 This is part of **Family API** which allow to create dual-os version of program runs under OS/2 and DOS

Note: This is legacy API call. It is recommended to use 32-bit equivalent

2021/09/17 04:47 · prokushev · 0 Comments
2021/08/20 03:18 · prokushev · 0 Comments

DosReallocHuge

This call changes the size of memory originally allocated by DosAllocHuge.

Syntax

```
DosReallocHuge (NumSeg, Size, Selector)
```

Parameters

- NumSeg (USHORT) - input : Number of 65536 byte segments requested.
- Size (USHORT) - input : Number of bytes requested in the last non-65536 byte segment. A value of 0 indicates none.
- Selector (SEL) - input : Selector returned on a previous DosAllocHuge.

Return Code

rc (USHORT) - return

Return code descriptions are:

- 0 NO_ERRORç
- 8 ERROR_NOT_ENOUGH_MEMORY
- 87 ERROR_INVALID_PARAMETER

Remarks

DosReallocHuge is called to change the size of unshared or shared huge memory allocated by DosAllocHuge. The selector used for this call must be the one returned by the DosAllocHuge request.

Normally, segments allocated as shared (AllocFlags bits 0 and 1 were set) cannot be decreased in size. However, if AllocFlags bit 3 was also set, the shared segment's size can be decreased.

DosReallocHuge is also called to reallocate a segment allocated as discardable (AllocFlags bit 2 set) after the segment is discarded by the system. The call to DosReallocHuge automatically locks the segment for access by the caller, the same as if a DosLockSeg had been issued.

'Note:' This request may be issued from privilege level 2 or 3. However, only a privilege level 3 huge segment is valid.

Family API Considerations

Some options operate differently in the DOS mode than in the OS/2 mode. Therefore, the following restriction applies to DosReallocHuge when coding for the DOS mode:

The requested Size value is rounded up to the next paragraph (16-byte).

Bindings

C Binding

```
#define INCL_DOSMEMMGR

USHORT  rc = DosReallocHuge(NumSeg, Size, Selector);

USHORT           NumSeg;        /* Number of 65536-byte segments
                                    requested */
USHORT           Size;          /* Number of bytes in last segment */
SEL              Selector;      /* Selector */

USHORT           rc;            /* return code */
```

MASM Binding

```
EXTRN  DosReallocHuge:FAR
INCL_DOSMEMMGR      EQU 1

PUSH   WORD    NumSeg        ;Number of 65536-byte segments requested
PUSH   WORD    Size          ;Number of bytes in last segment
PUSH   WORD    Selector      ;Selector
CALL   DosReallocHuge

Returns WORD
```

Note

Text based on http://www.edm2.com/index.php/DosReallocHuge

Family API		
DOS	Process Manager	DosBeep DosExit DosSleep DosExecPgm
	File Manager	DosChDir DosChgFilePtr DosClose DosDelete DosDupHandle DosMkDir DosMove DosQCurDir DosQCurDisk DosSetFileMode DosOpen DosQFileInfo DosRead DosQFileMode DosQFSInfo DosQVerify DosRmDir DosSelectDisk DosFindClose DosFindFirst DosFindNext DosSetFileInfo DosSetVerify DosWrite DosFileLocks DosSetFHandState DosNewSize DosBufReset DosQFHandState DosSetFSinfo
	Memory Manager	DosFreeSeg DosSubAlloc DosSubFree DosSubSet DosAllocHuge DosAllocSeg DosReallocHuge DosReallocSeg DosGetHugeShift DosCreateCSAlias
	NLS	DosCaseMap DosGetCtryInfo DosGetDBCSEv DosSetCtryCode DosGetCollate DosGetMessage DosInsMessage DosPutMessage
	Date and Time	DosSetDateTime DosGetDateTime
	Devices	DosDevConfig DosDevIOCtl DosDevIOCtl2
	Signals	DosHoldSignal DosSetSigHandler
	Misc	BadDynLink DosGetEnv DosGetMachineMode DosGetVersion DosError DosErrClass DosSetVec
KBD		KbdCharIn KbdFlushBuffer KbdGetStatus KbdSetStatus KbdStringIn KbdPeek
VIO		VioGetBuf VioGetConfig VioGetCurPos VioGetCurType VioGetPhysBuf VioReadCellStr VioReadCharStr VioScrollUp VioScrollDn VioScrollLf VioScrollRt VioScrUnLock VioSetCurPos VioSetCurType VioSetMode VioGetMode VioShowBuf VioWrtCellStr VioWrtCharStr VioWrtCharStrAtt VioWrtNAttr VioWrtNCell VioWrtNChar VioWrtTTY VioScrLock VioPopUp
Tools		BIND
Modules		DOSCALLS.DLL VIOCALLS.DLL KBDCALLS.DLL MSG.DLL
Libraries		API.LIB OS2386.LIB FAPI.LIB DOSCALLS.LIB SUBCALLS.LIB

2018/08/25 15:05 · prokushev · 0 Comments