<https://blog.csdn.net/caimouse/article/details/1947575>

当大家打开Windows任务管理器时，就会看到每个进程使用内存的分布情况，往往会发现有一些进程占用大量的内存，在这种情况也是一种异常情况，可以作为是否恶意软件的标志之一。下面就来使用API函数GetProcessMemoryInfo来获取内存的使用情况。

函数GetProcessMemoryInfo声明如下：

BOOL

WINAPI

GetProcessMemoryInfo(

    HANDLE Process,

    PPROCESS\_MEMORY\_COUNTERS ppsmemCounters,

    DWORD cb

    );

***Process***是获取内存使用情况的进程句柄。

***ppsmemCounters***是返回内存使用情况的结构。

***cb***是结构的大小。

调用函数的例子如下：

#001 //获取某一个进程的内存信息。

#002  //蔡军生 2007/12/18 QQ:9073204 深圳

#003  void TestGetProcessMemoryInfo(void)

#004  {

#005         //

#006         const int nBufSize = 512;

#007         TCHAR chBuf[nBufSize];

#008         ZeroMemory(chBuf,nBufSize);

#009

#010         //

#011         DWORD dwProcs[1024];

#012         DWORD dwNeeded;

#013

#014         //枚举所有进程ID。

#015         if ( !EnumProcesses( dwProcs, sizeof(dwProcs), &dwNeeded ) )

#016         {

#017               //输出出错信息。

#018               wsprintf(chBuf,\_T("EnumProcesses failed (%d)./n"), GetLastError() );

#019               OutputDebugString(chBuf);

#020

#021               return;

#022         }

#023

#024         // 计算有多少个进程ID。

#025         DWORD dwProcCount = dwNeeded / sizeof(DWORD);

#026

#027         wsprintf(chBuf,\_T("EnumProcesses Count(%d)./n"), dwProcCount );

#028         OutputDebugString(chBuf);

#029

#030         //遍历所有进程ID，打开进程。

#031         for (DWORD i = 0; i < dwProcCount; i++)

#032         {

#033               wsprintf(chBuf,\_T("EnumProcesses (%d)./r/n"), dwProcs[i] );

#034               OutputDebugString(chBuf);

#035

#036               //根据进程ID打开进程。

#037               HANDLE hProcess = OpenProcess( PROCESS\_QUERY\_INFORMATION |

#038                    PROCESS\_VM\_READ,

#039                    FALSE, dwProcs[i] );

#040

#041               if (hProcess)

#042               {

#043                    //

#044                    PROCESS\_MEMORY\_COUNTERS pmc;

#045                    pmc.cb = sizeof(PROCESS\_MEMORY\_COUNTERS);

#046

#047                    //获取这个进程的内存使用情况。

**#048                   if ( ::GetProcessMemoryInfo( hProcess, &pmc, sizeof(pmc)) )**

#049                    {

#050                          ZeroMemory(chBuf,nBufSize);

#051

#052                          wsprintf(chBuf,\_T("/t缺页中断次数: 0x%08X/n"), pmc.PageFaultCount );

#053                          OutputDebugString(chBuf);

#054

#055                          wsprintf(chBuf,\_T("/t使用内存高峰: 0x%08X/n"),

#056                               pmc.PeakWorkingSetSize );

#057                          OutputDebugString(chBuf);

#058

#059                          wsprintf(chBuf,\_T("/t当前使用的内存: 0x%08X/n"), pmc.WorkingSetSize );

#060                          OutputDebugString(chBuf);

#061

#062                          wsprintf(chBuf,\_T("/t使用页面缓存池高峰: 0x%08X/n"),

#063                               pmc.QuotaPeakPagedPoolUsage );

#064                          OutputDebugString(chBuf);

#065

#066                          wsprintf(chBuf,\_T("/t使用页面缓存池: 0x%08X/n"),

#067                               pmc.QuotaPagedPoolUsage );

#068                          OutputDebugString(chBuf);

#069

#070                          wsprintf(chBuf,\_T("/t使用非分页缓存池高峰: 0x%08X/n"),

#071                               pmc.QuotaPeakNonPagedPoolUsage );

#072                          OutputDebugString(chBuf);

#073

#074                          wsprintf(chBuf,\_T("/t使用非分页缓存池: 0x%08X/n"),

#075                               pmc.QuotaNonPagedPoolUsage );

#076                          OutputDebugString(chBuf);

#077

#078                          wsprintf(chBuf,\_T("/t使用分页文件: 0x%08X/n"), pmc.PagefileUsage );

#079                          OutputDebugString(chBuf);

#080

#081                          wsprintf(chBuf,\_T("/t使用分页文件的高峰: 0x%08X/n"),

#082                               pmc.PeakPagefileUsage );

#083                          OutputDebugString(chBuf);

#084                    }

#085

#086                    //

#087                    CloseHandle(hProcess);

#088               }

#089         }

#090

#091  }

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/caimouse/article/details/1947575