* [登录](https://passport.csdn.net/account/login?ref=toolbar)|[注册](http://passport.csdn.net/account/mobileregister?ref=toolbar&action=mobileRegister)

关闭

[**MoreWindows Blog**](http://blog.csdn.net/morewindows)

格物穷理，以求自由！

* [http://static.blog.csdn.net/images/ico_list.gif目录视图](http://blog.csdn.net/morewindows?viewmode=contents)
* [http://static.blog.csdn.net/images/ico_summary.gif摘要视图](http://blog.csdn.net/morewindows?viewmode=list)
* [http://static.blog.csdn.net/images/ico_rss.gif订阅](http://blog.csdn.net/morewindows/rss/list)

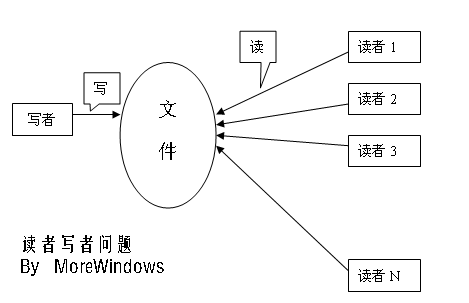
[想听课？来发话题吧](http://bbs.csdn.net/topics/391857175)    [CSDN APP 博客上线](http://www.csdn.net/app/)    [有奖征文：云服务器使用初体验](http://blog.csdn.net/blogdevteam/article/details/49612641)    [有奖试读—增长黑客，创业公司必知的“黑科技”](http://blog.csdn.net/blogdevteam/article/details/49758609)

[秒杀多线程第十一篇 读者写者问题](http://blog.csdn.net/morewindows/article/details/7596034)

分类： [Windows多线程](http://blog.csdn.net/MoreWindows/article/category/1115271) [Windows编程](http://blog.csdn.net/MoreWindows/article/category/862060)2012-05-28 09:44 35352人阅读 [评论](http://blog.csdn.net/morewindows/article/details/7596034#comments)(75) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/morewindows/article/details/7596034#report)

[多线程](http://www.csdn.net/tag/%e5%a4%9a%e7%ba%bf%e7%a8%8b)[null](http://www.csdn.net/tag/null)[list](http://www.csdn.net/tag/list)[output](http://www.csdn.net/tag/output)[c++](http://www.csdn.net/tag/c%2b%2b)[面试](http://www.csdn.net/tag/%e9%9d%a2%e8%af%95)

与上一篇《[秒杀多线程第十篇 生产者消费者问题](http://blog.csdn.net/morewindows/article/details/7577591)》的生产者消费者问题一样，读者写者也是一个非常著名的同步问题。读者写者问题描述非常简单，有一个写者很多读者，多个读者可以同时读文件，但写者在写文件时不允许有读者在读文件，同样有读者在读文件时写者也不去能写文件。



上面是读者写者问题示意图，类似于[生产者消费者问题](http://blog.csdn.net/morewindows/article/details/7577591)的分析过程，首先来找找哪些是属于“等待”情况。

第一．写者要等到没有读者时才能去写文件。

第二．所有读者要等待写者完成写文件后才能去读文件。

找完“等待”情况后，再看看有没有要互斥访问的资源。由于只有一个写者而读者们是可以共享的读文件，所以按题目要求并没有需要互斥访问的资源。类似于上一篇中美观的彩色输出，我们对生产者输出代码进行了颜色设置（在控制台输出颜色设置参见《[VC 控制台颜色设置](http://blog.csdn.net/morewindows/article/details/6789206)》）。因此在这里要加个互斥访问，不然很有可能在写者线程将控制台颜色设置还原之前，读者线程就已经有输出了。所以要对输出语句作个互斥访问处理，修改后的读者及写者的输出函数如下所示：

**[cpp]** [view plaincopy](http://blog.csdn.net/morewindows/article/details/7596034)

1. //读者线程输出函数
2. **void** ReaderPrintf(**char** \*pszFormat, ...)
3. {
4. **va\_list**   pArgList;
5. va\_start(pArgList, pszFormat);
6. EnterCriticalSection(&g\_cs);
7. vfprintf(stdout, pszFormat, pArgList);
8. LeaveCriticalSection(&g\_cs);
9. va\_end(pArgList);
10. }
11. //写者线程输出函数
12. **void** WriterPrintf(**char** \*pszStr)
13. {
14. EnterCriticalSection(&g\_cs);
15. SetConsoleColor(FOREGROUND\_GREEN);
16. printf("     %s\n", pszStr);
17. SetConsoleColor(FOREGROUND\_RED | FOREGROUND\_GREEN | FOREGROUND\_BLUE);
18. LeaveCriticalSection(&g\_cs);
19. }

读者线程输出函数所使用的可变参数详见《[C,C++中使用可变参数](http://blog.csdn.net/morewindows/article/details/6707662)》。

       解决了互斥输出问题，接下来再考虑如何实现同步问题。可以设置一个变量来记录正在读文件的读者个数，第一个开始读文件的读者要负责将关闭允许写者进入的标志，最后一个结束读文件的读者要负责打开允许写者进入的标志。这样第一种“等待”情况就解决了。第二种“等待”情况是有写者进入时所以读者不能进入，使用一个事件就可以完成这个任务了——所有读者都要等待这个事件而写者负责触发事件和设置事件为未触发。详细见代码中注释：

