

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=00H

Version

1.0

Brief

Get Coprocessor Status

Input

```
AX = 0E00H
```

Return

```
if function successful
Carry flag = clear (this function always succeeds in DPMI 1.0)
AX = coprocessor status
```

Bit	Significance
0	MPv (MP bit in the virtual MSW/CR0) 0 = numeric coprocessor is disabled for this client 1 = numeric coprocessor is enabled for this client
1	EMv (EM bit in the virtual MSW/CR0) 0 = client is not emulating coprocessor instructions 1 = client is emulating coprocessor instructions
2	MPr (MP bit from the actual MSW/CR0) 0 = numeric coprocessor is not present 1 = numeric coprocessor is present
3	EMr (EM bit from the actual MSW/CR0) 0 = host is not emulating coprocessorinstructions 1 = host is emulating coprocessor instructions
4-7	coprocessor type 00H = no coprocessor 02H = 80287 03H = 80387 04H = 80486 with numeric coprocessor 05H-0FH reserved for future numeric processors
8-15	not applicable

```
if function unsuccessful
Carry flag = set (this function always fails in DPMI 0.9)
```

Notes

Returns information about whether or not a numeric coprocessor exists, the type of coprocessor available (if any), and whether or not the host or client is providing coprocessor emulation.

If the real EM (EMr) bit is set, the host is supplying or is capable of supplying floating point emulation.

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.

MPr bit setting should be consistent with the setting of coprocessor type information. Ignore MPr bit information if it is in conflict with the coprocessor type information.

If the virtual EM (EMv) bit is set, the host delivers all coprocessor exceptions to the client, and the client is performing 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 virtual machine is active, and reflects coprocessor not present faults (Int 7) to the virtual machine.

A client can determine the CPU type with Int 31H Function 0400H, but a 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

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

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

From:
https://www.osfree.ru/doku/ - **osFree wiki**

Permanent link:
https://www.osfree.ru/doku/doku.php?id=en:docs:dpmi:api:int31:0e:00

Last update: **2021/08/27 07:01**