<https://blog.csdn.net/Qsir/article/details/72820207>

前面的《[Windows 各种计时函数总结](http://blog.csdn.net/morewindows/article/details/6854764)》介绍了Windows系统常用的5种计时函数——标准C/C++下的time()及clock()，在Windows系统下的API接口timeGetTime()、GetTickCount()及QueryPerformanceCounter()。下面来介绍下Windows系统中表示时间的两个结构体——FILETIME和SYSTEMTIME及相关函数。

先来看看这两个结构体的定义：

1.     FILETIME

//By MoreWindows-(http://blog.csdn[**.NET**](http://lib.csdn.net/base/dotnet)/MoreWindows)

typedef struct \_FILETIME {

    DWORDdwLowDateTime;

    DWORDdwHighDateTime;

} FILETIME, \*PFILETIME, \*LPFILETIME;

它在MSDN上的说明——Contains a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC时间).

2.     SYSTEMTIME

//By MoreWindows-(http://blog.csdn[**.Net**](http://lib.csdn.net/base/dotnet)/MoreWindows)

typedef struct \_SYSTEMTIME {

    WORDwYear;

    WORDwMonth;

    WORDwDayOfWeek;

    WORDwDay;

    WORDwHour;

    WORDwMinute;

    WORDwSecond;

    WORDwMilliseconds;

} SYSTEMTIME, \*PSYSTEMTIME, \*LPSYSTEMTIME;

这个就不用解释了，和大家平常表示时间的方法一样，都是日期（年-月-日）加时间（小时：分钟：秒）

与这两个结构体相关的函数主有6个——GetSystemTime、GetLocalTime、SystemTimeToFileTime、FileTimeToSystemTime、LocalFileTimeToFileTime、FileTimeToLocalFileTime。下面来看看这些函数的用法：

一．得到当前UTC时间

//By MoreWindows-(http://blog.csdn.net/MoreWindows)

void GetSystemTime(LPSYSTEMTIMElpSystemTime);

二．得到当地时间

void GetLocalTime(LPSYSTEMTIMElpSystemTime);

三．SYSTEMTIME转成FILETIME

//By MoreWindows-(http://blog.csdn.net/MoreWindows)

BOOLSystemTimeToFileTime(

    const SYSTEMTIME\* lpSystemTime,

    LPFILETIMElpFileTime

);

四．FILETIME转成SYSTEMTIME

BOOLFileTimeToSystemTime(

    const FILETIME\* lpFileTime,

    LPSYSTEMTIMElpSystemTime

);

五．当地时间转成UTC时间

//By MoreWindows-(http://blog.csdn.net/MoreWindows)

BOOLLocalFileTimeToFileTime(

    const FILETIME\* lpLocalFileTime,

    LPFILETIMElpFileTime

);

六．UTC时间转成当地时间

BOOLFileTimeToLocalFileTime(

       const FILETIME\* lpFileTime,

       LPFILETIMElpLocalFileTime

);

下面再给出一个示例，这个示例主要使用两个功能：

1．对获取当前当地系统时间

2．打开一个文件，并将文件的创建时间，修改时间，访问时间从FILETIME类型转成SYSTEMTIME类型。

完整代码如下：

**[cpp]** [view plain](http://blog.csdn.net/morewindows/article/details/8654298) [copy](http://blog.csdn.net/morewindows/article/details/8654298)

1. // Windows系统时间(FILETIME和SYSTEMTIME)
2. //By MoreWindows-(http://blog.csdn.net/morewindows/article/details/8654298)
3. #include <windows.h>
4. #include <stdio.h>
5. #include <conio.h>
7. **class** CWindowsDateAndTime
8. {
9. **public**:
10. **static** **void** GetCurrentLocalSystemTime(**char** \*pstrDate, **char** \*pstrTime);
11. **static** **void** FileTimeToLocalSystemTime(FILETIME &ft, **char** \*pstrDate, **char** \*pstrTime);
12. };
14. //得到当前当地时间
15. **void** CWindowsDateAndTime::GetCurrentLocalSystemTime(**char** \*pstrDate, **char** \*pstrTime)
16. {
17. SYSTEMTIME st;
18. GetLocalTime(&st);
20. **if** (pstrDate != NULL)
21. sprintf(pstrDate, "%d-%d-%d", st.wYear, st.wMonth, st.wDay);
22. **if** (pstrTime != NULL)
23. sprintf(pstrTime, "%02d:%02d:%02d", st.wHour, st.wMinute, st.wSecond);
24. }
26. //文件时间转成当地时间
27. **void** CWindowsDateAndTime::FileTimeToLocalSystemTime(FILETIME &ft, **char** \*pstrDate, **char** \*pstrTime)
28. {
29. FILETIME localft;
30. FileTimeToLocalFileTime(&ft, &localft);
31. SYSTEMTIME st;
32. FileTimeToSystemTime(&localft, &st);
34. **if** (pstrDate != NULL)
35. sprintf(pstrDate, "%d-%d-%d", st.wYear, st.wMonth, st.wDay);
36. **if** (pstrTime != NULL)
37. sprintf(pstrTime, "%02d:%02d:%02d", st.wHour, st.wMinute, st.wSecond);
38. }

