Note: This API call is for DOS and Win16 personality only. Use Family API for portability.

2018/09/07 05:04 · prokushev · 0 Comments

Int 25H

Version

1 and higher

Brief

Absolute disk read

Family API

Input

- AL = drive number (00h = A:, 01h = B:, etc)
- CX = number of sectors to read (not FFFFh)
  - DX = starting logical sector number (0000h - highest sector on drive)
  - DS:BX → buffer for data
- CX = FFFFh (DOS 3.31+)
  - DS:BX → disk read packet (see #02548)

Return

- CF clear if successful
- CF set on error
  - AH = status (see #02547)
  - AL = error code (same as passed to INT 24 in DI)

Macro

Notes

partition is potentially >32M (and requires this form of the call) if bit 1 of the device attribute word in the device driver is set

for FAT32 drives (which may be up to 2TB in size), use INT 21/AX=7305h

AX = 0207h if more than 64K sectors on drive or if FAT32 drive – use CX=FFFFH or INT 21/AX=7305h

may destroy all other registers except segment registers; Win9X always sets SI to 0000h due to an apparent coding bug

original flags are left on stack, and must be popped by caller

this call bypasses the DOS filesystem

examination of CPWIN386.CPL indicates that if this call fails with error 0408h on an old-style (<32M) call, one should retry the call with the high bit of the drive number in AL set

Novell DOS 7 decides whether the old-style or new-style (>32M) version of INT 25 must be used solely on the basis of the partition's size, thus forcing use of the new-style call even for data in the first 32M of the partition

PC Tools MIRROR as shipped with MS-DOS 5.0+ checks several signatures at the beginning of INT 25h and INT 26h before it starts to patch these vectors. The signatures it looks for are 83h, F9h, FFh, 74h (CMP CX,-01; JZ ????) at offset +1 from the INT 25h/26h entry points and 2Eh, FFh, 2Eh (JMP DWORD PTR CS:[????]) at the location pointed to by the JZ ????. If it finds these signatures it will use the target address of the far jump for its sub-sequent checks, otherwise it will just take the previous interrupt entry points when scanning for FAh, 2Eh, 8Ch, 16h (CLI; MOV CS:????,SS) or FAh, 2Eh, 89h, 26h (CLI; MOV CS:????,SP) right at the beginning. Hence, it seems the first two checks are to trace through a specific INT 25h/26h filter. However, the purpose of the whole patch is unknown.

A method to detect the actual assignments of logical drive numbers to physical BIOS drive units (for example to detect the boot drive), is to temporarily mount an INT 13h handler recording the used DL drive unit for any INT 13/AH=02h read operations and discarding any attempts to access actual floppy drives. Then call INT 25h for all the appropriate DOS drives and watch the results recorded by the INT 13h interceptor.

Although all registers except segment registers may be destroyed some software depends on some of the registers being preserved. For example some Flash disk drivers requires that DX is not trashed. DR-DOS 7.03 takes care of this.

BUGS: DOS 3.1 through 3.3 set the word at ES:[BP+1Eh] to FFFFh if AL is an invalid drive number

DR DOS 3.41 will return with a jump instead of RETF, leaving the wrong number of bytes on the stack; use the huge-partition version (INT 25/CX=FFFFh) for all partition sizes under DR DOS 3.41

DR DOS 6.0 original issues 05/1991 & 08/1991 reported wrong error codes for “drive not ready” and “write protect”. This was fixed with the DR DOS BDOS patch “PAT321” (1992/02/19, XDIR /C: 947Bh), and later “full” rebuilds (see INT 21/AX=4452h for details).

See Also

INT 26,**INT 21**/AX=7305h

Note

Text based on Ralf Brown Interrupt List Release 61

DOS API	
Process manager	INT 20H, **INT 21H**: 00H, 25H, 26H, 31H, 34H, 35H, 4BH, 4CH, 4DH, 50H, 51H, 52H, 55H, 62H, INT 22H, INT 27H, INT 28H
File manager	INT 25H, INT 26H, **INT 21H**: 0DH, 0EH, 0FH, 10H, 11H, 12H, 13H, 14H, 15H, 16H, 17H, 19H, 1AH, 1BH, 1CH, 21H, 22H, 23H, 24H, 27H, 28H, 29H, 2EH, 2FH, 32H, 3305H, 36H, 39H, 3AH, 3BH, 3CH, 3DH, 3EH, 3FH, 40H, 41H, 42H, 4300H, 4301H, 45H, 45H, 46H, 4EH, 4FH, 54H, 56H, 5700H, 5701H, 5AH, 5BH, 5c00H, 5c01H, 60H, 67H, 68H, 6900H, 6901H, 6AH, 6CH
Character Device I/O	INT 29H, **INT 21H**: 01H, 02H, 03H, 04H, 05H, 06H, 07H, 08H, 09H, 0AH, 0BH, 0AH, 0CH, 5D07H, 5D08H, 5D09H, 5D0AH
Signals	INT 23H, INT 24H, **INT 21H**: 3300H, 3301H, 3302H
Memory manager	**INT 21H**: 48H, 49H, 4AH, 5800H, 5801H, 5802H, 5803H
Date and Time	**INT 21H**: 2AH, 2BH, 2CH, 2DH
Misc	**INT 21H**: 30H, 3306H, 3700H, 3701H, 3702H, 3703H, 59H
NLS	**INT 21H**: 3303H, 3304H, 3800H, 3801H, 6300H, 6301H, 6301H, 6500H, 6501H, 6502H, 6503H, 6504H, 6505H, 6506H, 6507H, 6520H, 6521H, 6522H, 6523H, 65A0H, 65A1H, 65A2H, 6601H, 6602H
Devices	**INT 21H**: 4400H, 4401H, 4402H, 4403H, 4404H, 4405H, 4406H, 4407H, 4408H, 4409H, 440AH, 440BH, 440CH, 440DH, 440EH, 440FH, 4410H, 4411H, 53H
Network	**INT 21H**: 5E00H, 5E01H, 5E02H, 5E03H, 5E04H, 5E05H, 5F00H, 5F01H, 5F02H, 5F03H, 5F04H, 5F05H, 5F07H, 5F08H
osFree Macro Library	
Video I/O	@SetMode @SetCurSz @SetCurPos @GetCur @SetPage @ScrollUp @ScrollDn @Scroll @GetChAtr @PutChAtr @PutCh @SetPalet @SetColor @SetDot @GetDot @WrtTTY @VideoState @GetMode @GetDisplay @GetVideoState @GetEGAInfo @Cls
Hardware info	@Equipment @MemSize
Serial I/O	@AuxInit @AuxSendChar @AuxRecieveChar @AuxStatus
Tape I/O	@TapeOn @TapeOff @TapeRead @TapeWrite
Keyboard I/O	@KbdStatus @CharIn @CharPeek
Printer I/O	@PrnPrint @PrnInit @PrnStatus
Disk I/O	@DskReset @DskStatus @DskRead @DskWrite @DskVerify @DskFormat
Date and Time	@SetTime @GetTime
Mouse	@MouInit @MouShowPointer @MouStatus @MouSetPos @MouSetMickey @MouRegion
Memory manager	@ModBlok SET_BLOCK

2022/10/04 14:28 · prokushev · 0 Comments
2018/09/04 17:23 · prokushev · 0 Comments

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 DosShutdown
	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