bool tishenqx()

{

HANDLE ProcessHandle;

DWORD DesiredAccess;

PHANDLE TokenHandle;

ProcessHandle = GetModuleHandle(NULL);

DesiredAccess = TOKEN\_ADJUST\_PRIVILEGES;

OpenProcessToken(

ProcessHandle, //要修改访问权限的进程句柄

DesiredAccess, //指定你要进行的操作类型

TokenHandle //返回的访问令牌指针

);

struct quanxiannameclass{

LPCTSTR lpName1 = SE\_ASSIGNPRIMARYTOKEN\_NAME;

LPCTSTR lpName2 = SE\_AUDIT\_NAME;

LPCTSTR lpName3 = SE\_BACKUP\_NAME;

LPCTSTR lpName4 = SE\_CREATE\_GLOBAL\_NAME;

LPCTSTR lpName5 = SE\_DEBUG\_NAME;

} quanxianname;

struct quanxianluidclass{

PLUID lpLuid1;

PLUID lpLuid2;

PLUID lpLuid3;

PLUID lpLuid4;

PLUID lpLuid5;

} quanxianluid;

LookupPrivilegeValue(NULL, quanxianname.lpName1, quanxianluid.lpLuid1);

LookupPrivilegeValue(NULL, quanxianname.lpName1, quanxianluid.lpLuid2);

LookupPrivilegeValue(NULL, quanxianname.lpName1, quanxianluid.lpLuid3);

LookupPrivilegeValue(NULL, quanxianname.lpName1, quanxianluid.lpLuid4);

LookupPrivilegeValue(NULL, quanxianname.lpName1, quanxianluid.lpLuid5);

TOKEN\_PRIVILEGES NewState;

//填充LUID\_AND\_ATTRIBUTES

LUID\_AND\_ATTRIBUTES qxluid[5];

qxluid[0].Luid = \*quanxianluid.lpLuid1;

qxluid[1].Luid = \*quanxianluid.lpLuid2;

qxluid[2].Luid = \*quanxianluid.lpLuid3;

qxluid[3].Luid = \*quanxianluid.lpLuid4;

qxluid[4].Luid = \*quanxianluid.lpLuid5;

qxluid[0].Attributes = SE\_PRIVILEGE\_ENABLED;

qxluid[1].Attributes = SE\_PRIVILEGE\_ENABLED;

qxluid[2].Attributes = SE\_PRIVILEGE\_ENABLED;

qxluid[3].Attributes = SE\_PRIVILEGE\_ENABLED;

qxluid[4].Attributes = SE\_PRIVILEGE\_ENABLED;

//填充TOKEN\_PRIVILEGES

NewState.PrivilegeCount = 5;

NewState.Privileges[0] = qxluid[0];

NewState.Privileges[1] = qxluid[1];

NewState.Privileges[2] = qxluid[2];

NewState.Privileges[3] = qxluid[3];

NewState.Privileges[4] = qxluid[4];

AdjustTokenPrivileges(

TokenHandle, //包含特权的句柄

NULL, //禁用所有权限标志

&NewState, //新特权信息的指针(结构体)

NULL, //缓冲数据大小,以字节为单位的PreviousState的缓存区(sizeof)

NULL, //接收被改变特权当前状态的Buffer

NULL //接收PreviousState缓存区要求的大小

);

return 0;

}