Note: This API calls are shared between DOS and Win16 personality.

DPMI is a shared interface for DOS applications to access Intel 80286+ CPUs services. DOS DMPI host provides core services for protected mode applications. Multitasking OS with DOS support also provides DMPI in most cases. Windows standard and extended mode kernel is a DPMI client app. Standard and extended mode kernel differs minimally and shares common codebase. Standard Windows kernel works under DOSX extender. DOSX is a specialized version of 16-bit DPMI Extender (but it is standard DPMI host). Standard mode is just DPMI client, exnhanced mode is DPMI client running under Virtual Machime Manager (really, multitasker which allow to run many DOS sessions). Both modes shares DPMI interface for kernel communication. The OS/2 virtual DOS Protected Mode Interface (VDPMI) device driver provides Version 0.9 DPMI support for virtual DOS machines. Win16 (up to Windows ME) provides Version 0.9 DPMI support. Windows in Standard Mode provides DPMI services only for Windows Applications, not DOS sessions.

DPMI host often merged with DPMI extender. Usually DPMI extender provide DPMI host standard services and DOS translation or True DPMI services.

2021/08/05 10:15 · prokushev · 0 Comments

Int 31H, AH=05H, AL=00H

Version

0.9

Brief

Get Free Memory Information

Input

```
AX = 0500H
ES:(E)DI = selector:offset of 48-byte buffer
```

Return

```
Carry flag = clear (this function always succeeds)
and the buffer is filled in with the following information:
```

Offset	Length	Contents
00H	4	Largest available free block in bytes
04H	4	Maximum unlocked page allocation in pages
08H	4	Maximum locked page allocation in pages
0CH	4	Linear address space size in pages
10H	4	Total number of unlocked pages
14H	4	Total number of free pages
18H	4	Total number of physical pages
1CH	4	Free linear address space in pages
20H	4	Size of paging file/partition in pages
24H	0CH	Reserved, all bytes set to 0FFH

Notes

Returns information about the amount of available physical memory, linear address space, and disk space for page swapping. Since DPMI clients will often run in multitasking environments, the information returned by this function should only be considered as advisory. DPMI 1.0 clients should avoid use of this function (see the last note of the call).

32-bit programs must use ES:EDI to point to the buffer. 16-bit programs should use ES:DI.

Only the first field of the returned structure is guaranteed to contain a valid value. Any fields that are not supported by the DPMI host will be set by the host to -1 (0FFFFFFFFH) to indicate that the information is not available.

The field at buffer offset 00H specifies the largest block of contiguous linear memory in bytes that could be allocated if the memory were to be allocated and left unlocked.

The field at buffer offset 04H specifies the largest number of pages that could be allocated (the value at offset 00H divided by the page size).

The field at buffer offset 08H specifies the largest block of memory in pages that could be allocated and then locked.

The field at buffer offset 0CH specifies the size of the total linear address space in pages. This value includes all linear address space that has already been allocated.

The field at buffer offset 10H specifies the total number of pages that are currently unlocked and could be paged out. This value also contains any free pages.

The field at buffer offset 14H specifies the number of physical pages that currently are not in use.

The field at offset 18H specifies the total number of physical pages that the DPMI host manages. This value includes all free, locked, and unlocked physical pages.

The field at offset 20H specifies the size of the DPMI host's paging partition or file in pages.

The size of the pages used by the DPMI host can be obtained with the Get Page Size function (Int 31H Function 0604H).

DPMI 1.0 clients should use Int 31H Function 050BH in preference to this function. This function is

supported in DPMI 1.0 solely for backward compatibility with DPMI 0.9.

See also

Note

Text based on http://www.delorie.com/djgpp/doc/dpmi/

DPMI	
Process manager	**INT 2FH** 1680H, 1687H
Signals	
Memory manager	
Misc	**INT 2FH** 1686H, 168AH
Devices	

2021/08/13 14:23 · prokushev · 0 Comments