

Windows平台下常用进程间通信的实现方式计算机二级考试
PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/598/2021_2022_Windows_E5_B9_B3_c97_598150.htm 编辑特别推荐: 全国计算机等级考试 (等考) 指定教材 全国计算机等级考试学习视频 全国计算机等级考试网上辅导招生 全国计算机等级考试时间及科目预告 百考试题教育全国计算机等级考试在线测试平台 全国计算机等级考试资料下载 全国计算机等级考试论坛 计算机等级考试四级应用题解析汇总 2009年下半年全国计算机二级考试报名时
间从6月1日起已经开始报名。详情点击：2009年下半年全国计算机等级考试各地报名点汇总。2009年下半年全国计算机二级考试时间是2009年9月19日至23日。更多优质资料尽在百考试题论坛 百考试题在线题库。 Windows平台为我们提供了多种进程间通信的机制，主要包括：注册表方式、共享文件方式、共享内存方式、共享数据段、映射文件方式、管道方式、剪贴板方式、消息方式。其中注册表方式需要增加注册表表项，而注册表一般不应删改，所以此种方式不被推荐；共享数据段需要借助动态链接库，实现起来比较麻烦，这种方式也不被推荐。下面重点介绍一下其它几种进程间通信的实现方式。 1. 共享文件方式 (1) 数据发送 数据发送进程为通过Cfile类创建一个共享文件，然后调用Write()方法向文件中写入数据，具体代码如下：

```
void CSendDlg::OnSend() {  
//TODO: 在此添加控件通知处理程序代码 UpdateData(TRUE).  
//更新数据 CFile file. CString filename = _T("C:\\test.txt"). if  
(file.Open(filename , CFile::modeCreate | CFile::modeWrite |  
CFile::shareDenyRead)) { char * buf =
```

```
(char*)(LPCWSTR)m_strsend.file.Write(buf , strlen(buf)).  
file.Close(). } else { MessageBox(_T("创建文件失败！")). } } ( 2  
) 数据接收 数据在接收进程中，通过Cfile类打开以上创建的  
共享文件，然后调用Read方法读取数据，具体代码如下：  
void CRecieveDlg::Onrecieve() { //TODO: 在此添加控件通知处  
理程序代码 CFile file. CString filename = _T("C:\\test.txt"). if  
(file.Open(filename , CFile::modeRead|CFile::shareDenyWrite)) {  
char Buf[100]={0}. file.Read(Buf , 100). m_strrecieve=Buf.  
file.Close(). } else { MessageBox(_T("打开文件失败！")). }  
UpdateData(FALSE). //更新数据 } 2 . 共享内存方式 通过内存  
来传递数据，必须在内存中申请一定的空间。可以调  
用GlobalAlloc()或者VirtualAllocEx()来实现内存空间分配，使  
用内存读写函数ReadProcessMemory()和WriteProcessMemory()  
来读写进程的内存。要使接收程序获得发送程序的内存地址  
，可以通过发送消息方法来实现，即通过消息把内存地址从  
发送程序传递到接收程序。（1）数据发送 首先要使用发送  
消息的方法来传递指针，就需要定义一个用户消息。可用如  
下的自定义消息来传递指针： const UINT  
wm_nMemMsg=RegisterWindowMessage("mem_data"). 寻找接  
收数据的程序Recieve的窗口指针pWnd和进程句柄hProcess，  
用VirtualAllocEx()函数在这个进程中申请虚拟内存空间。然后  
通过WriteProcessMemory()把字符串m_strsend存放入虚拟内存  
中，并且通过消息wm_nMemMsg把所申请的内存空间起始地  
址发送给数据接收程序。最后，当数据接收程序接收到数据  
后，用VirtualFreeEx()释放所申请的虚拟内存。 数据发送函数  
具体代码如下： void CSendDlg::OnSend() { //TODO: 在此添加
```

```
控件通知处理程序代码 UpdateData(TRUE). //更新数据 CWnd
*pWnd=CWnd::FindWindow(NULL, _T("Recieve")). //查
找Recieve进程 if(pWnd==NULL){ MessageBox(_T("寻找接收消
息窗口失败！")). return. } DWORD PID. //获取进程号
GetWindowThreadProcessId(pWnd-&PID ). HANDLE
hProcess = OpenProcess (PROCESS_ALL_ACCESS , FALSE
, PID). LPVOID lpBaseAddress. //分配虚拟内存 lpBaseAddress
= VirtualAllocEx(hProcess , 0 , BUFFER_SIZE ,
MEM_COMMIT , PAGE_READWRITE). char
data[BUFFER_SIZE]. strcpy(data , m_strsend). //把字符串写
入hProcess进程的内存 WriteProcessMemory(hProcess ,
lpBaseAddress , data , BUFFER_SIZE , NULL). //发送基址
给Recieve进程 pWnd->SendMessage(WM_NMEMMSG , NULL
, (LPARAM)lpBaseAddress). Sleep(100). //等待接收程序接收数
据 VirtualFreeEx(hProcess , lpBaseAddress , 0 ,
MEM_RELEASE). //释放虚拟内存 } (2) 数据接收 首先需要
定义一个用户消息 , 如下代码所示 : const UINT
WM_NMEMMSG=RegisterWindowMessage("mem_data"). 然后在
头文件中添加消息映射函数定义 : afx_msg void
OnRegMemMsg(WPARAM wParam , LPARAM lParam). 接着需
要定义WM_NMEMMSG消息映射 , 它在消息映射表中的表示方
法如下 : BEGIN_MESSAGE_MAP(CDataRecvDlg , CDialog)
ON_REGISTERED_MESSAGE(WM_NMEMMSG
, OnRegMemMsg) END_MESSAGE_MAP() 最后在源文件中
添加实现消息映射函数 , 具体代码如下 : LRESULT
CRecieveDlg::OnRegMemMsg(WPARAM wParam , LPARAM
```

```
IParam) { //TODO: 在此添加控件通知处理程序代码 LPVOID  
lpBaseAddress=(LPVOID)IParam. HANDLE  
hProcess=GetCurrentProcess(). //把字符串写入hProcess进程的  
内存 char data[BUFFER_SIZE]. ReadProcessMemory(hProcess ,  
lpBaseAddress , data , BUFFER_SIZE , NULL).  
m_strrecieve=data. UpdateData(FALSE). //更新数据 return 0. }  
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
```