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=06H

Version

1.0

Brief

Get Page Attributes

Input

```
AX = 0506H
ESI = memory block handle
EBX = base offset in memory block of page (or of first page, if requesting
attributes for multiple pages)
ECX = number of pages
ES:EDX = selector:offset of a buffer to receive page attributes, 1 word (16-
bits) per page (see Note)
```

Return

```
if function successful
Carry flag = clear
and buffer at ES:EDX filled in with page attributes (see Note)

if function unsuccessful
Carry flag = set
AX = error code
8001H   unsupported function (16-bit host)
8023H   invalid handle (in ESI)
8025H   invalid linear address (Specified range is not within specified
block)
```

Notes

Returns the attributes of one or more pages within a linear memory block previously allocated with Int 31H Function 0504H.

A DPMI 1.0 host that is 16-bit only will not support this function.

A 16-bit client of a 32-bit DPMI 1.0 host can use this function.

If EBX is not aligned, it will be rounded down to the next lower page boundary.

The specified buffer is filled in by the DPMI host with the attributes of the requested pages, 1 word (16-bits) per page, in the following format:

Bits	**Significance**	
0-2	page type (0-7)	
	Value	Meaning
	0	uncommitted page
	1	committed page
	2	mapped page
	3-7	currently unused
3	0 = page is read-only 1 = page is read/write	
4	0 = accessed/dirty bits not available for this page 1 = accessed/dirty bits are supplied for this page in bits 5-6	
5	0 = page has not been accessed (if bit 4=1) 1 = page has been accessed (if bit 4=1)	
6	0 = page has not been modified (if bit 4=1) 1 = page has been modified (if bit 4=1)	
7-15	reserved, currently zero	

Mapped pages can only occur in memory blocks under DPMI hosts that support the Device Mapping capability or the Conventional Memory Mapping capability. See Int 31H Functions 0401H, 0508H, and 0509H.

The dirty and accessed bits are only supplied if the DPMI host supports the Page Accessed/Dirty capability. DPMI hosts that support this capability are required to return dirty and accessed bits for all committed pages and for mapped pages created with the Map Conventional Memory call (Int 31H Function 0509H). However, dirty and accessed bits may not be returned for individual mapped pages created with the Map Device call (Int 31H Function 0508H) if the host is using page table entries (PTEs) to virtualize the device.

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