
Subject: Re: memory consumption of my application
Posted by [BetoValle](#) on Fri, 05 May 2023 22:57:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

In stackoverflow i have located this code but I don't know if it works on linux. My question is, is this the way to go? Or if there is a code already implemented in U++?

```
#include <windows.h>
#include <stdio.h>
#include <psapi.h>

void PrintMemoryInfo( DWORD processID )
{
    HANDLE hProcess;
    PROCESS_MEMORY_COUNTERS pmc;

    // Print the process identifier.

    printf( "\nProcess ID: %u\n", processID );

    // Print information about the memory usage of the process.

    hProcess = OpenProcess( PROCESS_QUERY_INFORMATION |
                           PROCESS_VM_READ,
                           FALSE, processID );
    if (NULL == hProcess)
        return;

    if ( GetProcessMemoryInfo( hProcess, &pmc, sizeof(pmc)) )
    {
        printf( "\tPageFaultCount: 0x%08X\n", pmc.PageFaultCount );
        printf( "\tPeakWorkingSetSize: 0x%08X\n",
                pmc.PeakWorkingSetSize );
        printf( "\tWorkingSetSize: 0x%08X\n", pmc.WorkingSetSize );
        printf( "\tQuotaPeakPagedPoolUsage: 0x%08X\n",
                pmc.QuotaPeakPagedPoolUsage );
        printf( "\tQuotaPagedPoolUsage: 0x%08X\n",
                pmc.QuotaPagedPoolUsage );
        printf( "\tQuotaPeakNonPagedPoolUsage: 0x%08X\n",
                pmc.QuotaPeakNonPagedPoolUsage );
        printf( "\tQuotaNonPagedPoolUsage: 0x%08X\n",
                pmc.QuotaNonPagedPoolUsage );
        printf( "\tPagefileUsage: 0x%08X\n", pmc.PagefileUsage );
        printf( "\tPeakPagefileUsage: 0x%08X\n",
                pmc.PeakPagefileUsage );
    }
}
```

```
    CloseHandle( hProcess );
}

int main( )
{
    // Get the list of process identifiers.

    DWORD aProcesses[1024], cbNeeded, cProcesses;
    unsigned int i;

    if ( !EnumProcesses( aProcesses, sizeof(aProcesses), &cbNeeded ) )
        return 1;

    // Calculate how many process identifiers were returned.

    cProcesses = cbNeeded / sizeof(DWORD);

    // Print the memory usage for each process

    for ( i = 0; i < cProcesses; i++ )
        PrintMemoryInfo( aProcesses[i] );

    return 0;
}
```

File Attachments

1) [2023-05-05_194900.jpg](#), downloaded 166 times
