

```

/*****
/*
/*----- M E M S T A T . C -----*/
/* Task      : Display of memory status information in a
/*            console window
/*-----*/
/* Authors       : Michael Tischer and Bruno Jennrich
/* developed on  : 09/06/1995
/* last update   : 09/06/1995
/*****/
#include <windows.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

/*****
/* main : Here we go
/*****/
void main( void )
{
    MEMORYSTATUS ms;
    SYSTEM_INFO  si;

    // MEMORYSTATUS -----
    printf("GlobalMemoryStatus:\n");
    printf("=====\n");

    ms.dwLength = sizeof( MEMORYSTATUS );           // must be initialized
    GlobalMemoryStatus( &ms );

    // Percentage of used memory
    printf( "MemoryLoad      : %ld%%\n", ms.dwMemoryLoad );
    // Size of physical memory
    printf( "TotalPhys       : 0x%08lx\n", ms.dwTotalPhys );
    // Free physical memory
    printf( "AvailPhys        : 0x%08lx\n", ms.dwAvailPhys );
    // Size of page file
    printf( "TotalPageFile: 0x%08lx\n", ms.dwTotalPageFile );
    // Free bytes in page file
    printf( "AvailPageFile: 0x%08lx\n", ms.dwAvailPageFile );
    // Size of virtual address space
    printf( "TotalVirtual : 0x%08lx\n", ms.dwTotalVirtual );
    // Free bytes in address space of current process
    printf( "AvailVirtual : 0x%08lx\n", ms.dwAvailVirtual );

    // SYSTEM_INFO -----
    printf("\nGetSystemInfo:\n");
    printf("=====\n");

    GetSystemInfo( &si );

    // Size of a memory page
    printf("PageSize          : 0x%04lx\n", si.dwPageSize );
    // Smallest address that application can address
    printf("Min-App-Address   : 0x%08lx\n",
        si.lpMinimumApplicationAddress );
    // Largest address that application can address
    printf("Max-App-Address   : 0x%08lx\n", si.lpMaximumApplicationAddress );
    // Which processor?
    printf("Processor Type    : ");
    switch( si.dwProcessorType )
    {
        case PROCESSOR_INTEL_386:    printf("i386\n");    break;
        case PROCESSOR_INTEL_486:    printf("i486\n");    break;
        case PROCESSOR_INTEL_PENTIUM: printf("Pentium\n"); break;
        default:                     printf("? \n");      break;
    }
    // Granularity for memory allocation
    printf("Alloc-Granularity : 0x%08lx\n",
        si.dwAllocationGranularity );

    printf("<Key>");
    _getch();
}

```