**[cpp]** [view plaincopy](http://blog.csdn.net/morewindows/article/details/7596034)

1. //读者与写者问题
2. #include <stdio.h>
3. #include <process.h>
4. #include <windows.h>
5. //设置控制台输出颜色
6. **BOOL** SetConsoleColor(**WORD** wAttributes)
7. {
8. **HANDLE** hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);
9. **if** (hConsole == INVALID\_HANDLE\_VALUE)
10. **return** FALSE;
12. **return** SetConsoleTextAttribute(hConsole, wAttributes);
13. }
14. **const** **int** READER\_NUM = 5;  //读者个数
15. //关键段和事件
16. CRITICAL\_SECTION g\_cs, g\_cs\_writer\_count;
17. **HANDLE** g\_hEventWriter, g\_hEventNoReader;
18. **int** g\_nReaderCount;
19. //读者线程输出函数(变参函数的实现)
20. **void** ReaderPrintf(**char** \*pszFormat, ...)
21. {
22. **va\_list**   pArgList;
24. va\_start(pArgList, pszFormat);
25. EnterCriticalSection(&g\_cs);
26. vfprintf(stdout, pszFormat, pArgList);
27. LeaveCriticalSection(&g\_cs);
28. va\_end(pArgList);
29. }
30. //读者线程函数
31. unsigned **int** \_\_stdcall ReaderThreadFun(**PVOID** pM)
32. {
33. ReaderPrintf("     编号为%d的读者进入等待中...\n", GetCurrentThreadId());
34. //等待写者完成
35. WaitForSingleObject(g\_hEventWriter, INFINITE);
37. //读者个数增加
38. EnterCriticalSection(&g\_cs\_writer\_count);
39. g\_nReaderCount++;
40. **if** (g\_nReaderCount == 1)
41. ResetEvent(g\_hEventNoReader);
42. LeaveCriticalSection(&g\_cs\_writer\_count);
44. //读取文件
45. ReaderPrintf("编号为%d的读者开始读取文件...\n", GetCurrentThreadId());
47. Sleep(rand() % 100);
49. //结束阅读,读者个数减小,空位增加
50. ReaderPrintf(" 编号为%d的读者结束读取文件\n", GetCurrentThreadId());
52. //读者个数减少
53. EnterCriticalSection(&g\_cs\_writer\_count);
54. g\_nReaderCount--;
55. **if** (g\_nReaderCount == 0)
56. SetEvent(g\_hEventNoReader);
57. LeaveCriticalSection(&g\_cs\_writer\_count);
59. **return** 0;
60. }
61. //写者线程输出函数
62. **void** WriterPrintf(**char** \*pszStr)
63. {
64. EnterCriticalSection(&g\_cs);
65. SetConsoleColor(FOREGROUND\_GREEN);
66. printf("     %s\n", pszStr);
67. SetConsoleColor(FOREGROUND\_RED | FOREGROUND\_GREEN | FOREGROUND\_BLUE);
68. LeaveCriticalSection(&g\_cs);
69. }
70. //写者线程函数
71. unsigned **int** \_\_stdcall WriterThreadFun(**PVOID** pM)
72. {
73. WriterPrintf("写者线程进入等待中...");
74. //等待读文件的读者为零
75. WaitForSingleObject(g\_hEventNoReader, INFINITE);
76. //标记写者正在写文件
77. ResetEvent(g\_hEventWriter);
79. //写文件
80. WriterPrintf("  写者开始写文件.....");
81. Sleep(rand() % 100);
82. WriterPrintf("  写者结束写文件");
84. //标记写者结束写文件
85. SetEvent(g\_hEventWriter);
86. **return** 0;
87. }
88. **int** main()
89. {
90. printf("  读者写者问题\n");
91. printf(" -- by MoreWindows( http://blog.csdn.net/MoreWindows ) --\n\n");
93. //初始化事件和信号量
94. InitializeCriticalSection(&g\_cs);
95. InitializeCriticalSection(&g\_cs\_writer\_count);
97. //手动置位,初始已触发
98. g\_hEventWriter = CreateEvent(NULL, TRUE, TRUE, NULL);
99. g\_hEventNoReader  = CreateEvent(NULL, FALSE, TRUE, NULL);
100. g\_nReaderCount = 0;
102. **int** i;
103. **HANDLE** hThread[READER\_NUM + 1];
104. //先启动二个读者线程
105. **for** (i = 1; i <= 2; i++)
106. hThread[i] = (**HANDLE**)\_beginthreadex(NULL, 0, ReaderThreadFun, NULL, 0, NULL);
107. //启动写者线程
108. hThread[0] = (**HANDLE**)\_beginthreadex(NULL, 0, WriterThreadFun, NULL, 0, NULL);
109. Sleep(50);
110. //最后启动其它读者结程
111. **for** ( ; i <= READER\_NUM; i++)
112. hThread[i] = (**HANDLE**)\_beginthreadex(NULL, 0, ReaderThreadFun, NULL, 0, NULL);
113. WaitForMultipleObjects(READER\_NUM + 1, hThread, TRUE, INFINITE);
114. **for** (i = 0; i < READER\_NUM + 1; i++)
115. CloseHandle(hThread[i]);
117. //销毁事件和信号量
118. CloseHandle(g\_hEventWriter);
119. CloseHandle(g\_hEventNoReader);
120. DeleteCriticalSection(&g\_cs);
121. DeleteCriticalSection(&g\_cs\_writer\_count);
122. **return** 0;
123. }

运行结果如下所示：



根据结果可以看出当有读者在读文件时，写者线程会进入等待状态中。当写者线程在写文件时，读者线程也会排队等待，说明读者和写者已经完成了同步。

本系列通过[经典线程同步问题](http://blog.csdn.net/morewindows/article/details/7442333)来列举线程同步手段的[关键段](http://blog.csdn.net/morewindows/article/details/7442639)、[事件](http://blog.csdn.net/morewindows/article/details/7445233)、[互斥量](http://blog.csdn.net/morewindows/article/details/7470936)、[信号量](http://blog.csdn.net/morewindows/article/details/7481609)，并作对这四种方法进行了[总结](http://blog.csdn.net/morewindows/article/details/7538247)。然后又通过二个著名的线程同步实例——[生产者消费者问题](http://blog.csdn.net/morewindows/article/details/7577591)和[读者写者问题](http://blog.csdn.net/morewindows/article/details/7596034)来强化对多线程同步互斥的理解与运用。希望读者们能够熟练掌握，从而在笔试面试中能够顺利的“秒杀”多线程的相关试题，获得自己满意的offer。

从《[秒杀多线程第十篇](http://blog.csdn.net/morewindows/article/details/7577591)生产者消费者问题》到《[秒杀多线程第十一篇](http://blog.csdn.net/morewindows/article/details/7596034)读者写者问题》可以得出多线程问题的关键在于找到所有“等待”情况和判断有无需要互斥访问的资源。那么如何从实际问题中更好更快更全面的找出这些了？请看《[秒杀多线程第十二篇多线程同步内功心法——PV操作上](http://blog.csdn.net/morewindows/article/details/7650470)》和《秒杀多线程第十三篇多线程同步内功心法——PV操作下》这二篇以加强解决多线程同步问题的“内功”。

另外，读者写者问题可以用读写锁SRWLock来解决，请看《[秒杀多线程第十四篇 读者写者问题继 读写锁SRWLock](http://blog.csdn.net/morewindows/article/details/7650574)》

转载请标明出处，原文地址：<http://blog.csdn.net/morewindows/article/details/7596034>

版权声明：本文为博主原创文章，未经博主允许不得转载。

* 上一篇[秒杀多线程第十篇 生产者消费者问题](http://blog.csdn.net/morewindows/article/details/7577591)
* 下一篇[秒杀多线程第十二篇 多线程同步内功心法——PV操作上](http://blog.csdn.net/morewindows/article/details/7650470)

顶

61

踩

0

主题推荐

[函数](http://www.csdn.net/tag/%E5%87%BD%E6%95%B0)[多线程](http://www.csdn.net/tag/%E5%A4%9A%E7%BA%BF%E7%A8%8B)[控制台](http://www.csdn.net/tag/%E6%8E%A7%E5%88%B6%E5%8F%B0)[线程](http://www.csdn.net/tag/%E7%BA%BF%E7%A8%8B)[classclass](http://www.csdn.net/tag/class)[pre](http://www.csdn.net/tag/pre)[cpp](http://www.csdn.net/tag/cpp)[91](http://www.csdn.net/tag/91)[vc](http://www.csdn.net/tag/vc)

猜你在找

[秒杀多线程第十篇 读者写者问题](http://blog.csdn.net/chenglinhust/article/details/8683143)

[秒杀多线程第十四篇 读者写者问题继 读写锁SRWLock](http://blog.csdn.net/JXH_123/article/details/23450225)

[秒杀多线程第十四篇 读者写者问题继 读写锁SRWLock](http://blog.csdn.net/fkfang/article/details/17342053)

[秒杀多线程第十四篇 读者写者问题继 读写锁SRWLock](http://blog.csdn.net/guoduhua/article/details/9095161)

