typedef struct \_SYSTEM\_INFO {
  
 union {
  
 DWORD dwOemId;
  
 struct {
  
 WORD wProcessorArchitecture;
  
 WORD wReserved;
  
 };
  
 };
  
 DWORD dwPageSize; //内存页的大小
  
 LPVOID lpMinimumApplicationAddress; //指向应用程序和动态链接库(DLL)可以访问的最低内存地址。
  
 LPVOID lpMaximumApplicationAddress; //进程可用最大地址
  
 DWORD dwActiveProcessorMask;
  
 DWORD dwNumberOfProcessors; //处理机数量
  
 DWORD dwProcessorType;
  
 DWORD dwAllocationGranularity; //虚拟内存的粒度
  
 WORD wProcessorLevel;
  
 WORD wProcessorRevision;
  
 } SYSTEM\_INFO;

typedef struct \_MEMORYSTATUSEX {
  
 DWORD dwLength;
  
 DWORD dwMemoryLoad; //内存占用率
  
 DWORDLONG ullTotalPhys; //总物理内存
  
 DWORDLONG ullAvailPhys; //剩余物理内存
  
 DWORDLONG ullTotalPageFile;
  
 DWORDLONG ullAvailPageFile;
  
 DWORDLONG ullTotalVirtual; //虚拟内存总量
  
 DWORDLONG ullAvailVirtual; //虚拟内存剩余量
  
 DWORDLONG ullAvailExtendedVirtual;
  
 } MEMORYSTATUSEX, \*LPMEMORYSTATUSEX;

typedef struct \_PERFORMANCE\_INFORMATION {
  
 DWORD cb;
  
 SIZE\_T CommitTotal; //系统当前提交的页面总数
  
 SIZE\_T CommitLimit;
  
 SIZE\_T CommitPeak;
  
 SIZE\_T PhysicalTotal; //全部物理内存
  
 SIZE\_T PhysicalAvailable;//可用物理内存
  
 SIZE\_T SystemCache; //系统缓存
  
 SIZE\_T KernelTotal; //内核占内存总大小(当前分页和非分页内核池中的内存总和)
  
 SIZE\_T KernelPaged; //当前分页内核池中的内存
  
 SIZE\_T KernelNonpaged; //当前在非分页内核池中的内存
  
 SIZE\_T PageSize;
  
 DWORD HandleCount; //句柄计数
  
 DWORD ProcessCount; //进程计数
  
 DWORD ThreadCount; //线程计数
  
} PERFORMANCE\_INFORMATION;

HANDLE OpenProcess(
  
 DWORD dwDesiredAccess, //渴望得到的访问权限（标志）
  
 BOOL bInheritHandle, // 是否继承句柄
  
 DWORD dwProcessId // 进程标示符
  
);

HANDLE WINAPI CreateToolhelp32Snapshot(
  
 DWORD dwFlags, //用来指定“快照”中需要返回的对象
  
 DWORD th32ProcessID //进程ID号
  
);

typedef struct \_MEMORY\_BASIC\_INFORMATION {
  
 PVOID BaseAddress; // 区域基地址。
  
 PVOID AllocationBase; // 分配基地址。
  
 DWORD AllocationProtect; // 区域被初次保留时赋予的保护属性。
  
 SIZE\_T RegionSize; // 区域大小（以字节为计量单位）。
  
 DWORD State; // 状态（MEM\_FREE、MEM\_RESERVE 或 MEM\_COMMIT）。
  
 DWORD Protect; // 保护属性。
  
 DWORD Type; // 类型。
  
} MEMORY\_BASIC\_INFORMATION;