实现方法:

   1.  通过CreateToolhelpSnapshot函数创建一线程快照;

   2.  使用Thread32First以及Thread32Next函数遍历系统中所有线程，根据比较线程父进程ID来过滤指定进程的所有线程;

   3.  关闭线程快照句柄.

API说明:

   1. typedef struct tagTHREADENTRY32 {   
          DWORD dwSize;       //   sizeof(THREADENTRY32)   
          DWORD [cntUsage](ms-help://ms.vscc.v90/);    //   不用了   
          DWORD [th32ThreadID](ms-help://ms.vscc.v90/);  // 线程ID   
          DWORD [th32OwnerProcessID](ms-help://ms.vscc.v90/);  // 线程所有的进程ID   
          LONG [tpBasePri](ms-help://ms.vscc.v90/);     // 基础线程优先级   
          LONG [tpDeltaPri](ms-help://ms.vscc.v90/);     // 不用了   
          DWORD [dwFlags](ms-help://ms.vscc.v90/);   // 不用了

        } THREADENTRY32

        线程描述信息

   1.  HANDLE WINAPI CreateToolhelp32Snapshot(DWORD dwFlags, DWORD th32ProcessID);

        创建系统快照，当dwFlags=TH32CS\_SNAPTHREAD时，仅生成系统中线程部分快照

   2.  BOOL WINAPI Thread32First(HANDLE hSnapshot,  LPTHREADENTRY32 lpt2)

        获取线程快照中第一个线程信息

   3.  BOOL WINAPI Thread32Next(HANDLE hSnapshot, LPTHREADENTRY32 lpte);

        获取线程快照中下一个线程信息

关键代码:

DWORD GetThreadIDFromName(DWORD ProcessID) //ProcessID为要取线程id的进程id

{

HANDLE hSnapThread = CreateToolhelp32Snapshot(TH32CS\_SNAPTHREAD, NULL); //获得目标进程id的线程快照

if (INVALID\_HANDLE\_VALUE != hSnapThread) //如果快照有效

{

THREADENTRY32 te32 = { sizeof(te32) }; //线程消息结构

if (Thread32First(hSnapThread, &te32)) //从快照中取出线程消息

{

if (te32.th32OwnerProcessID == ProcessID) //如果线程所有者与进程id相同

return te32.th32ThreadID; //返回线程id

}

while (Thread32Next(hSnapThread, &te32)) {

if (te32.th32OwnerProcessID == ProcessID)

return te32.th32ThreadID;

}

}

CloseHandle(hSnapThread);

}