[秒杀多线程第十四篇 读者写者问题继 读写锁SRWLock](http://blog.csdn.net/jxhgh/article/details/43761043)

**查看评论**

37楼 [aitp2011](http://blog.csdn.net/aitp2011) 2015-08-11 11:56发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/aitp2011)

感谢lz无私分享，看了帖子跟回复，理下自己的思路：  
lz的基本思路是，用关键段维持读者数目增减的互斥操作，用两个Event表示文中的两种“等待情况”，刚开始都是触发的，当WaiteFor成功后，Reset另一个Event，让自己的互斥者WaiteFor阻塞；  
14楼跟23楼提出的bug是，在自己WaiteFor成功，但还没来得及Reset另一个Event时，自己的互斥者也WaiteFor成功了，简单说，这个bug还是因为WaiteFor与Reset之间不是原子操作；  
23、27、28楼还提出了优先级的解决办法，如果是写优先，即把g\_hEventNoReader初始置为触发态，g\_hEventWriter 置为为触发态，因为只有一个写线程，所有的读线程都等着它完成，这个不会有啥问题；但如果是读优先，在写线程ResetEvent(g\_hEventWriter)之前，读线程WaiteFor成功，那么问题又来了；  
对等优先级的读者写者问题咋办？  
36楼给了条链接，测了下代码，也是不行，  
还是用读写锁吧

Re: [aitp2011](http://blog.csdn.net/aitp2011) 2015-08-11 12:38发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/aitp2011)

回复aitp2011：研究了下36楼的代码，互斥锁不能解决问题的原因在于它的线程拥有权问题；  
当第一个读线程WaiteFor成功后就有了锁的拥有权，其他读线程无需WaiteFor；  
但是当第一个读线程退出时，若还有其他读线程，按照代码逻辑不会调用ReleaseMutex；  
但实际上，该锁已经释放（锁的拥有权线程退出），写线程可以得到锁，而此刻其他读线程还在读，问题来了。

36楼 [FENGYEJINGXIANG](http://blog.csdn.net/FENGYEJINGXIANG) 2015-07-21 23:17发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/FENGYEJINGXIANG)

32楼的 hEvent\_writer改为hMutex(互斥量)就容易理解得多  
  
使用互斥量, 即锁，来表示文件的所有权即可解决这个问题。  
  
不管读者还是写者，在处理文件前都要先获取这把锁。读者只要第一个人获取这把锁就可以了。只有有锁的才可以处理文件。由于锁不是归读者就是归写者，所以可以保证读者和写者不交错。  
  
当第一个读者想读取文件时，先测试这把锁，如果写者占用了锁，则等待。  
  
当读者数量归0时，释放锁。这样，接下来读者和写者都可以公平竞争这把锁。  
  
当写者试图写文件时，也先测试这把锁，如果读者占用了锁，等待。  
  
当写者写完文件后，释放锁。这样，接下来读者和写者又都可以公平竞争这把锁。  
  
引自：http://www.cnblogs.com/dplearning/p/4052080.html

35楼 [cychocolate](http://blog.csdn.net/cychocolate) 2014-11-20 09:03发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/cychocolate)

32楼的伪代码解决了LZ不能锁住同时读写的BUG.

34楼 [cychocolate](http://blog.csdn.net/cychocolate) 2014-11-20 09:02发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/cychocolate)

第32楼是正解。

Re: [szszszcw](http://blog.csdn.net/szszszcw) 2015-05-07 17:14发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/szszszcw)

回复cychocolate：不太理解32楼的伪代码。因为此代码一执行wait\_for后线程不就阻塞了吗？不管是reader还是writer都是这样。求教~

33楼 [ch2leo](http://blog.csdn.net/kdok123) 2014-09-28 16:18发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/kdok123)

博主您好!  
請問在您的例子裡如果把Critical section都拿掉會怎樣嗎?  
我試過都拿掉(因為我不知道為什麼要加)  
發現出來的結果也是題目想要的結果  
不知道這裡Critical section的作用是什麼呢??

32楼 [黑面瘪三](http://blog.csdn.net/wei_MM) 2014-03-07 22:43发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/wei_MM)

void ReadThread()  
{  
lock();  
g\_reader++;  
if(g\_reader == 1)  
wait\_for(hEvent\_writer);  
unlock();  
  
// do something  
  
lock();  
--g\_reader;  
if (g\_reader == 0)  
{  
set\_event(hEvent\_writer);  
}  
unlock();  
}  
  
void writerThread()  
{  
wait\_for(hEvent\_writer);  
  
// do somting  
  
set\_event(hEvent\_writer);  
}  
  
我这里是伪代码

31楼 [Corner](http://blog.csdn.net/Corner514) 2013-08-19 11:09发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/B/B/4/3_corner514.jpg](http://blog.csdn.net/Corner514)

谢谢您的讲解，但是有一个问题不太明白，如果读者线程全部结束了，而写者（多个写者或者写多次）还在执行，那么没有读者触发事件，写者是不是就无法执行了？还请指点，谢谢了。

30楼 [wodownload2](http://blog.csdn.net/wodownload2) 2013-05-29 15:47发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/wodownload2)

读14楼帖子的一些体会：  
当写者线程设置g\_hEventWriter以标记写者结束时，如果此时线程被读者抢到，假如进入到了读者代码的：WaitForSingleObject，读者在执行了EnterCriticalSection并且对g\_nReaderCount++之后，写线程又抢到了执行的机会，因为在读者没有执行ResetEvent(g\_hEventNoReader)之前，也就是说写者的WaitForSingleObject仍然能够触发，然后执行写入操作，当然再假设写者还没有写完，线程又转到了读者，读者此时执行RestEvent操作，标记写者不能工作，但是我们知道此时写者仍然在写，因为它已经获得了g\_hEventNoReader了，这样读者在读，而写者在写，完全木有互斥呀！！！！这是本文的一个bug所在。

Re: [彼岸花](http://blog.csdn.net/touchthefuturer) 2014-09-28 09:59发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/touchthefuturer)

回复wodownload2：g\_hEventNoReader创建的时候设置为未触发就好了吧。。。

Re: [StrokeAce](http://blog.csdn.net/gaga392464782) 2014-09-16 17:32发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/gaga392464782)

回复wodownload2：你说的是有多个写线程的情况吗

Re: [ohmygodohyeah](http://blog.csdn.net/ohmygodohyeah) 2013-08-13 23:39发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/ohmygodohyeah)

回复wodownload2：谢谢讲解，这下懂了。

29楼 [超然\_烟火](http://blog.csdn.net/sun2043430) 2013-04-12 10:16发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/sun2043430)

感谢LZ辛苦的码字，不过各位读者要注意，本文的代码并不能保证读写是互斥的，原因参看14楼。14楼 Little\_H 2012-06-15 17:56发表 [回复] [引用] [举报]您好，读了您的文章我受益匪浅，本篇文章有一处bug如下[cpp] view plaincopyWaitForSingleObject(g\_hEventWriter, INFINITE); //读者个数增加 EnterCriticalSection(&g\_cs\_writer\_enter); g\_nReaderCount++; if (g\_nReaderCount == 1) ResetEvent(g\_hEventNoReader); LeaveCriticalSection(&g\_cs\_writer\_enter);

