk_adjustment_new(0, 0, 100, 0, 0, 0);
156.
157.     Bar = gtk_progress_bar_new_with_adjustment(adj);
158.
159.     gtk_container_add(GTK_CONTAINER(Frame2), Bar);
160.
161.     gtk_widget_show_all(Window);
162.
163.     if (!g_thread_create(&my_thread, argv[1], FALSE, NULL) != 0)
164.
165.         g_warning( "can't create the thread" );
166.
167.     gdk_threads_enter();
168.
169.     gtk_main();
170.
171.     gdk_threads_leave();
172.
173.     return 0;
174.
175. }

```

編譯 `export PKG_CONFIG_PATH=/usr/lib/pkgconfig/`

`gcc progress.c -o progress `pkg-config --libs --cflags gtk+-2.0` -lcurl -lgthread-2.0`

`/ progress http://software.sky-union.cn/index.asp`

4、斷點續傳實例

[cpp]

```

01. //採用CURLOPT_RESUME_FROM_LARGE 實現文件斷點續傳功能
02.
03. #include <stdlib.h>
04.
05. #include <stdio.h>
06.
07. #include <sys/stat.h>
08.
09. #include <curl/curl.h>
10.
11. //這個函數為CURLOPT_HEADERFUNCTION參數構造
12.
13. /* 從http頭部獲取文件size*/

```

```

14.
15. size_t getcontentlengthfunc( void *ptr, size_t size, size_t nmemb, void *stream)
16. {
17.     int r;
18.
19.     long len = 0;
20.
21.     /* _snscanf() is Win32 specific */
22.
23.     // r = _snscanf(ptr, size * nmemb, "Content-Length: %ld/n", &len);
24.
25.     r = sscanf(ptr, "Content-Length: %ld/n", &len);
26.
27.     if (r) /* Microsoft: we don't read the specs */
28.
29.         *(( long *) stream) = len;
30.
31.     return size * nmemb;
32. }
33.
34.
35. /* 保存下載文件 */
36.
37. size_t wirtefunc( void *ptr, size_t size, size_t nmemb, void *stream)
38. {
39.
40.
41.     return fwrite(ptr, size, nmemb, stream);
42.
43. }
44.
45. /*讀取上傳文件 */
46.
47. size_t readfunc( void *ptr, size_t size, size_t nmemb, void *stream)
48. {
49.
50.
51.     FILE *f = stream;
52.
53.     size_t n;
54.
55.     if (ferror(f))
56.
57.         return CURL_READFUNC_ABORT;
58.
59.     n = fread(ptr, size, nmemb, f) * size;
60.
61.     return n;
62. }
63.
64.
65. // 下載 或者上傳文件函數
66.
67. int download(CURL *curlhandle, const char * remotepath, const char * localpath,
68.
69.     long timeout, long tries)
70. {
71.
72.
73.     FILE *f;
74.
75.     curl_off_t local_file_len = -1 ;
76.
77.     long filesize =0 ;
78.
79.     CURLcode r = CURLE_GOT_NOTHING;
80.
81.     int c;
82.
83.     struct stat file_info;
84.
85.     int use_resume = 0;
86.
87.     /* 得到本地文件大小 */
88.
89.     //if(access(localpath,F_OK) ==0)
90.
91.     if (stat(localpath, &file_info) == 0)

```



```

92.
93.     {
94.
95.         local_file_len = file_info.st_size;
96.
97.         use_resume = 1;
98.
99.     }
100.
101.     //採用追加方式打開文件，便於實現文件斷點續傳工作
102.
103.     f = fopen(localpath, "ab+" );
104.
105.     if (f == NULL) {
106.
107.         perror(NULL);
108.
109.         return 0;
110.
111.     }
112.
113.     //curl_easy_setopt(curlhandle, CURLOPT_UPLOAD, 1L);
114.
115.     curl_easy_setopt(curlhandle, CURLOPT_URL, remotepath);
116.
117.     curl_easy_setopt(curlhandle, CURLOPT_CONNECTTIMEOUT, timeout); //設置連接
    超時，單位秒
118.
119.     //設置http 頭部處理函數
120.
121.     curl_easy_setopt(curlhandle, CURLOPT_HEADERFUNCTION, getcontentlengthfunc);
122.
123.     curl_easy_setopt(curlhandle, CURLOPT_HEADERDATA, &filesize);
124.
125.     // 設置文件續傳的位置給libcurl
126.
127.     curl_easy_setopt(curlhandle, CURLOPT_RESUME_FROM_LARGE, use_resume?
    local_file_len:0);
128.
129.     curl_easy_setopt(curlhandle, CURLOPT_WRITEDATA, f);
130.
131.     curl_easy_setopt(curlhandle, CURLOPT_WRITEFUNCTION, wirtefunc);
132.
133.     //curl_easy_setopt(curlhandle, CURLOPT_READFUNCTION, readfunc);
134.
135.     //curl_easy_setopt(curlhandle, CURLOPT_READDATA, f);
136.
137.     curl_easy_setopt(curlhandle, CURLOPT_NOPROGRESS, 1L);
138.
139.     curl_easy_setopt(curlhandle, CURLOPT_VERBOSE, 1L);
140.
141.     r = curl_easy_perform(curlhandle);
142.
143.     fclose(f);
144.
145.     if (r == CURLE_OK)
146.
147.         return 1;
148.
149.     else {
150.
151.         fprintf(stderr, "%s/n" , curl_easy_strerror(r));
152.
153.         return 0;
154.
155.     }
156.
157. }
158.
159. int main( int c, char **argv) {
160.
161.     CURL *curlhandle = NULL;
162.
163.     curl_global_init(CURL_GLOBAL_ALL);
164.
165.     curlhandle = curl_easy_init();
166.
167.     //download(curlhandle, "ftp://user:pass@host/path/file", "C://file", 0, 3);
168.

```

```
169.         download(curlhandle , "http://software.sky-union.cn/index.asp" , "/work/index.asp"
      ,1,3);
170.
171.         curl_easy_cleanup(curlhandle);
172.
173.         curl_global_cleanup();
174.
175.         return 0;
176.
177.     }
```

編譯gcc resume.c -o resume -lcurl

./resume

上一篇：[linux samba詳細配置及使用](#)

分享到：

下一篇：[nothing's gonna change my love for you](#)



[查看評論](#)

1樓PenglueR 2011-04-15 14:55發表



[e01]

您還沒有登錄,請[登錄](#)或[註冊](#)

* 以上用戶言論只代表其個人觀點，不代表CSDN網站的觀點或立場

专区推荐内容

英特尔 至强 服务器上的最佳实践...
Intel MKL函数,如何得到...
跟燕青一起学Windows8应用...
在学习HTML5时，别忘了CSS...
C#多线程参数传递
Android 多线程编程

<< >>

更多招聘职位

我公司职位也要出现在这里

【CIC 济南】诚聘售前工程师、商务经理、项目经理等
【CIC 北京】诚聘高级测试工程师、GIS高级开发工程师等
【澳客网】诚聘web app移动开发工程师、架构师（JS前端方向）
【全景赛斯】诚聘 高级软件工程师
【上海振华】诚聘软件工程师

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帳號](#) | [聯繫方式](#) | [版權聲明](#) | [法律顧問](#) | [問題報告](#)

京 ICP 證 070598 號

北京創新樂知信息技術有限公司版權所有

✉ 聯繫郵箱：webmaster@csdn.net

Copyright © 1999-2012, CSDN.NET, All Rights Reserved

