 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

DosRead

This call reads the specified number of bytes from a file, pipe, or device to a buffer location.

Syntax

DosRead (FileHandle, BufferArea, BufferLength, BytesRead)

Parameters

;FileHandle (HFILE) - input : File handle obtained from DosOpen. ;BufferArea (PVOID) - output : Address of the input buffer. ;BufferLength (USHORT) - input : Length, in bytes, to be read. ; BytesRead (PUSHORT) - output : Address of the number of bytes read.

Return Code

rc (USHORT) - return Return code descriptions are: * 0 NO_ERROR * 5 ERROR_ACCESS_DENIED * 6 ERROR_INVALID_HANDLE * 26 ERROR_NOT_DOS_DISK * 33 ERROR_LOCK_VIOLATION * 109 ERROR_BROKEN_PIPE * 234 ERROR_MORE_DATA

Remarks

The requested number of bytes may not be read. If the value returned in BytesRead is zero, the process has tried to read from the end of the file.

A BufferLength value of zero is not considered an error. In the case where BufferLength is zero, the system treats the request as a null operation.

The file pointer is moved to the desired position by reading, writing, or issuing DosChgFilePtr.

Family API Considerations

Some options operate differently in the DOS mode than in the OS/2 mode. Therefore, the following

restrictions apply to DosRead when coding for the DOS mode: * Only single-byte DosRead calls may be made to the COM device, because the COM device driver for the DOS environment does not support multiple-byte I/O. * When DosRead is called with a handle that is open to CON, the read goes directly through KbdStringIn using the buffer and length that are provided to DosRead. This causes a DOS mode DosRead to behave differently than an OS/2 mode DosRead. Because an OS/2 mode DosRead buffers the call to KbdStringIn, this allows the user to enter many more characters.

For example, suppose DosRead is called with a buffer of length 10 from a handle opened to CON: * In OS/2 mode, the user is allowed to enter a large number of characters before KbdStringIn begins beeping (indicating the buffer is full). After carriage return is pressed, KbdStringIn returns to DosRead. DosRead returns the first 10 characters to the caller and the remaining characters on subsequent calls to DosRead from CON. * In DOS mode, the user is allowed to enter only eight characters (DOS mode DosRead reserves two characters for CR and LF) before KbdStringIn begins beeping. DosRead returns the eight characters entered, followed by CR and LF to the calling program.

Named Pipe Considerations

A named pipe is read as one of the following: * A byte pipe in byte read mode. * A message pipe in message read mode. * A message pipe in byte read mode.

A byte pipe must be in byte read mode to be read; an error is returned if it is in message read mode. All currently available data, up to the size requested, is returned.

A message pipe can be read in either message read mode or byte read mode. When the message pipe is in message read mode, a read that is larger than the next available message returns only that message. BytesRead is set to indicate the size of the message returned.

A read that is smaller than the next available message returns with the number of bytes requested and an ERROR_MORE_DATA return code. When resuming the reading of a message after ERROR_MORE_DATA is returned, a read always blocks until the next piece (or the rest) of the message can be transferred. DosPeekNmPipe may be used to determine how many bytes are left in the message.

A message pipe in byte read mode is read as if it were a byte stream, skipping over message headers. This is like reading a byte pipe in byte read mode.

When blocking mode is set for a named pipe, a read blocks until data is available. In this case, the read never returns with BytesRead = 0 except at EOF. In message read mode, messages are always read in their entirety, except in the case where the message is bigger than the size of the read.

Non-blocking mode allows a return with BytesRead = 0 only when no data is available at the time of the read.

Example Code

C Binding

<PRE> #define INCL_DOSFILEMGR

USHORT rc = DosRead(FileHandle, BufferArea, BufferLength, BytesRead);

HFILE FileHandle; /* File Handle */ PVOID BufferArea; /* User buffer (returned) */ USHORT BufferLength; /* Buffer length */ PUSHORT BytesRead; /* Bytes read (returned) */

USHORT rc; /* return code */ </PRE>

MASM Binding

<PRE> EXTRN DosRead:FAR INCL_DOSFILEMGR EQU 1

PUSH WORD FileHandle ;File Handle PUSH@ OTHER BufferArea ;User buffer (returned) PUSH WORD BufferLength ;Buffer length PUSH@ WORD BytesRead ;Bytes read (returned) CALL DosRead

Returns WORD </PRE>

Note

Text based on http://www.edm2.com/index.php/DosRead_(FAPI)

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