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2013-04-12 12:16发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复超然\_烟火：对的，这个地方会有点问题，因此规范的做法应该使用《秒杀多线程第十四篇 读者写者问题继 读写锁SRWLock》http://blog.csdn.net/morewindows/article/details/7650574

28楼 [ch\_andy](http://blog.csdn.net/ch_andy) 2013-04-09 21:40发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/ch_andy)

发现  
g\_hEventWriter = CreateEvent(NULL, TRUE, TRUE, NULL);   
可改为  
g\_hEventWriter = CreateEvent(NULL, TRUE, FALSE, NULL);   
那么写者线程不必放在读者线程后面了，是不是？

27楼 [Unix\_Architect](http://blog.csdn.net/Unix_Architect) 2013-02-15 02:02发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/Unix_Architect)

**[cpp]** [view plaincopy](http://blog.csdn.net/morewindows/article/details/7596034)

1. g\_hEvent = CreateEvent(NULL,FALSE,TRUE,NULL);
2. g\_hReader = CreateEvent(NULL,FALSE,FALSE,NULL);
3. **DWORD** WINAPI WriterThread(**LPVOID** lpParamter)
4. {
5. WaitForSingleObject(g\_hEvent,INFINITE);
6. SetEvent(g\_hEvent);
7. **return** 0;
8. }
10. **DWORD** WINAPI ReaderThread(**LPVOID** lpParamter)
11. {
12. **bool** first = **false**;
13. EnterCriticalSection(&g\_cs);
14. g\_ReaderCount++;
15. **if** ( g\_ReaderCount == 1 )
16. {
17. first = **true**;
18. }
19. LeaveCriticalSection(&g\_cs);
21. **if** ( first  )
22. {
23. WaitForSingleObject(g\_hEvent,INFINITE);
24. SetEvent(g\_hReader);
25. }
26. **else**
27. {
28. WaitForSingleObject(g\_hReader,INFINITE);
29. SetEvent(g\_hReader);
30. }
31. EnterCriticalSection(&g\_cs);
32. g\_ReaderCount--;
33. **if** ( g\_ReaderCount == 0 )
34. {
35. SetEvent(g\_hEvent);
36. ResetEvent(g\_hReader);
37. }
38. LeaveCriticalSection(&g\_cs);
39. **return** 0;
40. }

26楼 [Unix\_Architect](http://blog.csdn.net/Unix_Architect) 2013-02-15 02:02发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/Unix_Architect)

g\_hEvent = CreateEvent(NULL,FALSE,TRUE,NULL);  
g\_hReader = CreateEvent(NULL,FALSE,FALSE,NULL);  
DWORD WINAPI WriterThread(LPVOID lpParamter)  
{  
WaitForSingleObject(g\_hEvent,INFINITE);  
SetEvent(g\_hEvent);  
return 0;  
}  
  
DWORD WINAPI ReaderThread(LPVOID lpParamter)  
{  
bool first = false;  
EnterCriticalSection(&g\_cs);  
g\_ReaderCount++;  
if ( g\_ReaderCount == 1 )  
{  
first = true;  
}  
LeaveCriticalSection(&g\_cs);  
  
if ( first )  
{  
WaitForSingleObject(g\_hEvent,INFINITE);  
SetEvent(g\_hReader);  
}  
else  
{  
WaitForSingleObject(g\_hReader,INFINITE);  
SetEvent(g\_hReader);  
}  
EnterCriticalSection(&g\_cs);  
g\_ReaderCount--;  
if ( g\_ReaderCount == 0 )  
{  
SetEvent(g\_hEvent);  
ResetEvent(g\_hReader);  
}  
LeaveCriticalSection(&g\_cs);  
return 0;  
}  
这个方法好使吗？

25楼 [Unix\_Architect](http://blog.csdn.net/Unix_Architect) 2013-02-04 13:04发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/Unix_Architect)

我想问问，图中的有箭头的直线是控制流呢，还是数据流呢？

24楼 [DayThinking](http://blog.csdn.net/sszgg2006) 2013-01-16 16:53发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/sszgg2006)

感觉楼主的讲解非常的细致，之前也接触到这些东西，但是看完楼主的杰作后对这些的理解更加深入，带着复习的心态阅读了一篇，感觉还是收获蛮大的，特别是网友跟你的互动，很有价值。感谢勤劳的楼主，或许看的过于过瘾，期待楼主将进程间的几种通信机制也细细的给我们讲讲吧，期待啊……

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2013-03-03 09:37发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复DayThinking：已发表命名管道，欢迎浏览。http://blog.csdn.net/morewindows/article/details/8260087

23楼 [CheckMate](http://blog.csdn.net/ggfaith) 2012-12-14 21:31发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/ggfaith)

楼主您好，我有个疑问。  
假设  
WaitForSingleObject(g\_hEventWriter, INFINITE);   
执行完后，立刻切换到写者线程，这时g\_hEventNoReader还是触发态，于是会执行  
WaitForSingleObject(g\_hEventNoReader, INFINITE);   
这样读者读的时候写者也在写了。  
这个bug怎么解决？   
假设多写者多读者加上要写者优先的话又该怎么写呢，是不是只能用信号量了？

Re: [CaptianSlow](http://blog.csdn.net/terminatorqi) 2014-02-12 16:50发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/terminatorqi)

回复CheckMate：这里应该并没有切换到写者线程，只是此读者线程阻塞了。

22楼 [coFinder](http://blog.csdn.net/lilien1010) 2012-10-26 15:07发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/A/D/3/3_lilien1010.jpg](http://blog.csdn.net/lilien1010)

**[cpp]** [view plaincopy](http://blog.csdn.net/morewindows/article/details/7596034)

1. //写者线程函数
2. unsigned **int** \_\_stdcall WriterThreadFun(**PVOID** pM)
3. {
4. WriterPrintf("作者 %d 准备写入···\n",GetCurrentThreadId() );
5. WaitForSingleObject(g\_hEventNoReader,INFINITE);
6. ResetEvent(g\_hEventWrited);
7. Sleep( 30);
8. WriterPrintf("作者 %d 正在写书---\n",GetCurrentThreadId() );
9. Sleep( 200);
10. WriterPrintf("作者 %d 写入完毕---\n",GetCurrentThreadId() );
11. SetEvent(g\_hEventWrited);
12. **return** 0;
13. }

这样的写法为什么有两个写入的线程还可以，3 4个写入的线程就不行了呢，

21楼 [bsnry](http://blog.csdn.net/bsnry) 2012-09-18 09:42发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/3/E/4/3_bsnry.jpg](http://blog.csdn.net/bsnry)

博主，你好，你还在吗  
  
有个疑问，关于计数器，这么多人，难道没有看出来吗？  
  
//读者个数增加  
EnterCriticalSection(&g\_cs\_writer\_count);  
g\_nReaderCount++;  
if (g\_nReaderCount >= 1) //你的是==1  
ResetEvent(g\_hEventNoReader);  
LeaveCriticalSection(&g\_cs\_writer\_count);  
  
不应该是>=1吗？  
  
问题2： 请问博文代码是不是最新版的?  
  
谢谢回答.

