
Subject: memory consumption of my application
Posted by [BetoValle](#) on Fri, 05 May 2023 16:15:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

is there a function that can tell how much my application is consuming?

Thanks

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 38 times

Subject: Re: memory consumption of my application
 Posted by [BetoValle](#) on Sat, 06 May 2023 20:34:21 GMT

Hi,

I found these 2 codes for windows and linux. I think they are reasonable to have a base. On windows I did the test and the values don't match with the manager since they use different structure (as the note in the example). On linux the values are already close to the test application!

windows:

```
void defP(){
    // Task Manager windows 10 uses VM_COUNTERS_EX2 similar structure is defined (in
ntddk.h)
    MEMORYSTATUSEX memInfo;
    memInfo.dwLength = sizeof(MEMORYSTATUSEX);
    GlobalMemoryStatusEx(&memInfo);
    DWORDLONG totalVirtualMem = memInfo.ullTotalPageFile;

    DWORDLONG virtualMemUsed = memInfo.ullTotalPageFile - memInfo.ullAvailPageFile;

    PROCESS_MEMORY_COUNTERS_EX pmc;
    GetProcessMemoryInfo(GetCurrentProcess(), (PROCESS_MEMORY_COUNTERS*)&pmc,
sizeof(pmc));
    SIZE_T virtualMemUsedByMe = pmc.PrivateUsage;

    DWORDLONG totalPhysMem = memInfo.ullTotalPhys;

    DWORDLONG physMemUsed = memInfo.ullTotalPhys - memInfo.ullAvailPhys;

    SIZE_T physMemUsedByMe = pmc.WorkingSetSize;

    SIZE_T x=pmc.PagefileUsage;
    SIZE_T y=pmc.PrivateUsage;
    Cout() <<"m virtual usada-----> " << virtualMemUsedByMe/1024 << EOL;
    Cout() <<"m fisica usada-----> " << physMemUsed/1024/1024 << EOL;
    Cout() <<"total m usada-----> " << totalPhysMem << EOL;
    Cout() <<"m privada usada-----> " << y/1024 << EOL;

}

CONSOLE_APP_MAIN
{
    SetLanguage("pt-br");
    SetDateFormat("%3:02d/%2:02d/%1:4d"); // format code in editor
    SetDateScan("dmy");
}
```

```

defP();

}

linux:
#include <iostream>
#include <Core/Core.h>
#include <SysInfo/SysInfo.h>

using namespace Upp;

int parseLine(char* line){
    // This assumes that a digit will be found and the line ends in " Kb".
    int i = strlen(line);
    const char* p = line;
    while (*p <'0' || *p >'9') p++;
    line[i-3] = '\0';
    i = atoi(p);
    return i;
}

int getValue(){ //Note: this value is in KB!
    FILE* file = fopen("/proc/self/status", "r");
    int result = -1;
    char line[128];

    while (fgets(line, 128, file) != NULL){
        if (strncmp(line, "VmRSS:", 6) == 0){
            result = parseLine(line);
            break;
        }
    }
    fclose(file);
    return result;
}

CONSOLE_APP_MAIN
{
    Cout() << "--> " << getValue() << EOL;
}

```

File Attachments

1) [2023-05-06_171011.jpg](#), downloaded 39 times

Subject: Re: memory consumption of my application
Posted by [mirek](#) on Sun, 14 May 2023 08:34:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

For U++ heap, you can use MemoryUsedKb, MemoryUseKbMax, MemoryProfile, or PeakMemoryProfile.

There is even MemoryProfileInfo function in CtrlLib that show it in Prompt.
