struct \_guandaocaozuo

{

HANDLE hReadPipe;

HANDLE hWritePipe;

}guandaocaozuo;

bool guandaoguojian()

{

//创建管道

SECURITY\_ATTRIBUTES sa;

sa.nLength = sizeof(SECURITY\_ATTRIBUTES);

sa.lpSecurityDescriptor = NULL;

sa.bInheritHandle = TRUE;

CreatePipe(&guandaocaozuo.hReadPipe, &guandaocaozuo.hWritePipe, &sa, 0);

//创建cmd进程，并指明输入输出为管道返回的读写句柄

STARTUPINFO si;

PROCESS\_INFORMATION pi;

//取得决定新进程的主窗体如何显示的STARTUPINFO基本结构内容体并填充，使cmd的输入输出和管道关联

GetStartupInfo(&si);

si.dwFlags = STARTF\_USESHOWWINDOW | STARTF\_USESTDHANDLES;

si.hStdInput = &guandaocaozuo.hReadPipe;

si.hStdError = &guandaocaozuo.hWritePipe;

si.hStdOutput = &guandaocaozuo.hWritePipe;

si.wShowWindow = SW\_HIDE;

//得到系统路径，即cmd路径

long cmdlen;

TCHAR cmdline[256] = { 0 };

cmdlen = GetSystemDirectory(cmdline, sizeof(cmdline));

wcscat(cmdline,TEXT("\\cmd.exe"));

//创建进程

if (CreateProcess(cmdline, NULL, NULL, NULL, TRUE, NULL, NULL, NULL, &si, &pi) == 0)

{

printf("CreateProcess Error \n");

return 1;

}

return 0;

}