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-09-19 22:12发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复bsnry：老兄，你要仔细看下代码。  
第一个读者线程执行ResetEvent(g\_hEventNoReader);表示现在已经有读者在文件了，写者须要等待。再进来的读者不须要再去执行ResetEvent(g\_hEventNoReader);了。

20楼 [stevenfute](http://blog.csdn.net/stevenfute) 2012-09-06 09:59发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/stevenfute)

**[cpp]** [view plaincopy](http://blog.csdn.net/morewindows/article/details/7596034)

1. WaitForSingleObject(g\_hEventNoReader, INFINITE);
2. //标记写者正在写文件
3. ResetEvent(g\_hEventWriter);

貌似这里应该是先ResetEvent后WaitForSingleObject要不然中间可能有读者进去，不知对否

Re: [sy5837387](http://blog.csdn.net/sy5837387) 2012-09-13 19:14发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/2/A/B/3_sy5837387.jpg](http://blog.csdn.net/sy5837387)

回复stevenfute：这里不是最后一个读者出来之后才会触发然后执行到Reset么?既然是最后一个读者已经读完了，应该没有其他读者进去了吧，个人理解~

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-09-14 10:44发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复sy5837387：等待函数中好像有个可以以原子的方式执行这2步。  
74. //等待读文件的读者为零   
75. WaitForSingleObject(g\_hEventNoReader, INFINITE);   
76. //标记写者正在写文件   
77. ResetEvent(g\_hEventWriter);   
我回学校后查下看看。

Re: [stevenfute](http://blog.csdn.net/stevenfute) 2012-09-15 16:30发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/stevenfute)

回复MoreWindows：就是最后一个读者出来后，如果不是原子操作，这时写的还没锁住，读者又进去了。。。  
我查了win32多线程编程里面好像没有这个的原子操作，为了防止调整后出现死锁，我加了个critical section好像可以解决，还是期待楼主大牛给力

19楼 [小强\_加油](http://blog.csdn.net/aini201) 2012-08-21 17:15发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/aini201)

真是太感谢楼主了，看到这个真是有一种豁然开朗的感觉啊。

18楼 [Emiyasstar\_\_](http://blog.csdn.net/Emiyasstar__) 2012-07-18 16:03发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/6/8/8/3_emiyasstar__.jpg](http://blog.csdn.net/Emiyasstar__)

为什么不用条件变量呢，个人觉得核心编程里用条件变量配合读写锁的例子很经典

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-07-18 19:42发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复Emiyasstar\_\_：后面有读写锁SRWLock来解决，请看http://blog.csdn.net/morewindows/article/details/7650574。像条件变量，读写锁这些API对系统要求高，个人估计在实际开发中会用的少，所以基本上就没去写文章了。

17楼 [shizhouhu](http://blog.csdn.net/szh2048) 2012-07-12 17:26发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/szh2048)

14楼提的问题楼主怎么解决的啊。期待。。。

16楼 [yah606060](http://blog.csdn.net/yah606060) 2012-07-03 17:45发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/yah606060)

37. //读者个数增加   
38. EnterCriticalSection(&g\_cs\_writer\_enter);   
39. g\_nReaderCount++;   
40. if (g\_nReaderCount == 1)   
41. ResetEvent(g\_hEventNoReader);   
42. LeaveCriticalSection(&g\_cs\_writer\_enter);   
52. //读者个数减少   
53. EnterCriticalSection(&g\_cs\_writer\_leave);   
54. g\_nReaderCount--;   
55. if (g\_nReaderCount == 0)   
56. SetEvent(g\_hEventNoReader);   
57. LeaveCriticalSection(&g\_cs\_writer\_leave);   
这里为什么要用两个不同的临界区结构对象啊？

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-07-04 09:55发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复yah606060：已经改好了,谢谢指正错误.

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-07-04 09:50发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复yah606060：可以参考9楼horsewww评论，读者个数增加和减少显然也要是原子操作，不然会造成计数不准，所以就分别加上关键段来保证其执行的完整性。  
不过确实应该只用一个关键段，这个关键段就是用来确保g\_nReaderCount被互斥的修改与访问。文章中代码马上就改，谢谢指出。

15楼 [yah606060](http://blog.csdn.net/yah606060) 2012-07-02 00:18发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/yah606060)

为什么这里要是用互斥啊。。？  
24. va\_start(pArgList, pszFormat);   
25. EnterCriticalSection(&g\_cs);   
26. vfprintf(stdout, pszFormat, pArgList);   
27. LeaveCriticalSection(&g\_cs);   
28. va\_end(pArgList);

Re: [yah606060](http://blog.csdn.net/yah606060) 2012-07-03 17:35发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/yah606060)

37. //读者个数增加   
38. EnterCriticalSection(&g\_cs\_writer\_enter);   
39. g\_nReaderCount++;   
40. if (g\_nReaderCount == 1)   
41. ResetEvent(g\_hEventNoReader);   
42. LeaveCriticalSection(&g\_cs\_writer\_enter);   
52. //读者个数减少   
53. EnterCriticalSection(&g\_cs\_writer\_leave);   
54. g\_nReaderCount--;   
55. if (g\_nReaderCount == 0)   
56. SetEvent(g\_hEventNoReader);   
57. LeaveCriticalSection(&g\_cs\_writer\_leave);   
这里为什么要用两个不同的临界区结构对象啊？

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-07-02 09:26发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复yah606060：输出屏幕其实也是一个需要互斥访问的资源。

Re: [yah606060](http://blog.csdn.net/yah606060) 2012-07-02 11:36发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/yah606060)

回复MoreWindows：printf()内部其实也是这么做的？

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-07-02 12:58发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复yah606060：printf内部也使用了va\_start(),vfprintf()和va\_end()，但不需要作同步互斥处理。这个同步互斥处理操作由使用库函数的人去考虑，而不用printf的编写者来考虑。

Re: [yah606060](http://blog.csdn.net/yah606060) 2012-07-03 11:00发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/yah606060)

回复MoreWindows：不是太明白啊， printf()内部不需要做这个操作？ 能不能讲的稍微详细点哈。。 亲

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-07-03 13:10发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复yah606060：printf作为库函数，它的功能是将数据输出到控制台窗口（不考虑重定向问题），因此在printf内部不需要考虑多线程这一情况。而使用printf的程序员要根据程序运行的实际环境来决定是否需要加同步互斥处理。  
如一个单线程的程序就不用考虑对printf进行同步互斥处理，在本例中，读者写者会对同一个控制台窗口进行输出，所以就需要作互斥处理。

Re: [yah606060](http://blog.csdn.net/yah606060) 2012-07-03 14:36发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/yah606060)

回复MoreWindows：嗯，， 这下清楚了。。 这个也是我们在设计函数接口时也需要考虑的。。

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-07-03 15:38发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复yah606060：是的，设计库函数要考虑的东西比较多，有时为了效率也会牺牲其它特性，如库函数strcpy就不对指针是否为空进行检查。

14楼 [Little\_H](http://blog.csdn.net/hy_jones) 2012-06-15 17:56发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/1/4/3/3_hy_jones.jpg](http://blog.csdn.net/hy_jones)

您好，读了您的文章我受益匪浅，本篇文章有一处bug如下

**[cpp]** [view plaincopy](http://blog.csdn.net/morewindows/article/details/7596034)

1. WaitForSingleObject(g\_hEventWriter, INFINITE);
3. //读者个数增加
4. EnterCriticalSection(&g\_cs\_writer\_enter);
5. g\_nReaderCount++;
6. **if** (g\_nReaderCount == 1)
7. ResetEvent(g\_hEventNoReader);
8. LeaveCriticalSection(&g\_cs\_writer\_enter);

