

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=06H, AL=02H

Version

0.9

Brief

Mark Real Mode Region as Pageable

Input

```
AX = 0602H
BX:CX = starting linear address of memory to mark as pageable
SI:DI = size of region to be marked (bytes)
```

Return

```
if function successful
Carry flag = clear
```

```
if function unsuccessful
Carry flag = set
AX = error code
8002H   invalid state (region already marked as pageable)
8025H   invalid linear address (region is above 1 MB boundary)
```

Notes

Advises the DPMI host that the specified memory below the 1 MB boundary may be paged to disk.

If the function returns an error, none of the memory has been marked as pageable.

If the specified region overlaps part of a page at the beginning or end of the region, the page(s) will not be marked as pageable.

Pageability information for a real mode region is maintained as a binary state, not a count. Therefore, multiple calls to this function for the same region have no effect.

For compatibility with DPMI version 0.9 hosts, a client must call the Relock Real Mode Region function (Int 31H Function 0603H) to relock the memory region before terminating. Memory that remains unlocked after the client has terminated could result in fatal page faults when another program is executed in the same address space. DPMI 1.0 hosts automatically relock real mode memory at client termination.

Under some DPMI hosts, all conventional memory may be locked by default. If a protected mode program is using memory in the first megabyte of address space, it is recommended that this function be used to turn off automatic page locking for regions of memory that will not be touched at interrupt time.

The client must not mark memory as pageable in regions that it does not own; i.e. it may only mark as pageable memory that it has previously allocated with Int 31H Function 0100H or by a direct call to DOS via the translation functions. For example, marking all free DOS memory as pageable under some DPMI hosts could cause a page fault to occur while inside of DOS, resulting in a crash. Also, a client should not mark the DPMI host data area as pageable.

Note that address space marked as pageable by this function can still be locked using the Lock Linear Region function (Int 31H Function 0600H). This function is just an advisory service to allow memory that does not need to be locked to be paged out; it disables any automatic locking of real mode memory performed by the DPMI host.

This function is ignored by DPMI implementations that do not support virtual memory; the function will return the Carry flag clear to indicate success, but has no other effect. DPMI hosts which support virtual memory may also choose to ignore this function, but such hosts must be able to handle page faults transparently at arbitrary points during a client's execution, including within interrupt and exception handlers.

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

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

Permanent link:
https://osfree.ru/doku/doku.php?id=en:docs:dpmi:api:int31:06:02

Last update: **2021/08/27 05:17**