<https://blog.csdn.net/dpsying/article/details/19038999>

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/dpsying/article/details/19038999

目录[(?)](https://blog.csdn.net/dpsying/article/details/19038999" \o "系统根据文章中H1到H6标签自动生成文章目录)[[+]](https://blog.csdn.net/dpsying/article/details/19038999)

## 1，原理

CPU资源看做是一个个的时间片，统计CPU使用率也是计算在一段时间内忙碌的时间占比。

我们知道GetSystemTime可以得到当前系统时间，另外一个名字类似的函数，GetSystemTimes可以得到三种不同的时间（自开机以来）：空闲时间,内核时间和用户时间。

函数原型：

**[cpp]** [view plain](https://blog.csdn.net/dpsying/article/details/19038999) [copy](https://blog.csdn.net/dpsying/article/details/19038999)

1. **BOOL** WINAPI GetSystemTimes(
2. \_Out\_opt\_  LPFILETIME lpIdleTime,
3. \_Out\_opt\_  LPFILETIME lpKernelTime,
4. \_Out\_opt\_  LPFILETIME lpUserTime
5. );

CPU要么是在内核态，要么是在用户态。相加就是CPU总时间。

所以有以下公式：

CPU使用率 = （内核时间 + 用户时间 - 空闲时间）/（内核时间 + 用户时间）。

在较短的间隔时间内，先后两次调用GetSystemTimes，然后相减，再使用上面的公式可得出这段时间内的CPU使用率。

## 2，使用

**[cpp]** [view plain](https://blog.csdn.net/dpsying/article/details/19038999) [copy](https://blog.csdn.net/dpsying/article/details/19038999)

1. #include "stdafx.h"
2. #include <Windows.h>
3. #include <iostream>
5. **using**   **namespace**   std;
7. **\_\_int64** CompareFileTime ( FILETIME time1, FILETIME time2 )
8. {
9. **\_\_int64** a = time1.dwHighDateTime << 32 | time1.dwLowDateTime ;
10. **\_\_int64** b = time2.dwHighDateTime << 32 | time2.dwLowDateTime ;
11. **return**   (b - a);
12. }
14. **int** \_tmain(**int** argc, \_TCHAR\* argv[])
15. {
16. **HANDLE** hEvent;
18. FILETIME preidleTime;
19. FILETIME prekernelTime;
20. FILETIME preuserTime;
21. GetSystemTimes( &preidleTime, &prekernelTime, &preuserTime );
23. hEvent = CreateEvent (NULL,FALSE,FALSE,NULL); // 初始值为 nonsignaled ，并且每次触发后自动设置为nonsignaled
24. **while** (1){
26. WaitForSingleObject( hEvent,1000 ); //等待500毫秒
28. FILETIME idleTime;
29. FILETIME kernelTime;
30. FILETIME userTime;
31. GetSystemTimes( &idleTime, &kernelTime, &userTime );
33. **int** idle = CompareFileTime( preidleTime,idleTime);
34. **int** kernel = CompareFileTime( prekernelTime, kernelTime);
35. **int** user = CompareFileTime(preuserTime, userTime);
37. **int** cpu = (kernel +user - idle) \*100/(kernel+user);
38. cout << "CPU利用率:" << cpu << "%" <<endl;
40. preidleTime = idleTime;
41. prekernelTime = kernelTime;
42. preuserTime = userTime ;
43. }
44. **return** 0;
45. }

效果如图：