在WaitForSingleObject(g\_hEventWriter, INFINITE); 之后，在 ResetEvent(g\_hEventNoReader); 之前，写线程函数是有可能执行的，若加入Sleep（）代码可见此bug。因为博主是在先开启两个读线程立即执行，在开启写线程，这个bug就十分隐蔽了，不知我说的对不对

Re: [超然\_烟火](http://blog.csdn.net/sun2043430) 2013-04-12 10:13发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/sun2043430)

回复Little\_H：说的对，本文的代码并不能保证读写是互斥的。

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-06-16 11:06发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复Little\_H：非常感谢，我马上看下这个问题。

Re: [JulianShang](http://blog.csdn.net/JulianShang) 2013-11-01 09:31发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/JulianShang)

确实有这个问题存在。 g\_hEventNoReader = CreateEvent(NULL, FALSE, TRUE, NULL); 这句使得写线程一开始就得到 WaitForSingleObject(g\_hEventNoReader, INFINITE)成功执行的机会，如果在读线程对 g\_nReaderCount进行修改之前让读线程Sleep(),则会出现既在读又在写的情况。

Re: [zhuyf87](http://blog.csdn.net/zhuyf87) 2013-06-17 06:57发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/8/D/C/3_zhuyf87.jpg](http://blog.csdn.net/zhuyf87)

回复MoreWindows：这块确实存在问题。在读者线程的Wait...函数后面加一个微小的延时，比如Sleep(10);，问题就可以呈现出来了。延时不能加的太大，加的太大，写者线程已经写文件完毕，读者线程才继续，问题就呈现不出来了。

Re: [JulianShang](http://blog.csdn.net/JulianShang) 2013-11-01 09:25发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/JulianShang)

确实有这个问题存在。 g\_hEventNoReader = CreateEvent(NULL, FALSE, TRUE, NULL); 这句使得写线程一开始就得到 WaitForSingleObject(g\_hEventNoReader, INFINITE)成功执行的机会，如果在读线程对 g\_nReaderCount进行修改之前让读线程Sleep(),则会出现既在读又在写的情况。

13楼 [wuhz2006](http://blog.csdn.net/wuhz2006) 2012-06-15 15:17发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/wuhz2006)

感谢博主写这么好的文

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-06-15 15:46发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复wuhz2006：读者写者问题用读写锁SRWLock来解决是最方便的，欢迎参阅《秒杀多线程第十四篇 读者写者问题继 读写锁SRWLock》

12楼 [songbeirui](http://blog.csdn.net/songbeirui) 2012-06-04 10:43发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/songbeirui)

学习。

11楼 [rwxdfbb](http://blog.csdn.net/rwxdfbb) 2012-06-03 23:33发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/rwxdfbb)

恩，一直关注博主的多线程秒杀系列篇，学到了不少知识！针对本例的情况，个人感觉单线程是不是要更好一点，反正有一个人干事，别人都得等着，不过作为学习，还是不错的例程。  
另外，博主要是有时间，希望最好能详细分析一下第一篇的那些面试题，让我们对这些面试题有一个确切的答案，也算对博主所讲的知识有全面的认识和回顾。

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-06-05 17:38发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复rwxdfbb："反正有一个人干事，别人都得等着"，读者可以共享的读的。  
最近非常忙，过后有空会整理发布的，欢迎再来参阅，谢谢。

10楼 [horsewww](http://blog.csdn.net/horsewww) 2012-06-02 22:44发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/horsewww)

在《Windows核心编程》里面已经提供了一个读写锁的类，大牛写的，代码简洁易懂高效。建议参考一下。

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-06-03 18:34发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复horsewww：读写锁使用很方便的，后面将会有继篇来讲解读写锁的。

9楼 [horsewww](http://blog.csdn.net/horsewww) 2012-06-02 22:40发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/horsewww)

看了一下，这个算法有Bugs，比如 ReaderThreadFun 函数里  
  
//读者个数增加  
InterlockedIncrement(PLONG(&g\_nReaderCount));  
  
看这里。在一个线程里执行完上面一句，但是还没有执行下面的if语句的时候，g\_nReaderCount可能被另一个线程修改，导致下面的判断语句不正确，进而ResetEvent(g\_hEventNoReader)得不到执行。  
多线程同步互斥很容易犯错而且错误不容易被发现，务必小心。  
  
if (g\_nReaderCount == 1)  
ResetEvent(g\_hEventNoReader);

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-06-03 19:07发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复horsewww：已经改正了。

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-06-03 18:37发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复horsewww：你说的对，  
34. //读者个数增加   
35. InterlockedIncrement(PLONG(&g\_nReaderCount));   
36. if (g\_nReaderCount == 1)   
37. ResetEvent(g\_hEventNoReader);   
和  
46. InterlockedDecrement(PLONG(&g\_nReaderCount));   
47. if (g\_nReaderCount == 0)   
48. SetEvent(g\_hEventNoReader);   
应该要分别用关键段来确保下执行的完整性。

8楼 [雷格](http://blog.csdn.net/leigema) 2012-05-29 22:20发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/leigema)

大神，好期待你的《进程通信之二 管道技术第三篇 命名管道》。  
什么时候出来啊！

Re: [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-06-01 19:55发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

回复雷格：这个月内。

Re: [雷格](http://blog.csdn.net/leigema) 2012-06-05 16:53发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/leigema)

回复MoreWindows：大神能不能帮小弟看一个命名管道连接的问题！跪求啊！  
http://topic.csdn.net/u/20120605/16/06491f4d-af78-4d3b-be1c-beaca358bb76.html  
用命名管道实现两个进程的通信，实在是不知道为什么？  
在一台机子上可以，在局域网内就不行了，客户端打开管道时错误代码为53：找不到指定网络路径。

7楼 [psuvtk](http://blog.csdn.net/psuvtk) 2012-05-28 13:20发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/2/E/E/3_psuvtk.jpg](http://blog.csdn.net/psuvtk)

顶一个

6楼 [李达\_August](http://blog.csdn.net/lidaasky) 2012-05-28 11:36发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/lidaasky)

[e01]

5楼 [strongcurrent](http://blog.csdn.net/strongcurrent) 2012-05-28 10:08发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/strongcurrent)

不错，顶！

4楼 [pangpangguai](http://blog.csdn.net/pangpangguai) 2012-05-28 10:02发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/pangpangguai)

顶一个

3楼 [jamcode](http://blog.csdn.net/jamcode) 2012-05-28 10:00发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/jamcode)

多线程就是浮云

2楼 [goodboy99199](http://blog.csdn.net/goodboy99199) 2012-05-28 09:54发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[](http://blog.csdn.net/goodboy99199)

不错，蛮详细的，先收藏了。

1楼 [MoreWindows](http://blog.csdn.net/MoreWindows) 2012-05-28 09:49发表 [[回复]](http://blog.csdn.net/morewindows/article/details/7596034#reply)

[http://avatar.csdn.net/C/F/7/3_morewindows.jpg](http://blog.csdn.net/MoreWindows)

