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 21H, AH=4BH

Version

2 and higher

Brief

“EXEC” - LOAD AND/OR EXECUTE PROGRAM

Family API

Input

```
AH = 4Bh
AL = type of load
    00h load and execute
    01h load but do not execute
    03h load overlay (see #01591)
    04h load and execute in background (European MS-DOS 4.0 only)
    "Exec & Go" (see also AH=80h)
DS:DX -> ASCIZ program name (must include extension)
ES:BX -> parameter block (see #01590,#01591,#01592)
CX = mode (subfunction 04h only)
    0000h child placed in zombie mode after termination
    0001h child's return code discarded on termination
```

Return

```
 CF clear if successful
    BX,DX destroyed
    if subfunction 01h, process ID set to new program's PSP; get with
    INT 21/AH=62h
CF set on error
```

```
    AX = error code (01h,02h,05h,08h,0Ah,0Bh) (see #01680 at AH=59h)
```

Notes

```
DOS 2.x destroys all registers, including SS:SP
under ROM-based DOS, if no disk path characters (colons or slashes)
  are included in the program name, the name is searched for in the
  ROM module headers (see #01595) before searching on disk
for functions 00h and 01h, the calling process must ensure that there
  is enough unallocated memory available; if necessary, by releasing
  memory with AH=49h or AH=4Ah
for function 01h, the AX value to be passed to the child program is put
  on top of the child's stack
for function 03h, DOS assumes that the overlay is being loaded into
  memory allocated by the caller
function 01h was undocumented prior to the release of DOS 5.0
some versions (such as DR DOS 6.0) check the parameters and parameter
  block and return an error if an invalid value (such as an offset of
  FFFFh) is found
background programs under European MS-DOS 4.0 must use the new
  executable format
this function ignores the filename extension, instead checking the
  first two bytes of the file to determine whether there is a valid
  .EXE header (see #01594); if not, the file is assumed to be in .COM
  format.  If present, the file may be in any of several formats which
  are extensions of the original .EXE format (see #01593)
.COM-format executables begin running with the following register
  values:
    AL = 00h if first FCB has valid drive letter, FFh if not
    AH = 00h if second FCB has valid drive letter, FFh if not
    CS,DS,ES,SS = PSP segment
    SP = offset of last word available in first 64K segment
    (note: AX is always 0000h under DESQview)
old-format executables begin running with the following register
  values:
    AL = 00h if first FCB has valid drive letter, FFh if not
    AH = 00h if second FCB has valid drive letter, FFh if not
    DS,ES = PSP segment
    SS:SP as defined in .EXE header
    (note: AX is always 0000h under DESQview)
new executables begin running with the following register values
    AX = environment segment
    BX = offset of command tail in environment segment
    CX = size of automatic data segment (0000h = 64K)
    ES,BP = 0000h
    DS = automatic data segment
    SS:SP = initial stack
  the command tail corresponds to an old executable's PSP:0081h and
  following, except that the 0Dh is turned into a NUL (00h); new
```

```
  format executables have no PSP
under the FlashTek X-32 DOS extender, only function 00h is supported
  and the pointers are passed in DS:EDX and ES:EBX
DR DOS 6 always loads .EXE-format programs with no fixups and
  .COM-format programs starting with 9Ch 55h (PUSHF/PUSH BP) above the
  64K mark to avoid the EXEPACK bug, by extending the memory block
  containing the program's environment; this code is disabled if the
  name of the parent program as stored in the MCB is 'WIN'.
DR DOS 3.41 and 5.0 check for a valid filename before testing the
  subfunction number, so the otherwise invalid subfunction 02h will
  only return error code 01h if the given filename actually exists;
  otherwise, errors 02h, 03h, or 05h are returned
BUGS:   DOS 2.00 assumes that DS points at the current program's PSP
Load Overlay (subfunction 03h) loads up to 512 bytes too many if the
  file contains additional data after the actual overlay
Load but Do Not Execute (subfunction 01h) is reported to corrupt the
  top word of the caller's stack if the loaded module terminates with
  INT 21/AH=4Ch in some versions of MS-DOS, including v5.00.
```

See also

```
AX=4B05h,AH=4Ch,AH=4Dh,AH=64h/BX=0025h,AH=8Ah,INT 2E,INT 60/DI=0604h
```

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

DOS API	
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