<https://blog.csdn.net/zacklin/article/details/7728482>

PROCESSENTRY32　　PROCESSENTRY32 Structure  
　　Describes an entry from a list of the processes residing in the system address space when a snapshot was taken.  
　　用来存放快照进程信息的一个结构体。（存放进程信息和调用成员输出进程信息）  
　　用来 **Process32First指向第一个进程信息，并将进程信息抽取到PROCESSENTRY32中。用 Process32Next指向下一条进程信息。**  
　　Syntax  
　　C++  
　　typedef struct tagPROCESSENTRY32   
　　{  
　　DWORD dwSize;   
　　DWORD cntUsage;   
　　DWORD th32ProcessID;  
　　ULONG\_PTR th32DefaultHeapID;  
　　DWORD th32ModuleID;   
　　DWORD cntThreads;  
　　DWORD th32ParentProcessID;  
　　LONG pcPriClassBase;   
　　DWORD dwFlags;   
　　TCHAR szExeFile[MAX\_PATH];  
　　} PROCESSENTRY32, \*PPROCESSENTRY32;  
　　Members  
　　**dwSize** （结构的大小）The size of the structure, in bytes. Before calling the [**Process32First**](http://baike.baidu.com/view/992064.htm) function, set this member to sizeof(PROCESSENTRY32). If you do not initialize **dwSize**, **Process32First** fails.  
　　(这个结构的长度，以字节为单位，初始化一个实例以后调用Process32First函数，设置成员的大小sizeof(PROCESSENTRY32).如果你没用PROCESSENTRY32中的成员dwSize初始化，pricess32First将会失败。)  
　　**cntUsage** （此进程的引用计数）This member is no longer used and is always set to zero.  
　　（这个成员已经很久没有使用，总是设置为零。）  
　　**th32ProcessID** 进程ID=process identifier=PIDThe process identifier.  
　　（这个就是任务管理器里面的进程的PID，打开任务管理器--查看---选择列---PID（勾选）就可以显示进程的标示符（PID））  
　　**th32DefaultHeapID** 进程默认堆IDThis member is no longer used and is always set to zero.  
　　（这个成员已经很久没有使用，总是设置为零。）  
　　**th32ModuleID** 进程模块IDThis member is no longer used and is always set to zero.  
　　（这个成员已经很久没有使用，总是设置为零。）  
　　**cntThreads** 此进程开启的线程计数The number of execution threads started by the process.  
　　（这个成员执行线程开始的进程。）  
　　**th32ParentProcessID** （父进程的ID）The identifier of the process that created this process (its parent process).  
　　**pcPriClassBase** .(线程优先权)The base priority of any threads created by this process  
　　当前进程创建的任何一个线程的基础优先级，即在当前进程内创建线程的话，其基本优先级的值。   
　　**dwFlags** This member is no longer used, and is always set to zero.  
　　（这个成员已经很久没有使用，总是设置为零。）  
　　**szExeFile** （一个数组）进程全名The name of the executable file for the process. To retrieve the full path to the executable file, call the **Module32First** function and check the **szExePath** member of the **MODULEENTRY32** structure that is returned. However, if the calling process is a 32-bit process, you must call the **QueryFullProcessImageName** function to retrieve the full path of the executable file for a 64-bit process.  
　　（进程的可执行文件名称。要获得可执行文件的完整路径，应调用Module32First函数，再检查其返回的MODULEENTRY32结构的szExePath成员。但是，如果被调用进程是一个32位程序，您必须调用QueryFullProcessImageName函数去获取64位进程的可执行文件完整路径名。）