读者写者还有很多解法，有兴趣的同学可以试试读写锁。

您还没有登录,请[[登录]](javascript:void(0);)或[[注册]](http://passport.csdn.net/account/register?from=http%3A%2F%2Fblog.csdn.net%2Fmorewindows%2Farticle%2Fdetails%2F7596034)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[**核心技术类目**](http://www.csdn.net/tag/)

[全部主题](http://www.csdn.net/tag) [Hadoop](http://g.csdn.net/5272865) [AWS](http://g.csdn.net/5272866) [移动游戏](http://g.csdn.net/5272870) [Java](http://g.csdn.net/5272871) [Android](http://g.csdn.net/5272872) [iOS](http://g.csdn.net/5272873) [Swift](http://g.csdn.net/5272868) [智能硬件](http://g.csdn.net/5272869) [Docker](http://g.csdn.net/5272867) [OpenStack](http://g.csdn.net/5272925)[VPN](http://www.csdn.net/tag/vpn) [Spark](http://g.csdn.net/5272924) [ERP](http://www.csdn.net/tag/erp) [IE10](http://www.csdn.net/tag/ie10) [Eclipse](http://www.csdn.net/tag/eclipse) [CRM](http://www.csdn.net/tag/crm) [JavaScript](http://www.csdn.net/tag/javascript) [数据库](http://www.csdn.net/tag/%E6%95%B0%E6%8D%AE%E5%BA%93) [Ubuntu](http://www.csdn.net/tag/ubuntu) [NFC](http://www.csdn.net/tag/nfc) [WAP](http://www.csdn.net/tag/wap) [jQuery](http://www.csdn.net/tag/jquery)[BI](http://www.csdn.net/tag/bi) [HTML5](http://www.csdn.net/tag/html5) [Spring](http://www.csdn.net/tag/spring) [Apache](http://www.csdn.net/tag/apache) [.NET](http://www.csdn.net/tag/.net) [API](http://www.csdn.net/tag/api) [HTML](http://www.csdn.net/tag/html) [SDK](http://www.csdn.net/tag/sdk) [IIS](http://www.csdn.net/tag/iis) [Fedora](http://www.csdn.net/tag/fedora) [XML](http://www.csdn.net/tag/xml) [LBS](http://www.csdn.net/tag/lbs) [Unity](http://www.csdn.net/tag/unity)[Splashtop](http://www.csdn.net/tag/splashtop) [UML](http://www.csdn.net/tag/uml) [components](http://www.csdn.net/tag/components) [Windows Mobile](http://www.csdn.net/tag/windowsmobile) [Rails](http://www.csdn.net/tag/rails) [QEMU](http://www.csdn.net/tag/qemu) [KDE](http://www.csdn.net/tag/kde) [Cassandra](http://www.csdn.net/tag/cassandra) [CloudStack](http://www.csdn.net/tag/cloudstack)[FTC](http://www.csdn.net/tag/ftc) [coremail](http://www.csdn.net/tag/coremail) [OPhone](http://www.csdn.net/tag/ophone)[CouchBase](http://www.csdn.net/tag/couchbase) [云计算](http://www.csdn.net/tag/%E4%BA%91%E8%AE%A1%E7%AE%97) [iOS6](http://www.csdn.net/tag/iOS6) [Rackspace](http://www.csdn.net/tag/rackspace)[Web App](http://www.csdn.net/tag/webapp) [SpringSide](http://www.csdn.net/tag/springside) [Maemo](http://www.csdn.net/tag/maemo)[Compuware](http://www.csdn.net/tag/compuware) [大数据](http://www.csdn.net/tag/%E5%A4%A7%E6%95%B0%E6%8D%AE) [aptech](http://www.csdn.net/tag/aptech) [Perl](http://www.csdn.net/tag/perl) [Tornado](http://www.csdn.net/tag/tornado) [Ruby](http://www.csdn.net/tag/ruby) [Hibernate](http://www.csdn.net/hibernate) [ThinkPHP](http://www.csdn.net/tag/thinkphp) [HBase](http://www.csdn.net/tag/hbase) [Pure](http://www.csdn.net/tag/pure) [Solr](http://www.csdn.net/tag/solr)[Angular](http://www.csdn.net/tag/angular) [Cloud Foundry](http://www.csdn.net/tag/cloudfoundry) [Redis](http://www.csdn.net/tag/redis) [Scala](http://www.csdn.net/tag/scala) [Django](http://www.csdn.net/tag/django) [Bootstrap](http://www.csdn.net/tag/bootstrap)

**与君共勉**

未曾清贫难成人  
不经打击老天真  
自古英雄出炼狱  
从来富贵入凡尘！  
------------------------  
[欢迎关注左丞的微博](http://weibo.com/MoreWindows)  
个人邮箱：morewindows#126.com

**个人资料**

[[http://avatar.csdn.net/C/F/7/1_morewindows.jpg](http://my.csdn.net/MoreWindows)](http://my.csdn.net/MoreWindows)  
[**MoreWindows**](http://my.csdn.net/MoreWindows)

1 1 1

* + 访问：3986817次
  + 积分：30193
  + 等级： http://c.csdnimg.cn/jifen/images/xunzhang/jianzhang/blog8.png
  + 排名：第93名
  + 原创：156篇
  + 转载：0篇
  + 译文：0篇
  + 评论：4408条

**博客专栏**

|  |  |
| --- | --- |
| [http://avatar.csdn.net/blogpic/20130107165742475.jpg](http://blog.csdn.net/column/details/algorithm-easyword.html) | [白话经典算法](http://blog.csdn.net/column/details/algorithm-easyword.html)  文章：17篇  阅读：994892 |

|  |  |
| --- | --- |
| [http://avatar.csdn.net/blogpic/20120330165623256.jpg](http://blog.csdn.net/column/details/killthreadseries.html) | [秒杀多线程面试题系列](http://blog.csdn.net/column/details/killthreadseries.html)  文章：15篇  阅读：862063 |

|  |  |
| --- | --- |
| [http://avatar.csdn.net/blogpic/20120317155612183.jpg](http://blog.csdn.net/column/details/morewindows-program.html) | [Windows C/C++/C# 编程](http://blog.csdn.net/column/details/morewindows-program.html)  文章：130篇  阅读：3690132 |

|  |  |
| --- | --- |
| [http://avatar.csdn.net/blogpic/20120105135242570.jpg](http://blog.csdn.net/column/details/stl-morewindows.html) | [C++ STL](http://blog.csdn.net/column/details/stl-morewindows.html)  文章：11篇  阅读：316777 |

**阅读排行**

* [【OpenCV入门指南】第一篇 安装OpenCV](http://blog.csdn.net/morewindows/article/details/8225783)(194613)
* [白话经典算法系列之六 快速排序 快速搞定](http://blog.csdn.net/morewindows/article/details/6684558)(183633)
* [白话经典算法系列之七 堆与堆排序](http://blog.csdn.net/morewindows/article/details/6709644)(177409)
* [白话经典算法系列之五 归并排序的实现](http://blog.csdn.net/morewindows/article/details/6678165)(148181)
* [秒杀多线程第二篇 多线程第一次亲密接触 CreateThread与\_beginthreadex本质区别](http://blog.csdn.net/morewindows/article/details/7421759)(117714)
* [秒杀多线程第一篇 多线程笔试面试题汇总](http://blog.csdn.net/morewindows/article/details/7392749)(112852)
* [白话经典算法系列之一 冒泡排序的三种实现](http://blog.csdn.net/morewindows/article/details/6657829)(109208)
* [秒杀多线程第四篇 一个经典的多线程同步问题](http://blog.csdn.net/morewindows/article/details/7442333)(102079)
* [STL系列之十 全排列(百度迅雷笔试题)](http://blog.csdn.net/morewindows/article/details/7370155)(70353)
* [秒杀多线程第三篇 原子操作 Interlocked系列函数](http://blog.csdn.net/morewindows/article/details/7429155)(68033)

