|  |
| --- |
| 1. 实验目的及实验内容   （本次实验所涉及并要求掌握的知识；实验内容；必要的原理分析） |
| 实验目的：  学习并应用Win32 API实现共享内存  实验内容：  使用Win32 API设计两个利用共享内存通信的程序。生产者进程生成Catalan序列，并写入内存共享对象。消费者进程从共享内存读取并输出序列。  生成的Catalan数的数目由命令行指定。 |
| 1. 实验环境及实验步骤   （本次实验所使用的器件、仪器设备等的情况；具体的实验步骤） |
| 实验环境：   1. IDE: CodeBlocks 2. 系统：Win10 家庭版   实验步骤：   1. 学习Catalan数的知识，并提炼出求Catalan数的比较简单的方法。   Catalan数的性质：  h(n)=h(n-1)\*(4\*n-2)/(n+1); h(n)=C(2n,n)/(n+1) (n=0,1,2,...) h(n)=c(2n,n)-c(2n,n-1)(n=0,1,2,...)  h(0)=1,h(1)=1 h(n)= h(0)\*h(n-1)+h(1)\*h(n-2) + ... + h(n-1)\*h(0) (n>=2)  简单的求法：若令f(2)=f(3)=1, 则f(n+1)=f(n)\* (4\*n-6)/n 。但是这里4\*n-6/n会有除不尽。  可以catalan[i]\*(4\*j-6)/j   1. 学习有关内存共享和进程创建的Win32 API函数   主要涉及CreateFile，CreateFileMapping，GetLastError，MapViewOfFile，sprintf，OpenFileMapping，CreateProcess函数。  函数的作用：  CreateFile 获得文件Handle  CreateFileMapping创建Handle的映射  MapViewOfFile在进程的虚拟地址空间，部分的表示上面的映射（可以是一部分，最大是完全的映射，不会再大）  CreateProcess创建进程。  各函数的参数及作用：  HANDLE CreateFile(  LPCTSTR lpFileName,    // 指向文件名的指针   DWORD dwDesiredAccess,    // 访问模式（写 / 读）   DWORD dwShareMode,    // 共享模式   LPSECURITY\_ATTRIBUTES lpSecurityAttributes, // 指向安全属性的指针   DWORD dwCreationDisposition,   // 如何创建   DWORD dwFlagsAndAttributes,   // 文件属性   HANDLE hTemplateFile    // 用于复制文件句柄  );  HANDLE WINAPI CreateFileMapping(  \_In\_ HANDLE hFile,  \_In\_opt\_ LPSECURITY\_ATTRIBUTES lpAttributes,  \_In\_ DWORD flProtect,  \_In\_ DWORD dwMaximumSizeHigh,  \_In\_ DWORD dwMaximumSizeLow,  \_In\_opt\_ LPCTSTR lpName  );  HANDLE CreateFileMapping(    HANDLE hFile,                       //物理文件句柄    LPSECURITY\_ATTRIBUTES lpAttributes, //安全设置    DWORD flProtect,                    //保护设置    DWORD dwMaximumSizeHigh,            //高位文件大小(因为我们的游戏是32位的，所以此处设为 0)    DWORD dwMaximumSizeLow,             //低位文件大小    LPCTSTR lpName                      //共享内存名称  );  BOOL CreateProcess(  　LPCTSTR lpApplicationName, // 应用程序名称  　LPTSTR lpCommandLine, // 命令行字符串  　LPSECURITY\_ATTRIBUTES lpProcessAttributes, // 进程的安全属性  　LPSECURITY\_ATTRIBUTES lpThreadAttributes, // 线程的安全属性  　BOOL bInheritHandles, // 是否继承父进程的属性  　DWORD dwCreationFlags, // 创建标志  　LPVOID lpEnvironment, // 指向新的环境块的指针  　LPCTSTR lpCurrentDirectory, // 指向当前目录名的指针  　LPSTARTUPINFO lpStartupInfo, // 传递给新进程的信息  　LPPROCESS\_INFORMATION lpProcessInformation // 新进程返回的信息  );   1. 代码实现   实际思路：  设计两个程序Catalan.exe（生产者） 以及Soncatalan.exe(消费者)，实现实验目的中的功能。  Catalan.cpp结构：  检查输入输出，若无错误将输入的catalan数个数转化为整型  按照设计好的简单方法计算Catalan序列，存储到vector<int>Catalan  CreateFile返回handle。（文件C:\\Users\\lenovo\\Desktop\\a.txt，采用了绝对路径）  CreateFileMapping创建映射  MapViewOfFile创建映射的视图  //在上述创建过程中均检查函数返回值确定是否创建成功，成功则继续，否则打印错误信息并退出  Sprintf打印信息到视图  CreateProcess打开soncatalan程序  //这里路径为C:\\Users\\lenovo\\Desktop\\soncatalan.exe，也采用绝对路径  睡眠一段时间等待soncatalan程序执行完毕  UnMapViewOfFile接触映射，CloseHandle接触句柄  结束  Soncatalan.cpp结构：  OpenFileMapping得到句柄  MapViewOfFile打开映射  Printf打印映射中内容  打印结束信息”Son finished!”  UnMapViewOfFile接触映射，CloseHandle接触句柄  结束   1. 具体过程   代码编写完成后，编译得到exe文件。  放在C:\\Users\\lenovo\\Desktop  命令行输入 Catalan.exe 9运行  观察命令行输出  观察是否在桌面创建了a.txt以及其中内容是否与命令行输出一致  以及检查产生的Catalan序列是否正确   1. **具体代码**   **见附件中的源文件** |
| 1. 实验过程分析   （实验分工，详细记录实验过程中发生的故障和问题，进行故障分析，说明故障排除的过程及方法。根据具体实验，记录、整理相应的数据表格等） |
| 遇到的问题及解决：   1. CreateFileMapping出错   解决：GetLastError获得错误信息，发现是文件大小的参数错误，不应该为0.调整后解决   1. 如何按输入到视图遇到了问题   解决：  for(int i=0;i<sizek;i++)  {  Int;  t=sprintf((char\*)lpMapAddress+j,"%d\r\n",catalan[i]); j+=t; }   1. 如何输出的问题   解决：直接采用printf %s输出  实验结果记录：    IMG_256 |