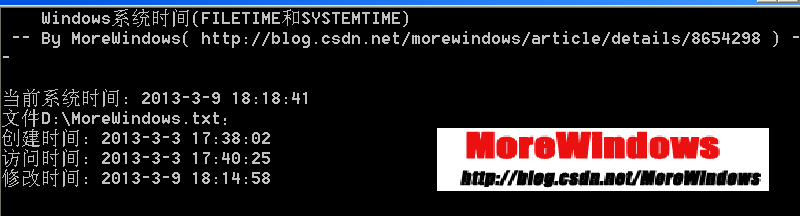
41. **int** main(**int** argc, **char** \*argv[])
42. {
44. printf("    Windows系统时间(FILETIME和SYSTEMTIME) \n");
45. printf(" -- By MoreWindows( http://blog.csdn.net/morewindows/article/details/8654298 ) --\n\n");
47. **const** **int** MAX\_LEN = 30;
48. **char** strDate[MAX\_LEN], strTime[MAX\_LEN];
49. CWindowsDateAndTime::GetCurrentLocalSystemTime(strDate, strTime);
50. printf("当前系统时间: %s %s\n", strDate, strTime);
52. **const** **char**\* pstrFileName = "D:\\MoreWindows.txt";
53. printf("文件%s：\n", pstrFileName);
54. **HANDLE** handleFile = CreateFile(pstrFileName, GENERIC\_READ,
55. FILE\_SHARE\_READ, NULL, OPEN\_EXISTING, 0, NULL);
56. FILETIME ftCreationTime, ftLastAccessTime, ftLastWriteTime;
57. GetFileTime(handleFile, &ftCreationTime, &ftLastAccessTime, &ftLastWriteTime);
58. CWindowsDateAndTime::FileTimeToLocalSystemTime(ftCreationTime, strDate, strTime);
59. printf("创建时间: %s %s\n", strDate, strTime);
60. CWindowsDateAndTime::FileTimeToLocalSystemTime(ftLastAccessTime, strDate, strTime);
61. printf("访问时间: %s %s\n", strDate, strTime);
62. CWindowsDateAndTime::FileTimeToLocalSystemTime(ftLastWriteTime, strDate, strTime);
63. printf("修改时间: %s %s\n", strDate, strTime);
64. getch();
65. **return** 0;
66. }

**[cpp]** [view plain](https://blog.csdn.net/Qsir/article/details/72820207) [copy](https://blog.csdn.net/Qsir/article/details/72820207)

1. // Windows系统时间(FILETIME和SYSTEMTIME)
2. //By MoreWindows-(http://blog.csdn.net/morewindows/article/details/8654298)
3. #include <windows.h>
4. #include <stdio.h>
5. #include <conio.h>
7. **class** CWindowsDateAndTime
8. {
9. **public**:
10. **static** **void** GetCurrentLocalSystemTime(**char** \*pstrDate, **char** \*pstrTime);
11. **static** **void** FileTimeToLocalSystemTime(FILETIME &ft, **char** \*pstrDate, **char** \*pstrTime);
12. };
14. //得到当前当地时间
15. **void** CWindowsDateAndTime::GetCurrentLocalSystemTime(**char** \*pstrDate, **char** \*pstrTime)
16. {
17. SYSTEMTIME st;
18. GetLocalTime(&st);
20. **if** (pstrDate != NULL)
21. sprintf(pstrDate, "%d-%d-%d", st.wYear, st.wMonth, st.wDay);
22. **if** (pstrTime != NULL)
23. sprintf(pstrTime, "%02d:%02d:%02d", st.wHour, st.wMinute, st.wSecond);
24. }
26. //文件时间转成当地时间
27. **void** CWindowsDateAndTime::FileTimeToLocalSystemTime(FILETIME &ft, **char** \*pstrDate, **char** \*pstrTime)
28. {
29. FILETIME localft;
30. FileTimeToLocalFileTime(&ft, &localft);
31. SYSTEMTIME st;
32. FileTimeToSystemTime(&localft, &st);
34. **if** (pstrDate != NULL)
35. sprintf(pstrDate, "%d-%d-%d", st.wYear, st.wMonth, st.wDay);
36. **if** (pstrTime != NULL)
37. sprintf(pstrTime, "%02d:%02d:%02d", st.wHour, st.wMinute, st.wSecond);
38. }

41. **int** main(**int** argc, **char** \*argv[])
42. {
44. printf("    Windows系统时间(FILETIME和SYSTEMTIME) \n");
45. printf(" -- By MoreWindows( http://blog.csdn.net/morewindows/article/details/8654298 ) --\n\n");
47. **const** **int** MAX\_LEN = 30;
48. **char** strDate[MAX\_LEN], strTime[MAX\_LEN];
49. CWindowsDateAndTime::GetCurrentLocalSystemTime(strDate, strTime);
50. printf("当前系统时间: %s %s\n", strDate, strTime);
52. **const** **char**\* pstrFileName = "D:\\MoreWindows.txt";
53. printf("文件%s：\n", pstrFileName);
54. **HANDLE** handleFile = CreateFile(pstrFileName, GENERIC\_READ,
55. FILE\_SHARE\_READ, NULL, OPEN\_EXISTING, 0, NULL);
56. FILETIME ftCreationTime, ftLastAccessTime, ftLastWriteTime;
57. GetFileTime(handleFile, &ftCreationTime, &ftLastAccessTime, &ftLastWriteTime);
58. CWindowsDateAndTime::FileTimeToLocalSystemTime(ftCreationTime, strDate, strTime);
59. printf("创建时间: %s %s\n", strDate, strTime);
60. CWindowsDateAndTime::FileTimeToLocalSystemTime(ftLastAccessTime, strDate, strTime);
61. printf("访问时间: %s %s\n", strDate, strTime);
62. CWindowsDateAndTime::FileTimeToLocalSystemTime(ftLastWriteTime, strDate, strTime);
63. printf("修改时间: %s %s\n", strDate, strTime);
64. getch();
65. **return** 0;
66. }

程序运行结果如下：



转载请标明出处，原文地址：<http://blog.csdn.net/morewindows/article/details/8654298>