

```

#include "stdio.h"
#include "windows.h"
#include "winbase.h"
#include "tlhelp32.h"
#pragma comment(lib,"kernel32.lib")
#pragma comment(lib,"advapi32.lib")
void EnableDebugPriv( void )
{
HANDLE hToken;
TOKEN_PRIVILEGES tkp;
OpenProcessToken(GetCurrentProcess(),TOKEN_ADJUST_PRIVILEGES |
TOKEN_QUERY, &hToken);
LookupPrivilegeValue(NULL, SE_SHUTDOWN_NAME,&tkp.Privileges
[0].Luid);
tkp.PrivilegeCount = 1;
tkp.Privileges[0].Attributes = SE_PRIVILEGE_ENABLED;
AdjustTokenPrivileges(hToken, FALSE, &tkp, 0,
(PTOKEN_PRIVILEGES)NULL, 0);
CloseHandle( hToken );
}
int pskill(int id)    //根据进程 ID 杀进程
{
HANDLE hProcess=NULL;
//打开目标进程
hProcess=OpenProcess(PROCESS_TERMINATE,FALSE,id);
if (hProcess==NULL){
printf("\nOpen Process fAiled:%d\n",GetLastError());
return -1;
}
//结束目标进程
DWORD ret=TerminateProcess(hProcess,0);
if(ret==0){
printf("%d",GetLastError());
}
return -1;
}
int main(void)
{
//进程列举
HANDLE hSnApsht= CreateToolhelp32Snapshot
(TH32CS_SNAPPROCESS,0);
if(hSnApsht!=INVALID_HANDLE_VALUE){

```

```
PROCESSENTRY32 te={sizeof(te)};
BOOL f0k = Process32First(hSnApshot,&te);
for(;f0k;f0k=Process32Next(hSnApshot,&te)){
printf("Pid: %d %s\n",te.th32ProcessID ,te.szExeFile );
}
}
CloseHandle(hSnApshot);
//杀进程
printf("the process's id which you want to kill:");
int id=0;
scanf("%d",&id);
EnableDebugPriv(); //提升权限
pskill(id);
return 0;
}
```