# 枚举进程方法1，参考：[用系统快照枚举所有进程](Windows%20Api学习指南学习笔记062-枚举进程1.docx)

# 枚举进程方法2，参考：[用psapi函数枚举所有进程](Windows%20Api学习指南学习笔记063-枚举进程2.docx)

# 这一节我们来学习枚举进程方法3

|  |
| --- |
|  |

# API语法说明

## WTSOpenServer函数的用法

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 打开指定远程桌面会话主机 (RD 会话主机) 服务器的句柄。 语法 C++复制  HANDLE WTSOpenServerW(  [in] LPWSTR pServerName  ); 参数 [in] pServerName  指向以 null 结尾的字符串的指针，该字符串指定 RD 会话主机服务器的 NetBIOS 名称。 返回值 如果函数成功，则返回值是指定服务器的句柄。  如果函数失败，它将返回无效的句柄。 可以通过在另一个函数调用中使用它来测试句柄的有效性。 注解 使用 **完 WTSOpenServer** 返回的句柄后，通过调用 [WTSCloseServer](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtscloseserver) 函数释放它。  无需打开在运行应用程序的 RD 会话主机服务器上执行的操作的句柄。 请改用 **常量WTS\_CURRENT\_SERVER\_HANDLE** 。  **备注**  wtsapi32.h 标头将 WTSOpenServer 定义为别名，该别名根据 UNICODE 预处理器常量的定义自动选择此函数的 ANSI 或 Unicode 版本。 将非特定编码别名与非非特定编码的代码混合使用可能会导致不匹配，从而导致编译或运行时错误。 有关详细信息，请参阅 [**函数原型的约定**](https://learn.microsoft.com/zh-cn/windows/win32/intl/conventions-for-function-prototypes)。 要求  |  |  | | --- | --- | | **标头** | wtsapi32.h | | **Library** | Wtsapi32.lib | | **DLL** | Wtsapi32.dll |  另请参阅 [WTSCloseServer](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtscloseserver) |

## WTSEnumerateProcess函数的用法

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 检索有关指定远程桌面会话主机 (RD 会话主机) 服务器上的活动进程的信息。 语法 C++复制  BOOL WTSEnumerateProcessesA(  [in] HANDLE hServer,  [in] DWORD Reserved,  [in] DWORD Version,  [out] PWTS\_PROCESS\_INFOA \*ppProcessInfo,//这是指针的指针  [out] DWORD \*pCount  ); 参数 [in] hServer  RD 会话主机服务器的句柄。 指定 [由 WTSOpenServer](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtsopenservera) 函数打开的句柄，或指定 **WTS\_CURRENT\_SERVER\_HANDLE** 以指示运行应用程序的 RD 会话主机服务器。  [in] Reserved  保留;必须为零。  [in] Version  指定枚举请求的版本。 必须为 1。  [out] ppProcessInfo  指向变量的指针，该变量接收指向 [WTS\_PROCESS\_INFO](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/ns-wtsapi32-wts_process_infoa) 结构的数组的指针。 数组中的每个结构都包含有关指定 RD 会话主机服务器上的活动进程的信息。 若要释放返回的缓冲区，请调用 [WTSFreeMemory](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtsfreememory) 函数。  [out] pCount  指向一个变量的指针，该变量接收 *ppProcessInfo* 缓冲区中返回**的WTS\_PROCESS\_INFO**结构的数目。 返回值 如果函数成功，则返回值为非零值。  如果函数失败，则返回值为零。 要获得更多的错误信息，请调用 GetLastError。 注解 调用方必须是管理员组的成员，才能枚举在不同用户的上下文下运行的进程。  **备注**  wtsapi32.h 标头将 WTSEnumerateProcesses 定义为别名，该别名根据 UNICODE 预处理器常量的定义自动选择此函数的 ANSI 或 Unicode 版本。 将非特定编码别名的使用与非非特定编码的代码混合使用可能会导致不匹配，从而导致编译或运行时错误。 有关详细信息，请参阅 [**函数原型的约定**](https://learn.microsoft.com/zh-cn/windows/win32/intl/conventions-for-function-prototypes)。 要求  |  |  | | --- | --- | | **标头** | wtsapi32.h | | **Library** | Wtsapi32.lib | | **DLL** | Wtsapi32.dll | | **API 集** | 在 Windows 8) 中引入的 ext-ms-win-session-wtsapi32-l1-1-0 ( |  另请参阅 [WTS\_PROCESS\_INFO](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/ns-wtsapi32-wts_process_infoa) |

## WTSCloseServer函数的用法

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 关闭远程桌面会话主机 (RD 会话主机) 服务器的打开句柄。 语法 C++复制  void WTSCloseServer(  [in] HANDLE hServer  ); 参数 [in] hServer  通过调用 [WTSOpenServer 或 WTSOpenServerEx](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtsopenservera) 函数打开的 RD 会话主机服务器的句柄。  不要为此参数传递 **WTS\_CURRENT\_SERVER\_HANDLE** 。 返回值 无 备注 在程序的清理例程中调用 **WTSCloseServer** 函数，以关闭通过调用 [WTSOpenServer](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtsopenservera) 或 [WTSOpenServerEx](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtsopenserverexa) 函数打开的所有服务器句柄。  关闭句柄后，它不能与任何其他 WTS API 一起使用。 要求  |  |  | | --- | --- | | **标头** | wtsapi32.h | | **Library** | Wtsapi32.lib | | **DLL** | Wtsapi32.dll |  另请参阅 [WTSOpenServer](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtsopenservera) |

