This call returns any available character data record from the keyboard without removing it from the buffer.

Syntax

KbdPeek (CharData, KbdHandle)

Parameters

;CharData (PKBDKEYINFO) - output : Address of the character data information: :asciicharcode (UCHAR) : ASCII character code. The scan code received from the keyboard is translated to the ASCII character code. :scancode (UCHAR) : Code received from the keyboard hardware. :status (UCHAR) : State of the keystroke event: 'Bit Description' 7-6 00 = Undefined.

```
          01 = Final character, interim character flag off.
          10 = Interim character.
          11 = Final character, interim character flag on.
```

5 1 = Immediate conversion requested. 4-2 Reserved, set to zero. 1 0 = Scan code is a character.

```
           1 = Scan code is not a character; it is an extended key code from
the keyboard.
```

0 1 = Shift status returned without character. :reserved (UCHAR) : NLS shift status. Reserved, set to zero. :shiftkeystat (USHORT) : Shift key status. 'Bit Description' 15 SysReq key down 14 CapsLock key down 13 NumLock key down 12 ScrollLock key down 11 Right Alt key down 10 Right Ctrl key down 9 Left Alt key down 8 Left Ctrl key down 7 Insert on 6 CapsLock on 5 NumLock on 4 ScrollLock on 3 Either Alt key down 2 Either Ctrl key down 1 Left Shift key down 0 Right Shift key down :time (ULONG) : Time stamp indicating when a key was pressed. It is specified in milliseconds from the time the system was started. ; KbdHandle (HKBD) - input : Default keyboard or the logical keyboard.

Return Code

rc (USHORT) - return Return code descriptions are: * 0 NO_ERROR * 439 ERROR_KBD_INVALID_HANDLE * 445 ERROR_KBD_FOCUS_REQUIRED * 447 ERROR_KBD_KEYBOARD_BUSY * 464 ERROR_KBD_DETACHED * 504 ERROR_KBD_EXTENDED_SG

Remarks

On an enhanced keyboard, the secondary enter key returns the normal character 0DH and a scan code of E0H.

Double-byte character codes (DBCS) require two function calls to obtain the entire code.

If shift report is set with KbdSetStatus the CharData record returned, reflects changed shift

information only.

Extended ASCII codes are identified with the status byte, bit 1 on and the ASCII character code being either 00H or E0H. Both conditions must be satisfied for the character to be an extended keystroke. For extended ASCII codes, the scan code byte returned is the second code (extended code). Usually the extended ASCII code is the scan code of the primary key that was pressed.

A thread in the foreground session that repeatedly polls the keyboard with KbdCharIn (with no wait), can prevent all regular priority class threads from executing. If polling must be used and a minimal amount of other processing is being performed, the thread should periodically yield the CPU by issuing a DosSleep call for an interval of at least 5 milliseconds.

Family API Considerations

Some options operate differently in the DOS mode than in the OS/2 mode. Therefore, the following restrictions apply to KbdPeek when coding for the DOS mode: * The CharData structure includes everything except the time stamp. * Interim character is not supported. * Status can be 0 or 1. * KbdHandle is ignored.

Example Code

C Binding

<PRE> typedef struct _KBDKEYINFO { /* kbci */

```
UCHAR    chChar;             /* ASCII character code */
UCHAR    chScan;             /* Scan Code */
UCHAR    fbStatus;           /* State of the character */
UCHAR    bNlsShift;          /* Reserved (set to zero) */
USHORT   fsState;            /* State of the shift keys */
ULONG    time;               /* Time stamp of keystroke (ms since ipl) */
```

}KBDKEYINFO;

#define INCL_KBD

USHORT rc = KbdPeek(CharData, KbdHandle);

PKBDKEYINFO CharData; /* Buffer for data */ HKBD KbdHandle; /* Keyboard handle */

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

MASM Binding

<PRE> KBDKEYINFO struc

```
kbci_chChar    db  ? ;ASCII character code
```

```
kbci_chScan    db  ? ;Scan Code
kbci_fbStatus  db  ? ;State of the character
kbci_bNlsShift db  ? ;Reserved (set to zero)
kbci_fsState   dw  ? ;state of the shift keys
kbci_time      dd  ? ;time stamp of keystroke (ms since ipl)
```

KBDKEYINFO ends

EXTRN KbdPeek:FAR INCL_KBD EQU 1

PUSH@ OTHER CharData ;Buffer for data PUSH WORD KbdHandle ;Keyboard handle CALL KbdPeek

Returns WORD </PRE>

Note

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