This is part of **Win16 API** which allow to create versions of program from one source code to run under OS/2 and Win16. Under OS/2 program can be running under Win-OS/2 if program is Windows NE executable, and with help on Windows Libraries for OS/2, if it is OS/2 NE executable. Here is a WLO to OS/2 API mapping draft

2021/09/01 04:23 · prokushev · 0 Comments

LocalReAlloc

Brief

Changes the size or the attributes of a specified local memory object. The size can increase or decrease. This function can also be used to discard a movable discardable block.

Syntax

```
HLOCAL WINAPI LocalReAlloc(
HLOCAL hMem,
UINT uBytes,
UINT uFlags
);
```

Parameters

hMem – Handle to the local memory object to be reallocated. This handle is returned by the LocalAlloc or a previous LocalReAlloc function.

uBytes – New size, in bytes, of the memory block. If uFlags specifies the LMEM_MODIFY flag, this parameter is ignored.

uFlags – Specifies how to reallocate the local memory object. If the LMEM_MODIFY flag is specified, this parameter modifies the attributes of the memory object, and the uBytes parameter is ignored. Otherwise, this parameter controls the reallocation of the memory object.

The LMEM_MODIFY flag can be combined with the following flags:

Flag	Description
LMEM_DISCARDABLE	Makes a movable block discardable. This flag is ignored if the object was not previously allocated as movable .
LMEM_MOVEABLE	You cannot combine LMEM_MOVEABLE with LMEM_MODIFY to change a fixed memory object into a movable one. The function returns an error if an application attempts this .

If this parameter does not specify LMEM_MODIFY, it can be any combination of the following flags:

Flag	Description
LMEM_MOVEABLE	

If uBytes is zero: Discards a previously movable and discardable memory block. The function fails if the object's lock count is not zero or the block is not movable and discardable. The handle remains valid.\

If uBytes is non-zero: Enables the system to move the reallocated block to a new location without changing the movable or fixed attribute of the memory object. If the object is fixed, the handle returned may be different from the handle specified in the hMem parameter. If the object is movable, the block can be moved without invalidating the handle, even if the object is currently locked. |

Flag	Description
LMEM_ZEROINIT	If the memory object is growing in size, the contents of the additional memory are initialized to zero .
LMEM_NOCOMPACT	Prevents memory from being compacted or discarded to satisfy the allocation request. This flag is provided only for compatibility with 16-bit Windows and is ignored in Win32 .
LMEM_NODISCARD	Prevents discardable blocks from being discarded to satisfy the allocation request. This flag is provided only for compatibility with 16-bit Windows and is ignored in Win32 .

Return Value

If the function succeeds, the return value is a handle to the reallocated memory object.

If the function fails, the return value is NULL. To get extended error information, call GetLastError.

Notes

If LocalReAlloc fails, the original memory is not freed, and the original handle and pointer are still valid .

If the function reallocates a movable object, the return value is a handle. To convert the handle to a pointer, use the LocalLock function.

If the function reallocates a fixed object, the value of the handle returned is the address of the first byte of the memory block. To access the memory, a process can simply cast the return value to a pointer .

When uBytes is zero and the LMEM_MOVEABLE flag is used, the function discards the block. The handle remains valid and can be used later to re-allocate the block .

In 16-bit Windows, if the data segment that contains the heap is moveable, calling this function may cause the data segment to move if Windows must increase the size of the heap and cannot do so in its current location. An application can prevent this by calling the LockData function .

Example Code

C Binding

```
#include <windows.h>

HLOCAL hMem = NULL;
LPSTR lpData = NULL;

// Allocate a fixed block of 100 bytes
hMem = LocalAlloc(LPTR, 100);

// Increase the size to 200 bytes, allow the block to move, and zero-init
new memory
hMem = LocalReAlloc(hMem, 200, LMEM_MOVEABLE | LMEM_ZEROINIT);

// Convert a fixed block to movable (this is done by reallocating with the
same size)
// Note: This does not use LMEM_MODIFY with LMEM_MOVEABLE, which is illegal.
hMem = LocalReAlloc(hMem, LocalSize(hMem), LMEM_MOVEABLE);

// Discard a movable discardable block
LocalReAlloc(hMem, 0, LMEM_MOVEABLE);
```

MASM Binding

```
; AX = hMem, BX = new size, CX = flags
push ax
push bx
push cx
call LocalReAlloc
```

See also

LocalAlloc

LocalFree

LocalLock

LocalDiscard

LocalSize

LockData

Group	Functions
Module manager	GETVERSION GETMODULEHANDLE GETMODULEUSAGE GETMODULEFILENAME GETPROCADDRESS MAKEPROCINSTANCE FREEPROCINSTANCE GETINSTANCEDATA CATCH THROW GETCODEHANDLE LOADLIBRARY
Global Memory Manager	GlobalAlloc GlobalCompact GlobalDiscard GlobalFree GlobalLock GlobalReAlloc GlobalSize GlobalUnlock GlobalFlags
Local Memory Manager	LocalInit LocalAlloc LocalCompact LocalDiscard LocalFree LocalLock LocalFreeze LocalMelt LocalReAlloc LocalSize LocalUnlock LocalHandleDelta LockData UnlockData LocalFlags
Task Scheduler	GetCurrentTask Yield SetPriority
Resource Manager	AddFontResource RemoveFontResource LoadBitmap LoadCursor LoadIcon LoadMenu LoadString LoadAccelerators FindResource LoadResource AllocResource LockResource FreeResource AccessResource SizeofResource SetResourceHandler
String Translation	AnsiUpper AnsiLower AnsiNext AnsiPrev
Atom Manager	InitAtomTable AddAtom DeleteAtom FindAtom GetAtomName
Windows Initialization File	GetProfileInt GetProfileString WriteProfileString
Debugging	FatalExit
File I/O	_lopen _lcreat _llseek _lread _lwrite _lclose OpenFile GetTempFileName GetTempDrive
Registry	RegOpenKey RegCreateKey RegCloseKey RegDeleteKey RegSetValue RegQueryValue RegEnumKey

2022/11/17 15:22 · prokushev · 0 Comments