## WTS\_PROCESS\_INFO结构体

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 包含有关远程桌面会话主机 (RD 会话主机) 服务器上运行的进程的信息。 语法 C++复制  typedef struct \_WTS\_PROCESS\_INFOA {  DWORD SessionId;  DWORD ProcessId;  LPSTR pProcessName;  PSID pUserSid;  } WTS\_PROCESS\_INFOA, \*PWTS\_PROCESS\_INFOA; 成员 SessionId  与进程关联的会话的远程桌面服务会话标识符。  ProcessId  唯一标识 RD 会话主机服务器上的进程的进程标识符。  pProcessName  指向以 null 结尾的字符串的指针，其中包含与进程关联的可执行文件的名称。  pUserSid  指向进程的主要访问令牌中的用户 [安全标识符](https://learn.microsoft.com/zh-cn/windows/desktop/SecAuthZ/security-identifiers) 的指针。 有关 SID 和访问令牌的详细信息，请参阅 [访问控制](https://learn.microsoft.com/zh-cn/windows/desktop/SecAuthZ/access-control)。 注解 **备注**  wtsapi32.h 标头将 WTS\_PROCESS\_INFO 定义为别名，该别名根据 UNICODE 预处理器常量的定义自动选择此函数的 ANSI 或 Unicode 版本。 将非特定编码别名的使用与非非特定编码的代码混合使用可能会导致不匹配，从而导致编译或运行时错误。 有关详细信息，请参阅 [**函数原型的约定**](https://learn.microsoft.com/zh-cn/windows/win32/intl/conventions-for-function-prototypes)。 要求 展开表   | **要求** | **值** | | --- | --- | | **最低受支持的客户端** | Windows Vista | | **最低受支持的服务器** | Windows Server 2008 | | **标头** | wtsapi32.h |  另请参阅 [WTSEnumerateProcesses](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtsenumerateprocessesa) |

## WTSFreeMemory函数的语法

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 释放由远程桌面服务函数分配的内存。 语法 C++复制  void WTSFreeMemory(  [in] PVOID pMemory  ); 参数 [in] pMemory  指向要释放的内存的指针。 返回值 无 备注 多个远程桌面服务函数分配缓冲区以返回信息。 使用 **WTSFreeMemory** 函数释放这些缓冲区。 要求  |  |  | | --- | --- | | **标头** | wtsapi32.h | | **Library** | Wtsapi32.lib | | **DLL** | Wtsapi32.dll | | **API 集** | 在 Windows 8) 中引入的 ext-ms-win-session-wtsapi32-l1-1-0 ( |  另请参阅 [WTSEnumerateProcesses](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtsenumerateprocessesa)  [WTSEnumerateSessions](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtsenumeratesessionsa)  [WTSQuerySessionInformation](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtsquerysessioninformationa)  [WTSQueryUserConfig](https://learn.microsoft.com/zh-cn/windows/desktop/api/wtsapi32/nf-wtsapi32-wtsqueryuserconfiga) |

# 演练

## 准备：这个程序需要电脑的NetBIOS名称，获取方法，在计算机上面点击右键-》点击下面的重命名电脑，注意是下面的那个不是上面那个，然后点击计算机名，点击其他，就可以看到NetBios名称，把它复制粘贴到我们的程序中

|  |
| --- |
|  |

## 1.新建一个常规空项目，取名Lesson64-wtsapi-enum-proc然后新建一个c语言源文件，取名：wtsapi-enum-proc

|  |
| --- |
|  |

## 2.完整的代码如下：

|  |
| --- |
| #include<Windows.h>  #include<WtsApi32.h>  #include<stdio.h>  #include<stdlib.h>  #pragma comment(lib,"WtsApi32.lib")//导入动态链接库，没有这一句，程序会报错  int main()  {  HANDLE hServer;  PWTS\_PROCESS\_INFO ppi;//这是一个指针  PWTS\_PROCESS\_INFO ppiTemp;//先声明然后在下面赋值，否则报错  DWORD dwCounts;  int i;  char\* serverName = "WIN-20240813BOX";//自己电脑的NetBIOS名称，如果修改了，这里必须修改  //打开终端服务器，也就是把自己的电脑当中终端服务器  hServer = WTSOpenServer(serverName);  if(NULL == hServer)  {  printf("打开终端服务器失败\n");  return 1;  }  if(!WTSEnumerateProcesses(hServer,0,1,&ppi,&dwCounts))  {  printf("枚举进程失败\n");  return 2;  }  ppiTemp = ppi;//用一个变量保存结构体变量，因为下面需要删除它，防止内存泄漏，在C语言中变量需要先声明再赋值  printf("一共有%d个进程\n",dwCounts);  printf("Process name\t ProcessId\n");  for(i=0;i<dwCounts;i++)  {  printf("%s \t %d\n",ppi->pProcessName,ppi->ProcessId);  ppi++;  }  WTSFreeMemory(ppiTemp);//释放内存  WTSCloseServer(hServer);//使用完毕需要改变 服务器  system("pause");  return 0;  } |

### 效果

|  |
| --- |
|  |

# 这一节的学习到此为止，这种方式比较简单。