

VC 中使用内存映射文件处理大文件（1）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/294/2021_2022_VC___E4_B8_AD_E4_BD_BF_c67_294126.htm 引言 文件操作是应用程序最为基本的功能之一，Win32 API和MFC均提供有支持文件处理的函数和类，常用的有Win32 API的CreateFile()、WriteFile()、ReadFile()和MFC提供的CFile类等。一般来说，以上这些函数可以满足大多数场合的要求，但是对于某些特殊应用领域所需要的动辄几十GB、几百GB、乃至几TB的海量存储，再以通常的文件处理方法进行处理显然是行不通的。目前，对于上述这种大文件的操作一般是以内存映射文件的方式来加以处理的，本文下面将针对这种Windows核心编程技术展开讨论。

内存映射文件 内存映射文件与虚拟内存有些类似，通过内存映射文件可以保留一个地址空间的区域，同时将物理存储器提交给此区域，只是内存文件映射的物理存储器来自一个已经存在于磁盘上的文件，而非系统的页文件，而且在该文件进行操作之前必须首先对文件进行映射，就如同将整个文件从磁盘加载到内存。由此可以看出，使用内存映射文件处理存储于磁盘上的文件时，将不必再对文件执行I/O操作，这意味着在对文件进行处理时将不必再为文件申请并分配缓存，所有的文件缓存操作均由系统直接管理，由于取消了将文件数据加载到内存、数据从内存到文件的回写以及释放内存块等步骤，使得内存映射文件在处理大数据量的文件时能起到相当重要的作用。另外，实际工程中的系统往往需要在多个进程之间共享数据，如果数据量小，处理方法是灵活多变的，如果共享数据容量巨大，那么就需要借助于内存

映射文件来进行。实际上，内存映射文件正是解决本地多个进程间数据共享的最有效方法。内存映射文件并不是简单的文件I/O操作，实际用到了Windows的核心编程技术--内存管理。所以，如果想对内存映射文件有更深刻的认识，必须对Windows操作系统的内存管理机制有清楚的认识，内存管理的相关知识非常复杂，超出了本文的讨论范畴，在此就不再赘述，感兴趣的读者可以参阅其他相关书籍。下面给出使用内存映射文件的一般方法：首先要通过CreateFile()函数来创建或打开一个文件内核对象，这个对象标识了磁盘上将要用作内存映射文件的文件。在用CreateFile()将文件映像的物理存储器的位置通告给操作系统后，只指定了映像文件的路径，映像的长度还没有指定。为了指定文件映射对象需要多大的物理存储空间还需要通过CreateFileMapping()函数来创建一个文件映射内核对象以告诉系统文件的尺寸以及访问文件的方式。在创建了文件映射对象后，还必须为文件数据保留一个地址空间区域，并把文件数据作为映射到该区域的物理存储器进行提交。由MapViewOfFile()函数负责通过系统的管理而将文件映射对象的全部或部分映射到进程地址空间。此时，对内存映射文件的使用和处理同通常加载到内存中的文件数据的处理方式基本一样，在完成了对内存映射文件的使用时，还要通过一系列的操作完成对其的清除和使用过资源的释放。这部分相对比较简单，可以通过UnmapViewOfFile()完成从进程的地址空间撤消文件数据的映像、通过CloseHandle()关闭前面创建的文件映射对象和文件对象。

内存映射文件相关函数 在使用内存映射文件时，所使用的API函数主要就是前面提到过的那几个函数，下面分别对其

进行介绍：HANDLE CreateFile(LPCTSTR lpFileName, DWORD dwDesiredAccess, DWORD dwShareMode, LPSECURITY_ATTRIBUTES lpSecurityAttributes, DWORD dwCreationDisposition, DWORD dwFlagsAndAttributes, HANDLE hTemplateFile). 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com