<https://blog.csdn.net/zhouzhenhe2008/article/details/53768854>

 一、下载

           官网下载地址 ：  https://curl.haxx.se/download.html

           选择最新的一个即可。



二、安装

    1、解压

         下载到的压缩包为curl-7.51.0.tar.gz，使用命令 **tar -zxvf curl-7.51.0.tar.gz**   解压。

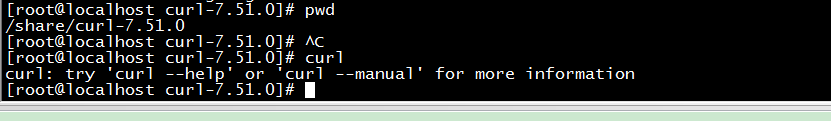
     2、进入解压出的目录curl-7.51.0，执行

**（1）sudo ./configure**

**（2）sudo make**

**（3）sudo make install**

         安装完成，在命令行测试是否可用，直接输入curl，出现如下情况安装基本成功



         如果提示该命令不存在。则执行以下命令：

**export PATH=$PATH:/usr/local/curl/bin**

        检查/usr/include/里有没有crul没有的话，需要把解压出的include文件夹下的文件复制过去。

**cp -r curl-7.51.0/include/curl/ /usr/include/**

         检查库和头文件

**[root@localhost curl-7.51.0]# curl-config --cflags**

**-I/usr/local/include**

**[root@localhost curl-7.51.0]# curl-config --libs**

**-L/usr/local/lib -lcurl**

            以后使用gcc编译使用libcurl的c代码，都需加上**-lcurl。**

三、编译代码

          使用libcurl官网的例子，只稍微修改。

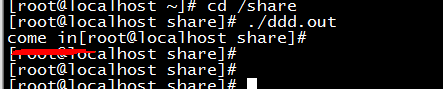
**[cpp]** [view plain](https://blog.csdn.net/zhouzhenhe2008/article/details/53768854) [copy](https://blog.csdn.net/zhouzhenhe2008/article/details/53768854)

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. \*                                  \_   \_ \_\_\_\_  \_
3. \*  Project                     \_\_\_| | | |  \_ \| |
4. \*                             / \_\_| | | | |\_) | |
5. \*                            | (\_\_| |\_| |  \_ <| |\_\_\_
6. \*                             \\_\_\_|\\_\_\_/|\_| \\_\\_\_\_\_\_|
7. \*
8. \* Copyright (C) 1998 - 2015, Daniel Stenberg, <daniel@haxx.se>, et al.
9. \*
10. \* This software is licensed as described in the file COPYING, which
11. \* you should have received as part of this distribution. The terms
12. \* are also available at https://curl.haxx.se/docs/copyright.html.
13. \*
14. \* You may opt to use, copy, modify, merge, publish, distribute and/or sell
15. \* copies of the Software, and permit persons to whom the Software is
16. \* furnished to do so, under the terms of the COPYING file.
17. \*
18. \* This software is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY
19. \* KIND, either express or implied.
20. \*
21. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
23. #include <stdio.h>
25. #include <curl/curl.h>
27. /\* <DESC>
28. \* Get a single file from an FTPS server.
29. \* </DESC>
30. \*/
32. **struct** FtpFile {
33. **const** **char** \*filename;
34. **FILE** \*stream;
35. };
37. **static** **size\_t** my\_fwrite(**void** \*buffer, **size\_t** size, **size\_t** nmemb,
38. **void** \*stream)
39. {
40. **struct** FtpFile \*out=(**struct** FtpFile \*)stream;
41. **if**(out && !out->stream) {
42. /\* open file for writing \*/
43. out->stream=fopen(out->filename, "wb");
44. **if**(!out->stream)
45. **return** -1; /\* failure, can't open file to write \*/
46. }
47. **return** fwrite(buffer, size, nmemb, out->stream);
48. }

51. **int** main(**void**)
52. {
53. CURL \*curl;
54. CURLcode res;
55. **struct** FtpFile ftpfile={
56. "yourfile.bin", /\* name to store the file as if successful \*/
57. NULL
58. };
60. printf("come in");
61. **return** 0;
63. curl\_global\_init(CURL\_GLOBAL\_DEFAULT);
65. curl = curl\_easy\_init();
66. **if**(curl) {
67. /\*
68. \* You better replace the URL with one that works! Note that we use an
69. \* FTP:// URL with standard explicit FTPS. You can also do FTPS:// URLs if
70. \* you want to do the rarer kind of transfers: implicit.
71. \*/
72. curl\_easy\_setopt(curl, CURLOPT\_URL,
73. "ftp://user@server/home/user/file.txt");
74. /\* Define our callback to get called when there's data to be written \*/
75. curl\_easy\_setopt(curl, CURLOPT\_WRITEFUNCTION, my\_fwrite);
76. /\* Set a pointer to our struct to pass to the callback \*/
77. curl\_easy\_setopt(curl, CURLOPT\_WRITEDATA, &ftpfile);
79. /\* We activate SSL and we require it for both control and data \*/
80. curl\_easy\_setopt(curl, CURLOPT\_USE\_SSL, CURLUSESSL\_ALL);
82. /\* Switch on full protocol/debug output \*/
83. curl\_easy\_setopt(curl, CURLOPT\_VERBOSE, 1L);
85. res = curl\_easy\_perform(curl);
87. /\* always cleanup \*/
88. curl\_easy\_cleanup(curl);
90. **if**(CURLE\_OK != res) {
91. /\* we failed \*/
92. fprintf(stderr, "curl told us %d\n", res);
93. }
94. }
96. **if**(ftpfile.stream)
97. fclose(ftpfile.stream); /\* close the local file \*/
99. curl\_global\_cleanup();
101. **return** 0;
102. }

编译：gcc ftpsget.c -o ddd.out -lcurl

执行：

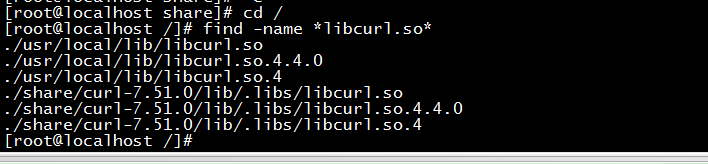


可能会出现的问题：error while loading shared libraries: libcurl.so.4: cannot open shared object file: No such file or directory

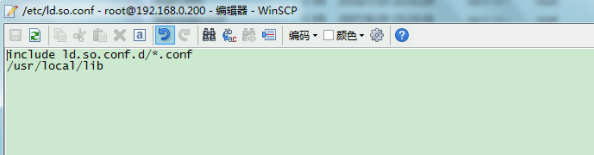
解决方法：

查找 libcurl所有相关的so库所在位置，在/etc/ld.so.conf中加入libcurl有关so库所在目录。

进入根目录/，执行find -name \*libcurl.so\*，就会发现所在目录。



在/etc/ld.so.conf中加入   /usr/local/lib    这一行，



保存/etc/ld.so.conf

执行命令  /sbin/ldconfig -v  生效。