**MoreWindows微博**

**文章分类**

* [白话经典算法系列](http://blog.csdn.net/MoreWindows/article/category/859207)(16)
* [Windows多线程](http://blog.csdn.net/MoreWindows/article/category/1115271)(15)
* [STL 他山之石](http://blog.csdn.net/MoreWindows/article/category/923923)(11)
* [C/C++/C#基础](http://blog.csdn.net/MoreWindows/article/category/868661)(18)
* [HTML/javascript/PHP](http://blog.csdn.net/MoreWindows/article/category/946696)(12)
* [Linux编程](http://blog.csdn.net/MoreWindows/article/category/1654717)(1)
* [MoreWindows工作笔记](http://blog.csdn.net/MoreWindows/article/category/1830595)(12)
* [Windows界面编程](http://blog.csdn.net/MoreWindows/article/category/1830593)(13)
* [Windows编程](http://blog.csdn.net/MoreWindows/article/category/862060)(87)
* [VC6.0及VS2008使用技巧](http://blog.csdn.net/MoreWindows/article/category/863841)(7)
* [OpenCV入门指南](http://blog.csdn.net/MoreWindows/article/category/1291764)(13)
* [生活](http://blog.csdn.net/MoreWindows/article/category/3168519)(1)

**评论排行**

* [白话经典算法系列之六 快速排序 快速搞定](http://blog.csdn.net/morewindows/article/details/6684558)(191)
* [白话经典算法系列之七 堆与堆排序](http://blog.csdn.net/morewindows/article/details/6709644)(165)
* [秒杀多线程第六篇 经典线程同步 事件Event](http://blog.csdn.net/morewindows/article/details/7445233)(145)
* [位操作基础篇之位操作全面总结](http://blog.csdn.net/morewindows/article/details/7354571)(139)
* [秒杀多线程第三篇 原子操作 Interlocked系列函数](http://blog.csdn.net/morewindows/article/details/7429155)(126)
* [白话经典算法系列之五 归并排序的实现](http://blog.csdn.net/morewindows/article/details/6678165)(125)
* [秒杀多线程第五篇 经典线程同步 关键段CS](http://blog.csdn.net/morewindows/article/details/7442639)(124)
* [STL系列之十 全排列(百度迅雷笔试题)](http://blog.csdn.net/morewindows/article/details/7370155)(117)
* [【白话经典算法系列之十四】腾讯2012年实习生笔试加分题](http://blog.csdn.net/morewindows/article/details/8742666)(116)
* [秒杀多线程第十篇 生产者消费者问题](http://blog.csdn.net/morewindows/article/details/7577591)(114)

**文章搜索**

窗体顶端



窗体底端

**最新评论**

* [【OpenCV入门指南】第三篇Canny边缘检测](http://blog.csdn.net/morewindows/article/details/8239625#comments)

[qq\_32746221](http://blog.csdn.net/qq_32746221): 你好 出现这个canny边缘检测.exe 中的 0x74afc42d 处未处理的异常: Micros...

* [白话经典算法系列之七 堆与堆排序](http://blog.csdn.net/morewindows/article/details/6709644#comments)

[Crisi](http://blog.csdn.net/ZEROKISSU): @ClutchFans\_OO:不好意思，给你回复错了，今天逛到这儿才看到，是（最后一个位置的索引-1...

* [白话经典算法系列之七 堆与堆排序](http://blog.csdn.net/morewindows/article/details/6709644#comments)

[Crisi](http://blog.csdn.net/ZEROKISSU): @ClutchFans\_OO:(数组长度-1)/2,这就是最后一个节点的父节点,从这儿开始

* [开博了](http://blog.csdn.net/morewindows/article/details/45462145#comments)

[暮婉歌](http://blog.csdn.net/qq_27831145): 非常感谢博主整理出通俗易懂的内容供小白学习~

* [白话经典算法系列之五 归并排序的实现](http://blog.csdn.net/morewindows/article/details/6678165#comments)

[qq2495664502](http://blog.csdn.net/qq2495664502): @awawfwfw:因为除法会向下取整假如你有一个集合1-6这样mid就等于（1+6）/2 =3所以...

* [进程通信之二 管道技术第二篇 匿名管道](http://blog.csdn.net/morewindows/article/details/7390441#comments)

[小文件](http://blog.csdn.net/x5942110): 给力啊！看了这么多，还是你写的清楚~

* [白话经典算法系列之五 归并排序的实现](http://blog.csdn.net/morewindows/article/details/7390441#comments)

[小黄难听大黄不雅](http://blog.csdn.net/ckhl1985): void mergearray(int a[],int m,int b[],int n,int c[...

* [白话经典算法系列之五 归并排序的实现](http://blog.csdn.net/morewindows/article/details/7390441#comments)

[小黄难听大黄不雅](http://blog.csdn.net/ckhl1985): void mergearray(int a[],int m,int b[],int n,int c[...

* [白话经典算法系列之六 快速排序 快速搞定](http://blog.csdn.net/morewindows/article/details/6684558#comments)

[十一期盖丽男](http://blog.csdn.net/u013036688): 讲解的很详细，明白了

* [白话经典算法系列之七 堆与堆排序](http://blog.csdn.net/morewindows/article/details/6684558#comments)

[十一期盖丽男](http://blog.csdn.net/u013036688): 讲解的挺好哒，谢谢~

[公司简介](http://www.csdn.net/company/about.html)|[招贤纳士](http://www.csdn.net/company/recruit.html)|[广告服务](http://www.csdn.net/company/marketing.html)|[银行汇款帐号](http://www.csdn.net/company/account.html)|[联系方式](http://www.csdn.net/company/contact.html)|[版权声明](http://www.csdn.net/company/statement.html)|[法律顾问](http://www.csdn.net/company/layer.html)|[问题报告](mailto:webmaster@csdn.net)|[合作伙伴](http://www.csdn.net/friendlink.html)|[论坛反馈](http://bbs.csdn.net/forums/Service)

[网站客服](http://wpa.qq.com/msgrd?v=3&uin=2355263776&site=qq&menu=yes)[杂志客服](http://wpa.qq.com/msgrd?v=3&uin=2251809102&site=qq&menu=yes)[微博客服](http://e.weibo.com/csdnsupport/profile)[webmaster@csdn.net](mailto:webmaster@csdn.net)400-600-2320|北京创新乐知信息技术有限公司 版权所有|江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 070598 号|Copyright © 1999-2014, CSDN.NET, All Rights Reserved [GongshangLogo](http://www.hd315.gov.cn/beian/view.asp?bianhao=010202001032100010)

[量子统计](http://tongji.linezing.com/report.html?unit_id=2819552)