

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=0EH, AL=01H

Version

1.0

Brief

Set Coprocessor Emulation

Input

```
AX = 0E01H
BX = coprocessor bits
```

Bit	Significance
0	new value of MPv bit for client's virtual CR0 0 = disable numeric coprocessor for this client 1 = enable numeric coprocessor for this client
1	new value of EMv bit for client's virtual CR0 0 = client will not supply coprocessor emulation 1 = client will supply coprocessor emulation
2-15	not applicable

Return

```
if function successful
Carry flag = clear

if function unsuccessful
Carry flag = set
AX = error code
8026H   invalid request (client requested disabling coprocessor on a
processor which does not support this)
```

Notes

Enables or disables the numeric coprocessor for this virtual machine and the reflection of coprocessor exceptions to the client.

If the MPv bit is not set, the host may not need to save the coprocessor state for this virtual machine to improve system performance.

If the virtual EM (EMv) bit is set, the host delivers all coprocessor exceptions to the client, so that the client can provide its own floating point emulation (whether or not a coprocessor is present or the host also has a floating point emulator). In other words, if the EMv bit is set, the host sets the EM bit in the real CR0 while the client is active, and reflects coprocessor not present faults (Int 7) to the client.

Floating point emulation can be tested on a system with a numeric coprocessor by using this function to enable client handling of coprocessor exceptions and disable the coprocessor.

The client should use Int 31H Function 0212H to register an exception handler for coprocessor not present faults (Int 7) prior to setting the EMv bit with this function.

A client can determine the CPU type with Int 31H Function 0400H, and the presence or absence of a coprocessor with Int 31H Function 0E00H. The client should not draw any conclusions about the presence or absence of a coprocessor based on the CPU type alone.

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

DPMI	
Devices	

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

From:
http://osfree.su/doku/ - **osFree wiki**

Permanent link:
http://osfree.su/doku/doku.php?id=en:docs:dpmi:api:int31:0e:01

Last update: **2021/08/27 07:03**