



**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, enhanced mode is DPMI client running under Virtual Machine 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](#)

**Note: This API implemented by DPMI extenders under DOS and by DOSX.EXE under Windows. Under OS/2 MVDM this API implemented by VDPMI**

Interrupt	AX	Description	Version	32-bit host	16-bit host
2f	1680	Release Current Virtual Machine's Time Slice	1.0	Yes	Yes
2f	1686	Get CPU Mode	0.9	Yes	Yes
2f	1687	Obtain Real-to-Protected Mode Switch Entry Point	0.9	Yes	Yes
2f	168A	Get Vendor-Specific API Entry Point	1.0	Yes	Yes
31	0000	Allocate LDT Descriptors	0.9	Yes	Yes
31	0001	Free LDT Descriptor	0.9	Yes	Yes
31	0002	Segment to Descriptor	0.9		
31	0003	Get Selector Increment Value	0.9		
31	0006	Get Segment Base Address	0.9		
31	0007	Set Segment Base Address	0.9		
31	0008	Set Segment Limit	0.9		
31	0009	Set Descriptor Access Rights	0.9		
31	000A	Create Alias Descriptor	0.9		
31	000B	Get Descriptor	0.9		
31	000C	Set Descriptor	0.9		
31	000D	Allocate Specific LDT Descriptor	0.9		
31	000E	Get Multiple Descriptors	1.0		
31	000F	Set Multiple Descriptors	1.0		
31	0100	Allocate DOS Memory Block	0.9		
31	0101	Free DOS Memory Block	0.9		
31	0102	Resize DOS Memory Block	0.9		
31	0200	Get Real Mode Interrupt Vector	0.9		

<b>Interrupt</b>	<b>AX</b>	<b>Description</b>	<b>Version</b>	<b>32-bit host</b>	<b>16-bit host</b>
31	<a href="#">0201</a>	Set Real Mode Interrupt Vector	0.9		
31	<a href="#">0202</a>	Get Processor Exception Handler Vector	0.9		
31	<a href="#">0203</a>	Set Processor Exception Handler Vector	0.9		
31	<a href="#">0204</a>	Get Protected Mode Interrupt Vector	0.9		
31	<a href="#">0205</a>	Set Protected Mode Interrupt Vector	0.9		
31	<a href="#">0210</a>	Get Extended Processor Exception Handler Vector (Protected Mode)	1.0		
31	<a href="#">0211</a>	Get Extended Processor Exception Handler Vector (Real Mode)	1.0		
31	<a href="#">0212</a>	Set Extended Processor Exception Handler Vector (Protected Mode)	1.0		
31	<a href="#">0213</a>	Set Extended Processor Exception Handler Vector (Real Mode)	1.0		
31	<a href="#">0300</a>	Simulate Real Mode Interrupt	0.9		
31	<a href="#">0301</a>	Call Real Mode Procedure With Far Return Frame	0.9		
31	<a href="#">0302</a>	Call Real Mode Procedure With IRET Frame	0.9		
31	<a href="#">0303</a>	Allocate Real Mode Callback Address	0.9		
31	<a href="#">0304</a>	Free Real Mode Callback Address	0.9		
31	<a href="#">0305</a>	Get State Save/Restore Addresses	0.9		
31	<a href="#">0306</a>	Get Raw Mode Switch Addresses	0.9		
31	<a href="#">0400</a>	Get Version	0.9		
31	<a href="#">0401</a>	Get DPMI Capabilities	1.0		
31	<a href="#">0500</a>	Get Free Memory Information	0.9		
31	<a href="#">0501</a>	Allocate Memory Block	0.9		
31	<a href="#">0502</a>	Free Memory Block	0.9		
31	<a href="#">0503</a>	Resize Memory Block	0.9		
31	<a href="#">0504</a>	Allocate Linear Memory Block	1.0		
31	<a href="#">0505</a>	Resize Linear Memory Block	1.0		
31	<a href="#">0506</a>	Get Page Attributes	1.0		
31	<a href="#">0507</a>	Set Page Attributes	1.0		
31	<a href="#">0508</a>	Map Device in Memory Block	1.0		
31	<a href="#">0509</a>	Map Conventional Memory in Memory Block	1.0		
31	<a href="#">050A</a>	Get Memory Block Size and Base	1.0		
31	<a href="#">050B</a>	Get Memory Information	1.0		
31	<a href="#">0600</a>	Lock Linear Region	0.9		
31	<a href="#">0601</a>	Unlock Linear Region	0.9		
31	<a href="#">0602</a>	Mark Real Mode Region as Pageable	0.9		
31	<a href="#">0603</a>	Relock Real Mode Region	0.9		
31	<a href="#">0604</a>	Get Page Size	0.9		
31	<a href="#">0702</a>	Mark Page as Demand Paging Candidate	0.9		
31	<a href="#">0703</a>	Discard Page Contents	0.9		
31	<a href="#">0800</a>	Physical Address Mapping	0.9		
31	<a href="#">0801</a>	Free Physical Address Mapping	1.0		
31	<a href="#">0900</a>	Get and Disable Virtual Interrupt State	0.9		
31	<a href="#">0901</a>	Get and Enable Virtual Interrupt State	0.9		

Interrupt	AX	Description	Version	32-bit host	16-bit host
31	0902	Get Virtual Interrupt State	0.9		
31	0A00	Get Vendor-Specific API Entry Point	0.9		
31	0B00	Set Debug Watchpoint	0.9		
31	0B01	Clear Debug Watchpoint	0.9		
31	0B02	Get State of Debug Watchpoint	0.9		
31	0B03	Reset Debug Watchpoint	0.9		
31	0C00	Install Resident Service Provider Callback	1.0		
31	0C01	Terminate and Stay Resident	1.0		
31	0D00	Allocate Shared Memory	1.0		
31	0D01	Free Shared Memory	1.0		
31	0D02	Serialize on Shared Memory	1.0		
31	0D03	Free Serialization on Shared Memory	1.0		
31	0E00	Get Coprocessor Status	1.0		
31	0E01	Set Coprocessor Emulation	1.0		

DPMI	
Process manager	<b>INT 2FH 1680H, 1687H</b>
Signals	
Memory manager	
Misc	<b>INT 2FH 1686H, 168AH</b>
Devices	

2021/08/13 14:23 · prokushev · [0 Comments](#)

From:

<http://osfree.org/doku/> - **osFree wiki**

Permanent link:

<http://osfree.org/doku/doku.php?id=en:docs:dpmi>

Last update: **2023/03/25 03:17**

